

**Development of a Low-Cost Postural Measurement  
System to Assist during Assessment, Optimal  
Correction, and Casting of the Spinal Orthosis for  
Scoliosis Patients**

By

**Pitchaya Rayothee**

B.Sc. in Prosthetics and Orthotics (1<sup>st</sup> class honor),

M.Eng. in Biomedical Engineering

This thesis is submitted in partial fulfilment of the requirements for the  
degree of PhD in Biomedical Engineering

The Department of Biomedical Engineering

University of Strathclyde

Glasgow, UK

2024

## **Declaration of Authenticity and Author's Rights**

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed:

A handwritten signature in blue ink, consisting of stylized initials and a surname, followed by a horizontal line.

Date: 22 March 2024

## **Acknowledgements**

Firstly, I would like to thank my supervisor, Professor Philip Rowe, for his guidance and support throughout my PhD study. It was difficult to pursue a PhD during the COVID-19 pandemic. However, with his support and love, I was able to overcome all obstacles.

In particular, I would like to thank my colleagues at Sirindhorn School of Prosthetics and Orthotics (SSPO), especially Assoc. Prof. Dr. Nisarath Opatkiattikul (former SSPO director), Assoc. Prof. Gulapar Srisawasdi (SSPO director), Asist. Prof. Dr. Manunchaya Samala (Deputy director of research) and the SSPO research team for supporting and guiding the clinical experiment and medical related.

I gratefully acknowledge the funding for my PhD from the Faculty of Medicine Siriraj Hospital, Mahidol University, Thailand, for making it possible for me to study here. Furthermore, the clinical experiment of this research project is supported by the Siriraj Research Development Fund, Grant number (IO) R016532021, Faculty of Medicine Siriraj Hospital, Mahidol University.

I would like to thank all scoliosis participants and their relatives who participated in this thesis. Although the recruitment period overlapped with the COVID-19 pandemic in Thailand, all of them were willing to join and well cooperate.

I would like to thank my parents and friends for their love and support, which helped me overcome all the difficulties.

Pitchaya Rayothee

## **Abstract**

Adolescent idiopathic scoliosis (AIS) is a spine pathology in teenagers that causes 3-dimensional deformities. Radiographs are a standard outcome measurement method that can be evaluated for coronal and sagittal plane deformities but lack 3-dimensional information. Casting is a process of capturing the trunk shape under force correction by clinicians. Scoliosis casting frames are uncommon in clinical practice due to difficulty in usage and lack of evidence. There is a lack of information about the force magnitude, force locations and directions to correct deformity in three dimensions during casting.

The study objective was to develop and validate a system that could quantify the spinal deformity of AIS patients in 3 dimensions, apply forces to correct the spinal deformity in three dimensions, measure the magnitude of forces and illustrate those force directions in three dimensions. This study also sought to clarify scoliosis deformity in 3 dimensions and how it changed on applying forces during casting.

A low-cost postural measurement system was developed using 8 Raspberry Pi mini-computers with integrated cameras arranged in a circle that communicated wirelessly with a main computer. The software was written in C++ and ran on Visual Studio 2019 and Windows 10. The stereo camera concept was used and implemented with OpenCV for camera calibration and marker position calculations in 3 dimensions. Each Raspberry Pi captured an image of the marker, which was stored on the central computer where the marker position in 3D space was calculated and used to quantify relevant spinal parameters. Six load cells were calibrated and used to measure the magnitude of the forces applied during casting. A Scoliosis casting apparatus was designed in the SolidWorks program and built following the design to apply the casting forces using manipulator arms. Force measuring software was written in Python and ran on Visual Studio 2017 and Windows 10. Validation of the data obtained was demonstrated by a series of experiments during the development process. After completing development, the systems were tested with AIS patients, and the data were analysed using descriptive and inferential statistics. The RMSE of the developed postural measurement system when locating markers was 2.42 mm, appropriate for quantifying scoliosis deformity in clinical practice. The validity of the system for



clinical practice was examined in a clinical experiment that recruited ten AIS participants. The experiment was approved by ethical committees from the University of Strathclyde and Mahidol University. The assessment result was that the postural measurement system had a high concurrent validity compared to radiographs for CSA (r-value: 0.57 - 0.96), SSA (r-value: 0.35 - 0.94) and trunk balance (r-value = 0.91). In coronal spinal parameters for the postural measurement system in assessment VS optimal correction and assessment VS casting, there was a reduction percentage of apical translation of approximately 50% with a statistically significant difference in reduction of apical translation. However, there was no statistically significant difference in CSA. For optimal correction VS casting, there was a statistically significant equivalence when the margin of equivalence (M) was 5°. In SSA, there was a statistically significant equivalence when M = 9° for assessment VS optimal correction and 8° for assessment VS casting and optimal correction VS casting. In 3DSA, there was no statistically significant difference for assessment VS optimal correction and assessment VS casting. There was statistically significant equivalence when M = 8° for optimal correction VS casting. There was a high reduction of trunk balance from assessment (-8.19 mm, SD = 11.58) to optimal correction (-1.25 mm, SD = 4.56) and casting (-0.71 mm, SD = 3.32). However, there was no statistically significant difference. There was a statistically significant equivalence when M = 3 mm for optimal correction VS casting. Trunk asymmetry (POTSI) improved from 33.54% (SD = 16.23) in assessment to 22.80% (SD = 12.41) in casting. The mean of the total reduction of the horizontal trunk rotation angle was 14.83° (SD = 12.44). The force to correct the deformity at each area was approximately 30 N, and the total force each patient had to tolerate during optimal correction was approximately 150 N.

In conclusion, we produced a system that could quantify, in three dimensions, the spinal deformity of AIS patients, produce relevant spinal parameters and quantify casting force magnitude and direction. This could be done before and after casting and hence quantify the effects of casting. The scoliosis casting apparatus itself could be suitably adjusted to apply forces to correct deformity in three dimensions as part of clinical practice. The system as a whole has the potential to quantify spinal orthotic practice and hence base practice on a scientific evidential basis.

## List of Presentations at Peer Reviewed Conferences

- Pitchaya Rayothee, Philip Rowe. (2021). A new low-cost motion capture system for assisting scoliosis treatment. Oral Presentation at International Society for Prosthetics and Orthotics (ISPO) 18<sup>th</sup> World Congress, Virtual presentation. November 2021.
- Pitchaya Rayothee, Philip Rowe. (2023). Quantifying spinal parameters for scoliosis using a low-cost 3-dimensional motion capture system. Oral Presentation at International Society of Biomechanics 2023, Fukuoka, Japan. August 2023.

# Contents

Declaration of Authenticity and Author's Rights .....	i
Acknowledgements .....	ii
Abstract .....	iii
List of Presentations at Peer Reviewed Conferences .....	v
Contents .....	vi
List of Figures .....	xi
List of Tables.....	xxvi
List of Abbreviations.....	xxxii
1. Chapter 1 Introduction .....	1
1.1. Introduction to scoliosis .....	2
1.2. Introduction to scoliotic treatment .....	4
1.3. Introduction to orthotic treatment .....	4
1.4. Introduction to spinal outcome measurement .....	6
1.5. Introduction to the need for alternative technology to assess scoliosis and scoliotic treatment .....	7
1.6. General outline of this thesis.....	8
2. Chapter 2 Aims, Objectives, and Thesis Outlines .....	10
2.1. Introduction .....	11
2.2. Summary of scoliosis research gap .....	11
2.3. Aim of the Thesis .....	12
2.4. The objectives of the Thesis.....	13
2.5. Thesis outline and Scope.....	14
3. Chapter 3 Literature Review .....	17
3.1. Introduction .....	18
3.2. Scoliosis .....	18
3.3. Spinal orthotic designs and its effectiveness.....	23
3.4. Radiographic outcome measurement in AIS.....	28
3.5. Alternative technologies to measure the outcome in AIS .....	34
3.5.1. Ultrasound system.....	34
3.5.2. Surface topography (ST) .....	41

3.5.3. Magnetic resonance imaging (MRI) .....	45
3.5.4. Computerized tomography (CT) scan .....	46
3.5.5. EOS imaging .....	48
3.5.6. Motion capture system .....	49
3.6. Accuracy of motion capture systems .....	57
3.6.1. Accuracy of high-cost motion capture systems .....	58
3.6.2. Accuracy of low-cost motion capture systems.....	61
3.7. Casting frames for scoliosis .....	72
3.8. Conclusion .....	77
4. Chapter 4 Development of Scoliosis Casting Apparatus .....	84
4.1. Introduction .....	85
4.2. Conceptual design of scoliosis casting apparatus .....	87
4.3. Function of scoliosis casting apparatus.....	90
4.4. How to adjust the scoliosis casting apparatus during optimal correction and casting process with scoliosis patient.....	99
4.5. Conclusion .....	101
5. Chapter 5 Development of Force Measuring System .....	103
5.1. Introduction .....	104
5.2. Components for load cell calibration .....	105
5.3. Procedure for load cell calibration .....	107
5.4. Calibration result.....	109
5.5. Simple linear regression for load cell calibration .....	121
5.6. Root-mean-square error (RMSE) in load cell calibration.....	123
5.7. Simple linear regression results and errors for another five load cells .....	123
5.8. Development and procedure of a force-measuring system in clinical experiments .....	125
5.9. Conclusion .....	127
6. Chapter 6 Development of A Low-Cost Postural Measurement System: Using 2 and 3 Raspberry Pis and Cameras to Estimate Marker Positions .....	128
6.1. Introduction .....	129
6.2. Components of a low-cost postural measurement system .....	130

6.3.	Image processing technique to calculate the centre of a marker on an image .....	132
6.4.	Experiment 1: A low-cost postural measurement system using 2 Raspberry Pis with parallel cameras.....	135
6.5.	Experiment 2: A low-cost postural measurement system using 2 Raspberry Pi cameras with 45° horizontally tilting cameras.....	148
6.6.	Experiment 3: A low-cost postural measurement system using 3 Raspberry Pi cameras .....	158
6.7.	Discussion and conclusion of Experiments 1, 2, and 3.....	167
7.	Chapter 7 Development of A Low-Cost Postural Measurement System based on OpenCV and Stereo Camera Calibration methods.....	171
7.1.	Introduction .....	172
7.2.	General concept of stereo camera to calculate marker positions .....	173
7.3.	Experiment 4: A low-cost postural measurement system based on stereo camera concept .....	184
7.4.	Discussion and conclusion for Experiment 4.....	202
7.5.	Limitations and recommendations .....	206
8.	Chapter 8 Clinical research methodology and Outcome measurements for Adolescent Idiopathic Scoliosis .....	208
8.1.	Introduction .....	209
8.2.	The objectives of the clinical study.....	210
8.3.	Clinical research methodology.....	214
8.4.	Clinical outcome measurement and Data analysis.....	221
8.5.	Formula to calculate spinal parameters.....	226
8.6.	Conclusion .....	244
9.	Chapter 9 Evaluation of The Developed System to Treat Adolescent Idiopathic Scoliosis During Assessment, Optimal Correction, and Casting.....	247
9.1.	Introduction .....	248
9.2.	Experimental results for individual participants .....	250
9.3.	Conclusion .....	318
10.	Chapter 10 Statistical Analysis and Result Interpretation from Clinical Experient with Adolescent Idiopathic Scoliosis .....	322

10.1. Introduction .....	323
10.2. Coronal spinal angles .....	326
10.3. Sagittal spinal angles.....	336
10.4. 3-dimensional spinal angles .....	346
10.4. Trunk balance.....	352
10.5. Trunk asymmetry in coronal plane between the low-cost postural measurement system in assessment and casting .....	358
10.6. Trunk horizontal rotation between the low-cost postural measurement system in assessment and casting.....	360
10.7. Force magnitude during optimal correction process.....	361
10.8. Conclusion .....	362
11. Chapter 11 Discussion, Limitation, Recommendation, and Conclusion .....	367
11.1. Introduction.....	368
11.2. Discussion .....	369
11.3. Limitations and recommendations .....	394
11.4. Conclusion .....	400
References .....	408
Appendix .....	426
Appendix 1 Coding for load cell calibration for experiment analysis (Chapter 5) .....	427
Appendix 2 Coding for load cell calibration for GUI (Chapter 5).....	427
Appendix 3 Coding for force-measuring program for GUI (Chapter 5).....	467
Appendix 4 Coding for xy-value calculation on Raspberry Pi (Chapter 6).....	476
Appendix 5 Coding for xy-value calculation on computer (Chapter 6).....	479
Appendix 6 Coding for camera calibration on Raspberry Pi (Chapter 6).....	481
Appendix 7 Coding for camera calibration, stereo camera calibration, and marker calculation (Chapter 7).....	482
Appendix 8 Certificate of approval by Siriraj Institutional Review Board .....	522
Appendix 9 Ethical approval from University of Strathclyde .....	524
Appendix 10 Participant information sheet and Poster invitation .....	525
Appendix 11 Consent form .....	532
Appendix 12 Case record form during clinical experiment.....	534

Appendix 13 Statistical analysis table.....	539
Appendix 14 Abstract book for presentations at peer reviewed conferences .....	549

## List of Figures

Figure 3.1 Spinal deformity of scoliosis in 3 dimensions (Wang <i>et al.</i> , 2014).....	19
Figure 3.2 Trunk asymmetry observed from the back of a scoliosis patient (Janicki and Alman, 2007b).....	21
Figure 3.3 The example of Boston brace (Fayssoux, Cho and Herman, 2010) .....	25
Figure 3.4 The example of Chêneau brace, where a). posterior and b). anterior views (Zaborowska-Sapeta <i>et al.</i> , 2011a). .....	27
Figure 3.5 Cobb angle measurement in the coronal plane. ....	30
Figure 3.6 Cobb angle measurement of thoracic kyphosis in the sagittal plane. ....	31
Figure 3.7 Cobb angle measurement of lumbar lordosis in the sagittal plane. ....	31
Figure 3.8 Vertebral rotation grading by Nash and Moe's (Greenwood and Bogar, 2014) .....	32
Figure 3.9 Vertebral rotation angle by Perdriolle, where a) Perdriolle torsionometer, and b) the torsionometer's outer margins are aligned over the vertebra's lateral borders (Richards, 1992).....	32
Figure 3.10 The stage of the Risser sign (Hacquebord and Leopold, 2012).....	33
Figure 3.11 a) In-brace ultrasound scanning, and b) the locations for corrective pad (Li <i>et al.</i> , 2010, Li <i>et al.</i> , 2012).....	36
Figure 3.12 a) The opening at the back of the frame, and b) a subject stands on a frame with a custom Providence brace design system (Edmond <i>et al.</i> , 2017).....	38
Figure 3.13 a) The standing pre-brace radiographs with Cobb angle 37°, b) The baseline US scan (Cobb angle 35°), c) The first trial US scan (Cobb angle 25°), and d) The 2nd trial US scan (Cobb angle 23°) (Edmond <i>et al.</i> , 2017) .....	38
Figure 3.14 a) Coronal Cobb angle measured by X-ray, b) automatic spinous process angle measurement, c) manual spinous process angle measurement, and d) manual transverse process angle measurement (Brink <i>et al.</i> , 2018).....	39
Figure 3.15 (a) The Scolioscan (b) Ultrasound assessment for a patient (Wong <i>et al.</i> , 2019). .....	40
Figure 3.16 The deviation colour map (DCM) to analyse the spinal parameters in Surface Topography (ST) torso scans (Minolta laser scanner) (Komeili <i>et al.</i> , 2015a, Komeili <i>et al.</i> , 2015b) .....	43



Figure 3.17 Example of image created by MRI (Schmitz <i>et al.</i> , 2001).....	46
Figure 3.18 Example of image created by CT scan (Pierre-Aurelien <i>et al.</i> , 2018)...	47
Figure 3.19 Example of images taken by the EOS system in 2 dimensions and reconstructed in 3 dimensions (Illés and Somoskeöy, 2012).....	48
Figure 3.20 Postural geometry in the transverse and coronal planes from the study of Zabjek and colleagues (Zabjek <i>et al.</i> , 2008). .....	50
Figure 3.21 Marker locations from the study of Solomito, Lee and Peterson .....	51
Figure 3.22 Marker locations from the study of Jang (Jang, 2018).....	53
Figure 3.23 The spinal parameters of the spinal column, including a) coronal angles, b) sagittal angles, and c) 3-dimensional angles relative to the horizontal plane (Jang, 2018). .....	54
Figure 3.24 Marker locations and used to calculate (a angle) thoracic Cobb angle, (b) lumbar Cobb angle, (c) thoracic kyphosis, (d) lumbar lordosis, (e) pelvic tilt, (f) CVA, and (g) SVA (Pesenti <i>et al.</i> , 2020) .....	55
Figure 3.25 Marker locations from the study of Leu and colleagues (Lau <i>et al.</i> , 2022) .....	56
Figure 3.26 SAMSA device (Richards, 1999, Topley and Richards, 2020).....	58
Figure 3.27 The absolute error (mm) between the high-cost motion capture systems in 1999 and 2020 (Topley and Richards, 2020).....	59
Figure 3.28 Geometry to calculate the error for a motion capture system in a scoliosis application. ....	71
Figure 3.29 EDF Cotrel’s frame created by Cotrel (Mauroy <i>et al.</i> , 2014, Canavese <i>et al.</i> , 2016) .....	73
Figure 3.30 Risser frame (Fayssoux, Cho and Herman, 2010).....	74
Figure 3.31 Apparatus created by Barry McCoy (McCoy and Barry, 1996).....	75
Figure 3.32 A custom Providence brace standing frame (Edmond <i>et al.</i> , 2015, Edmond <i>et al.</i> , 2017) .....	75
Figure 3.33 Scoli-Standing Frame created by Jang (Jang, 2018). .....	76
Figure 3.34 The conceptual design of a new development system for this thesis in front view. ....	82
Figure 3.35 The conceptual design of a new development system for this thesis in oblique view.....	83

Figure 4.1 Consideration of a 3-point force system to correct spinal deformity in the coronal plane, a) to correct the lumbar curve, b) to correct the thoracic curve. ....	85
Figure 4.2 Traditional method to perform casting with a scoliosis patient.....	86
Figure 4.3 The conceptual design of a scoliosis casting apparatus, a). oblique view, b). front view, c). back view, d). side view, and e) top view. ....	89
Figure 4.4 Design of main structural element in the oblique view. ....	91
Figure 4.5 Design of sitting bar in the oblique view.....	91
Figure 4.6 Design of arm-rest area, a). oblique view and b). top view.....	92
Figure 4.7 Design of abdominal manipulator, a) oblique view and b) top view. ....	93
Figure 4.8 Design of gluteus medius manipulators, a) oblique view and b) top view. ....	94
Figure 4.9 Design of axillary manipulators in oblique view.....	95
Figure 4.10 Design of left lumbar or thoracic-corrective manipulator, a) oblique view and b) top view.....	96
Figure 4.11 Design of right lumbar or thoracic-corrective manipulator, a) oblique view and b) top view.....	97
Figure 4.12 Scoliosis casting apparatus, a) posterior view and b) oblique view. ....	98
Figure 4.13 Scoliosis casting apparatus, a) arm-rest area, b) sitting bar, c) left axillary manipulator, d) gluteus medius manipulators, e) abdominal manipulator, and f) left and right lumbar or thoracic-corrective manipulators.....	99
Figure 4.14 Scoliosis patient in the optimal correction process, a) before applying the forces, b) while applying the forces in the posterior view, c) in the lateral view, and d) in the anterior view. ....	101
Figure 5.1 Load cell (ID: 3136_0) connected with the PhidgetBridge.....	105
Figure 5.2 PhidgetBridge (ID: 1046_1).....	106
Figure 5.3 Example of required components for load cell calibration, a) testing mass, b) calibration machine and support base of load cell, and c) computer and program for load cell calibration. ....	107
Figure 5.4 Example of data collection sheet used for calculating a simple linear regression formula, a) increasing mass condition, and b) decreasing mass condition .....	107

Figure 5.5 Graphical user interface (GUI) of the program for load cell calibration .....	108
Figure 5.6 Example of the line graph of the calibration result.....	109
Figure 5.7 Calibration result of the load cell in condition 1. ....	110
Figure 5.8 Calibration result of the load cell in condition 2 .....	111
Figure 5.9 Calibration result of the load cell in condition 3 .....	111
Figure 5.10 Calibration result of the load cell in condition 4. ....	112
Figure 5.11 Calibration result of the load cell in condition 5. ....	112
Figure 5.12 Calibration result of the load cell in condition 6. ....	113
Figure 5.13 Calibration result of the load cell in condition 7 .....	113
Figure 5.14 Calibration result of the load cell in condition 8 .....	114
Figure 5.15 Calibration result of the load cell in condition 9 .....	115
Figure 5.16 Calibration result of the load cell in condition 10 .....	115
Figure 5.17 Calibration result of the load cell in condition 11 .....	116
Figure 5.18 Calibration result of the load cell in condition 12. ....	116
Figure 5.19 Calibration result of the load cell in condition 13. ....	117
Figure 5.20 Calibration result of the load cell in condition 14. ....	117
Figure 5.21 Calibration result of the load cell in condition 15. ....	118
Figure 5.22 Calibration result of the load cell in condition 16. ....	118
Figure 5.23 Calibration result of load cell when the mass increased and decreased between 0 kg and 40 kg.....	120
Figure 5.24 The scatter plot between applied force and voltage ratio during increasing and decreasing force conditions. ....	122
Figure 5.25 The scatter plot between applied force and voltage ratio for six load cells .....	124
Figure 5.26 The graphical user interface (GUI) of the force-measuring program. ....	125
Figure 5.27 The scoliosis casting apparatus applies multiple forces to correct the scoliosis deformity, and the program is measuring the magnitude of the force at each area. ....	126
Figure 6.1 Raspberry Pi 3 Model B+ (Raspberry Pi, 2023).....	131
Figure 6.2 Raspberry Pi Camera Module 2 NoIR (Raspberry Pi, 2023). ....	132
Figure 6.3 The original image after opening the Raspberry Pi camera. ....	133

Figure 6.4 The grayscale image after applying cv2.cvtColor function.....	133
Figure 6.5 The thresholded image after applying cv2.threshold function. ....	134
Figure 6.6 The image after applying cv2.bitwise_and, cv2.findContours and cv2.drawContours functions, respectively. ....	134
Figure 6.7 The result of the centre of the marker in x and y values.....	135
Figure 6.8 Example of required components for Experiment 1.....	136
Figure 6.9 Twenty-five markers on the flatboard used in Experiment 1. ....	136
Figure 6.10 The component setting for Experiment 1, a) from the side view, and b) from the top view. ....	137
Figure 6.11 Geometry of right triangle similarity to calculate the focal length (f) for Experiment 1 in the top view .....	138
Figure 6.12 Geometry for a larger triangle for calculating a focal length (f) in Experiment 1. ....	139
Figure 6.13 The geometry for a smaller triangle for calculating a focal length (f) in Experiment 1. ....	139
Figure 6.14 Geometry of right triangle similarity for Experiment 1 in the side view .....	142
Figure 6.15 The range of red colour to represent the range of the error in Experiment 1.....	143
Figure 6.16 A different tone of red colour to represent the X error compared with Reference 1. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.....	144
Figure 6.17 A different tone of red colour to represent the Y error compared with Reference 1. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.....	144
Figure 6.18 A different tone of red colour to represent the Z error compared with Reference 1. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.....	145
Figure 6.19 A different tone of red colour to represent the X error compared with Reference 2. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.....	146

Figure 6.20 A different tone of red colour to represent the Y error compared with Reference 2. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.....	146
Figure 6.21 A different tone of red colour to represent the Z error compared with Reference 2. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.....	147
Figure 6.22 The VNC Viewer program installed on the computer.....	149
Figure 6.23 The FileZilla program installed on the computer. ....	150
Figure 6.24 Markers attached on the flatboard for Experiment 2.....	151
Figure 6.25 The camera position for Experiment 2.....	151
Figure 6.26 The geometry used to calculate the marker position of Experiment 2 in the top view. ....	151
Figure 6.27 The geometry used to calculate the marker position of Experiment 2 in the side view.....	153
Figure 6.28 The image of the chessboard, a) original shape of chessboard and b) distorted image taken by distorted lens (Medium, 2023).....	159
Figure 6.29 The chessboard pattern used in Experiment 3. ....	160
Figure 6.30 The program to capture the image in Experiment 3. ....	160
Figure 6.31 An example of images used for camera calibration.....	161
Figure 6.32 The result after drawing the chessboard corners. ....	162
Figure 6.33 The example of calibration results saved in the Raspberry Pi.....	162
Figure 6.34 The geometry used to calculate the marker position of Experiment 3 in the top view. ....	163
Figure 6.35 The camera frame and volume experiment in Experiment 3.....	164
Figure 6.36 The surface experiment in Experiment 3.....	164
Figure 7.1 The pinhole camera model to create an image. ....	174
Figure 7.2 The pinhole camera model after switching between the image plane and pinhole plane. ....	174
Figure 7.3 Recorded images for camera calibration using the “getpicameraimage” program. ....	177
Figure 7.4 Example of image, a) the original image and b) the undistorted image.	177

Figure 7.5 Recorded images for stereo camera calibration using the “getpistereoisimages” program.....	178
Figure 7.6 Recorded images for reprojection process using the “getimagepairs” program.....	180
Figure 7.7 User specify the camera number for running the "reconstructfromimages" program.....	180
Figure 7.8 Image of a scoliosis patient with markers attached to anatomical landmarks obtained from cameras 1 and 2, a) undistorted image and b) converted image from colour to grayscale. ....	181
Figure 7.9 Thresholding the images.....	182
Figure 7.10 Find and draw contour of marker in images.....	182
Figure 7.11 Stereo vision concept after completing all processes. ....	183
Figure 7.12 All programs for calibrating the cameras and calculating marker positions. The programs were kept on the C-drive of the computer. ....	184
Figure 7.13 Folders on C-drive that contain image folders for camera calibration, the image folders for stereo camera calibration, and the camera calibration result.....	184
Figure 7.14 Cameras setting on the frame with the chessboard in the middle of camera frame. ....	186
Figure 7.15 The experimental setting to calculate marker positions using a stereo camera pair.....	188
Figure 7.16 The camera frame, Raspberry Pis with cameras positioned on the frame, and experimental setting for Experiment 4.4. ....	192
Figure 7.17 The camera positions and orientations on the camera frame, and points O and P in the top view.....	194
Figure 7.18 Experimental setting for Experiment 4.5.....	195
Figure 7.19 Experimental setting for Experiment 4.6, a) a high-cost motion system, b) a low-cost postural measurement system, and c) a 3D scanning system.....	197
Figure 8.1 The base and marker used for data collection, where a) a base (a piece of magnet), b) a marker (a white circle marker attached with another piece of magnet), c) place the base on the landmark and secure with tape, and d) attach the marker to the base.....	217

Figure 8.2 Participant stand in the middle of the camera frame and the low-cost postural measurement program collects the data during assessment process.....	218
Figure 8.3 The spinal deformity is corrected by scoliosis casting apparatus and the programs (force measuring and postural measurement systems) collect the data during optimal correction process. ....	218
Figure 8.4 The spinal deformity is corrected by scoliosis casting apparatus and researcher performs casting.....	219
Figure 8.5 Re-attach markers to the positive model, and researcher run program to collect marker positions. ....	220
Figure 8.6 a) a removable pad that can be removed from the load cell, and b) a long wooden stick that can fit along the tip of the removable pad to represent the force direction.....	221
Figure 8.7 Spinal parameters from radiographic evaluation.....	222
Figure 8.8 Location of markers, where a) the list of marker positions, b) markers for spinal column, c) markers for trunk asymmetry, d) markers for horizontal rotation, and e) markers illustration with the horizontal rotation.....	224
Figure 8.9 Geometry and formula for calculating the trigonometry.....	226
Figure 8.10 Marker locations and the 3D spinal angles, including 3D-UTA, 3D- LTA, 3D-ULA, and 3D-LLA. These angles are relative to the horizontal plane.....	227
Figure 8.11 Marker positions and geometry to calculate 3D-UTA.....	228
Figure 8.12 Marker positions and geometry to calculate 3D-LTA.....	228
Figure 8.13 Marker positions and geometry to calculate 3D-ULA.....	229
Figure 8.14 Marker positions and geometry to calculate 3D-LLA.....	230
Figure 8.15 Marker positions and geometry to calculate CUTA.....	231
Figure 8.16 Marker positions and geometry to calculate CLTA.....	231
Figure 8.17 Marker positions and geometry to calculate CULA.....	232
Figure 8.18 Marker positions and geometry to calculate CLLA.....	232
Figure 8.19 Marker positions and geometry to calculate SUTA.....	233
Figure 8.20 Marker positions and geometry to calculate SLTA.....	234
Figure 8.21 Marker positions and geometry to calculate SULA.....	234
Figure 8.22 Marker positions and geometry to calculate SLLA.....	235

Figure 8.23 Geometry to calculate the reduction percentage of the apical translation. ....	236
Figure 8.24 Marker positions and geometry to calculate trunk balance. ....	237
Figure 8.25 Marker positions and geometry to calculate shoulder level. ....	238
Figure 8.26 Marker positions and geometry to calculate axilla level. ....	238
Figure 8.27 Marker positions and geometry to calculate waist level.....	239
Figure 8.28 Marker positions and geometry to calculate the distance of left and right side at axilla.....	240
Figure 8.29 Marker positions and geometry to calculate the distance of left and right side at waist. ....	240
Figure 8.30 Geometry and formula to calculate the POsterior Trunk Symmetry Index (POTSI).....	242
Figure 8.31 Marker locations at PSIS and left and right markers at T7.....	243
Figure 8.32 Geometry to calculate the horizontal rotation of the trunk at T7 with respect to the PSIS level, where a) is the geometry for PSIS level and b) is the geometry for markers at T7 level. ....	243
Figure 9.1 Spinal column in the coronal plane of participant 1, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	252
Figure 9.2 Spinal column in the sagittal plane of participant 1, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	252
Figure 9.3 The trunk horizontal rotation of participant 1, where a) and b) the results from a postural measurement system during the assessment and casting.....	254
Figure 9.4 The trunk asymmetry of participant 1, where a) and b) the results from a postural measurement system during the assessment and casting. ....	255
Figure 9.5 Forces applied to participant 1's trunk in, a) the transverse, b) coronal, and c) sagittal planes. ....	256
Figure 9.6 Spinal column in the coronal plane of participant 2, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	258



Figure 9.7 Spinal column in the sagittal plane of participant 2, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	259
Figure 9.8 Spinal parameters in the transverse plane of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting. ....	261
Figure 9.9 Spinal parameters in the trunk asymmetry of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting. ....	262
Figure 9.10 Forces applied to participant 2's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	263
Figure 9.11 Spinal column in the coronal plane of participant 3, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	265
Figure 9.12 Spinal column in the sagittal plane of participant 3, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	266
Figure 9.13 Spinal parameters in the transverse plane of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting. ....	268
Figure 9.14 Spinal parameters in the trunk asymmetry of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting. ....	269
Figure 9.15 Forces applied to participant 3's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	270
Figure 9.16 Spinal column in the coronal plane of participant 4, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	272
Figure 9.17 Spinal column in the sagittal plane of participant 4, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	272

Figure 9.18 Spinal parameters in the transverse plane of participant 4, where a) and b) the results from a postural measurement system during the assessment and casting. ....	275
Figure 9.19 Spinal parameters in the trunk asymmetry of participant 4, where a) and b) the results from a postural measurement system during the assessment and casting. ....	275
Figure 9.20 Forces applied to participant 4's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	277
Figure 9.21 Spinal column in the coronal plane of participant 5, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	279
Figure 9.22 Spinal column in the sagittal plane of participant 5, where a) X-ray image and b) to d) the results from a postural measurement capture system during assessment, optimal correction, and casting. ....	279
Figure 9.23 Spinal parameters in the transverse plane of participant 5, where a) and b) the results from a postural measurement system during the assessment and casting. ....	282
Figure 9.24 Spinal parameters in the trunk asymmetry of participant 5, where a) and b) the results from a postural measurement system during the assessment and casting. ....	282
Figure 9.25 Forces applied to participant 5's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	284
Figure 9.26 Spinal column in the coronal plane of participant 6, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	286
Figure 9.27 Spinal column in the sagittal plane of participant 6, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	286
Figure 9.28 Spinal parameters in the transverse plane of participant 6, where a) and b) the results from a postural measurement system during the assessment and casting. ....	289

Figure 9.29 Spinal parameters in the trunk asymmetry of participant 6, where a) and b) the results from a postural measurement system during the assessment and casting. ....	289
Figure 9.30 Forces applied to participant 6's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	291
Figure 9.31 Spinal column in the coronal plane of participant 7, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	293
Figure 9.32 Spinal column in the sagittal plane of participant 7, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	293
Figure 9.33 Spinal parameters in the transverse plane of participant 7, where a) and b) the results from a postural measurement system during the assessment and casting. ....	295
Figure 9.34 Spinal parameters in the trunk asymmetry of participant 7, where a) and b) the results from a postural measurement system during the assessment and casting. ....	296
Figure 9.35 Forces applied to participant 7's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	297
Figure 9.36 Spinal column in the coronal plane of participant 8, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	299
Figure 9.37 Spinal column in the sagittal plane of participant 8, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	300
Figure 9.38 Spinal parameters in the transverse plane of participant 8, where a) and b) the results from a postural measurement system during the assessment and casting. ....	302
Figure 9.39 Spinal parameters in the trunk asymmetry of participant 8, where a) and b) the results from a postural measurement system during the assessment and casting. ....	303
Figure 9.40 Forces applied to participant 8's trunk in the transverse plane. ....	304

Figure 9.41 Spinal column in the coronal plane of participant 9, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	306
Figure 9.42 Spinal column in the sagittal plane of participant 9, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	307
Figure 9.43 Spinal parameters in the transverse plane of participant 9, where a) and b) the results from a postural measurement system during the assessment and casting. ....	309
Figure 9.44 Spinal parameters in the trunk asymmetry of participant 9, where a) and b) the results from a postural measurement system during the assessment and casting. ....	310
Figure 9.45 Forces applied to participant 9's trunk in a) the transverse, b) coronal, and c) sagittal planes. ....	311
Figure 9.46 Spinal column in the coronal plane of participant 10, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	313
Figure 9.47 Spinal column in the sagittal plane of participant 10, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting. ....	314
Figure 9.48 Spinal parameters in the transverse plane of participant 10, where a) and b) the results from a postural measurement system during the assessment and casting. ....	316
Figure 9.49 Spinal parameters in the trunk asymmetry of participant 10, where a) and b) the results from a postural measurement system during the assessment and casting. ....	317
Figure 9.50 Forces applied to participant 10's trunk in the transverse plane. ....	318
Figure 10.1 The line graph of the coronal spinal angles between radiographs (Black line) and the postural measurement system in assessment (Red line), where a) CUTA, b) CLTA, c) CULA, and d) CLLA. ....	329
Figure 10.2 The scatter plot of the CSA between radiographs and the postural measurement system in assessment, a) CUTA, b) CLTA, c) CULA, d) CLLA, and e)	

all CSA. The X-axis is the data from radiographs, and the Y-axis is the data from a postural measurement system in assessment.....	331
Figure 10.3 The line graph of the CSA between the postural measurement system in assessment (Red line) and optimal correction (Green line), a) CUTA, b) CLTA, c) CULA, and d) CLLA. ....	333
Figure 10.4 The line graph of the CSA between the postural measurement system in assessment (Red line) and casting (Blue line), a) CUTA, b) CLTA, c) CULA, and d) CLLA. ....	334
Figure 10.5 The line graph of the CSA between the postural measurement system in optimal correction (Green line) and casting (Blue line), a) CUTA, b) CLTA, c) CULA, and d) CLLA. ....	336
Figure 10.6 The line graph of the SSA between radiographs (Black line) and the postural measurement system for assessment (Red line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.....	338
Figure 10.7 The scatter plot of the SSA between radiographs and the postural measurement system in assessment, a) SUTA, b) SLTA, c) SULA, d) SLLA, and e) all SSA. The X-axis is the data from radiographs, and the Y-axis is the data from the postural measurement for assessment. ....	340
Figure 10.8 The line graph of the SSA between the postural measurement system in assessment (Red line) and optimal correction (Green line), a) SUTA, b) SLTA, c) SULA, and d) SLLA. ....	342
Figure 10.9 The line graph of the SSA between the postural measurement system in assessment (Red line) and casting (Blue line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.....	344
Figure 10.10 The line graph of the SSA between the postural measurement system in optimal correction (Green line) and casting (Blue line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.....	345
Figure 10.11 The line graph of the 3DSA between the postural measurement system in assessment (Red line) and optimal correction (Green line), a) 3D-UTA, b) 3D-LTA, c) 3D-ULA, and d) 3D-LLA. ....	348

Figure 10.12 The line graph of the 3DSA between the postural measurement system in assessment (Red line) and casting (Blue line), a) 3D-UTA, b) 3D-LTA, c) 3D-ULA, and d) 3D-LLA.....	349
Figure 10.13 The line graph of the 3DSA between the postural measurement system in optimal correction (Green line) and casting (Blue line), a) 3D-UTA, b) 3D-LTA, c) 3D-ULA, and d) 3D-LLA. ....	351
Figure 10.14 The line graph of the trunk balance between radiographs (Black line) and the postural measurement system in assessment (Red line).....	353
Figure 10.15 The scatter plot of the trunk balance between radiographs and the postural measurement in assessment. The X-axis is the data from radiographs, and the Y-axis is the data from the postural measurement system in assessment. ....	354
Figure 10.16 The line graph of the trunk balance between the postural measurement system in assessment (Red line) and optimal correction (Green line). ....	355
Figure 10.17 The line graph of the trunk balance between the postural measurement system in assessment (Red line) and casting (Blue line). ....	356
Figure 10.18 The line graph of the trunk balance between the postural measurement system in optimal correction (Green line) and casting (Blue line). ....	357

## List of Tables

Table 3.1 Summary of the intra- and inter-rater reliability of the ultrasound system and the correlation between ultrasound and radiograph results from the relevant literature. ....	40
Table 3.2 Summary of the Meta-analysis of correlation of thoracic coronal Cobb angle, lumbar coronal Cobb angle, thoracic kyphosis, and lumbar lordosis for ST system (Navarro, Candotti and Rosa, 2019). ....	44
Table 3.3 Summary of the correlation and difference distance error between radiographs and motion capture systems from the relevant literature. ....	57
Table 3.4 Summary of the accuracy of low-cost motion capture systems.....	65
Table 3.5 The possible errors of a motion capture system that is still practical for a maximum of 5° of error. ....	71
Table 5.1 Calibration result of the load cell in conditions 1 to 8 (Increasing mass condition). ....	114
Table 5.2 Calibration result of the load cell in conditions 9 to 16 (Decreasing mass condition). ....	119
Table 5.3 Comparison of the difference in voltage change at each testing point between increasing and decreasing mass conditions. ....	120
Table 5.4 Simple linear regression results and RMSE of voltage and force for another 5 load cells.....	124
Table 6.1 RMSE comparison between Reference 1 and the system in Experiment 1 .....	143
Table 6.2 Root-mean-square error comparison between Reference 2 and the system in Experiment 1 .....	145
Table 6.3 Simple linear regression formula used in Experiment 2. ....	154
Table 6.4 Formula of SLR with multiple conditions used in Experiment 2. ....	154
Table 6.5 MLR formula used in Experiment 2. ....	155
Table 6.6 The results of RMSEs before and after applying SLR, SLR with multiple conditions, and MLR in Experiment 2 (n = 125).....	156
Table 6.7 RMSE of the volume experiment in Experiment 3 compared before and after MLR.....	165

Table 6.8 The RMSE of the surface experiment in Experiment 3 compared before and after MLR.....	166
Table 7.1 The camera calibration results of camera 1 with different image conditions. Condition 1 is when the images are taken by servo motors, and condition 2 is by the user. ....	187
Table 7.2 The mean and SD of the difference distance in X, Y, Z, and Euclidean distance for Experiment 4.2. ....	189
Table 7.3 The mean and SD of the difference distance in X, Y, Z, and Euclidean distance for Experiment 4.3. ....	190
Table 7.4 The RMSE in X, Y, Z, and Euclidean distance for Experiment 4.5. ....	195
Table 7.5 The absolute maximum, minimum, and mean errors with SD by comparing the Euclidian distance between a low-cost postural measurement system, a high-cost motion capture system, and a 3-dimensional scanning system.....	197
Table 7.6 The absolute maximum, minimum, mean, and RMSE for Experiments 4.7 and 4.8. ....	200
Table 7.7 The absolute maximum, minimum, mean, and RMSE for Experiments 4.9. ....	202
Table 7.8 The deviation angle when the RMSE of the postural measurement system was 2.42 mm. ....	205
Table 9.1 Participant characteristics .....	248
Table 9.2 Spinal angles in 3 dimensions of non-scoliosis persons proposed by Jang (n = 20) .....	250
Table 9.3 Results of spinal parameters in the sagittal and coronal planes from an X-ray and a low-cost postural measurement system for participant 1. ....	252
Table 9.4 Results of a low-cost postural measurement system for participant 1 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	253
Table 9.5 Results of the forces that were applied to participant 1's trunk. ....	257
Table 9.6 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 2.....	259
Table 9.7 Results of a low-cost postural measurement system for participant 2 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	260



Table 9.8 Result of the forces that were applied to participant 2's trunk.....	264
Table 9.9 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 3.....	266
Table 9.10 Results of a low-cost postural measurement system for participant 3 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	267
Table 9.11 Result of the forces that were applied to participant 3's trunk.....	270
Table 9.12 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 4.....	272
Table 9.13 Results of a low-cost postural measurement system for participant 4 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	274
Table 9.14 Result of the forces that were applied to participant 4's trunk.....	277
Table 9.15 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 5.....	279
Table 9.16 Results of a low-cost postural measurement system for participant 5 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	281
Table 9.17 Result of the forces that were applied to participant 5's trunk.....	284
Table 9.18 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 6.....	286
Table 9.19 Results of a low-cost postural measurement system for participant 6 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	288
Table 9.20 Result of the forces that were applied to participant 6's trunk.....	291
Table 9.21 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 7.....	293
Table 9.22 Results of a low-cost postural measurement system for participant 7 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	294
Table 9.23 Result of the forces that were applied to participant 7's trunk.....	298

Table 9.24 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 8.....	300
Table 9.25 Results of a low-cost postural measurement system for participant 8 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	301
Table 9.26 Result of the forces that were applied to participant 8's trunk.....	305
Table 9.27 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 9.....	307
Table 9.28 Results of a low-cost postural measurement system for participant 9 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	308
Table 9.29 Result of the forces that were applied to participant 9's trunk.....	311
Table 9.30 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 10.....	314
Table 9.31 Results of a low-cost postural measurement system for participant 10 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.....	315
Table 9.32 Result of the forces that were applied to participant 10's trunk.....	318
Table 10.1 Mean and SD of CSA from radiographs and the low-cost postural measurement system in assessment, optimal correction, and casting.....	326
Table 10.2 Reduction percentage of the apical translation between assessment, optimal correction, and casting process. ....	327
Table 10.3 Comparison the reduction of the apical translation between assessment VS optimal correction, and assessment VS casting, using Paired sample t-test. ....	328
Table 10.4 Pearson correlation coefficient between CSA from radiographs and the postural measurement in assessment.....	331
Table 10.5 Comparison between CSA from the postural measurement in assessment and optimal correction, using Paired sample t-test. ....	333
Table 10.6 Comparison between CSA from the postural measurement in assessment and casting, using Paired sample t-test. ....	334
Table 10.7 Equivalence testing between CSA from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.....	336

Table 10.8 Mean and SD of SSA from radiographs and the postural measurement in assessment, optimal correction, and casting. ....	337
Table 10.9 Pearson correlation coefficient between SSA from radiographs and the postural measurement in assessment.....	340
Table 10.10 Equivalence testing between SSA from the postural measurement in assessment and optimal correction, using Paired t-test for equivalence. ....	343
Table 10.11 Equivalence testing between SSA from the postural measurement in assessment and casting, using Paired t-test for equivalence. ....	344
Table 10.12 Equivalence testing between SSA from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.....	346
Table 10.13 Mean and standard deviation of 3DSA from the postural measurement in assessment, optimal correction, and casting. ....	347
Table 10.14 Comparison between 3DSA from the postural measurement in assessment and optimal correction, using Paired sample t-test. ....	348
Table 10.15 Comparison between 3DSA from the postural measurement in assessment and casting, using Paired sample t-test. ....	350
Table 10.16 Equivalence testing between 3DSA from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.....	352
Table 10.17 Mean and SD of trunk balance from radiographs and the postural measurement in assessment, optimal correction, and casting. ....	353
Table 10.18 Pearson correlation coefficient between trunk balance from radiographs and the postural measurement in assessment. ....	354
Table 10.19 Comparison between trunk balance from the postural measurement in assessment and optimal correction, using Paired sample t-test.....	355
Table 10.20 Comparison between trunk balance from the postural measurement in assessment and casting, using Paired sample t-test.....	357
Table 10.21 Equivalence testing between trunk balance from the postural measurement in optimal correction and casting, using Paired t-test for equivalence. ....	358
Table 10.22 The results of the POTSI in 6 parameters compared between the assessment and casting. ....	359

Table 10.23 The summary results of the POTSI compared between the assessment and casting.....	359
Table 10.24 Total angles of horizontal trunk rotation of individual participants during the assessment and casting process .....	361
Table 10.25 Mean and standard deviation of forces in various areas to correct scoliosis during the optimal correction (n=10).....	362
Table 10.26 Summary of results from the clinical experiment.....	364
Table 11.1 CSA of non-scoliosis persons proposed by Jang (n = 20) .....	374
Table 11.2 SSA of non-scoliosis persons proposed by Jang (n = 20).....	379
Table 11.3 Three-dimensional spinal angles of non-scoliosis persons proposed by Jang (n = 20).....	383

## List of Abbreviations

3D	3-dimensional, 3 dimensions
3DSA	3-dimensional-spinal angle
3DLLA	3-dimensional-lower-lumbar angle
3DLTA	3-dimensional-lower-thoracic angle
3DULA	3-dimensional-upper-lumbar angle
3DUTA	3-dimensional-upper-thoracic angle
AAOP	American Academy of Orthotists and Prosthetists
AIS	Adolescent idiopathic scoliosis
C	Cervical
CLLA	Coronal-lower-lumbar angle
CLTA	Coronal-lower-thoracic angle
CSA	Coronal spinal angles
CT scan	Computerised tomography scan
CULA	Coronal-upper-lumbar angle
CUS	Clinical ultrasound system
CUTA	Coronal-upper-thoracic angle
CVA	Coronal vertical axis
CVSL or CSL	Central-vertical-sacral line
DCM	Deviation colour map
EDF	Elongation, derotation, and flexion
GM	Gluteus medius
ICC	Intraclass correlation coefficient
IC	Initial contact
ISIS	Integrated Shape Imaging System
IS	Idiopathic scoliosis
JIS	Juvenile idiopathic scoliosis
JPO	Journal of Prosthetics & Orthotics
Kg	Kilograms
L	Lumbar
LT	Lower thoracic

LL	Lower lumbar
MCIC	Minimal clinically important change
mm	millimetre
MLR	Multiple linear regression
MRI	Magnetic resonance imaging
N	Newtons
n/a	Not applicable
OM	Outcome measure, outcome measurement
PA	Posterior–Anterior
PF	Peak flexion
QSYS	Quantec spinal image system
RGB	Red-green-blue
RMSE	Root-mean-square error
ROM	Range of motion
rpm	Rounds per minute
r-value	Pearson correlation coefficient value
SAMSA	Standard Assessment of Motion System Accuracy
SD	Standard deviation
SLLA	Sagittal-lower-lumbar angle
SLR	Simple linear regression
SLTA	Sagittal-lower-thoracic angle
SO	Spinal orthosis, spinal orthotics
SOSORT	International Society on Scoliosis Orthopaedic and Rehabilitation Treatment
SPA	Spinous process angle
SRS	Scoliosis Research Society
SSA	Sagittal spinal angle
ST	Surface Topography
SULA	Sagittal-upper-lumbar angle
SUTA	Sagittal-upper-thoracic angle
SVA	Sagittal vertical axis
T	Thoracic

TBCM	Total body centre of mass
TLSO	Thoraco-lumbo- sacral orthosis
TPA	Transverse process angle
US	Ultrasound system
UL	Upper lumbar
UT	Upper thoracic
VS	versus

**1. Chapter 1 Introduction**



## 1.1. Introduction to scoliosis

Idiopathic scoliosis is a common pathology of the spine that causes deformities in three dimensions (3D): lateral deviation in the coronal plane, increasing or decreasing kyphotic or lordotic curvature in the sagittal plane, and horizontal rotation of vertebrae in the transverse plane (Stokes, Bigalow and Moreland, 1987, Hattori *et al.*, 2011, Kotwicki, 2008). Scoliosis is diagnosed when the lateral curvature exceeds 10° of the Cobb angle (Janicki and Alman, 2007b). Idiopathic scoliosis is more common in girls than in boys (Tay, Kornberg and Cook, 2007).

Regarding the aetiology of scoliosis, there are many certain factors, such as, genetic, muscles imbalance, arthritis, and neurological impairment. However, the most common cause is still unknown called Idiopathic scoliosis (or IS), representing at 80% (Negrini *et al.*, 2018). Scoliosis is classified according to age range, with infantile scoliosis occurring between the ages of 0 and 3 years, juvenile scoliosis occurring between the ages of 4 and 10 years, and adolescent scoliosis occurring between the ages of 10 and 18 years.

The prevalence of adolescent idiopathic scoliosis (AIS) ranges from 0.93% to 12% (Burwell *et al.*, 1983, Brooks *et al.*, 1975, Wong *et al.*, 2005, Grivas *et al.*, 2006a, Dickson, 1983, Soucacos *et al.*, 1997, Pin *et al.*, 1985, Huang, 1997, Nissinen *et al.*, 1993, Willner and Udén, 1982, Laulund, Søjbjerg and Hørlyck, 1982, Morais, Bernier and Turcotte, 1985, Yawn and Yawn, 2001, Gore *et al.*, 1981, Rogala, Drummond and Gurr, 1978, Shands and Eisberg, 1955, Koukourakis *et al.*, 1997, Grivas *et al.*, 2007). However, most studies report a prevalence of 2% to 3%. Surprisingly, the incidence varies and depends on latitude (Grivas *et al.*, 2006a, Grivas *et al.*, 2006b).

Three primary characteristics are commonly seen in AIS patients: 1) vertebral rotation in the horizontal plane, 2) lateral displacement of the spinal column in the coronal plane, and 3) thoracic hypokyphosis in the sagittal plane. Concerning the first characteristic, the spinal column and each vertebra in the segment is rotated towards the convex side of the spinal curve, resulting in prominence (Adam, Askin and Percy, 2008, Stokes, 1989, Jaremko *et al.*, 2002, Parent, Newton and Wenger, 2005, Gum *et al.*, 2007, Kotwicki and Cheneau, 2008). If the rotation and prominence occur at the rib or thoracic area, it is referred to as a rib hump, which can present an asymmetrical

shape of the trunk and rib cage. Additionally, if the rotation and prominence occur at the lumbar area, it is referred to as lumbar prominence.

According to the second characteristic, the spinal column is laterally displaced from the spinal midline to the convex side of the curve, called the coronal curve, with the most displacement presenting at the curve's apex, called apical translation. Regarding the final characteristic, kyphosis is a typical curve at the thoracic of the human spine in sagittal plane, but many scoliosis patients have a decrease in this angle, known as thoracic hypokyphosis or a flattening of thoracic curve in the sagittal alignment of spinal column (Dickson *et al.*, 1984).

There are many studies classify the location and pattern of the spinal curve. For instance, the side of spinal curve bases on the convexity of curve, therefore, if the convexity is on the right side, the spinal curve is the right curve. Furthermore, the spinal location can be classified based on the location of the apical vertebra, such as, between C7 and T1 called cervicothoracic curve, between T2 and T11 called thoracic (T) curve, between T12 and L1 called thoracolumbar (Maclean *et al.*) curve, and between L2 and L4 called lumbar (L) curve (Kotwicki, 2008). Then, the spinal curve can also be categorized by the number of curves, such as, single, double, and triple curve. In addition, there are other well-known spinal curve based on the different pattern of spinal column, such as, King classification (King *et al.*, 1983) and Lenke classification (Lenke *et al.*, 2001a).

The magnitude of the scoliotic curve or curve progression show a high risk progression during the growing phase with skeleton immaturity, especially in adolescence (Weinstein and Ponseti, 1983, Dickson and Weinstein, 1999, Richards *et al.*, 2005), and girls are more likely to have higher curve progression than boys (Negrini *et al.*, 2018). If the magnitude of the coronal curve is between  $10^{\circ}$  and  $20^{\circ}$ , the curve progression of both groups is similar, and the ratio of girls to boys is 1.3:1. Interestingly, if the Cobb angle is between  $20^{\circ}$  and  $30^{\circ}$ , the girl-boy ratio increases to 5.4:1, and the difference is even greater if the curve magnitude is greater than  $30^{\circ}$ , representing a 7:1 ratio (Parent, Newton and Wenger, 2005, Lonstein, 2006).

Other factors contributing to the high risk of curve progression include the severity of the coronal curve magnitude and the Risser sign grading. The high-risk curve progression occurs when the Cobb angle is between  $20^{\circ}$  and  $29^{\circ}$  with a Risser sign

grading of 0 to 1 (Richards *et al.*, 2005). It should be noted that the Risser sign indicates skeleton maturity. This sign illustrates the ossification of the iliac apophysis on the pelvis. A standard tool for evaluating this grading result is an X-ray image.

Finally, if the magnitude of the curve is high and no proper treatment is provided, some patients may experience health problems. According to the literature (Lonstein, 2006, Weinstein *et al.*, 2003, Negrini *et al.*, 2006), if the curve angle is between 30° and 50°, it can be considered a critical threshold with a high risk of health problems such as cosmetic appearance, decreased self-esteem, decreased quality of life, back pain, and functional limitations.

## **1.2. Introduction to scoliotic treatment**

In the general treatment of the AIS patient (Richards *et al.*, 2005), there are two main types of treatment which are conservative treatment and surgical treatment and the choice between them is based on the severity of deformity. Many studies use conservative treatment including observation with activity modification, physical therapy, exercise, acupuncture, and orthosis. The conservative treatment by observation with activity modification is recommended for patient who has the coronal curve smaller than 25°. Orthotic treatment is prescribed for the coronal curve between 25° and 40°. However, the surgical treatment is recommended if the severity of deformity is high (more than 40° of coronal curve) and to prevent a secondary complication, such as pulmonary dysfunction.

## **1.3. Introduction to orthotic treatment**

Focusing on orthotic treatment, the spinal orthosis (SO) is the most commonly prescribed for AIS treatment (Richards *et al.*, 2005, Danielsson *et al.*, 2007, Yrjönen *et al.*, 2007, Katz *et al.*, 2010, Schiller, Thakur and Ebersson, 2010) if the coronal curve is between 25° and 40° with a Risser sign grading of 0 to 3 (Negrini *et al.*, 2015), and the goal of using SO is to delay or prevent curve progression and reduce surgical

treatment (Richards *et al.*, 2005). Many different types of spinal orthotic designs and categories are available today based on time use, biomechanical principles, and fabrication materials, such as the Milwaukee brace (Lonstein and Winter, 1994, Keiser and Shufflebarger, 1976, Carr *et al.*, 1980, Fisher, Rapp and Emkes, 1987), Boston brace (Emans *et al.*, 1986, Katz *et al.*, 1997, Vijvermans, Fabry and Nijs, 2004, Wiley *et al.*, 2000, Yrjönen *et al.*, 2007, Xu *et al.*, 2019, Grivas *et al.*, 2003), Cheneau brace (Pham *et al.*, 2007, Zabrowska-Sapeta *et al.*, 2010, Zaborowska-Sapeta *et al.*, 2011b, Fang *et al.*, 2015b, Pasquini *et al.*, 2016), Charleston bending brace (Price *et al.*, 1990, Katz *et al.*, 1997, Lee *et al.*, 2012), Lyon brace (Aulisa *et al.*, 2015), and SpineCor dynamic elastic brace (Coillard *et al.*, 2007, Coillard, Circo and Rivard, 2014). Essentially, all of them are based on the biomechanical force correction system known as the three-point force system, with the goal of re-aligning the spine in the neutral or optimal alignment/position.

Regarding the orthotic procedure to treat AIS patients, the rehabilitation doctor or orthopaedic doctor performs the assessment and evaluation, such as subjective assessment, physical assessment (range of motion, manual muscle testing, special clinical testing etc.), and radiographic (X-ray image) evaluation in one or two orthogonal planes. After that, the doctor makes the treatment plan and orthotic prescription and refers a patient to an orthotist to make a device. Then, a professional orthotist performs the subjective and objective assessment in detail before designing the orthosis to match the individual patient's condition. The orthotist performs a casting (a process of capturing the shape of a human body) using plaster or synthetic bandages. The patient stands or slightly sits in place, and 2 – 3 orthotists apply the corrective force on the patient's trunk by hand to re-align the spine in neutral or optimal alignment. Patients and orthotists remain in the casting position for 10 – 15 minutes or until the bandage is completely set. The important landmarks and reference lines must be taken on the cast before removing the negative cast (the negative cast is the bandage after setting). Then, the negative cast is removed from the patient's trunk, and cleaning must be done immediately.

The negative cast is then evaluated on its quality, such as the overall shape, the alignment, and the strength of the negative cast. However, re-casting may be necessary if the quality of the negative cast is not appropriate. After adjusting and preparing the

negative cast, the cast is filled to create a positive cast (the positive cast is a model of a human's body) using plaster or synthetic materials (Polyurethane foam or rigid foam). Rectification or modification process is performed after positive cast setting (Rectification or modification process is a process to modify the positive cast to get the optimal shape). After positive cast modification, the orthosis's main structure is fabricated using thermoplastic, such as polypropylene (PP), polyethene (PE), and copolymer materials. Then, the trimline (the edge of the device) is determined and cut by following the orthotic prescription. The padding (soft materials posted at the corrective area to apply the corrective force and provide comfort), straps, and fitting equipment are prepared before fitting.

In the fitting process, the patient wears the spinal orthosis on their trunk, and the temporal strap is attached to the orthosis to provide hydrostatic compression on the spine and suspend the orthosis. The paddings also attach inside the orthosis. Moreover, increasing the padding thickness is commonly required to increase the corrective force at that area. The trimline is then adjusted to provide the best contour and to eliminate the body movement during standing, walking and sitting.

After the orthotist is satisfied with the fitting outcome based on their experience, the permanent straps and paddings are attached to the orthosis. Then, the patient is referred back to a doctor for final checking as well as referring the patient to take an X-ray immediately to check the corrected position. The treatment outcome from the orthosis is evaluated by X-ray images taken with and without the orthosis. The doctor makes an appointment with the patient to follow up on the treatment. If some orthotic adjustment is needed, the doctor will refer the patient to adjust the device with the orthotist. Finally, an X-ray must be taken every follow-up until the patient stops wearing the orthosis.

#### **1.4. Introduction to spinal outcome measurement**

Evaluating radiographic or X-ray images is the standard tool for measuring and assessing spinal parameters in the coronal and sagittal planes (Kotwicki, 2008). Before the X-ray, the patient is asked to change clothes to the in-patient hospital uniform and

take accessories off because it can interfere with the X-ray image. The patient is then placed in a standing position in front of the X-ray machine, and the radiologist operates the machine to obtain the image. All images are saved in the medical record for the doctor and other healthcare professionals to see. Cobb's method (Cobb, 1948b) is a standard method for determining and evaluating spinal parameters on an X-ray image in the coronal and sagittal planes. Many parameters, including the location of the curve's apex, the distance between the body's midline and the apex (apical translation), the upper and lower end vertebrae of the curve, the magnitude or degrees of the curve, trunk balance or coronal decompensation, the type of curve, the grade of the Risser sign, the status of triradiate cartilage, and the indirect measurement of vertebral rotation, will be measured and evaluated in the coronal plane. Thoracic kyphosis and lumbar lordosis angles will be measured in the sagittal plane.

### **1.5. Introduction to the need for alternative technology to assess scoliosis and scoliotic treatment**

As can be seen from the introduction above, the historical approach to the assessment and treatment of scoliosis has been heavily influenced by the use of planar X-rays. These planar X-rays first showed and defined scoliosis, and as a result of their adoption in medical practice, they have also been adopted in orthotic practice. However, scoliosis is essentially a three-dimensional deformity, and orthotic treatment applies a three-dimensional force system to attempt to control and correct this three-dimensional deformity. Outcome measurements from standard X-ray images can only describe a two-dimensional deformity, which is insufficient to represent the result of orthotic treatment. Furthermore, repeated X-ray exposure throughout the treatment and follow-up period may result in health complications in the future.

To address these issues and improve the quality of outcome measurement in orthotics, many research groups are developing and experimenting with other technologies, such as ultrasound images (Li *et al.*, 2010, Edmond *et al.*, 2015, Edmond *et al.*, 2017, Wong *et al.*, 2019), surface topography (Komeili *et al.*, 2014, Komeili *et al.*, 2015a, Komeili *et al.*, 2015b), and most recently the use of motion capture system,

to aid assessment and treatment. The goal of employing these alternative technologies is to reduce the number of X-rays required and to give a truer representation of the scoliotic deformity.

The casting process has been heavily influenced by this planar approach to treatment with a three-point force system applied in the coronal plane to correct coronal deformity and a second one in the sagittal plane to correct sagittal deformity. What is really needed is to treat the spine as a three-dimensional entity and apply a three-dimensional force system which corrects the three-dimensional deformity.

A number of clinical researchers have attempted to improve the casting process by using a casting bar, apparatus, or frame. Examples include the Risser table (Ballhause *et al.*, 2019), Cotrel's frame (Cotrel and Morel, 1964, Mahajan *et al.*, 2020), a custom Providence brace standing frame (Edmond *et al.*, 2015), apparatus for forming a scoliosis brace (McCoy and Barry, 1996), and an infant scoliosis casting frame (Sanders and D'Astous, 2009). In addition to the casting frame, ultrasound images have been utilised to evaluate the spinal result in the coronal plane following casting (Edmond *et al.*, 2015, Edmond *et al.*, 2017). However, this technology can only measure the outcome in one plane after casting and not in real-time. Lastly, a technology utilising ultrasound images to aid in locating the optimal position of corrective padding and verifying the spinal outcome in the coronal plane is introduced to facilitate the fitting process (Li *et al.*, 2010, Li *et al.*, 2012). However, this technology can only measure the result on one plane and not in real-time.

## **1.6. General outline of this thesis**

This study aims to quantify spinal parameters using a motion capture or postural measurement system. Previous research by Jang (Jang, 2018) developed and validated a method for using a motion capture system to quantify the spinal parameters during the assessment process. However, that study used a high-cost system that is not generally available in orthotic clinical settings and quantified the spine using typical clinical planar measures. Moreover, Jang's study focused only on the assessment process and did not use the system to assist during the casting process.

The current study describes the development of a low-cost system to assist orthotic practitioners during the assessment, optimal correction and casting process for AIS treatment. The system aims to capture the three-dimensional shape of the spine and the three-dimensional force system being applied to it. Our contention is that if we can assess and treat scoliosis in three dimensions rather than two independent planes and in an inexpensive way in the clinic, then in the long run, we will better understand scoliosis and how to control it than if we continue to adopt the two-plane approach. There were three main parts involved in this development.

The first part was a mechanical frame to assist during the optimal correction and casting process called the "**Scoliosis casting apparatus**". This apparatus aimed to apply the corrective, counter, and stabilising forces to re-align the spine during the optimal correction and casting process as an alternative to hand pressure from multiple orthotists.

The second part was a "**Force-measuring system**" using load cells attached at the end of manipulator rods bolted to the scoliosis casting apparatus, which contacted and compressed the spine to give the orthotic effect. The forces measured along each rod were used for reporting the result during the optimal correction and casting process.

The final part was a "**Low-cost postural measurement system**" used to analyse and represent the spinal outcome during the assessment, optimal correction and casting process.

The typical casting process generally requires a team of two to three practitioners. Practitioners must apply force with their hands on the corrective and counterforce areas while also controlling the patient's posture and alignment. They must judge the correction being achieved by eye and as observed from their position.

In contrast, the apparatus proposed to be developed in this thesis requires only one practitioner during the process for establishing optimal correction and the assistance of a second person during the casting process. The scoliosis casting apparatus can adjust and maintain the forces throughout the process. Furthermore, the load cell measures the magnitude of force applied to each area, and a low-cost postural measurement system calculates spinal parameters and force direction to realign the spine and can provide this data as real-time feedback to the casting team.



## **2. Chapter 2 Aims, Objectives, and Thesis Outlines**

## **2.1. Introduction**

The introduction to this thesis in the previous chapter described the primary research gap of scoliosis research, which in our opinion, based on the literature review presented in Chapter 3, is our collective inability to consider the problem of scoliosis in three dimensions. This research gap is explored and expanded in the literature review (Chapter 3) and was the main driving force behind the development of this PhD project and thesis. This chapter summarises the thesis by presenting the research gap and introducing the main content of the thesis, which includes the development of a low-cost postural measurement system, validity testing, and quantifying the spinal parameters in scoliosis treatment during the assessment, optimal correction, and casting process. Finally, this chapter states the research project's goals and the objectives of the study.

## **2.2. Summary of scoliosis research gap**

- There are few research studies of scoliosis that quantify spinal curvature in three dimensions including using motion capture.
- Currently, there is insufficient evidence to support the use of motion capture to quantify spinal parameters in scoliosis applications.
- Only one recent PhD study by Jang, used a motion capture system to quantify spinal parameters by dividing the spinal column into several sections, including lower cervical, upper thoracic, lower thoracic, upper lumbar, and lower lumbar. This study evaluated scoliosis in three dimensions during the assessment process but used a high-cost motion capture system.
- No low-cost motion capture system is currently available for quantifying spinal parameters in scoliosis applications.
- While the Raspberry Pi has been widely used in computer vision applications, there has been no development of a low-cost motion capture system that uses multiple Raspberry Pis and cameras to estimate the location of markers used in human motion analysis.

- Casting frames for scoliosis have been used in clinical practice but this is not common. Casting frames add expense, can be complicated to use, and lack evidence to support the effectiveness of the frame. Moreover, all currently available frames apply forces to correct the deformity in one or two planes only, and not in 3 dimensions.
- There is still a lack of knowledge about the magnitude of forces applied during casting. Previous research mostly studied the force or pressure inside the spinal orthosis, not that developed during the casting process.

To overcome the research gaps, a low-cost motion capture system or a low-cost postural measurement system that can quantify spinal parameters in scoliosis applications must be developed. Furthermore, the system's accuracy should be validated compared to the standard high-cost system, and the error of the developed system should be within the acceptable range in routine clinical practice. A scoliosis casting apparatus that the user can adjust to apply the 3-dimensional forces to correct the spinal deformity must be developed. Furthermore, a force-measuring system that can measure the magnitude of forces applied during optimal correction and casting process must be developed. Finally, the developed system should be able to quantify spinal parameters during the assessment, optimal correction, and casting process to evaluate the change of spinal parameters while applying the forces to re-align the deformity in 3 dimensions.

### **2.3. Aim of the Thesis**

This study aims to develop a low-cost postural measurement system, a scoliosis casting apparatus, and a force-measuring system to assist practitioners during the assessment, optimal correction, and casting process of spinal orthosis for adolescent idiopathic scoliosis (AIS) treatment.

## 2.4. The objectives of the Thesis

The objectives of this study are as follows.

Scoliosis casting apparatus (Chapter 4):

- To develop a practical scoliosis casting apparatus to assist the clinicians during optimal correction and casting process in AIS treatment. The apparatus should be able to be manually adjusted by the clinicians to apply the 3-dimensional force correction to correct the 3-dimensional spinal deformity of AIS patients and should be inexpensive.

Force-measuring system (Chapter 5):

- To develop a force-measuring system to measure the magnitude of forces applied during optimal correction and casting process in AIS treatment. The system should be able to measure the force magnitudes and report the results on the graphical user interface (GUI) on the computer screen.

Low-cost postural measurement system (Chapters 6 and 7):

- To develop a low-cost postural measurement system using Raspberry Pis and their associated cameras to calculate marker positions in 3-dimensional space.
- To study the validity of a low-cost postural measurement system compared to the actual marker positions and high-cost motion capture system. The system's error should be within the acceptable range in routine clinical practice (approximately 5 mm).

Clinical experiment in AIS patients (Chapters 8 to 11):

- To demonstrate the feasibility of using a low-cost postural measurement system, scoliosis casting apparatus, and force-measuring system in clinical practice during the clinical experiment with AIS patients.
- To verify whether a low-cost postural measurement system can quantify the spinal parameters in coronal and sagittal planes and to study the validity of the

spinal parameters using a low-cost postural measurement system compared to the radiographic evaluation as a goal standard.

- To examine whether a low-cost postural measurement system can quantify the angles of the spinal column in 3 dimensions during the assessment, optimal correction, and casting process.
- To examine whether a low-cost postural measurement system can quantify the angle of trunk rotation in the horizontal plane during the assessment and casting process.
- To examine whether a low-cost postural measurement system can quantify the trunk asymmetry in the coronal plane during the assessment and casting process.
- To examine whether the scoliosis casting apparatus can be adjusted to correct the spinal deformity in 3 dimensions during optimal correction and casting process.
- To examine whether the force-measuring system can measure the force magnitude and whether the postural measurement system can illustrate the force directions when the forces were applied to correct the spinal deformity in 3 dimensions.

## **2.5. Thesis outline and Scope**

The following chapter, **Chapter 3**, presents an analysis and evaluation of the relevant literature concerning the following six aspects: 1) General information regarding scoliosis, 2) the design of the spinal orthotic and its efficacy, 3) the radiographic outcome measurement in AIS patients, 4) alternative technologies to measure the outcome of AIS, 5) accuracy of high-cost and low-cost postural measurement system, and 6) casting frame for scoliosis treatment.

**Chapter 4** discusses the development of scoliosis casting apparatus used to assist practitioners during the optimal correction and casting process.

**Chapter 5** details the development of a force-measuring system, including both hardware and software. The chapter describes a method for calibrating the sensors (load cell), as well as how to calculate the result and demonstrates how to collect data during the optimal correction and casting process with scoliosis patients.

**Chapter 6** presents the development history of a low-cost postural measurement system, including both hardware and software. Initial developments begin with developing and experimenting with two and three Raspberry Pis and cameras to estimate multiple marker locations on a flat board with varying depths and assess the accuracy of these methods. The chapter then introduces further development of the system with an experimental result using three Raspberry Pis and cameras perfectly moulded on a camera frame, which is then used to estimate multiple marker locations on a semi-circular shape (or ball shape) and to calculate the accuracy of the system.

**Chapter 7** introduces computer vision technology and software which is used to improve the accuracy of the system. The chapter explains the revised concept of a low-cost postural measurement system based on the OpenCV principles and implemented in Raspberry Pis that communicates wirelessly with a central computer. This chapter starts by developing and experimenting with two Raspberry Pis and cameras to estimate the location of multiple markers on a flat board and calculate the error. The chapter then explains how to estimate a marker location in 3-dimensional space using multiple Raspberry Pis and cameras (8 Raspberry Pis). Finally, this chapter discusses the various experimental tests and their accuracy, including multiple markers on a scoliosis manikin, moving the manikin in different positions and orientations, and then comparing the results between a high-cost motion capture system, a low-cost postural measurement system, and a 3D scanning system.

**Chapter 8** presents the clinical research methodology, clinical outcome measurement, how to calculate each spinal parameter, and how to analyse the data statistically.

**Chapter 9** discusses the clinical experiment conducted with AIS patients using the scoliosis casting apparatus, the force-measuring system, and the low-cost postural measurement system to quantify the spinal parameters during the assessment, optimal correction, and casting process, in three dimensions. This chapter discusses the results from individual participants to demonstrate the feasibility of using the developed system in clinical practice.

**Chapter 10** discusses the data analysis using descriptive and inferential statistics and reports the experiment's outcome.

The final chapter, **Chapter 11**, discusses the study's findings compared to the previous articles. The chapter then discusses the study's limitations, recommendations, and suggestions for future research and concludes the research project.

### **3. Chapter 3 Literature Review**



### **3.1. Introduction**

The previous chapters provided an overview of the thesis, including the introduction and background of scoliosis, research gaps, aims of study, and scope of this thesis. This chapter expands on the existing literature on scoliosis, especially adolescent idiopathic scoliosis (AIS), standard and alternative outcome measurements, the accuracy of high-cost and low-cost motion capture systems, and the frame to assist during casting. This chapter is divided into 7 sections. The first section begins with general scoliosis information. Then, the spinal orthotic treatment and its effectiveness are described in Section 2. Section 3 describes how to measure the outcome on the radiographs or X-ray images. Section 4 discusses alternative technologies for determining the outcome of scoliosis. The accuracy of high-cost and low-cost motion capture systems is discussed in Section 5. Section 6 describes the casting frames to assist practitioners during spinal casting. Section 7 describes the summary of the existing literature on scoliosis and introduces the developing system for this thesis.

### **3.2. Scoliosis**

Scoliosis is a spinal deformity characterised by lateral curvature and vertebral rotation (Janicki and Alman, 2007a), or a three-dimensional deformity (3D), including lateral deviation in the coronal plane, increasing or decreasing kyphotic or lordotic curvatures in the sagittal plane, and horizontal rotation of vertebrae in the transverse plane (Stokes, Bigalow and Moreland, 1987, Kotwicki, 2008, Hattori *et al.*, 2011), as shown in Figure 3.1. When the lateral curvature of the spine is greater than  $10^\circ$  of the coronal Cobb angle, scoliosis is diagnosed (Janicki and Alman, 2007a).

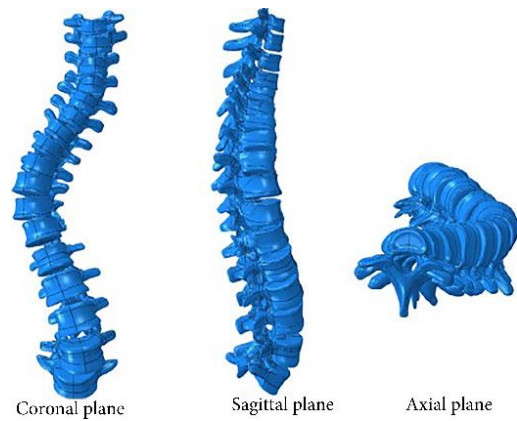


Figure 3.1 Spinal deformity of scoliosis in 3 dimensions (Wang *et al.*, 2014).

Idiopathic scoliosis affects females at a higher rate than males (Tay, Kornberg and Cook, 2007). Scoliosis is caused by a variety of factors, including genetics, muscle imbalance, arthritis, and neurological impairment. However, the most common cause is still unknown (called idiopathic scoliosis or IS), accounting for 80% of all cases (Negrini *et al.*, 2018). Scoliosis is classified based on age, with infantile scoliosis affecting children aged 0 to 3 years (0.5% of IS), juvenile scoliosis affecting children aged 4 to 10 years (10.5% of IS), and adolescent scoliosis affecting children aged 10 to 18 years (89% of IS) (Riseborough and Wynne-Davies, 1973). Adolescent idiopathic scoliosis (AIS) has a prevalence ranging from 0.93% to 12% (Burwell *et al.*, 1983, Brooks *et al.*, 1975, Wong *et al.*, 2005, Grivas *et al.*, 2006a, Dickson, 1983, Soucacos *et al.*, 1997, Pin *et al.*, 1985, Huang, 1997, Nissinen *et al.*, 1993, Willner and Udén, 1982, Laulund, Søjbjerg and Hørlyck, 1982, Morais, Bernier and Turcotte, 1985, Yawn and Yawn, 2001, Gore *et al.*, 1981, Rogala, Drummond and Gurr, 1978, Shands and Eisberg, 1955, Koukourakis *et al.*, 1997, Grivas *et al.*, 2007). However, most studies report a prevalence of 2% to 3%. Interestingly, the incidence may differ by latitude (Grivas *et al.*, 2006a, Grivas *et al.*, 2006b).

AIS patients illustrate 3 primary characteristics: 1) vertebral rotation in the transverse plane, 2) lateral displacement of the spinal column in the coronal plane, and 3) thoracic hypokyphosis in the sagittal plane. The first distinguishing characteristic is an outward rotation of the spinal column and each segmental vertebra towards the convex side of the spinal curve, resulting in prominence (Adam, Askin and Pearcy, 2008, Stokes, 1989, Jaremko *et al.*, 2002, Parent, Newton and Wenger, 2005, Gum *et*

*al.*, 2007, Kotwicky and Cheneau, 2008). If the rotation and prominence occur in the rib or thoracic region, it is known as a rib hump, which can present an asymmetrical shape of the trunk and rib cage. If the rotation and prominence occur in the lumbar region, it is known as a lumbar prominence. The second characteristic is a lateral translation of the spinal column from the midline to the convex side of the curve (the coronal curve), with the most remarkable translation occurring at the curve's apex, called the apical translation. In the final characteristic, many scoliosis patients demonstrate thoracic hypokyphosis or a flattening of the thoracic curve in the sagittal alignment (Dickson *et al.*, 1984).

Many studies have been conducted to classify the location and pattern of the spinal curve. For example, the side of the spinal curve is determined by the convexity of the curve. Thus, if the convexity is on the right side, the spinal curve is the right curve. Furthermore, the location of the apical vertebra can be used to classify the spinal location. The cervicothoracic curve, for example, is located between C7 and T1, the thoracic (T) curve is located between T2 and T11, the thoracolumbar (TL) curve is located between T12 and L1, and the lumbar (L) curve is located between L2 and L4 (Kotwicky, 2008). The curve of the spine can also be classified according to the number of curves present; for example, a single curve, a double curve, or a triple curve. In addition, other well-known spinal curve classifications are based on various patterns, such as the King classification (King *et al.*, 1983) and Lenke classification (Lenke *et al.*, 2001a).

Trunk asymmetry can be seen in scoliosis patients (fig. 3.2), and clinicians commonly evaluate it during physical examination. The patients are asked to stand upright, and the clinicians observe the trunk asymmetry on the posterior side of the patient. The common trunk asymmetry consists of uneven shoulder, uneven axilla, uneven waist, uneven scapula, uneven arm gap, and trunk leaning to the left or right side.

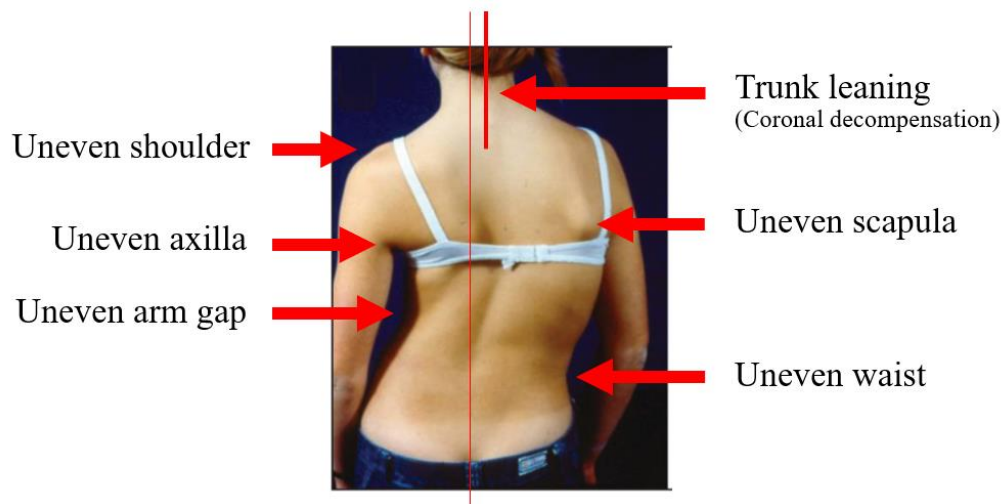


Figure 3.2 Trunk asymmetry observed from the back of a scoliosis patient (Janicki and Alman, 2007a).

The severity of the scoliotic curve, also known as the curve progression, indicates a high risk of progression during the growing of the skeleton, particularly during adolescence (Weinstein and Ponseti, 1983, Dickson and Weinstein, 1999, Richards *et al.*, 2005). Girls are more likely than boys to have a higher curve progression (Negrini *et al.*, 2018). Suppose the magnitude of the coronal curve is between  $10^\circ$  and  $20^\circ$ . In that case, the curve progression of both groups will be the same, and the ratio of girls to boys is 1.3:1. It is interesting to note that the ratio of females to males rises to 5.4:1 when the coronal Cobb angle is between  $20^\circ$  and  $30^\circ$ . The difference becomes even more noticeable when the curve magnitude is greater than  $30^\circ$ , representing a 7:1 ratio (Parent, Newton and Wenger, 2005, Lonstein, 2006).

Risser sign is an important indicator to indicate skeleton maturity status. This sign illustrates the ossification status of the iliac apophysis on the pelvis (Risser, 1958), and X-ray images are commonly used to assess this type of grading outcome. Two of the most critical factors contributing to the high risk of curve progression are the severity of the coronal curve magnitude and the grading of the Risser sign. When the Cobb angle is between  $20^\circ$  and  $29^\circ$ , and the Risser sign grading is between 0 and 1, the curve's progression is considered high risk (Richards *et al.*, 2005). Additionally, some patients may experience health problems if the magnitude of the curve is high and appropriate treatment is not provided. According to the literature, there is an increased risk of health problems like cosmetic appearance, low self-esteem, psychosocial issues

(Payne *et al.*, 1997), low quality of life, back pain, and functional limitations if the curve angle is between 30° and 50° (Lonstein, 2006, Weinstein *et al.*, 2003, Negrini *et al.*, 2006).

Generally, AIS treatment is determined by the severity of the curve magnitude, the patient's age, skeleton maturity status, and the risk of curve progression. The doctor considers the observation for an immutable patient with a coronal Cobb magnitude of 10° to 25° (Janicki and Alman, 2007a). If the magnitude of the curve is between 25° and 45°, an orthosis is prescribed (Janicki and Alman, 2007a). Orthotic treatment aims to prevent curve progression but not permanently improve or correct the deformity (Willers *et al.*, 1993). In immature patients, surgery is considered when the curve is greater than 45° and greater than 50° in mature patients (Janicki and Alman, 2007a). The operation aims to correct the deformity permanently, prevent curve progression, and improve spinal alignment and balance. Furthermore, the exercise has also been introduced for AIS treatment to improve muscle power and flexibility (Romano *et al.*, 2012, Romano *et al.*, 2013, Monticone *et al.*, 2014, Negrini *et al.*, 2014).

Scoliosis deformities can change the pattern while walking or performing trunk motions. During the physical assessment, the clinicians ask the scoliosis patients to perform Adam's forward bending test with a lateral trunk flexion test (Fairbank, 2004) to assess the curve's flexibility and how the spine, especially in transverse rotation, changes during the test. To perform Adam's forward bending test, the clinicians ask the scoliosis patients to flex the trunk forward, and the clinicians observe the prominent asymmetry of the trunk between the left and right sides of the trunk, which happens on the convex side of the curve. If the patients have apical vertebrae at the thoracic level, the rib cage will be prominent, commonly located on the right side. If the patients have apical vertebrae at the lumbar level, the lumbar area will be prominent, commonly located on the left side. Then, the patient continues performing a lateral flexion test to the convex side of the curve. If the prominent disappears, the scoliotic spine has a flexible curve, and the transverse rotation of the spine can change during trunk movement. If the prominent slight reduces, the scoliotic curve has more rigidity of the curve, and the transverse rotation of the spine is more difficult to change during movement.

Scoliosis deformities can also affect the gait. Schmid and colleagues used a motion capture system to quantify the spinal kinematics during walking (Schmid *et al.*, 2016). Twenty-nine participants were recruited (14 AIS patients and 15 normal subjects for the control group). The study concluded that the sagittal curve in the AIS group had less kyphosis on average of  $10.7^\circ$  compared to the control group during walking. In the coronal plane, the curvature angles indicated average lateral deviation in the AIS group compared to the control group or the spine deviation laterally during walking. However, their research did not study the spinal curvature change in a transverse plane or 3-dimensional during gait. Sung and Park studied the axial trunk rotation or horizontal trunk rotation in AIS patients (14 AIS patients and 18 normal subjects for the control group) by using a motion capture system (Sung and Park, 2020). Participants with right thoracic curves were recruited for the study. During the experiment, the participants had to perform an axial trunk rotation test while standing while holding a bar with a shoulder flexion of  $90^\circ$ . Participants then rotated the trunk to the left and then right. The result reported that the spine's range of motion in axial rotation had a positively moderate correlation with Cobb's angle magnitude in the AIS group, and the lumbar spine had significantly decreased the range of motion in horizontal rotation in the AIS group. The literature showed that scoliosis deformities could change while patients walk or perform trunk motions. Because of the rigidity of the scoliosis curve, the spine tended to have a limitation of range of motion during dynamic movement in 3 dimensions.

### **3.3. Spinal orthotic designs and its effectiveness**

Several variables must be considered when treating AIS patients, including the degree of the coronal curve, the patient's age, and the patient's and parents' preferences. A spinal orthosis is typically prescribed by a doctor for a patient with a coronal curve magnitude between  $25^\circ$  and  $40^\circ$  (Richards *et al.*, 2005, Danielsson *et al.*, 2007, Yrjönen *et al.*, 2007, Katz *et al.*, 2010, Schiller, Thakur and Eberson, 2010). The spinal orthosis (SO) is an external device that encompasses around the thoracic, lumbar, and

sacral areas, and the most common orthotic design is the thoraco-lumbo- sacral orthosis (TLSO) or underarm brace.

Regarding the orthotic guideline from the International Society on Scoliosis Orthopaedic and Rehabilitation Treatment (SOSORT) and Scoliosis Research Society (SRS), the criteria of spinal orthotic treatment is when the coronal curve magnitude between 25° and 40° and the grading of Risser sign between 0 and 3 (Negrini *et al.*, 2015) because this period is a high risk of curve progression. The objective of using spinal orthosis is to delay or prevent the curve progression (Richards *et al.*, 2005), reduce the surgical treatment, and restore the spinal column to neutral position in all planes (Edmond, Hill and Raso, 2008). In practice, many doctors may recommend a spinal orthosis to a patient who has a coronal curve magnitude more than 40° or a Risser sign greater than grade 3 in order to prevent and delay the progression of the curve.

There are many different types of spinal orthotic designs available today, and they can be classified based on a variety of factors such as time of use (day and night orthosis), biomechanical principle (three-point force system, end-point control, static or dynamic orthosis), and fabrication materials (low and high temperature thermoplastic, fabrics and elastic materials, conventional or metal design), for example, the Milwaukee brace (Lonstein and Winter, 1994, Keiser and Shufflebarger, 1976, Carr *et al.*, 1980, Fisher, Rapp and Emkes, 1987), Boston brace (Emans *et al.*, 1986, Katz *et al.*, 1997, Vijvermans, Fabry and Nijs, 2004, Wiley *et al.*, 2000, Yrjönen *et al.*, 2007, Xu *et al.*, 2019, Grivas *et al.*, 2003), Cheneau brace (Pham *et al.*, 2007, Zabrowska-Sapeta *et al.*, 2010, Zaborowska-Sapeta *et al.*, 2011a, Fang *et al.*, 2015b, Pasquini *et al.*, 2016), Charleston bending brace (Price *et al.*, 1990, Katz *et al.*, 1997, Lee *et al.*, 2012), Lyon brace (Aulisa *et al.*, 2015), and SpineCor dynamic elastic brace (Coillard *et al.*, 2007, Coillard, Circo and Rivard, 2014). Essentially, all of them are based on the biomechanical force correction strategy known as the three-point force system, with the goal of re-aligning the spine in the neutral or optimal position. This section discusses the two most popular spinal orthoses prescribed in clinic, which are the Boston brace and the Cheneua brace.

## **Boston brace**

Boston brace was invented in 1972 by Bill Miller and John Hall (Emans, 2003, Fayssoux, Cho and Herman, 2010). The Boston brace is a pre-made device with a posterior opening. It is a rigid orthosis made from thermoplastic material. It comes in a variety of sizes to accommodate each patient. The orthosis can be adjusted by cutting the trimline, opening the hole for trunk shifting, attaching the straps for suspension, and adding the pad to increase the force to correct the deformity. The Boston brace's biomechanical principle is based on a three-point force system. Nowadays, many orthotists use this concept to create custom-made orthoses for treating patients, known as custom-made TLSOs based on Boston principles. When the apex of the curve is at T8 or lower, the Boston brace is usually prescribed, and the most efficient location is when the apex is between T8 and L2. While wearing the orthosis, the coronal curve magnitude should be reduced less than 50% compared to without a brace (Boston Orthotics & Prosthetics, 2023). For example, if a patient has a coronal Cobb angle of 40°, the coronal Cobb angle should be reduced by at least 20° or more while wearing the Boston brace. Figure 3.3 illustrates the example of Boston brace.



Figure 3.3 The example of Boston brace (Fayssoux, Cho and Herman, 2010)

Emans and colleagues studied the effectiveness of the Boston brace in 295 scoliosis patients with a follow-up at least one year after weaning the brace (Emans *et al.*, 1986). According to the findings, 49% of the curve remained unchanged ( $\pm 5^\circ$ ), 39% achieved final correction between  $5^\circ$  and  $15^\circ$ , 4% achieved final correction greater than  $15^\circ$ , 4% of patients lost the correction between  $5^\circ$  and  $15^\circ$ , and 3% lost the



correction greater than 15°. During the bracing period, 11% underwent surgery, and 1% underwent surgery during the follow-up period.

Katz and colleagues investigated the Boston brace's effectiveness in 51 AIS patients (Katz *et al.*, 1997). According to the study, 61% of patients had a successful treatment, 16% progressed more than 5°, and 31% required surgery.

Wiley and colleagues investigated the Boston brace's effectiveness in 50 AIS patients with a large curve between 35° and 45° (Wiley *et al.*, 2000). This was a retrospective study that collected data after patients discontinued wearing the orthosis. The participants were divided into 3 groups based on the amount of time they spent wearing the orthosis, including more than 18 hours per day (Group 1), 12 to 18 hours per day (Group 2), and less than 12 hours per day (Group 3). The study concluded that wearing a Boston brace for more than 18 hours per day (Group 1) was more effective than other groups in preventing the progression of large curves.

Lange and colleagues also investigated the long-term outcomes of Boston brace treatment in late-onset juvenile (JIS) and adolescent idiopathic scoliosis (AIS) (Lange *et al.*, 2011). This was a prospective study. The study included 272 patients who met the SRS criteria (58 were late-onset JIS and 214 were AIS). The results showed that the majority of patients treated with the Boston brace were satisfied. At the time of follow-up, 9% of patients had surgery, and 13% had a curve progression greater than 45°.

### **Chêneau brace**

Chêneau brace was invented by Chêneau and Matthias in 1979 and several books were published in Germany and France by Jacques Chêneau and his collaborators (Chêneau, 1994, Weiß, Rigo and Chêneau, 2000, Weiss and Rigo, 2008, Rigo and Weiss, 2008). The Chêneau brace is a custom-made orthosis constructed from thermoplastic material. The seaming line, or opening side, is on the front side, called the anterior opening. Figure 3.4 illustrates the example of Chêneau brace.



Figure 3.4 The example of Chêneau brace, where a). posterior and b). anterior views (Zaborowska-Sapeta *et al.*, 2011a).

Zaborowska-Spofeta and colleagues investigated the treatment results of the Chêneau brace with physiotherapy in IS patients (Zaborowska-Sapeta *et al.*, 2011a). The study design was a prospective observational study. There were 79 patients (58 girls and 21 boys) following SOSORT and SRS criteria included in the study, and data collection was obtained after weaning the brace with a minimum of one-year follow-up. The study result was divided into 4 groups based on the severity of coronal Cobb angle after weaning, including improved (Cobb angle was less than  $6^{\circ}$ ), stable (Cobb angle was  $\pm 5^{\circ}$ ), progressed with below  $50^{\circ}$ , and progressed with more than  $50^{\circ}$ . Overall, Chêneau orthosis and physiotherapy effectively stopped spinal curve progression in 48.1% of patients. There were 25.3% of patients in the improved group, 22.8% in the stable group, 39.2% in the progressed group below  $50^{\circ}$ , and 12.7% in the progressed group with more than  $50^{\circ}$ .

Ming-Qiao Fang and colleagues studied the Chêneau brace's effectiveness on 32 AIS patients (Fang *et al.*, 2015a). When comparing before and after treatment, 50% of total participants decreased the coronal Cobb angle by more than  $5^{\circ}$ , 31% reported no change, and 19% increased the Cobb angle by more than  $5^{\circ}$ .

Pepke and colleagues investigated the efficacy of the Chêneau brace on 78 AIS patients (Pepke *et al.*, 2023). The result showed that there was no statistically significant difference in the coronal Cobb angle between pre-brace and post-brace. Furthermore, at the end of the treatment, 35% of total participants had a curve progression of less than  $5^{\circ}$ , 46% had an unchanged or within 5-degree curve

progression, and 19% had a curve progression of more than 5°. Sub-group analysis revealed that patients with thoracic curves, younger age, and Risser grade 0 had a higher rate of brace therapy failure.

Previous studies clearly examined the effectiveness of Boston and Chêneau braces. Boston brace was the first spinal orthosis for scoliosis treatment and was invented in the United States of America. In contrast, the Chêneau brace has grown in popularity over the last decade and was invented on the European side. Both designs have strong evidence that they can successfully treat scoliosis patients and help stop or delay the curve's progression. According to recent evidence, the Chêneau brace reported a slightly higher percentage of brace treatment success than Boston. However, it cannot be concluded that one is superior to the other because no research has been conducted to compare the effectiveness of two brace designs. Future research should compare the effectiveness of both devices and studies in various groups of scoliosis patients.

### **3.4. Radiographic outcome measurement in AIS**

Generally, the comparison of the individual's spine between before and after treatment is investigated to determine the success of orthotic treatments. Outcome measures (OM) are used to show the change during treatment in factors such as mobility, satisfaction, and quality of life (Robinson and Fatone, 2012). Outcome measures can be classified into three types (Fetters, 2012), including performance based outcome measure, self or patient reported outcome measure, and biomechanical analysis outcome measure.

Focusing on the biomechanical outcome measure, movement analysis, especially gait analysis, is typically used to identify clinical diagnosis and outcome measures for normal and disabled people. Movement analysis is a method to study biomechanics and primarily can be used to diagnose the pathomechanics related to musculoskeletal diseases (Harris *et al.*, 1996, Andriacchi and Alexander, 2000). It is not, however, typically used to measure scoliotic posture.

For outcome measures in scoliosis, radiographs or X-ray images of the whole spine are the most common outcome measure (Kotwicki, 2008). Radiographs well illustrate

the bone shape and spinal column alignment and do so at relatively low cost, and are widely available in hospitals. Radiographs can be used to evaluate the spinal deformity and skeletal maturity status so enabling diagnosis and a choice of treatment pathway. They can also be repeated when required during the entire treatment process. This outcome measure can be categorized as the current standard biomechanical analysis outcome measure.

As shown in the previous sections of this review, radiographs-based outcome measures are commonly used to determine the success of orthotic use. If the difference in the coronal curve magnitude before and after treatment is less than or equals  $5^{\circ}$ , it shows that the orthosis can stabilize the spine and prevent curve progression (Carman, Browne and Birch, 1990, Shaughnessy, 2007). On the other hand, if the difference in magnitude is greater than  $5^{\circ}$ , curve progression has occurred (Richards *et al.*, 2005).

The Cobb method (Cobb, 1948a) is a gold standard for measuring curve magnitude, diagnosing scoliosis, and evaluating treatment outcomes from X-rays. De Smet and colleagues investigated the difference in radiographs result between anterior-posterior (AP) and posterior-anterior (PA) views (DeSmet *et al.*, 1982). The result showed that there were no significant differences between AP and PA radiographs with a mean difference of  $2.4^{\circ}$ . The study recommended taking the radiographs in the PA view because it provided less radiation to the breast and thyroid area (Smet, 1985). Although the clinicians use the same method of measuring the Cobb angle, the results can differ by up to  $5^{\circ}$  (Morrissy *et al.*, 1990, Carman, Browne and Birch, 1990).

### **Cobb angle in coronal plane**

Coronal radiographs are used to directly measure the Cobb angle in the coronal plane and indirectly measure the horizontal rotation of the vertebrae. Sagittal radiographs are used to directly measure the Cobb angle in the sagittal plane, called kyphosis and lordosis. To measure the Cobb angle in the coronal plane, clinicians must first locate the mid-sacrum and L5 and then draw a vertical line to the C7. This line is known as the Central-vertical-sacral line (CVSL or CSL) (Kotwicki, 2008, Lenke *et al.*, 2001b). This line divides the body into left and right sides and observes the trunk's curve location and asymmetry. The clinicians then determine the location of the apical vertebra. The apical vertebra is the furthest away from the CSL line, the most

horizontally rotated, and the most deformed but not tilted (Kotwicki, 2008). The clinicians subsequently figure out the upper and lower-end vertebrae of each curve. The superior and inferior vertebrae are the most tilted, least deformed, and usually absent of any horizontal rotation (Kotwicki, 2008). Clinicians draw the lines that are parallel to the superior edge of the upper-end vertebrae and the inferior edge of the lower-end vertebrae. Clinicians then measure the angles from the intersection of 2 lines and this angle is called the Cobb angle in the coronal plane. Figure 3.5 illustrates the Cobb angle measurement in the coronal plane.

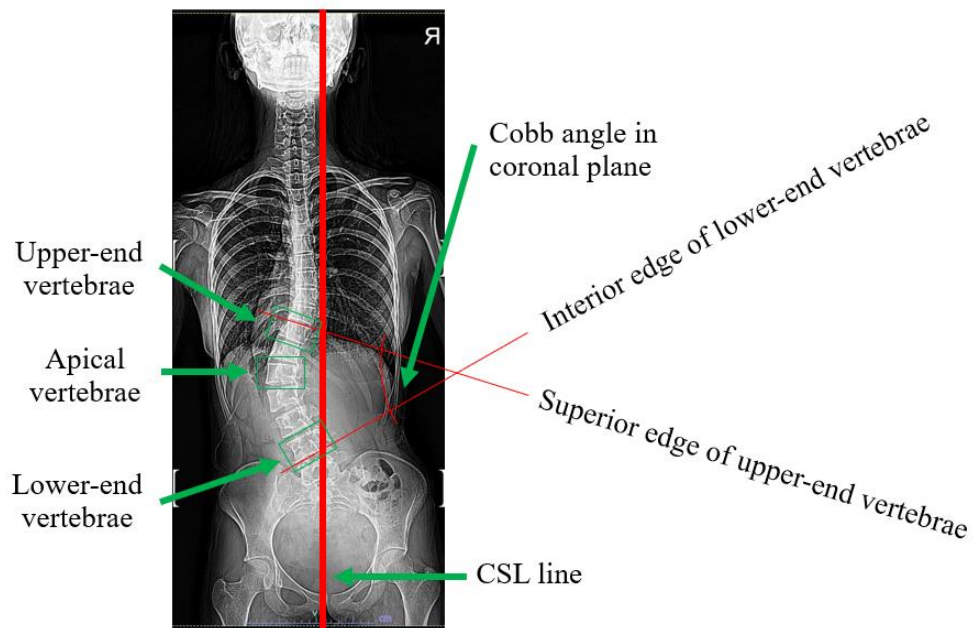


Figure 3.5 Cobb angle measurement in the coronal plane.

### **Cobb angle in sagittal plane**

To measure the Cobb angle in the sagittal plane, the clinicians locate T1, T12, L1, and L5 vertebrae. In thoracic kyphosis, the clinicians draw the lines that are parallel to the superior edge of the upper-end (T1) and the inferior edge of the lower-end (T12) vertebrae. Performing a similar method to lumbar lordosis, the clinicians draw the lines that are parallel to the superior edge of the upper-end (L1) and the superior edge of the lower-end (S1) vertebrae (Ploumis *et al.*, 2009, Findikcioglu *et al.*, 2013, Park *et al.*, 2013, Karaaslan *et al.*, 2013). Clinicians then measure the angles from the intersection

of 2 lines of each curve. Figure 3.6 and 3.7 illustrate the Cobb angle measurement of thoracic kyphosis and lumbar lordosis in the sagittal plane.

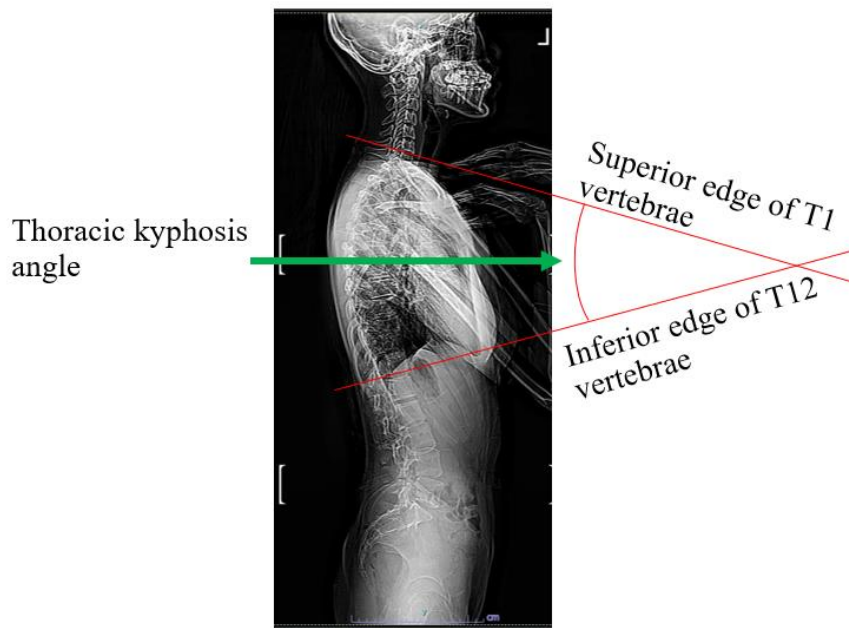


Figure 3.6 Cobb angle measurement of thoracic kyphosis in the sagittal plane.

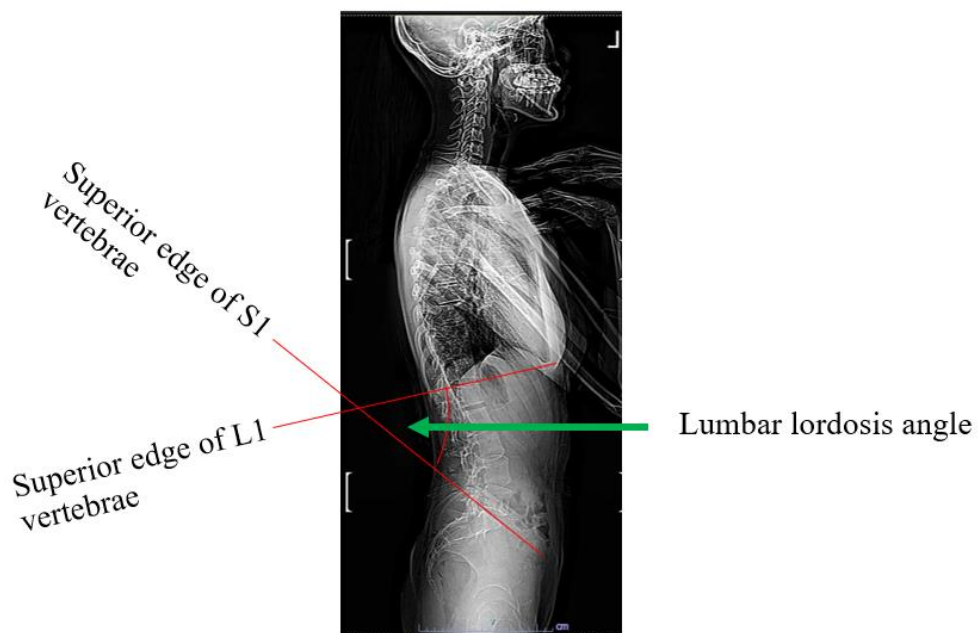


Figure 3.7 Cobb angle measurement of lumbar lordosis in the sagittal plane.

## Rotational angle in transverse plane

The radiographs can directly measure the spinal curvature in 2 planes, lateral deviation in the coronal plane and abnormalities in the kyphotic and lordotic curves in the sagittal plane, but there is a lack of the information in the horizontal plane. As a result, other researchers further investigated how to indirectly measure rotational angle using the radiographs from the coronal plane.

Nash and Moe's proposed an indirect method for measuring the relative position of the shadow of vertebral pedicles VS the shadow of the vertebral body, as shown in Figure 3.8 (Nash and Moe, 1969). Perdriolle also proposed the indirect method to measure vertebral rotation, as shown in Figure 3.9 (Perdriolle, 1979). The researcher developed the Perdriolle torsionometer, which can overlay the vertebral body and measure the rotation.

Although we can measure the horizontal rotation of the vertebral body, the angle is only for individual vertebrae, particularly the apical vertebrae, and there is no information for the whole spinal column.



Figure 2. Nash and Moe pedicle method for determining vertebral rotation

(Reproduced with permission of the Radiological Society of North America (RSNA). Kim H, Kim HS, Moon ES et al. Scoliosis imaging: What radiologists should know. *RadioGraphics*. 2010;30:1823-1842)

Figure 3.8 Vertebral rotation grading by Nash and Moe's (Greenwood and Bogar, 2014)



Figure 3.9 Vertebral rotation angle by Perdriolle, where a) Perdriolle torsionometer, and b) the torsionometer's outer margins are aligned over the vertebra's lateral borders (Richards, 1992)

### Skeletal maturity status

It is important to assess the skeletal maturity status for scoliosis patients because the curve progression occurs rapidly if the skeletal is still immature. Radiographs are a standard method for obtaining this information. Many body parts can be evaluated for skeletal maturity status, including the bone age of the hand and wrist (Bayer, 1959), pelvic triradiate cartilage (Parvaresh *et al.*, 2018, Fabricant *et al.*, 2013, Dimeglio, 2001, Kim *et al.*, 2019) and Risser sign (Risser, 1958). The Risser sign, which indicates the stage of development of iliac bone apophysis, has become a standard method for assessing growth in scoliosis patients. The Risser sign stage is generally divided into six grades ranging from 0 to 5. It grows from the anterior to the posterior of the iliac spine (Grade 0 to 4) and is eventually fused with the iliac bone (Grade 5). Figure 3.10 illustrates the stage of the Risser sign (Hacquebord and Leopold, 2012).

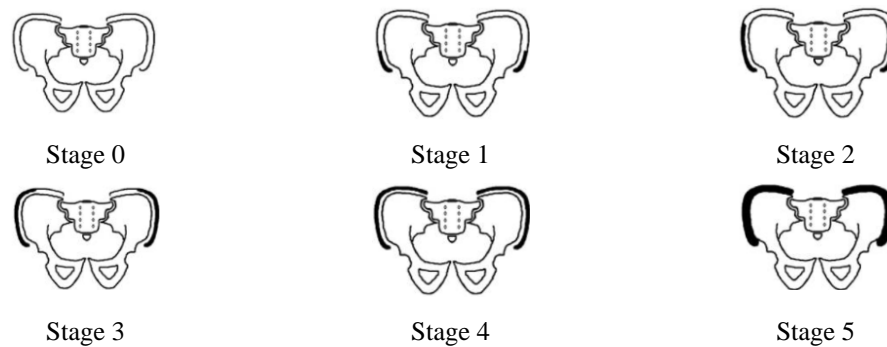


Figure 3.10 The stage of the Risser sign (Hacquebord and Leopold, 2012).

### Percent of coronal Cobb angle reduction

Coronal Cobb angle is usually measured before wearing the brace (called without a brace), while wearing the brace (called an in-brace correction), and after taking the brace off. The clinicians then calculate the percentage reduction of the coronal Cobb angle. The formula for calculating this percentage is shown in Equation 3.1, where (Rx Cobb) is the coronal curve magnitude without the brace, and (IB Cobb) is the coronal curve magnitude while wearing the brace. The Boston brace design is expected to achieve a 50% coronal Cobb angle reduction (Boston Orthotics & Prosthetics, 2023). However, several factors may influence treatment outcome, including the age of onset, the patient's age, gender, the quality of the spinal orthosis based on the skill of the



orthotist who fabricates and fits the device, patient compliance and motivation, wearing time, and strap tension.

$$\% \text{ Cobb angle reduction} = \frac{(\text{Rx Cobb} - \text{IB Cobb})}{\text{Rx Cobb}} \times 100\% \quad (3.1.)$$

### **3.5. Alternative technologies to measure the outcome in AIS**

The previous section discussed the definition of scoliosis, the characteristics of a spinal deformity, treatment with spinal orthoses, and standard outcome measurement using radiographs. However, because scoliosis is a three-dimensional deformity, there are numerous limitations to quantifying the spinal deformity using radiographs. Furthermore, being exposed to radiation multiple times during the treatment process may increase the risk of a health problem. Many research groups have recently been developing new technologies to quantify spinal parameters and reduce the number of radiographs taken. This chapter discusses alternative technologies for measuring spinal parameters in scoliosis, including ultrasound, surface topography, magnetic resonance imaging (MRI), computerised tomography (CT) scan, EOS imaging, and motion capture systems.

#### **3.5.1. Ultrasound system**

Ultrasound was the first technology to be used as an alternative to X-rays in AIS. Recently, many medical applications have used ultrasound for diagnosis and treatment. In general, the system can produce an image after scanning the target area. The ultrasound probe generates an ultrasound signal or wave, which travels through the scanned area. The system is ideal for visualising soft tissue images, and the ultrasound waves produce different signal responses depending on tissue density, which can be measured for diagnosis and treatment. Ultrasound is helpful for musculoskeletal imaging (Zheng *et al.*, 2016, Brink *et al.*, 2018). The system helps diagnose the cause of pain, swelling, and infection in the body's internal organs. It can

examine an unborn child in a pregnant woman. Moreover, the ultrasound can be used to treat the soft tissue injury. After generating the ultrasound wave in the target area, it causes vibration at the molecular level. It increases warmth within tissues to help reduce pain, improve stiff joint flexibility, and promote tissue healing.

For the general procedure to use this system, the clinicians apply an ultrasound gel to the patient's skin to prevent air pockets because it blocks the ultrasonic waves and may generate an unclear image or insufficient heat for treatment. Then, the clinicians place and move the ultrasound probe over the target area. The system generates the image and displays it on monitors or generates deep heat to injured soft tissue.

Numerous advantages can be gained from this system. This system is highly safe because it is radiation-free (Zheng *et al.*, 2016, Brink *et al.*, 2018, Edmond *et al.*, 2017). Unlike other image-diagnosis methods, such as MRI, this system does not require using substances known as contrast agents to emphasise the specific area to generate a clear image. Because some patients get allergies to this substance. This system is a non-invasive method to examine the internal organ, which can cause painlessness and no recovery time after scanning. Moreover, the system is easy to operate, portable system, relatively inexpensive (Zheng *et al.*, 2016, Brink *et al.*, 2018), available for most hospitals, and dynamic real-time response. However, the ultrasound waves cannot scan through the bone or the structure underneath the bone.

In scoliosis, Li and colleagues used ultrasound to assist during spinal orthotic fitting (Li *et al.*, 2010, Li *et al.*, 2012). The researchers used ultrasound to track the spinous process on the spine of scoliosis patients, measured the spinous process angle (SPA) for Cobb angle estimation, and compared it with radiographs results. The study design was a prospective study with 43 AIS patients (21 in the intervention and another 22 in the control groups). The study's objective was to improve the effectiveness of orthotic treatment by using a three-dimensional clinical ultrasound system (3D CUS) to find the best location of the corrective pad in the fitting process. The intervention group used the ultrasound system to find the best location of the corrective pad, which greatly reduced the curve magnitude. The control group followed the conventional method to position the corrective pad inside the orthosis. The result reported that the intra-rater reliability for measuring SPA from ultrasound was excellent or greater than 0.9 and p-value <0.05. The average of immediately in-brace correction in the

intervention group was significantly higher than the control group. There was an excellent positive correlation between Cobb angle from radiographs and SPA from the ultrasound system, representing 0.98 with p-value <0.01. The study concluded that the 3D CUS significantly improved the scoliosis deformity when locating the corrective pad in orthotic fitting. The system could be considered effective and further used in other spinal deformities. Figure 3.11 illustrates the various experimental locations of padding and ultrasound scanning used to determine the best position of the corrective pad.

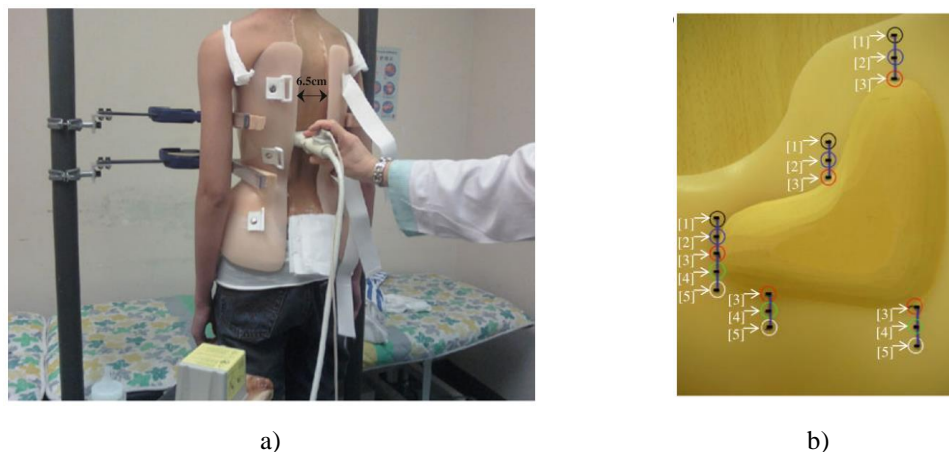


Figure 3.11 a) In-brace ultrasound scanning, and b) the locations for corrective pad (Li *et al.*, 2010, Li *et al.*, 2012)

Zheng and colleagues investigated the reliability and validity of using Scolioscan (3-dimensional ultrasound imaging) (Zheng *et al.*, 2016). The study design was a prospective divided into two sub-studies: 1) intra-rater and inter-operator reliability while using the Scolioscan (20 AIS patients), and 2) correlation between radiographs and ultrasound systems (49 AIS patients). The result reported that the intra-rater and inter-operator reliability was very good, with an intraclass correlation coefficient (ICC) of 0.94 and 0.88, respectively. Additionally, the angles from the ultrasound system were slightly smaller than the radiographs. The correlation between radiographs and ultrasound was moderate to strong positive correlation or greater than 0.72 overall.

Edmond Lou and colleagues used the ultrasound system to assist during assessment, optimal correction, and casting processes (Edmond *et al.*, 2017). Thirty-

four AIS patients were included in the study following the SRS criteria and divided into 2 groups: intervention (17 participants with 2 males and 15 females) and control (17 participants with 2 males and 15 females by retrospective method) groups. The study design was a prospective for the intervention group. The study's objective was to investigate the effectiveness of using ultrasound to assist orthotists in designing braces for AIS treatment. Participants from 2 groups were prescribed the same orthotic design (Providence brace) and casting method (using Providence brace standing frame). Additionally, in the intervention group, an ultrasound system, pressure measurement system, and software were used to assist during the casting process. The system took 1.5 min during the scanning process and another 2 min during coronal Cobb angle calculation before displaying on the computer screen. Figure 3.12 illustrates the ultrasound system and the scanning frame, as well as the Providence brace standing frame, which was used to assist in casting. The result reported that the intervention group could reduce the total number of radiographs taken from 28 times in the control group to 18 times in the intervention group. The intervention group spent slightly longer time during casting but significantly shorter during brace adjustment. The control group needed more brace adjustments (11 times) than the intervention group (1 time). The percentage of coronal Cobb angle reduction of the control group during the first in-brace follow-up was 33% (SD = 19), and the final in-brace follow-up was 40% (SD = 20), respectively. Additionally, the percentage of coronal Cobb angle reduction of the intervention group during the first in-brace follow-up was 29% (SD = 11), and the final in-brace follow-up was 48% (SD = 17). The study concluded that the ultrasound system could help to improve the quality of spinal orthotic casting, reduce the number of radiographs taken during follow-up, reduce the total time of brace adjustment, and improve the percentage of coronal Cobb angle reduction. Figure 3.13 shows an example result from radiographs and the ultrasound system without force being applied, first in-brace follow-up, and second in-brace follow-up.

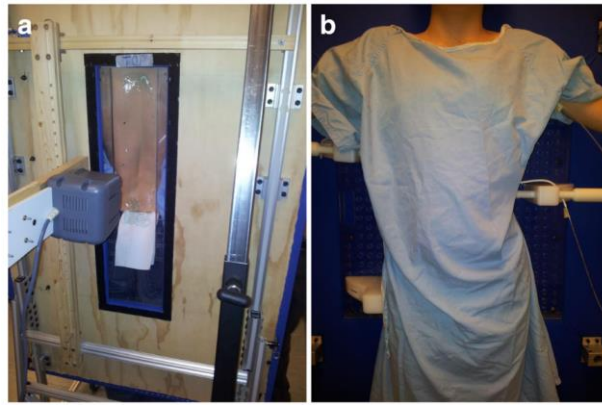


Figure 3.12 a) The opening at the back of the frame, and b) a subject stands on a frame with a custom Providence brace design system (Edmond *et al.*, 2017)

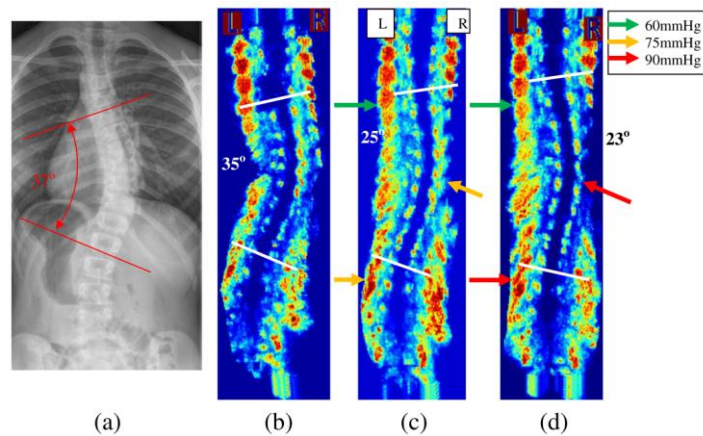


Figure 3.13 a) The standing pre-brace radiographs with Cobb angle 37°, b) The baseline US scan (Cobb angle 35°), c) The first trial US scan (Cobb angle 25°), and d) The 2nd trial US scan (Cobb angle 23°) (Edmond *et al.*, 2017)

Brink and colleagues investigated the reliability and validity of ultrasound in AIS patients (Brink *et al.*, 2018). The study compared the results of 2 methods for calculating the coronal angle from ultrasound, spinous process (SP) angle and transverse process (TP) angle, to the results from radiographs. The study recruited 33 AIS patients. The Scolioscan was used to collect ultrasound data. The data were gathered using four methods (fig. 3.14): automatic SP angle, manual SP angle, manual TP angle, and radiographs Cobb angle. According to the findings, ultrasound angles were 15% to 37% smaller than radiographs angles. All ultrasounds had an excellent positive correlation to radiographs with greater than 0.97. Furthermore, no statistically significant differences were found between the automatic SP angle, manual SP angle, and manual TP angle.

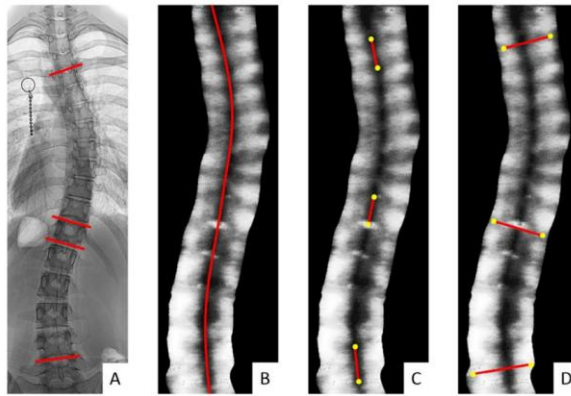


Figure 3.14 a) Coronal Cobb angle measured by X-ray, b) automatic spinous process angle measurement, c) manual spinous process angle measurement, and d) manual transverse process angle measurement (Brink *et al.*, 2018)

Yi-Shum Wong and colleagues used the ultrasound to measure the spinous process angle (SPA) and assist during the assessment process (Wong *et al.*, 2019). Nine hundred and fifty-two IS patients were recruited in the study (721 females and 231 males). The study's objective was to evaluate the reliability and validity of ultrasound for measuring coronal angle compared between radiographic measurement and automatic spinous process angle (SPA) measurement from an ultrasound system. The study design was a prospective. The automatic SPA measurement from ultrasound was previously developed by Zhou and colleagues (Zhou *et al.*, 2017). During scanning, participants were asked to stand in front of the Scolioscan and uncover the posterior side of the trunk. Figure 3.15 illustrates the scoliosis patient during ultrasound scanning by using the Scolioscan. The practitioners scanned the spine between T1 and L5 vertebrae. Then, the system calculated and reported the coronal curve magnitude automatically. The system took about 5 to 10 minutes, including data entry, data processing and patient positioning. Regarding the results, there were 1,625 curves measured from radiographs, while 1,432 curves (88.1%) were detected by ultrasound, 3 curves (0.2%) had mismatch of curve direction, and 190 curves (11.7%) were not detected by ultrasound. The correlation between radiographs and ultrasound was 0.629, 0.873 and 0.740 presenting at upper thoracic curves (apices at T6.5 or above), upper spinal curves (apices between T7 and T12/L1), and lower spinal curves (apices at L1 or below) respectively. The study suggested that the radiographic examination was still necessary for the first assessment to clarify the spinal evaluation and identify the anatomical landmarks. Ultrasound system could alternatively be used during the

follow-up period, and most appropriate for a patient with more than 30° coronal Cobb angle with the apex at T7 or below.

Table 3.1 summarises the intra- and inter-rater reliability of the ultrasound system and the correlation between ultrasound and radiograph results from the relevant literature.

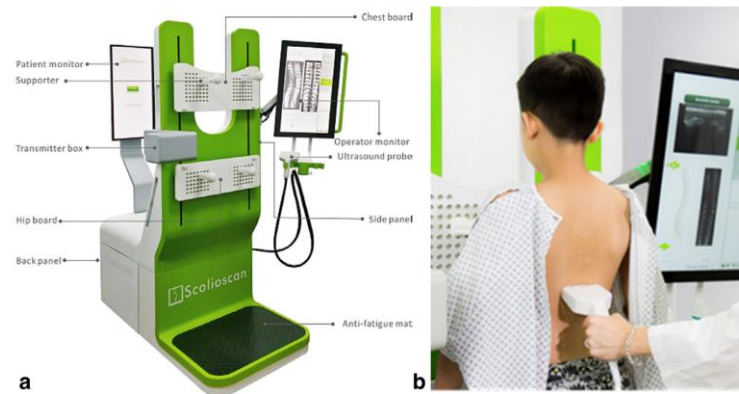


Figure 3.15 (a) The Scolioscan (b) Ultrasound assessment for a patient (Wong *et al.*, 2019).

Table 3.1 Summary of the intra- and inter-rater reliability of the ultrasound system and the correlation between ultrasound and radiograph results from the relevant literature.

Author, year	Number of Participants	Scanning system	Intra-rater reliability	Inter-rater reliability	Correlation with radiographs
Li et al., 2010 & 2012	43 AIS patients	3D CUS	> 0.90	n/a	0.98
Zheng et al., 2016	20 & 49 AIS patients	Scolioscan	0.94	0.88	0.72
Brink et al., 2018	33 AIS patients	Scolioscan	n/a	n/a	0.97
Wong et al., 2019	952 IS patients	Scolioscan	n/a	n/a	0.629 at above T6.5 0.873 at T7 to T12/L1 0.740 at below L1

**Note:** 3D CUS = Three-dimensional clinical ultrasound system, AIS = adolescent idiopathic scoliosis, IS = idiopathic scoliosis, n/a = not applicable, T = thoracic, L = lumbar.

Ultrasound systems, especially the Scolioscan, have recently become more popular in scoliosis applications. These systems have been used to quantify the spinal parameters in the coronal plane by measuring the spinous process angle (SPA). Ultrasound systems are radiation-free. Therefore, they can scan the trunk several times without harming the patient. The Scolioscan is an automatic system to calculate the SPA. It is convenient for the users to use the system. Moreover, previous research was



conducted on a large group of scoliosis patients using Scolioscan. The results showed that the correlation between the ultrasound system and radiographs varied and depended on the spinal level. Overall, the relationship level was between moderate to high positive correlation. The intra- and inter-rater reliability showed very high reliability or  $>0.90$ . The Scolioscan system is relatively expensive, and only a few hospitals have this system. Furthermore, the current research only studied the scoliosis deformity in the coronal plane, and no research studied the deformity on sagittal, transverse, and 3-dimensions.

### **3.5.2. Surface topography (ST)**

Body surface topography is a photogrammetric technique that uses photograms to reconstruct the shapes, sizes, and mutual positions of objects (Karpiel *et al.*, 2021). The photogram is an image created by placing an object directly onto the surface of a light-sensitive material and exposing it to the light. Many surface topography applications are currently being developed for medical treatment, such as scanning the upper and lower limbs to create prostheses and orthoses. However, this technology is relatively new for scoliosis treatment. The benefit of this technology is safety, as it is a radiation-free and non-invasive system. Additionally, it can illustrate the trunk in three dimensions.

In 1997, Theologis and colleagues evaluated the reliability of the Integrated Shape Imaging System (ISIS) scans in detecting the progression of scoliosis (Theologis *et al.*, 1997). The study design was retrospective, and 78 AIS patients with a right thoracic curve were included. The results from the ISIS scan were compared to coronal Cobb angle measurement from radiographs. The study reported that the scanning system could demonstrate a significant change in the patient with progressive scoliosis.

In 2001, Liu and colleagues used the Quantec Spinal Image System (QGIS) to classify the spinal deformity of mild AIS patients (Liu *et al.*, 2001). The study design was prospective, and 248 AIS patients were included. The QGIS could quantify the spinal parameters, including trunk height, pelvic tilt angle, Suzuki hump sum, axial surface rotation, kyphosis, and lordosis. The study reported that the scanning system could quantitatively assess mild spinal deformity.



In 2004, Lyon and colleagues further assessed the QGIS by focusing on intra-observer reproducibility (Lyon *et al.*, 2004). The study design was prospective, and 200 AIS patients were included. The result reported that the reliability of the parameters was excellent or greater than 80%. This study confirmed the previous study from Liu and colleagues that the scanning system could quantitatively assess mild spinal deformity.

In 2013, Schülein and colleagues investigated the inter-observer and intra-observer reliability of Raster-stereographic 3D back surface analysis system (Schülein *et al.*, 2013). The study design was prospective, and 39 postsurgical AIS patients were included (32 females and 7 males). The spinal parameters were trunk length, trunk inclination, trunk imbalance, kyphosis, and lordosis after scoliosis surgery, and 5 investigators collected the data. The result reported that the inter-observer and intra-observer reliability was excellent. The inter-observer reliability was between 0.918 and 0.988, while the intra-observer reliability was between 0.697 and 0.994. The study reported that the scanning system could investigate the spinal parameters after spinal surgery in scoliosis patients.

In 2015, Komeili and colleagues used the Surface Topography torso scans (Minolta laser scanner) to detect the curve progression of 100 AIS patients (Komeili *et al.*, 2015b). The scanning system could illustrate the deviation colour map (DCM) on the surface to represent the asymmetry of the trunk (Asymmetry visual analysis). Figure 3.16 illustrates the DCM to analyse the spinal parameters in the scanning system. The scanning results were compared to the coronal Cobb angle measured from radiographs, and the data were collected during a 1-year follow-up interval. The curve was considered as progress if the curve was greater than 5°. The result reported that the scanning system could detect 85.7% of the curve progression, 71.6% of the non-progression cases, and a false-negative rate of 4%.

Komeili and colleagues investigated the reliability of Surface Topography torso scans (Minolta laser scanner) and compared to the radiographs in the same year (Komeili *et al.*, 2015a). The spinal parameters were curve number, direction and location, apical vertebra location, and curve severity, and 3 examiners collected the data. A total of 124 AIS subjects were recruited (100 radiographs from the investigating group and 24 participants in the validation sample group). The curve

magnitude was divided into 3 groups, including mild ( $<25^\circ$ ), moderate ( $25^\circ$  to  $40^\circ$ ), and severe ( $>40^\circ$ ) curves. The curve location was divided into 3 groups, including single, double, and triple. For identifying the curve type, the average percentage agreement for single, double, and triple curves was 62%, 66%, and 23%, respectively, with a kappa coefficient of 0.32. For a mild curve group, the average percentage agreement for single, double, and triple curves was 72%, 77%, and 0%, respectively, with a kappa coefficient of 0.52. For identifying the curve location (proximal thoracic, thoracic and thoracolumbar, and lumbar), the percentage of prediction was 63%, 92%, and 62%, with a kappa coefficient of 0.67. For a mild curve group, the percentage of prediction was 87%, 95%, and 68%, with a kappa coefficient of 0.74. The study showed that the scanning system could identify the curve direction correctly and well in the mild curve group.

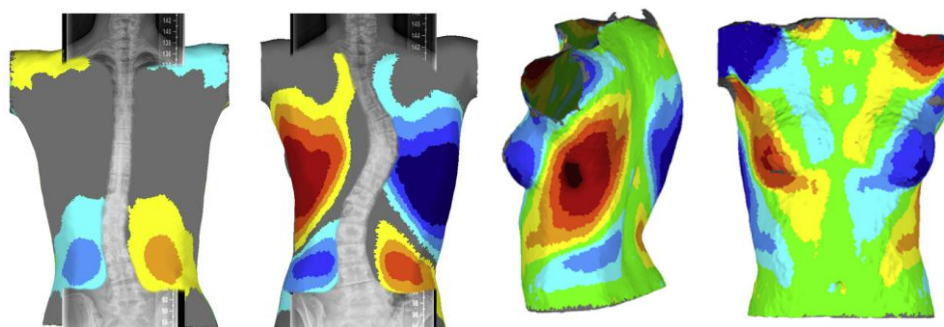


Figure 3.16 The deviation colour map (DCM) to analyse the spinal parameters in Surface Topography (ST) torso scans (Minolta laser scanner) (Komeili *et al.*, 2015a, Komeili *et al.*, 2015b)

In 2019, Navarro and colleagues published a systematic review with a meta-analysis on the evaluation and validity of surface topography for AIS patients (Navarro, Candotti and Rosa, 2019). Twenty-three articles were included for the qualitative synthesis. Only 4 articles examined the Pearson correlation coefficient (r-value) between the scanning system and radiographs. In 2000, Thometz and colleagues compared the QGIS with the results from radiographs (Thometz *et al.*, 2000). The study included 193 cases of pediatric scoliosis. The results reported that the r-values in the thoracic and lumbar coronal Cobb angles were moderate, or 0.65 and 0.63, respectively. In 2006, Knott and colleagues studied the accuracy of Orthoscan (Orthoscan Technologies, Inc.) by comparing it with radiographs (Knott *et al.*, 2006). Forty-two scoliosis patients were included in the study. The result reported that the r-

value was 0.65 in the thoracic coronal Cobb angle and 0.13 in the lumbar coronal Cobb angle. In 2012, Frerich and colleagues used the Formetric 4D surface topography to compare with radiographs (Frerich *et al.*, 2012). Fourteen AIS patients were included in the study. The result reported that the r-value was 0.87 in the thoracic coronal Cobb angle and 0.76 in the lumbar coronal Cobb angle, and the angle error of the thoracic and lumbar was 6.98° and 9.42°, respectively. Additionally, the r-value was 0.80 in thoracic kyphosis and 0.81 in lumbar lordosis. In 2016, Knott and colleagues studied the accuracy of the DIERS Formetric system with radiographs (Knott *et al.*, 2016). One hundred and ninety-three pediatric scoliosis patients were included in the study. The result reported that the r-value was 0.73 in the thoracic coronal Cobb angle and 0.49 in the lumbar coronal Cobb angle, and the thoracic and lumbar angle error was 5.8° and 8.8°, respectively. Additionally, the r-value was 0.87 in thoracic kyphosis and 0.82 in lumbar lordosis. Navarro and colleagues calculated a Meta-analysis of correlation (r-value) from 4 articles mentioned above (Navarro, Candotti and Rosa, 2019). The result reported that the r-value was 0.74 in the thoracic coronal Cobb angle and 0.55 in the lumbar coronal Cobb angle. Additionally, the r-value was 0.84 in thoracic kyphosis and 0.82 in lumbar lordosis. Table 3.2 summarises the Meta-analysis of correlation (r-value) of thoracic coronal Cobb angle, lumbar coronal Cobb angle, thoracic kyphosis, and lumbar lordosis.

Table 3.2 Summary of the Meta-analysis of correlation of thoracic coronal Cobb angle, lumbar coronal Cobb angle, thoracic kyphosis, and lumbar lordosis for ST system (Navarro, Candotti and Rosa, 2019).

Author, year	Number of Participants	Scanning system	Pearson correlation coefficient (r)			
			Thoracic Cobb	Lumbar Cobb	Kyphosis	Lordosis
Thometz, 2000	149 IS patients	QSI	0.65 (moderate)	0.63 (moderate)	n/a	n/a
Knott, 2006	42 scoliosis patients	Orthoscan	0.65 (moderate)	0.13 (poor)	n/a	n/a
Frerich, 2012	14 AIS patients	Formetric 4D surface topography	0.87 (strong) (6.98° error)	0.76 (strong) (9.42° error)	0.80	0.81
Knott, 2016	193 pediatric scoliosis patients	DIERS Formetric	0.73 (5.8° error)	0.49 (8.8° error)	0.87	0.82
Result from Meta-analysis			0.74	0.55	0.84	0.82

**Note:** IS = idiopathic scoliosis, QSI = Quantec Spinal Imaging System, D = Dimension, n/a = not applicable.

Surface topography systems use 3-dimensional scanning to scan the trunk to quantify the spinal parameters in scoliosis patients. The systems are safe for the patients because they are radiation-free and non-invasive. Therefore, these systems can scan the trunk several times without harming the patients. The previous research showed that the correlation between surface topography and radiographs varied and depended on scanning types and the spinal level. Overall, the relationship level was between poor to strong positive correlation, slightly lower than the ultrasound system. When comparing the correlation between coronal and sagittal spinal parameters, the sagittal plane showed a slightly higher correlation than the coronal plane. The current system is still used in the laboratory research but not in clinical practice. The current research only studied the scoliosis deformity in the coronal and sagittal planes. Even though the scanning could reconstruct the trunk in 3 dimensions, no research studied the scoliosis deformity in 3-dimensions or while casting or wearing an orthosis.

### **3.5.3. Magnetic resonance imaging (MRI)**

Magnetic resonance imaging (MRI) is an advanced imaging technology that creates detailed images for diagnosis and treatment planning. MRI system is a non-invasive medical examination, which creates a magnetic field around the patient and uses a computer to take pictures of the patient's body. Each image illustrates a few layers of body tissue, and combining several images can result in a three-dimensional structure. The benefit of MRI is to show the structure of soft tissues, such as the spinal cord and intervertebral discs.

In the scoliosis treatment, MRI may be recommended to check for brain and spinal cord abnormalities. MRI may also be recommended to diagnose JIS or AIS in patients with atypical scoliosis, such as congenital scoliosis and scoliosis associated with other neurological disorders. A scoliosis patient who presents numbness, weakness, pain, or an asymmetric loss of reflexes may be advised to undergo MRI scan to determine the cause of the problem. MRI helps investigate neural axis malformations and can be used to plan for surgical correction (Ozturk *et al.*, 2010, Ameri *et al.*, 2015). MRI may also be recommended for a child who has a severe curve with rapid progression in order to evaluate and prevent future neurological complications. In AIS patients, MRI

may be used to measure vertebral rotation (Birchall *et al.*, 1997). MRI produces detailed images with excellent tissue contrast without the use of ionising radiation. However, the scoliosis deformity changes in three dimensions or the spine moves in and out of the image plane, making it difficult to interpret the results (Wright, 2000) and requiring special software to solve this problem. Furthermore, the patient must lie inside a magnetic tube for about an hour to scan the whole spine, which is also time-consuming. Figure 3.17 illustrates the example of image created by MRI (Schmitz *et al.*, 2001).



Figure 3.17 Example of image created by MRI (Schmitz *et al.*, 2001).

Although the MRI is a radiation-free system that can create a 3-dimensional spinal column reconstruction in scoliosis, the system takes a long time to complete the process and is relatively expensive. Additionally, the patient must be in a lying position while scanning, which can not assess the true spinal curve magnitude compared to weight bearing or standing position. Therefore, the system is not commonly used to assess the typical AIS patients.

#### **3.5.4. Computerized tomography (CT) scan**

A Computerized tomography (CT) scan is a series of X-ray images taken from different angles around the body and uses a computer to process and create cross-

sectional images (slices) of the scanned area. The system can also reconstruct the scanned part to be a 3-dimensional shape. This system generates high radiation doses to the patients during scanning. MRI can provide excellent images of soft tissue, such as the spinal cord and canal, but not the bone structure. In contrast, the CT scan well shows the bone structure and is recommended to check the skeletal abnormalities of the vertebral body, particularly in the preoperative planning stage (Wright, 2000, Ng and Bettany-Saltikov, 2017).

In scoliosis application, the CT scan is mostly recommended for atypical scoliosis, such as congenital scoliosis and malformation of the vertebral body, not typical AIS patients. The CT scan is helpful in assessing the segmentation of vertebral abnormalities and axis rotation of vertebra and rib cage deformities (Aaro and Dahlborn, 1981, Ho *et al.*, 1993, Krismer *et al.*, 1996). Figure 3.18 illustrates the example of image created by CT scan (Pierre-Aurelien *et al.*, 2018).



Figure 3.18 Example of image created by CT scan (Pierre-Aurelien *et al.*, 2018).

Although a CT scan can create a 3D reconstruction of the spinal column in scoliosis, the system produces a high radiation dose, must be in a lying position while scanning, is relatively expensive, and takes a long time to complete the scanning process (Lam *et al.*, 2008). Therefore, the system is not a common tool for AIS patient assessment.

### 3.5.5. EOS imaging

Biplanar stereo-radiography, also known as EOS Imaging, is a low-dose radiographic method that captures anteroposterior and lateral 2 dimensional images at the same time (Illés and Somoskeöy, 2012). The system then generates a 3D reconstruction of the object (Melhem *et al.*, 2016, Illés and Somoskeöy, 2012). EOS system well illustrates the images of musculoskeletal system. The system reduces radiation dose by approximately 8 to 10 times when compared to traditional radiographs and by approximately 800 to 1,000 times when compared to high-resolution 3D CT scan systems (Girdler *et al.*, 2020). The system can create a full body images and can be scanned during sitting or standing (weight-bearing method). Recently, the EOS system has been widely used in a variety of applications, including scoliosis (Morel *et al.*, 2018, Ilharreborde *et al.*, 2016, Kato, Debaud and Zeller, 2017), to quantify spinal deformity in two and three dimensions. Figure 3.19. illustrates an example of images taken by the EOS system in 2 dimensions and reconstructed in 3 dimensions (Illés and Somoskeöy, 2012).

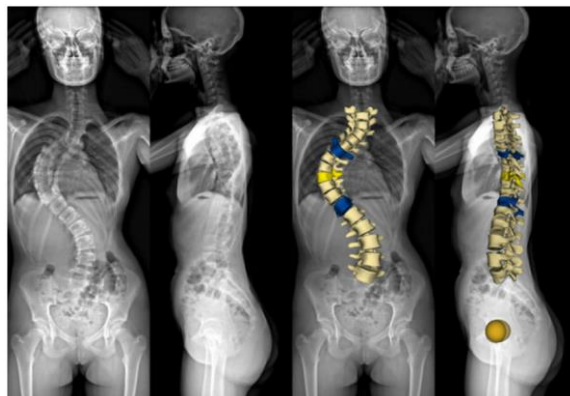


Figure 3.19 Example of images taken by the EOS system in 2 dimensions and reconstructed in 3 dimensions (Illés and Somoskeöy, 2012).

The EOS system is a new version of radiographs that can create images in two planes and reconstruct the object in 3 dimensions. The system reduces the radiation dose by approximately 8 to 10 times compared to traditional radiographs. However, the system is relatively expensive, and only a few hospitals have this system. No research has studied the EOS system compared to traditional radiographs in scoliosis

applications. Even though the system can reconstruct the spine in 3 dimensions, no research studied the scoliosis deformity in 3-dimensions.

### **3.5.6. Motion capture system**

Motion capture, also known as Mocap, was invented in the 1960s. Recently, such systems have been used in a variety of applications, including industries, military, sports, gaming, entertainment, robotics, and medicine. Basically, this system is used to record the movement of objects or people. Based on the technology used to calculate marker position, it is divided into two types: a visual record of body segment positions (optical system) and electron-magnetic sensors to determine segment position and orientation (Richards, 1999).

The Optical system is widely used in biomechanics and human movement analysis. The system utilizes the data (or marker) captured from the image to calculate the marker position in 3-dimensional space using at least two cameras. Numerous marker positions can then calculate the spatial-temporal and kinematic parameters during the movement. Three types of optical systems have been used recently. The first one is an optically passive system. Cameras with rings of infrared diodes are used to track retroreflective passive markers attached to the object. The marker does not emit the light itself but reflects the infrared light from the cameras back to the cameras. The second type of systems are optically active. The cameras tracks infrared LED markers attached to the subject. This marker emits the light itself. The third category and the newest is a markerless system. A marker is not required during capturing the data. The motion capture relies on the capability and the intelligence of the software to recognise the body and its orientation from the background of the image in a number of views and to triangulate these. Markerless motion capture is an emerging technology and so far has not been approved for routine medical use.

In medical applications, the first two categories, passive and active systems of Mocap has been used in rehabilitation in patients recovering from an accident or surgery and to improve their physical ability. The current Mocap systems can report movement in real-time and integrate the data with virtual reality which can be used to improve patient interaction during training and help improve the physical performance



of patients. Interestingly, most research studies have used the motion capture system to analyse the movement of upper and lower limbs. However, few have done so for the spine, especially in scoliosis patients.

In 2008, Zabjek and colleagues used a motion capture system to quantify the spinal deformity of AIS patients and compare the result to radiographs (Zabjek *et al.*, 2008). The study design was prospective, with 57 AIS patients. The objective of this study was to compare the results of posture in different types of spinal curve patterns between motion capture (Motion Analysis Corporation, 8 cameras) and radiographic techniques. In motion analysis, the markers were attached to anatomical bony landmarks, including the spinous process of T1 and S1, the left and right acromion processes, inferior angles of the scapula, ASIS, PSIS, calcaneus, and second metatarsal bones. The spinal parameters of this study were the tilting and rotation of the shoulder (acromion process) compared to the pelvis (ASIS and PSIS), tilting and rotation of shoulder blades (inferior angle of the scapula) compared to pelvis (ASIS and PSIS), and lateral shift (the spinous process of T1 and S1). The result showed that their motion capture technique reported a strong positive intraclass correlation to the radiographic technique. The study also suggested to future research that the new development in motion capture should focus on the monitoring application in the progression of spinal deformity. Figure 3.20 illustrates a postural geometry in the transverse and coronal planes (Zabjek *et al.*, 2008)

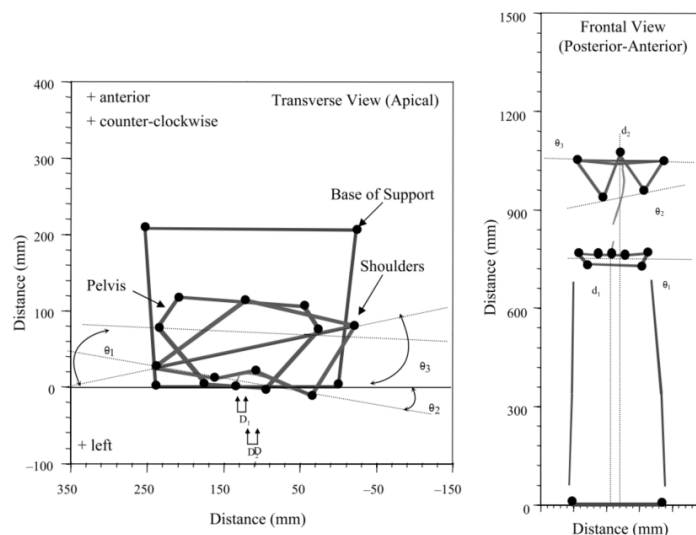


Figure 3.20 Postural geometry in the transverse and coronal planes from the study of Zabjek and colleagues (Zabjek *et al.*, 2008).

In 2011, Solomito, Lee and Peterson used a Vicon 512 system (12 cameras) to quantify the spinal parameters in AIS patients (Solomito, Lee and Peterson, 2011). This research aimed to study a correlation between radiographs and motion capture results in kinematics and postural analysis of the trunk. The study design was prospective, with 10 participants (5 AIS patients and 5 normal subjects). There were 2 main experiments involved in the study. The first experiment was to study static posture and divided the body into 4 sections, including the upper trunk, middle trunk, pelvis, and spine. Additionally, the second experiment was to study kinematics by asking participants to perform trunk movements, including lateral bending, axial rotation, flexion, and extension of the spine. Ten markers were attached to anatomical bony landmarks, as shown in Figure 3.21. The first experiment showed that the motion capture system reported a high positive correlation ( $r > 0.75$ ) in the postural experiment compared with the radiographic result. The second experiment showed that most of the AIS group's range of motion (ROM) was similar to the control group. However, individual scoliosis patients tended to have lower spinal ROM, especially in bending at the lower spine. The study concluded that the motion capture system could help diagnose and track scoliosis's progression. The researchers also suggested that future research add more markers along the spine to increase the model's sensitivity.

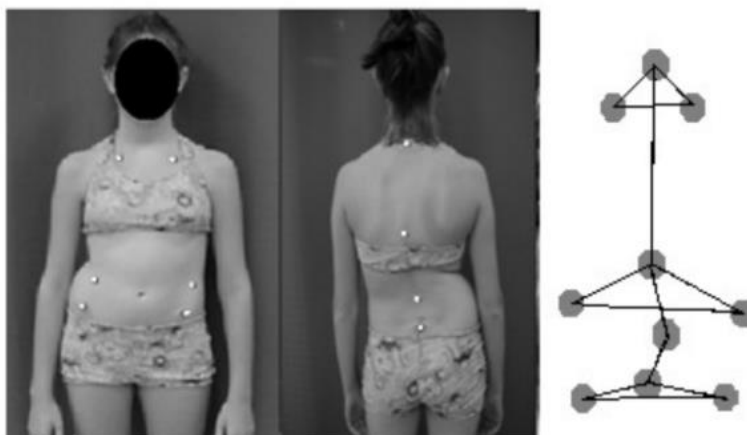


Figure 3.21 Marker locations from the study of Solomito, Lee and Peterson (Solomito, Lee and Peterson, 2011).

In 2015, Schmid and colleagues used a motion capture system (Vicon, Nexus software, 12 cameras, 300 Hz) to quantify the thoracic curve in 10 AIS patients (Schmid *et al.*, 2015). The research's objective was to compare the accuracy of a skin

marker attached to the trunk's surface with Cobb angle measurement from radiographs. The study design was prospective. Markers were placed at C7, T3, T5, T7, T9, T11, and L1 - L5. The result showed a moderate to strong positive correlation between the motion capture system and radiographs. However, the study did not report the correlation value. The highest correlation was presented at the thoracic curve in the sagittal plane, and the results were not underestimated or overestimated. However, the coronal curve results reported an underestimation of the result compared to radiographs. The study also reported that the distance difference between the markers in the motion capture system and the anatomical position in the radiographs was between 5 mm and 18 mm error in the vertical direction and up to 9 mm error in the horizontal direction. The study showed that the skin marker technique could quantify the spinal curve for AIS patients during assessment. The comparison of the skin marker technique and Cobb angle measurement in the lateral view (thoracic kyphosis) was reasonably accurate. However, the comparison between this technique and Cobb angle measurement in the posteroanterior view (coronal curve) was systematically underestimated.

In 2018, Jang used a motion capture system to quantify the spinal parameter in AIS patients in 3 dimensions (Jang, 2018). This study was the first study that used a motion system to quantify the scoliosis in 3 dimensions. The aim of this study was to develop a new spinal parameter to describe the spinal deformities in 3 dimensions. Furthermore, the study also developed a new application in motion capture system for AIS assessment and presented the result in real time which was the first study to report the result in real time. Figure 3.22 illustrates the location of marker placements on the anatomical bony landmarks. The spinal column was divided into 5 segments, including cervical, upper thoracic, lower thoracic, upper lumbar, and lower lumbar. The spinal angle in 3 dimensions was measured in all spinal segments. Figure 3.23 illustrates the spinal parameters of the spinal column, including a) coronal angles, b) sagittal angles, and c) 3-dimensional angles relative to the horizontal plane. Furthermore, the relationship between sagittal and coronal plane values from the motion capture system and radiographs results was investigated. The evaluation of the motion capture system was divided into 2 studies. The first study was to investigate the correlation between the motion capture system and radiographs. The study design was a prospective with

13 AIS patients. The result showed that the motion capture system reported a high correlation ( $r > 0.7$ ) in all spinal parameters. Pearson's correlation coefficients for coronal cervical, upper thoracic, lower thoracic, upper lumbar, and lower lumbar were 0.95, 0.87, 0.79, 0.81, and 0.70, respectively. The correlations for sagittal cervical, upper thoracic, lower thoracic, upper lumbar, and lower lumbar were 0.69, 0.92, 0.96, 0.89, and 0.83, respectively. The second study was then to evaluate the developed application. This study was a pilot study with 5 AIS patients. The data from developed application was compared and evaluated between with and without spinal correction by Scoliosis Apparatus. The research concluded that the developed application could help practitioners to provide the optimal placement of corrective forces while the application could illustrate the result of spinal parameters in 3 dimensions and in real time. Many spinal parameters were studied in this research. However, we adopted some parameters in our study, including the coronal spinal angle (Márkus *et al.*), sagittal spinal angle (SSA), 3-dimensional spinal angle (3DSA), trunk horizontal rotation at the scapula level, and trunk balance.

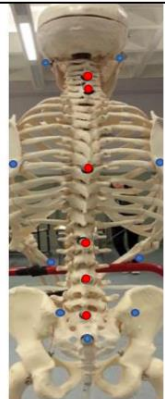


Name of Landmarks	19 Key Anatomical Landmarks of 11 3D Skin Level Parameters (3DSPs)	
LTC1		Left Mastoid Process at C1 Level
RTC1		Right Mastoid Process at C1 Level
C4C5		Just below the spinal process of C4
C7T1		Just below the spinal process of C7
T7T8		Just below the spinal process of T7
T12L1		Just below the spinal process of T12
L3L4		Just below the spinal process of L3
L5S1		Just below the spinal process of L5
S2S3		Just below the spinal process of S2
LTRO		Left inferior angle of scapular at T7/T8 level
RTRO		Right inferior angle of scapular at T7/T8 level
LLRO		Left paraspinous muscles at L3/L4 level
RLRO		Right paraspinous muscles at L3/L4 level
LPSI		Left posterior superior iliac spine
RPSI	Right posterior superior iliac spine	
STT5		Superior part of the sternum at T5 rib level
STT8		Highest point of the sternum
LASI		Left anterior superior iliac spine
RASI		Right anterior superior iliac spine



Figure 3.22 Marker locations from the study of Jang (Jang, 2018).

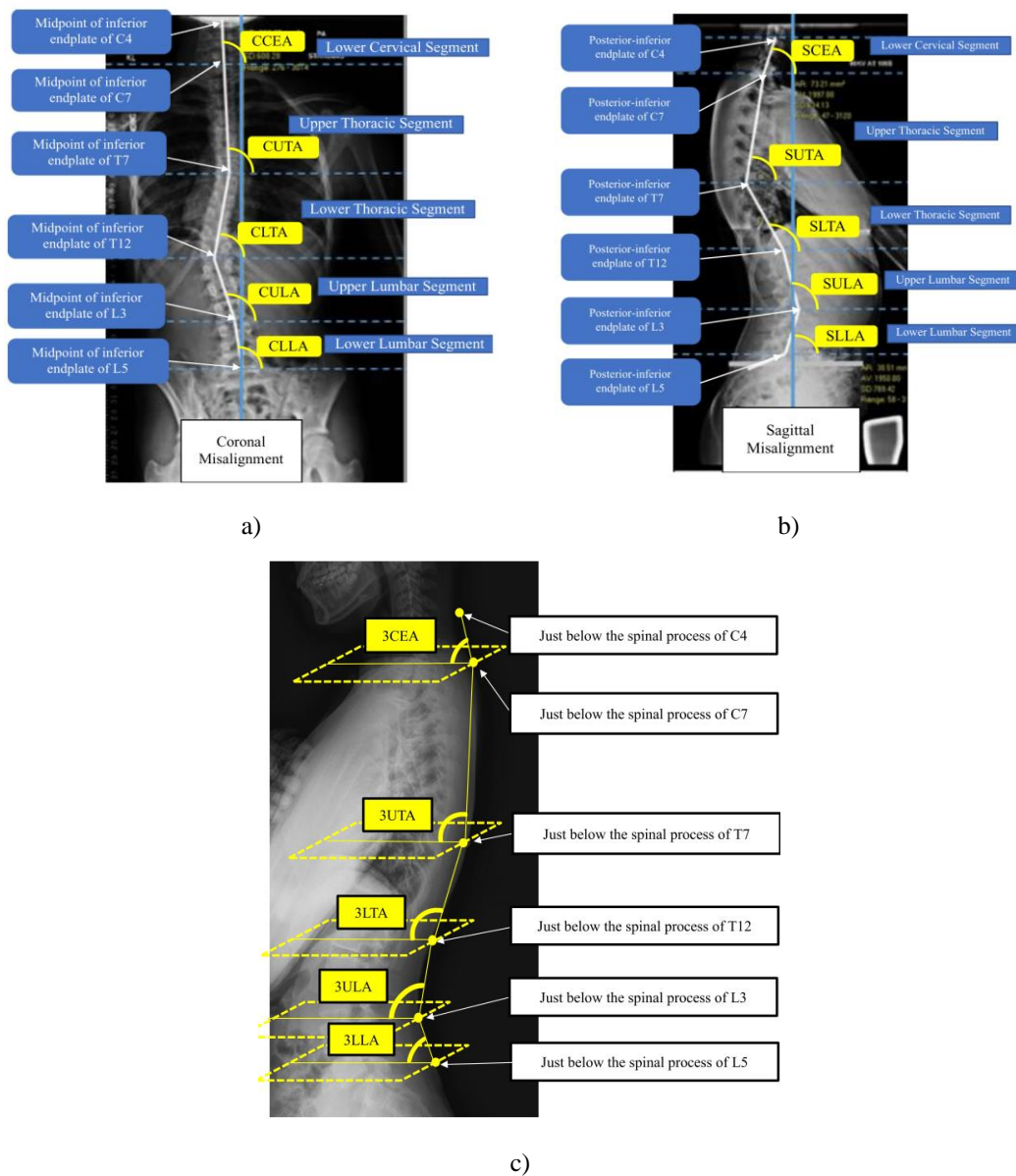


Figure 3.23 The spinal parameters of the spinal column, including a) coronal angles, b) sagittal angles, and c) 3-dimensional angles relative to the horizontal plane (Jang, 2018).

In 2020, Pesenti and colleagues used a motion capture system (6 high-resolution infrared cameras, 100 Hz) to quantify the spinal parameters during static and dynamic movements (Pesenti *et al.*, 2020). The study included 62 AIS patients and was prospective. The radiographs were compared to the static results (standing position) from motion analysis, and the static results were then compared to the dynamic movements (walking 9 metres at a comfortable speed). Figure 3.24a illustrates all marker locations. The spinal parameters of the study were coronal vertical axis (CVA), sagittal vertical axis (SVA), and coronal shoulder tilt. Figure 3.24b illustrates the

markers used to calculate thoracic Cobb angle, lumbar Cobb angle, thoracic kyphosis, lumbar lordosis, pelvic tilt, CVA, and SVA. The result reported a significant correlation between the radiographs and static movement from the motion capture system for most spinal parameters. The r-values for thoracic and lumbar coronal Cobb angles were 0.58 and -0.21, respectively. The r-values for thoracic kyphosis and lumbar lordosis angle were 0.57 and 0.44, respectively. However, the motion capture system's static and dynamic movement was not significantly correlated, and the r-values ranged from -0.109 to 0.229. The study concluded that the motion capture system during static movement well correlated to radiographic results. However, spinal parameters changed from static to dynamic movements in AIS patients.

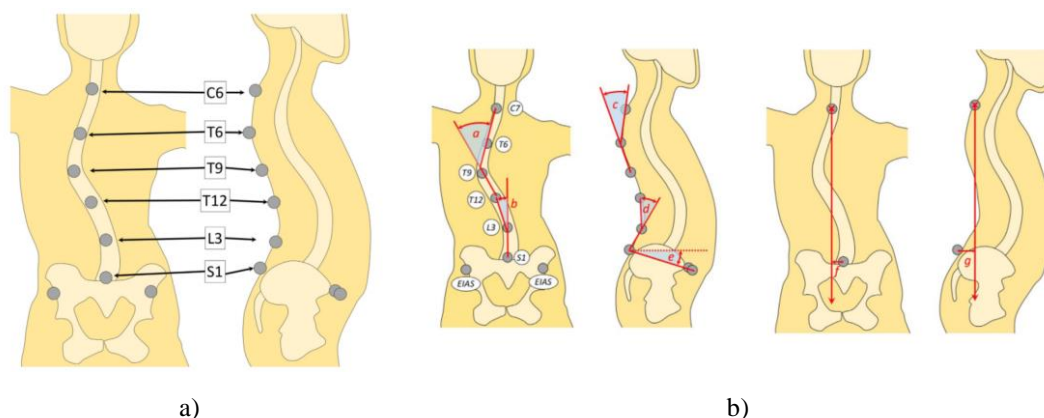


Figure 3.24 Marker locations and used to calculate (a) angle) thoracic Cobb angle, (b) lumbar Cobb angle, (c) thoracic kyphosis, (d) lumbar lordosis, (e) pelvic tilt, (f) CVA, and (g) SVA (Pesenti *et al.*, 2020)

In 2022, Lau and colleagues used a motion capture system (Vicon, Nexus 2.12, 8 cameras, 100 Hz) to study a 3-dimensional spinal proprioception assessment for AIS patients (Lau *et al.*, 2022). The study aimed to develop a new spinal proprioception assessment using 3-dimensional motion analysis. Fifty-nine AIS patients were included in the study. During data collection, participants were asked to perform 3 tasks: trunk flexion-extension, lateral flexion, and axial rotation. Twenty-two markers were attached to the spine, and other anatomical bony landmarks, including C7, T3, T5, T7, T9, T11, L1 to S1, ASIS, PSIS, lateral one-third of the clavicle, inferior angle of the scapula, and costal end of the 12th ribs (fig. 3.25).

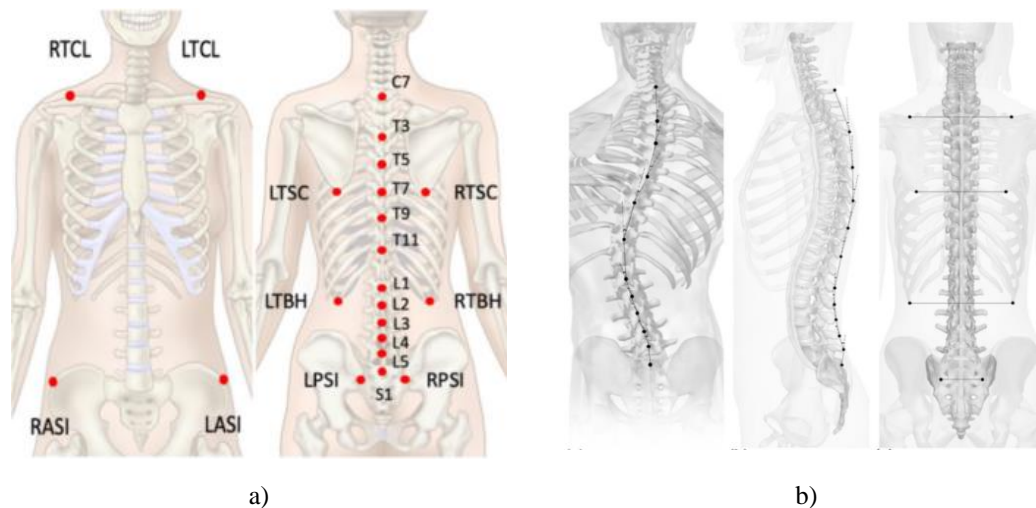


Figure 3.25 Marker locations from the study of Leu and colleagues (Lau *et al.*, 2022)

The outcome measure of this study was proprioceptive, which assessed the total absolute repositioning errors in all anatomical planes between the starting and ending positions of each test. The intraclass correlation coefficient (ICC) was used to assess trial consistency. The first three trials of each trunk test were tested for intra-examiner reliability. The test-retest reliability was performed for the two sets of the 3 trunk tests. In the coronal and sagittal planes, the angles were calculated between the upper and lower markers and performed along the spinal column (fig 3.25a and 3.25b). In the transverse plane, the system calculated the rotation and compared between the upper and lower levels, such as the PSIS relative to the rib cage, the rib cage relative to the scapula, and the scapula relative to the clavicle. The results reported that trunk flexion-extension's intra- and test-retest reliability were 0.464 and 0.722, while trunk lateral flexion were 0.435 and 0.698, and trunk axial rotation were 0.324 and 0.589, respectively. The study concluded that a new spinal proprioception assessment using a 3-dimensional motion capture system showed a high possibility of measuring the proprioception of AIS patients, especially the trunk flexion and extension. Future research should study the 3-dimensional spinal proprioception assessment for AIS patients in various groups.

Table 3.3 Summary of the correlation and difference distance error between radiographs and motion capture systems from the relevant literature.

Author, year	Number of Participants	Mocap system	Correlation with radiographs		Distance diff. error between radiographs and Mocap
			Coronal	Sagittal	
Solomito, Lee and Peterson, 2011	10 (5 AIS and 5 normal subjects)	Vicon	> 0.75		n/a
Schmid et al., 2015	10 AIS	Vicon	Moderate to strong, but not mentioned the exact number in the article		5 - 18 mm error in the vertical and 9 mm error in the horizontal directions
Jang, 2018	13 AIS patients	Vicon with Nexus software, and Motek D-Flow visualisation software	0.95, 0.87, 0.79, 0.81, and 0.70 for C, UT, LT, UL, and LL	0.69, 0.92, 0.96, 0.89, and 0.83 for C, UT, LT, UL, and LL	n/a
Pesenti et al., 2020	62 AIS	High-resolution infrared cameras	0.58 and -0.21 for thoracic and lumbar	0.57 and 0.44 for thoracic kyphosis and lumbar lordosis	n/a

**Note:** AIS = adolescent idiopathic scoliosis, C = cervical, UT = upper thoracic, LT = lower thoracic, UL = upper lumbar, LL = lower lumbar, n/a = not applicable, diff. = difference.

### 3.6. Accuracy of motion capture systems

The previous section introduced motion capture systems and their applications. Motion capture systems are widely used to analyse human movement, such as sitting, standing, jumping, walking, and running. Furthermore, some studies use this system to analyse spinal parameters in scoliosis patients. However, these systems are expensive and beyond the scope of many orthotists, so while current motion capture techniques can give a more scientific evaluation of AIS patients than radiographs and in three dimensions, they are not commonly deployed to do so. One of the aims of this thesis was therefore to develop a new low-cost postural measurement system which could be used for the assessment and treatment of AIS patients. The next section of the review, therefore, looks at the accuracy of existing motion capture systems to give



the levels of accuracy that such a new low-cost system would require. Therefore, this section begins to describe the accuracy of high-cost motion capture systems.

### 3.6.1. Accuracy of high-cost motion capture systems

In 1999, Richards evaluated the accuracy of high-cost motion capture systems. (Richards, 1999). A testing device called Standard Assessment of Motion System Accuracy (SAMSA) was designed and used to assess the accuracy, as shown in Figure 3.26. The SAMSA device had seven fixed markers at known distances from each other. The distance and angle calculated by the systems were compared to the known values. The SAMSA device could be rotated 360° in the transverse plane at 60 rounds per minute (rpm). The tested system consisted of Ariel, CODA, Elite, Motion Analysis, Peak, Qualisys, and Vicon. In 2020, Topley and Richards re-evaluated the accuracy of high-cost motion capture systems used recently (Topley and Richards, 2020). The tested system consisted of Vicon, Qualisys, OptiTrack, and Motion Analysis. The SAMSA device developed in the previous research continues to be used in this experiment (fig. 3.26).

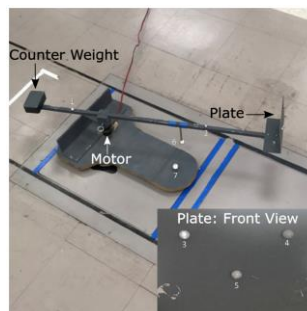


Figure 3.26 SAMSA device (Richards, 1999, Topley and Richards, 2020).

The dimension of SAMSA was measured by the 3D Fusion FaroArm digitizer to get an accurate measurement. The 3D Fusion FaroArm digitizer has an absolute accuracy of 0.035 mm (FARO Fusion tech sheet, 2017). Three sets of markers on SAMSA were analysed in this experiment.

In marker set 1, the top two rotating arm markers or between markers 1 and 2, the difference distance error of markers ranged from 0.539 mm (in Vicon, 16MP) to 0.585

mm (in Motion Analysis, 4MP). The maximum errors were between 0.417 mm (in OptiTrack, 4.1MP) and 1.030 mm (in Motion Analysis, 1.3MP). As a result, all high-cost motion capture errors were lower than 1 mm while measuring the top two rotating arm markers. The average difference distance error from all systems was 0.35 mm, with an average standard deviation of 0.087 mm and an average maximum error of 0.61 mm.

In marker set 2, the top two plate markers or between markers 3 and 4, the difference distance error of markers ranged from -0.003 mm (in Motion Analysis, 1.3MP) to 0.126 mm (in Vicon, 4MP). The maximum errors were between 0.182 mm (in OptiTrack, 4.1MP) and 0.991 mm (in Motion Analysis, 1.3MP). As a result, the errors in this marker set were also lower than 1 mm. The average difference distance error from all systems was 0.08 mm, with an average standard deviation of 0.12 mm and an average maximum error of 0.48 mm.

In marker set 3, for calculating the angle errors or between markers 3, 4, and 5, the difference angle error ranged from  $-0.235^{\circ}$  (in Qualisys, 12MP) to  $-0.012^{\circ}$  (in Motion Analysis, 1.3MP). The maximum errors were between  $0.284^{\circ}$  (in Vicon, 16MP) and  $2.083^{\circ}$  (in Motion Analysis, 1.3MP). The average difference angle error from all systems was  $0.16^{\circ}$ , with an average standard deviation of  $0.15^{\circ}$  and an average maximum error of  $0.74^{\circ}$ .

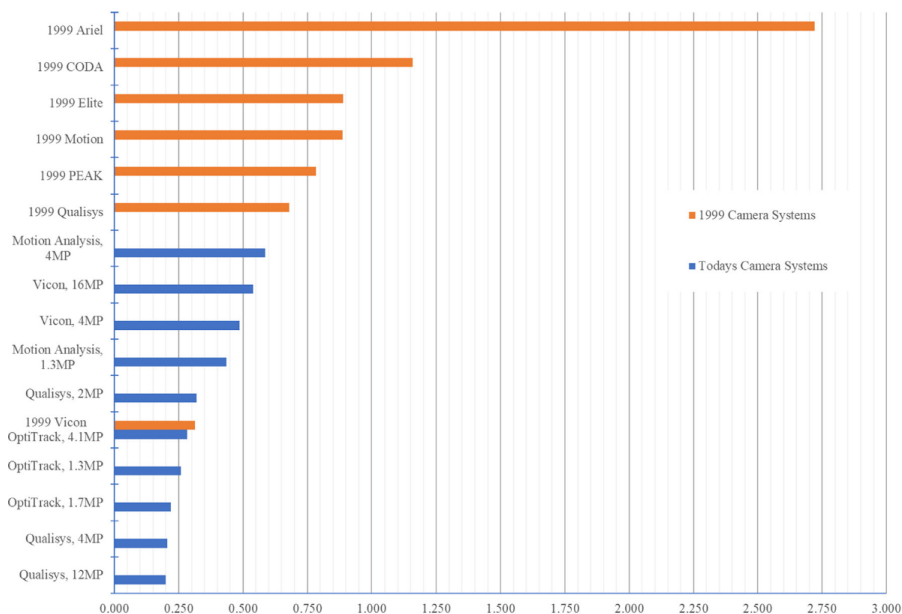


Figure 3.27 The absolute error (mm) between the high-cost motion capture systems in 1999 and 2020 (Topley and Richards, 2020)

The study mentioned that motion capture systems in 2020 significantly improved compared with motion capture systems in 1999. Several improvements were achieved in the high-cost motion capture system in 2020, such as the quality of cameras, the tracking software, gap filling, and the signal filtering technique. The errors were reduced from a few millimetres in 1999 to a half millimetre in 2020. Figure 3.27 illustrates the absolute error between the systems in 1999 and 2020.

Overall, the error of current motion capture systems when recording dynamic motion of the SAMSA device was lower than 0.5 mm for position and  $0.5^\circ$  for rotation. This was an order of magnitude better when compared to the systems available in and around 1999. This improvement is due to the increased image quality of the camera and the introduction of Megapixel cameras. During walking and high-speed movement, where the markers exhibit velocity, a comet tail effect can occur, spreading out the marker image on the camera image plate. Even though the systems were highly accurate, the effect still occurs in measuring dynamic motion.

In a scoliosis application, the spinal parameters were quantified in static position by asking the patients to stand upright, not to quantify the spinal parameters in dynamic movement. Hence comet tailing will not occur, so these values represent the maximum likely errors. The previous scoliosis articles mentioned that the progression of the coronal curve was considered clinically significant when the magnitude of change was greater than  $5^\circ$  (Emans *et al.*, 1986, Zaborowska-Sapeta *et al.*, 2011a, Pepke *et al.*, 2023). The mean difference between the Cobb angle in AP and PA views from repeated evaluation of radiographs was  $2.4^\circ$  (DeSmet *et al.*, 1982), and the difference in Cobb angle measurement between clinicians was up to  $5^\circ$  (Morrissy *et al.*, 1990, Carman, Browne and Birch, 1990). However, no research mentioned the minimal clinically important change (MCIC) in distance or error of the distance in scoliosis outcome measurement.

Current motion capture systems would appear excellent for quantifying the spinal parameters in scoliosis patients. However, the systems are expensive and, in the author's opinion, overkill for scoliosis applications.

### 3.6.2. Accuracy of low-cost motion capture systems

A low-cost motion capture system is an alternative system for capturing the movement of an object or people using a low-cost camera and developing software. Some systems can analyse movement in 2 dimensions, primarily in the sagittal plane, while others can analyse movement in 3 dimensions. A variety of camera types have been used in low-cost motion capture systems, including smartphone and tablet cameras, Microsoft Windows Kinect, and digital cameras. As a result, the following section describes the accuracy of low-cost motion captures that are currently being used to analyse human movement.

In 2012, Eltoukhy and colleagues investigated the accuracy of a low-cost motion capture system (Eltoukhy *et al.*, 2012). The study recorded the human movement using a digital video and imported the data to analyse the movement in the 2D software, Dartfish ProSuite 5.5 (60 Hz). The results from Dartfish software were compared with the Vicon MXF40 (10 cameras, 60 Hz). One male adult healthy subject was recruited in the study, and the data were collected during simple squats. Sixteen markers were attached to the anatomical landmarks following the plug-in-gait model. The variables of this study were joint angles and moments at the ankle and knee. The result reported that the marker positions differed by -10 to 20 mm, and the trajectory difference errors were between 5 and 10 mm. The study mentioned that the rough difference of marker position was plus or minus 5 mm, and Dartfish ProSuite 5.5 could track the sagittal movement at the ankle and knee during squatting.

In 2013, Clark and colleagues compared the accuracy of the Kinect V1 (30 Hz) with Vicon Nexus V1.5.2 (12 cameras, 120 Hz) (Clark *et al.*, 2013). Twenty healthy subjects (10 participants were male and another 10 were female) were recruited in the study, and the mean age was 27.1 years (SD = 4.5). The entire system was intended to be used for gait training, but this article only focused on measuring lateral trunk leaning while walking. Twenty-two markers were attached to the anatomical landmarks. The results showed that the agreement before calibrating the Kinect system was greater than 2° (mean error = 3.2°, SD = 2.2). The error was significantly lower than 2° after the global calibration of the Kinect system (mean error = 1.7°, SD = 1.5). According to the study, the low-cost system could be used effectively for a

lateral trunk lean gait training program, and the Kinect could provide real-time feedback during training.

In 2014, Yeung and colleagues compared the accuracy of the Anterior SDK Beta V2 Kinect (30 Hz) with Vicon MXF40 (8 cameras, 200 Hz) (Yeung *et al.*, 2014). Ten male healthy subjects were recruited in the study, and the mean age was 24.79 years (SD = 2.68). The study measured the total body centre of mass (TBCM) during standing sway. Forty-six markers were attached to the anatomical landmarks. The result reported that the Kinect system correlated highly to the Vicon system. The mean RMSE was 4.38 mm, and the mean Pearson correlation coefficient was 0.94.

In the same year, Pfister and colleagues (Pfister *et al.*, 2014) also compared the accuracy of the Kinect Xbox 360 (Brekel Kinect software) with Vicon MX (10 cameras, 120 Hz, Nexus 1.7). The data were collected during treadmill walking and 3 running velocity (3.0, 4.5, and 5.5 mph). Twenty healthy subjects (9 participants were male and another 11 were female) were recruited in the study, and the mean age was 27.4 years (SD = 10). Participants wore the body suit, and markers were attached to the anatomical landmarks following the plug-in-gait model. The variables of this study were sagittal peak angle at knee and hip flexion and extension and stride timing. The result reported that the Kinect could report the normal gait during data collection. However, the Kinect provided underestimation in joint flexion and overestimation in extension. The Pearson correlation coefficient varied and ranged between 0.30 and 0.80. The angle error was greater than 5° at both joints, ranging from -19.32° to 5.14°. When the participant walked slowly, the error was smaller. The study also showed that knee movement correlated better than hip movements.

Additionally, Childer and colleagues (Childer *et al.*, 2014) investigated the accuracy of Dartfish Express on iPad 2 (2 devices, 30 Hz) compared to Vicon 3D (8 cameras, 100 Hz, Vicon Bodybuilder 3.6.1). The data were collected during treadmill walking with 31 healthy subjects. Thirty-nine markers were attached to the anatomical landmarks following the plug-in-gait model. The variables of this study were the maximum knee flexion and extension angles during the stance and swing phases. The result reported that the angle error was approximately 10°. The study mentioned that this error did not preclude clinical use and was still a cost-effective tool.

In 2015, Kharazi and colleagues investigated the accuracy of the Anterior Kinect V2 (30 Hz) versus Vicon MX (4 cameras, 120 Hz) (Kharazi *et al.*, 2015). The study recruited 21 healthy volunteers. The systems were measured while walking at slow, normal, and fast self-selected speeds. The variables in this study were sagittal plane ankle, knee, and hip movements. The result reported that the Pearson correlation coefficient for the ankle, knee, and hip was 0.09, 0.95, and 0.98, respectively. Furthermore, the angle errors at the ankle, knee, and hip were 22.03°, 5.33°, and 5.29°, respectively. The study concluded that the Kinect V2 could track movement in the knee and hip but not in the ankle joints in sagittal plane movements.

Belyea and colleagues (Belyea *et al.*, 2015) investigated the accuracy of the Kinesio Capture (Spark Motion) 2D app on iPad 2 (30 Hz) to Vicon Nexus (7 cameras, 240 Hz, Visual 3D). The data were collected during the drop vertical jump. Twenty-two healthy subjects (11 participants were male and another 11 were female) were recruited in the study, and the mean age was 21 years (SD = 1.4). Twenty-nine markers were attached to the anatomical landmarks. The variables of this study were the joint angle at the knee and hip in sagittal and frontal planes. The result reported that the correlation ranged from 0.48 (frontal knee angle) to 0.77 (knee flexion angle). The study concluded that the handheld tablet system could be used to assess and evaluate joint alignment during drop vertical jumps. However, it is not appropriate to measure the absolute joint angle.

Albert Kim and colleagues (Kim *et al.*, 2015) compared between SmartGait (iPhone 5s and Mondizen Inc wide-angle lens) and GAITRite (CIR System) walkway. The data were collected during walking at slow, preferred, and fast speeds, and the variable of this study was spatiotemporal parameters, including step length, step width, step time, gait speed, and double support time. The result reported that the intra-class correlation coefficients (ICCs) were excellent, ranging from 0.731 to 0.982, and the absolute error ranged between 1 and 11 mm.

In 2017, Grey and colleagues compared the Anterior Kinect V2 (30 Hz) to the Vicon MX-T40S (8 cameras, 100 Hz, and Nexus) (Gray *et al.*, 2017). Thirty-eight healthy subjects (20 males and 18 females) were recruited in the study, with a mean age of 24.79 years (SD = 2.68). The data were collected during 5 trials of the drop vertical jump (total trials = 190). Sixteen markers were attached to the anatomical

landmarks. The result reported that the intraclass correlation coefficient (ICC) was excellent, with values of 0.84 for initial contact (IC) and 0.95 for peak flexion (PF). Furthermore, the Kinect V2 identified the movement with 95.8% reliability to Vicon (182 of 190 trials).

In the same year, Schurr and colleagues (Schurr *et al.*, 2017) compared the 2D system from Canon digital cameras (3 cameras, 60 Hz, Kinovea software) to Flock of Birds 6 degree of freedom (DOF) electromagnetic tracking system (144 Hz, Ascension Tech Inc). The systems were measured during single-leg squats on each leg. Twenty-six healthy subjects were recruited, and the mean age was 22.26 years (SD = 2.99). Fifteen markers were attached to the anatomical landmarks. The variables of this study were frontal and sagittal plane joint displacements at the ankle, knee, hip, and trunk. The result reported that the correlation in the sagittal plane was moderate to strong, ranging from 0.51 to 0.93, and the correlation in the frontal plane was 0.31 at the knee joint (p-value <0.05). Other variables were not statistically significant between both systems. The Bland-Altman analysis in the sagittal plane revealed strong agreement in the average mean difference of joint displacement, ranging between 0.74° and 3.12° and between -8.72° and 7.92° in the frontal plane. The study concluded that the 2D low-cost system was adequate for tracking the sagittal plane movement during single-leg squats.

In 2018, Reither and colleagues compared the Kinect V1 (30 Hz), Kinect V2 (30 Hz), and 3D video motion capture (8 cameras, 60 Hz, Cortex) (Reither *et al.*, 2018). One adult male healthy subject was recruited in the study. Eighteen markers were attached to the anatomical landmarks. The systems measured the upper limb extremity motions for 6 trials. The result reported that the forward reaching of the upper limb had a similar result between Kinect V2 and 3D video motion capture (VMC), but Kinect V1 was underestimated by 7 cm. In the shoulder motion, the Kinect V1 and V2 significantly overestimated the VMC by 5° to 27°. The study concluded that both Kinects could detect the movement pattern well but poorly measure an absolute range of motion magnitude.

In 2020, Pilar and colleagues compared the digital camera (Nikon D3200, 50Hz) with Kinovea software (version 0.8.15) to the Vicon system (8 cameras, 100Hz, Nexus 1.8.5 software) (Fernández-González *et al.*, 2020). The data were collected during the

initial contact phase of walking and focused on the kinematics of hip, knee and ankle joints in the sagittal plane. Fifty healthy subjects (26 females and 24 males) were recruited in the study, and the mean age was 21.62 years (SD =2.62). The result reported that the intra-rater reliability was a good correlation for all joints with greater than 0.85. The inter-rater reliability was excellent for all joints with greater than 0.90. The Bland–Altman analysis of the magnitude of disagreement was approximately  $\pm 5^\circ$  for intra-rater reliability,  $\pm 2.5^\circ$  for inter-rater reliability and around  $\pm 2.5^\circ$  to  $\pm 5^\circ$  for Kinovea versus Vicon.

Table 3.4 Summary of the accuracy of low-cost motion capture systems.

Author and Years	Low-Cost System	High-Cost System	Subject type	Activity	Accuracy
Eltoukhy et al., 2012	A digital video with Dartfish ProSuite 5.5	Vicon MXF40 (10 cameras, 60 Hz)	1 healthy	Simple squats	Marker position difference: -10 to 20 mm Trajectory difference: 5 to 10 mm
Clark et al., 2013	Kinect V1	Vicon Nexus V1.5.2 (12 cameras, 120 Hz)	20 healthy (10 males, 10 females)	Lateral trunk leaning while walking	The agreement: mean error before calibration was $3.2^\circ$ (SD=2.2). After calibration was $1.7^\circ$ (SD=1.5).
Yeung et al., 2014	Anterior SDK Beta V2 Kinect	Vicon MXF40 (8 cameras, 200 Hz)	10 healthy (10 males)	Total body centre of mass (TBCM) during standing sway	Mean of RMSE: 4.38 mm. Mean Pearson correlation: 0.94
Pfister et al., 2014	Kinect Xbox 360	Vicon MX (10 cameras, 120 Hz, Nexus 1.7)	20 healthy (9 males, 11 females)	Treadmill walking and three running velocity (3.0, 4.5,	Pearson correlation: between 0.30 and 0.80. Angle error: between $-19.32^\circ$ to $5.14^\circ$ .



Author and Years	Low-Cost System	High-Cost System	Subject type	Activity	Accuracy
				and 5.5 mph)	
Childers et al., 2014	Dartfish Express on iPad 2 (2 devices)	Vicon 3D (8 cameras, 100 Hz, Vicon Bodybuilder 3.6.1)	31 healthy	Treadmill walking	Angle error: 10°.
Kharazi et al., 2015	Anterior Kinect V2	Vicon MX (4 cameras, 120 Hz)	21 healthy	Walking at slow, normal, and fast self-selected speeds	Pearson correlation: 0.09 at ankle, 0.95 at knee, and 0.98 at hip. Angle errors: 22.03° at ankle, 5.33° at knee, and 5.29° at hip.
Belyea et al., 2015	Kinesio Capture (Spark Motion) 2D app on iPad 2	Vicon Nexus (7 cameras, 240 Hz, Visual 3D)	22 healthy (11 males, 11 females)	Drop vertical jump	Correlation: 0.48 (frontal knee angle) to 0.77 (knee flexion angle).
Kim et al., 2015	SmartGait (iPhone 5s and Mondizen Inc wide-angle lens)	GAITRite (CIR System) walkway	Not mention	Walking at slow, preferred, and fast speeds	Intra-class correlation: between 0.731 to 0.982. Absolute error: between 1 and 11 mm.
Grey et al., 2017	Anterior Kinect V2	Vicon MX-T40S (8 cameras, 100 Hz, and Nexus)	38 healthy (20 male, 18 female)	Drop vertical jump	Intra-class correlation: between 0.84 and 0.95. Reliability: 95.8%
Schurr et al., 2017	Canon digital cameras (3 cameras)	Flock of Birds 6 degree of freedom	26 healthy	Single-leg squats	Correlation in the sagittal plane: between 0.51 and 0.93.

Author and Years	Low-Cost System	High-Cost System	Subject type	Activity	Accuracy
	with Kinovea software	electromagnetic tracking system (144 Hz)			Correlation in the frontal plane: 0.31 at the knee. Agreement in the average mean difference of joint displacement: between 0.74 and 3.12° in sagittal and between -8.72° and 7.92° in the frontal plane.
Reither et al., 2018	Kinect V1 and Kinect V2	3D video motion capture (8 cameras, 60 Hz, Cortex)	1 healthy (1 male)	Upper limb extremity motions	In forward reaching movement: Kinect V2 and 3D video motion capture (VMC) had a similar result, but Kinect V1 was underestimated by 7 cm. In the shoulder motion: Kinect V1 and V2 overestimated 5° to 27°.
Pilar et al., 2020	Digital camera (Nikon D3200) with Kinovea software (version 0.8.15)	Vicon system (8 cameras, 100Hz, Nexus 1.8.5 software)	50 healthy (24 males, 26 females)	Initial contact phase of walking	Intra-rater reliability: >0.85. Inter-rater reliability: >0.90. Disagreement: ±5° for intra-rater reliability, ±2.5° for inter-rater reliability and around ±2.5° to ±5° for Kinovea versus Vicon

- **Raspberry Pi and its applications**

A Raspberry Pi is a small computer developed in the United Kingdom by the Raspberry Pi Foundation (Raspberry Pi, 2023). It can plug into a monitor, keyboard, mouse, electronic board, and Raspberry Pi camera. It is a capable little device that enables the user to write computer programming, connect to the internet and Bluetooth, play a game, and record video or pictures from a Raspberry Pi camera. Raspberry Pi has been used in many applications, such as robotics and weather monitoring, but not in human motion analysis.

Karthikeyan and colleagues (Karthikeyan *et al.*, 2023) conducted a systematic review of Raspberry Pi in terms of the use, challenges, benefits, and drawbacks of Raspberry Pi and its applications. The research summarized that the Raspberry Pi was developed for many applications, such as education, robot controllers, Web servers, motion capture security cameras, and healthcare applications.

Some researchers developed systems using Raspberry Pi in healthcare applications. Banerjee and colleagues used the Raspberry Pi to monitor the movement and body posture of elderly patients in a hospital. The system could detect the change and send the notification to caregivers' mobile phones (Banerjee *et al.*, 2013). Another research by Sachian and colleagues used the Raspberry Pi to measure and monitor the air quality in a hospital (Sachian *et al.*, 2020). Lavanya, Lavanya. and Divyabharathi used the Raspberry Pi to measure and monitor the heart rate of the patients. The doctors could assess the recorded data to monitor the heart rate and prevent the risk of health problems (Lavanya, Lavanya and Divyabharathi, 2017). Kamal and Ghosal conducted a similar research, they used the Raspberry Pi to measure and monitor the patient body temperature, heartbeat, and body position movements. The recorded data was wirelessly sent from the remote area to the healthcare services (Kamal and Ghosal, 2018).

Some researchers used Raspberry Pi to develop the smart home system. Sruthy and George used Raspberry Pi to develop a home security system to detect the event and send the notification to the users' mobile phone or email. The system could also make a phone call, capture the VDO of the event, and live stream the VDO on a webpage. The user could control the system remotely via an internet connection (Sruthy and George, 2017). Idris, Alkooheji and Jameel developed a smart home automation using Raspberry Pi. The system could detect and analyse visitors' faces when they were in front of the door. If the faces were authorised, the door could automatically open. An additional option for unauthorised faces was that the system could make an automatic video call with the user. The users could lock and unlock the door remotely. The system could also report the body temperature of the visitors for COVID-19 symptoms screening (Idris, Alkooheji and Jameel, 2022). Sornalatha and Kavitha used Raspberry Pi to develop the smart museum. The system could detect the movement of visitors

and automatically provide information about the artwork. The purpose of the system was to gain more attention of visitors (Sornalatha and Kavitha, 2017)

Some researchers used Raspberry Pi to develop a system to detect environmental changes. Balasubramaniyan and Manivannan used Raspberry Pi to measure, analyse, and monitor the air quality (Balasubramaniyan and Manivannan, 2016). A similar study from Kumar and Jasuja also used Raspberry Pi to measure, analyse, and monitor the air quality (Kumar and Jasuja, 2017). Ate and Abdelrahim used Raspberry Pi to develop a system to control and maintain the temperature constant in the chemical storage room (Ate and Abdelrahim, 2018).

Some researchers used Raspberry Pi to develop a smart agricultural system. Amer, Mudassir and Malik used Raspberry Pi to develop an autonomous robot for agriculture called AgriBot. The robot could perform various tasks, such as seeding, weeding, fertilizer spraying, and insecticide spraying (Amer, Mudassir and Malik, 2015). Chaudhari used Raspberry Pi to measure the temperature and moisture from a windmill at a farm (Chaudhari, 2019). Danita and colleagues conducted similar research. They used Raspberry Pi to measure and control moisture, temperature, and humidity inside the greenhouse (Danita *et al.*, 2018). Cabaccan, Cruz and Agulto also conducted similar research. They used Raspberry Pi to monitor the light, temperature and humidity around the agricultural environment (Cabaccan, Cruz and Agulto, 2017). Furthermore, Tian-xing Xu and Feng-ying used Raspberry Pi to monitor the temperature, humidity, and oxygen concentration in the granary. The system could also capture the image of pests in the grain pile, and the system then sent the warning to the users (Xu and Feng-ying, 2021).

Lastly, Sooryavanshi, Urganlawar and Bhosle used Raspberry Pi to develop the system for defence purposes. They developed an autonomous vehicle robot for spying, bomb disposal units, and threat detection (Sooryavanshi, Urganlawar and Bhosle, 2017).

In these current applications, the Raspberry Pi was integrated with other sensors or additional hardware to receive the information and perform the tasks. The sensors or additional hardware consisted of a humidity sensor, temperature sensor, air pressure sensor, gas sensor, movement sensor, and camera. Focusing on the applications that use the Raspberry Pi with a camera, some studies developed the system to capture

images and live VDO streaming and then send the information to users online or show it on the webpage. Furthermore, some studies have developed a system that performs image processing to detect the movement of humans or pests and face detection and recognition. OpenCV was used for the image processing technique. The current articles focused on system development and demonstrate the possibility of using the system to assist users in practice. However, no research has studied the system's accuracy or the error in the distance of the image processed by the system. Even though the Raspberry Pi and camera were widely used in many applications, surprisingly, no one has used them to analyse human motion and scoliosis applications.

## Summary

Current human motion capture systems are excellent for quantifying the spinal parameters in scoliosis patients. However, they are expensive and overkill for scoliosis applications. The error of the current low-cost motion capture systems varied and depended on the type of camera, software, and testing activities. The distance error ranged from 5 mm to 20 mm and  $-19.32^{\circ}$  to  $27^{\circ}$  across the history of motion capture development, and modern high-end systems are highly accurate. No research mentioned the standard error that a low-cost motion capture system must achieve. Another reason that influenced the error was the speed of movement. Pfister and colleagues mentioned that the error was smaller when the participant walked slowly. In contrast, the error was larger when the participant walked faster (Pfister *et al.*, 2014).

The error of current low-cost motion capture system was relatively high. Some systems could be used in clinical practice to assist clinicians in evaluating the outcome measurement in dynamic activities (such as sitting-to-standing and walking), but not for more general outcome measurement in clinical research. Furthermore, the current low-cost motion capture systems were unsuitable for scoliosis applications because the kinematics error was high and greater than the minimal clinically important change or curve progression of the coronal curve at  $5^{\circ}$ . Therefore, a new low-cost motion capture system or a new low-cost postural measurement system with suitable accuracy was required to measure static spinal pose. Technology was available to develop this

system (Raspberry Pis and megapixel cameras) with suitable accuracy for a scoliosis application and suitable for clinical use.

No research mentioned the error of measuring tools in motion capture systems, which is of suitable accuracy for scoliosis applications. This thesis suggested the error of measuring tools in motion capture systems by using a calculation from a maximum of  $5^\circ$  of curve progression and the spinal column's length. Figure 3.28 illustrates the geometry to calculate the error for a motion capture system in a scoliosis application. Point A to point B (X-value) is the distance of spinal segments, including the spinal segments of coronal-upper-thoracic-angle (CUTA), coronal-lower-thoracic-angle (CLTA), coronal-upper-lumbar-angle (CULA), and coronal-lower-lumbar-angle (CLLA). Y-value is the error in distance or how the data deviates from its position. If the Y-value is small, the error is small, and the system has high accuracy. In contrast, if the Y-value is high, the error is high, and the system has low accuracy. Theta ( $\Theta$ ) is an angle of a maximum of  $5^\circ$  of curve progression when the clinicians evaluate the curve progression after treatment. However, we used  $2.5^\circ$  in the calculation because the error can go around upward and downward. For the spinal segment length, we used the Euclidean distance of the spinal segments from participants in the Clinical experiment in this thesis. The average spinal segment length (Table 3.5) in CUTA, CLTA, CULA, and CLLA was 176.93 mm, 112.30 mm, 57.23 mm and 52.43 mm, respectively.

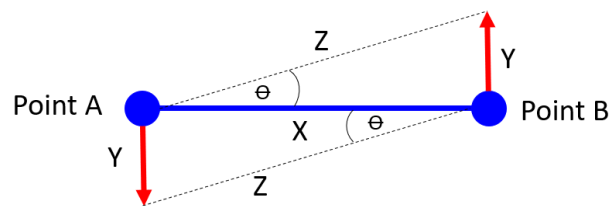


Figure 3.28 Geometry to calculate the error for a motion capture system in a scoliosis application.

Table 3.5 The possible errors of a motion capture system that is still practical for a maximum of  $5^\circ$  of error.

Spinal Segments	Spinal segment length (mm)	Possible Error, 2Y (mm)
CUTA	176.93	15.43
CLTA	112.30	9.80

Spinal Segments	Spinal segment length (mm)	Possible Error, 2Y (mm)
CULA	57.23	4.99
CLLA	52.43	4.57

Table 3.5 shows that the error varied between the spinal segment lengths. A longer length could allow a bigger error that is still practical for a maximum of 5° of curve progression. In contrast, a shorter length of the spinal segment could require a smaller error to measure the same accuracy of angle. The system should have sufficient accuracy to calculate spinal parameters in scoliosis to within 5°. Therefore, we proposed that the error of a motion capture system to quantify spinal parameters in scoliosis applications should be lower than 4.57 mm (or approximately 5 mm).

### 3.7. Casting frames for scoliosis

Should a suitable low-cost motion capture or low-cost postural measurement system be able to measure spinal outcome measurement with suitable accuracy, it would need to do so during the casting process for the orthotic treatment. As indicated in the introduction to this thesis, it would be helpful to deploy a casting frame to assist the clinicians while casting the AIS patients. Hence, the postural measurement system would need to work alongside the casting frame. For this reason, this review also includes a review of existing casting frames.

Casting is a method of capturing a patient's limb shape and using it to create prosthetic and orthotic devices. Traditionally, the practitioners use a plaster bandage to capture the patient's shape. The plaster bandage must be soaked in water before casting, and the water must be well absorbed inside the bandage. The plaster bandage is then twisted to remove excess water before being wrapped around the patient's limb. The practitioners shape the plaster bandage on the patient's limb and apply forces to correct the deformity. The practitioners wait until the plaster bandage is set before removing it from the patient's limb. It usually takes between 10 and 15 minutes. The cast is then removed, and the patient is cleaned. The cast after removing from the patient is called a negative cast. Practitioners evaluate the negative cast's quality. If the

quality is poor, the casting must be repeated. If the cast is of good quality, the practitioners will use it for subsequent processes such as plaster modification, device fabrication, and fitting the device to the patient.

In scoliosis casting, the practitioners perform techniques similar to those described above. Because several forces are required to correct the spine, 2 to 3 practitioners are usually required. Practitioners must be skilled, work well together, and understand the concept of 3-dimensional spinal correction for scoliosis deformity to produce a high-quality negative cast and spinal orthosis. Some researchers created the casting apparatus to help practitioners when casting scoliosis. As a result, the casting apparatuses are described in this section.

In 1964, Cotrel invented the casting frame called EDF (Elongation, derotation, and flexion) Cotrel's frame (Cotrel and Morel, 1964). The frame provides axial elongation or spinal traction by pulling the strap at the head, leg, or pelvis. In coronal plane correction, practitioners use the hand to apply the force to correct the deformity. The kyphosis and lordosis can be maintained in the sagittal plane while lying on the frame. In the horizontal plane, practitioners place the strap from one side of the casting frame, then roll the strap around the patient's trunk and attach the strap to another side of the frame (fig. 3. 29). Nowadays, the casting frame is used mostly in casting infantile and juvenile scoliosis (Sanders and D'Astous, 2009, Sanders, Johnston and D'Astous, 2012, Canavese *et al.*, 2016, Dede and Sturm, 2016, Sanders, 2016, Ballhause *et al.*, 2019, Mahajan *et al.*, 2020). However, it is not a common in AIS casting.

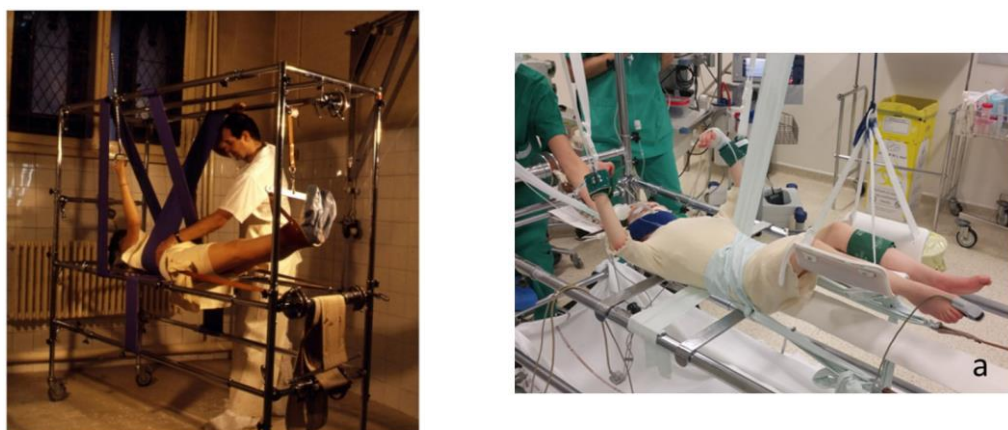


Figure 3.29 EDF Cotrel's frame created by Cotrel (Mauroy *et al.*, 2014, Canavese *et al.*, 2016)



A similar design and function of a casting frame is a Risser frame (Fayssoux, Cho and Herman, 2010). This casting frame (fig. 3. 30) was invented and is widely used in North America. Similar to Cotrel's frame, the patient must lay supine on the frame. The practitioners pull the straps at the head and pelvis to apply the traction force to strengthen the spine. After that, practitioners apply the corrective and counter forces on the patient's trunk to correct the deformity.



Figure 3.30 Risser frame (Fayssoux, Cho and Herman, 2010)

The advantages of these casting frames are to elongate or strengthen the spine and derotate the spine using the strap (Cotrel's frame) and forces applied from the practitioner's hands. However, the patient must be laid down on the frame, which might make it difficult to control the thoracic kyphosis and lumbar lordosis.

In 1994, Barry McCoy invented a casting frame called the Apparatus for Forming a Scoliosis Brace (McCoy and Barry, 1996). The apparatus was also granted a patent in the United States. Figure 3.31 illustrates the detailed structure of the casting apparatus. The apparatus consists of a bed and adjustable manipulators, each with a force sensor at the tip. During casting, the patient lies supine on the bed, and the practitioners place the adjustable manipulators on the patient's lateral side and at the corrective and counterforces' locations. The manipulator's length is then adjusted to compress the trunk slightly. After achieving the optimal spine alignment, practitioners measure the force at the corrective and counterforce areas. Although the casting apparatus can apply forces to correct the spine, it can only correct the spine in one plane (the coronal plane). Surprisingly, no further research has conducted on this casting apparatus in clinical use.

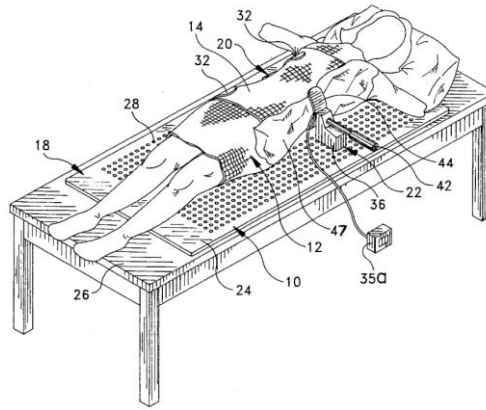


Figure 3.31 Apparatus created by Barry McCoy (McCoy and Barry, 1996).

Another casting frame called a custom Providence brace standing frame (fig. 3.32) was used to assist casting the Providence brace (Edmond *et al.*, 2015, Edmond *et al.*, 2017). Patient stand upright and put the back agent the casting board. After that, the practitioners place the adjustable manipulators on the patient's lateral side and at the corrective and counterforces' locations. The manipulator's length is then adjusted to compress the trunk slightly. Although the casting apparatus can apply forces to correct the spine, it can only correct the spine in one plane (the coronal plane) similar to the casting apparatus from Barry McCoy. There was no research study the effectiveness of this casting frame. However, the cast is widely used during casting for the Providence brace and some use this frame and integrate with ultrasound system to quantify the spinal deformity.

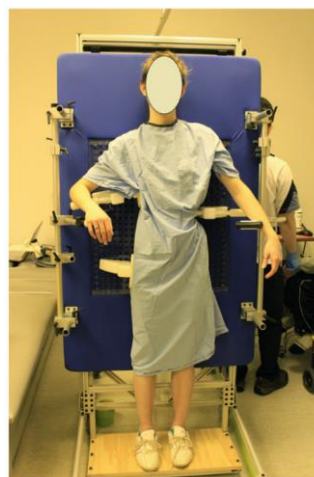


Figure 3.32 A custom Providence brace standing frame (Edmond *et al.*, 2015, Edmond *et al.*, 2017)

In 2018, Jang introduced a Scoli-Standing Frame (fig. 3.33) to evaluate the change of spinal parameters before and during applying the forces to correct the deformity (Jang, 2018). The manipulators attached to the frame can be adjusted in various positions and directions to correct the spine in 3 dimensions. Moreover, the load cells were attached to the tip of the manipulators, and the forces could be measured while compressing the load cells on the patient skin during the experiment.

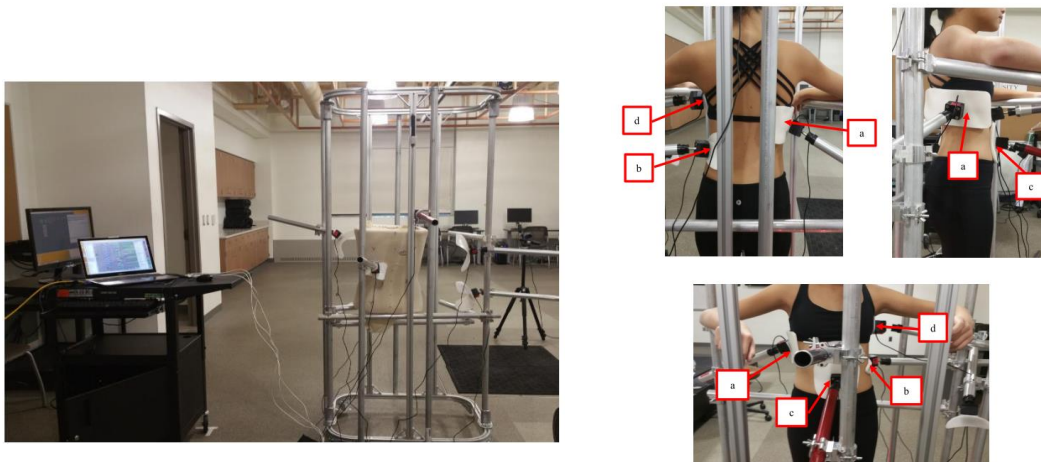


Figure 10.2 - Posterior, Lateral, and Anterior Views of a Subject in the Scoli-Corrective Frame while Corrective Forces are Applied: (a) Thoracic Corrective Force, (b) Lumbar Corrective Force, (c) Abdominal Pressure, and (d) Axillary Force

Figure 3.33 Scoli-Standing Frame created by Jang (Jang, 2018).

The casting frames for scoliosis were introduced to assist clinicians during casting. EDF Cotrel's frame and Risser frame had a similar function to provide a traction force to straighten the spine while the patients were in a lying position. The clinicians had to apply the forces to correct the spinal deformity in all directions using their hands. It requires the skill and teamwork of the casting team to obtain a good negative cast.

Apparatus for Forming a Scoliosis Brace and a custom Providence brace standing frame had a similar function in that the users could manually adjust the frame to correct the deformity in the coronal plane and lack of correction in the sagittal and transverse planes. There are two different points from both systems. The Apparatus for Forming a Scoliosis Brace had to perform casting when the patients were lying. However, the patients were cast in a standing position using a custom Providence brace standing frame. Furthermore, the Apparatus for Forming a Scoliosis Brace had a force sensor to measure the force during casting, but the Providence brace standing frame did not.

A Scoli-Standing Frame could manually adjust the frame to correct the deformity in 3 dimensions, and patients had to stand while applying the forces. The system also had load cells to measure the forces to correct the deformity. However, the evidence to support the effectiveness of the casting frame was not strong, and future research needs to investigate this in a big group of scoliosis patients.

No research has studied the position of scoliosis patients during casting. However, in clinical practice, the clinicians will decide the casting position based on the patient's condition. The casting in a lying position will be performed if patients cannot stand or can stand only for a short time. The advantage of casting in a lying position is that the gravity is eliminated, and the spinal curve is easier to correct than in a standing position. However, clinicians must be careful of the sagittal curve because it is easy to create a flat back during casting.

Casting in a standing position or perching on the casting bar has become more common in AIS casting because clinicians can move around the patient's trunk, and it is easier to apply the force corrections in 3 dimensions than in a lying position. There was no difference in terms of spinal angle and alignment between standing or perching on the casting bar. However, the patients feel more stable while perching on the casting bar. The body is not easily swayed compared to a standing position. As a result, a new casting frame should be adjusted in 3 dimensions to correct the 3-dimensional spinal deformity. Patients perch on the bar during casting, and the system can measure the magnitude of forces during casting and illustrate the force locations and directions in 3 dimensions.

### **3.8. Conclusion**

This chapter described the scoliosis literature review, which included the definition and characteristics of scoliosis, scoliosis treatment, standard and alternative scoliosis outcome measurement, high-cost and low-cost motion capture systems, and casting frames. Scoliosis is a three-dimensional spine deformity that requires a method to quantify the deformity in all planes in order to correct the spine appropriately in all three planes.

Radiographs are a standard tool for measuring outcomes because they illustrate the bone structure well, can measure the magnitude of the curve in the coronal and sagittal planes, are relatively inexpensive, and are common in hospitals. However, exposure to radiation throughout the treatment process may increase the risk of future health problems. Some research uses coronal radiographs to estimate the degree of vertebral rotation in the transverse plane. However, this method is not the direct measurement and it still lacks information about the whole spinal column in 3 dimensions.

Some studies use MRI and CT scans to quantify spinal parameters in scoliosis. Both systems can reconstruct and illustrate the spine in 3 dimensions. However, these technologies are not commonly used in AIS patients because they take a long time to complete scanning, only scan in the lying position, and are relatively expensive. As a result, doctors primarily use these systems to examine the spine in severe cases, such as congenital scoliosis, scoliosis with neurological complications, and surgical planning. Moreover, CT scan generates higher radiation compared to standard radiographs.

Ultrasound has become a popular alternative method for measuring spinal parameters in scoliosis. Because the system is radiation-free, it can illustrate the spinal column in coronal plane and is inexpensive. Many studies are now being conducted to investigate the effectiveness of using ultrasound in a large group of scoliosis patients, and there is strong evidence to support the effectiveness of this system. However, most current studies use ultrasound to quantify spinal deformity in the coronal plane, lacking research in the sagittal and transverse plane and spinal column in 3 dimensions.

Nowadays, EOS or low-dose radiographs are being used to quantify the spinal parameters in scoliosis. Because this system clearly shows bone structure and the system can reconstruct the spinal column in 3 dimensions. This system, however, is new and relatively expensive. As a result, there is still a lack of research using this system to quantify spinal deformity in the transverse plane and 3 dimensions.

Surface topography and motion analysis are two new technologies to quantify the spinal parameters in scoliosis. Both systems are currently in the research phase. The advantage of these technologies is that they consider the spine or trunk in 3 dimensions. Surface topography scans and reconstructs the object in 3 dimensions,

while motion analysis tracks the marker locations attached to anatomical bony landmarks and illustrates the marker location or spinal parameters in 3 dimensions.

Regarding the correlation studies between these technologies and radiographs, surface topography has a slightly lower correlation than the motion capture system. The motion capture system estimates spinal deformity from marker locations attached to the skin surface and anatomical bony landmarks that are relatively close to the spinal column inside the body, whereas the surface topography estimates spinal deformity from surface asymmetry. Furthermore, when compared to ultrasound, the motion capture system reports a slightly lower correlation. Like surface topography, motion analysis estimates spinal deformity from marker locations attached to the skin surface and anatomical bony landmarks. Meanwhile, ultrasound estimates spinal parameters from the spine inside the body. However, it cannot conclude which technology is superior to another. Surface topography and motion analysis should be studied further in a large group of scoliosis patients, and the following research should directly compare these 3 systems in the same study.

High-cost motion capture systems are widely used to measure biomechanical outcomes, particularly lower and upper limb movement, but not in scoliosis. An optical motion capture system, which uses multiple cameras to track the locations of markers, is the most common high-cost system used in the laboratory. This system is highly accurate, with less than 0.5 mm and 0.5° error in dynamic movement. A motion capture system is used in a few articles to quantify spinal parameters in scoliosis, mostly in coronal and sagittal motion. Some studies examine the shoulder rotation and the scapula's inferior angle in relation to the PSIS level to assess spinal parameters in the transverse plane. Only one study considers the entire spinal column in 3 dimensions. As a result, there is still a lack of evidence to support and clarify the spinal parameters in all planes, and this area needs to be studied in the following research.

Low-cost motion capture systems are become increasingly popular for use in motion analysis. Most of this research to date has used a Kinect to detect and analyse movement, such as drop vertical jump, walking, and running. Furthermore, some researchers create software and use digital or smartphone cameras to analyse movement. The current system's error varies depending on joint position and

movement velocity. Overall, the errors range from 5 mm to 20 mm and  $-19.32^{\circ}$  to  $27^{\circ}$ . Even though the Raspberry Pi and camera are widely used in many applications. Surprisingly, no one has used the Raspberry Pi and camera to analyse human motion.

Traditional casting requires the practitioner's skill and experience to obtain a good quality of the negative cast. If the negative cast is of poor quality, the prosthetic and orthotic devices will not fit the patients or cause other problems, such as excessive compression to the skin, device slipping, and insufficient control of the deformity. In scoliosis casting, many forces must be applied to correct the spine in 3 dimensions. Therefore, more than 2 practitioners are commonly involved during this process. To correct the spinal deformity, working as a team and having a thorough understanding of the three-dimensional force system is necessary. Some scoliosis casting frames have been introduced but have yet to become widely used in clinics because they are relatively expensive, lack research evidence to support their efficacy, are relatively difficult to use and only apply corrective force in one or two dimensions. Furthermore, there is a lack of evidence regarding the magnitude of force to correct the deformity during casting, and no research illustrates the direction of forces applied to correct the spine in 3 dimensions.

Only one study recently used a motion capture system for assessing spinal deformity in scoliosis patients in 3 dimensions. However, the technology has only been used during the assessment process. Furthermore, no other study has used a motion capture system to quantify spinal deformity in scoliosis patients during the optimal correction and casting processes. As a result, the new motion capture system or postural measurement system should be able to quantify the spinal deformity in 3 dimensions, be inexpensive with good accuracy, be appropriate for scoliosis applications, be a simple system and easy to use by clinicians, be integrated with a force-measuring system that can describe the amount of force applied and illustrate the force direction, assist clinicians during the assessment, optimal correction, and casting processes, and reduce the number of clinicians required during casting.

Figure 3.33 and 3.34 illustrate the conceptual design of a new development system for this thesis. The system is composed of 3 parts. The first part is a scoliosis casting apparatus that can adjust the forces in various directions and areas. The second part is a force-measuring system that can measure the magnitude of force applied to correct

spinal deformity. Finally, a low-cost postural measurement system that can quantify 3-dimensional spinal deformity. Furthermore, the postural measurement system collaborates with the force-measuring system to display the direction of forces used to correct the deformity in 3 dimensions. The entire system should help practitioners correct deformity and quantify scoliosis in 3 dimensions during the assessment, optimal correction, and casting phases.



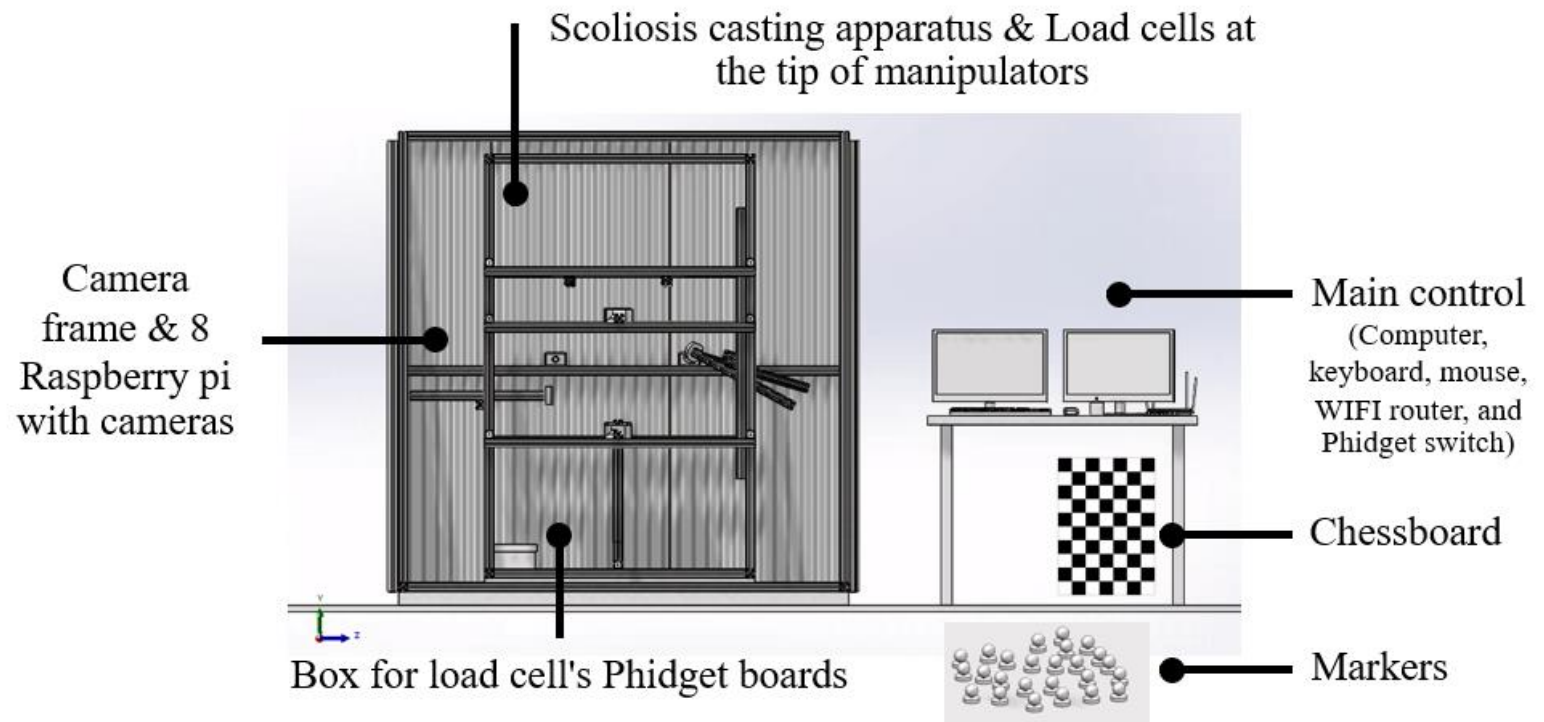


Figure 3.34 The conceptual design of a new development system for this thesis in front view.

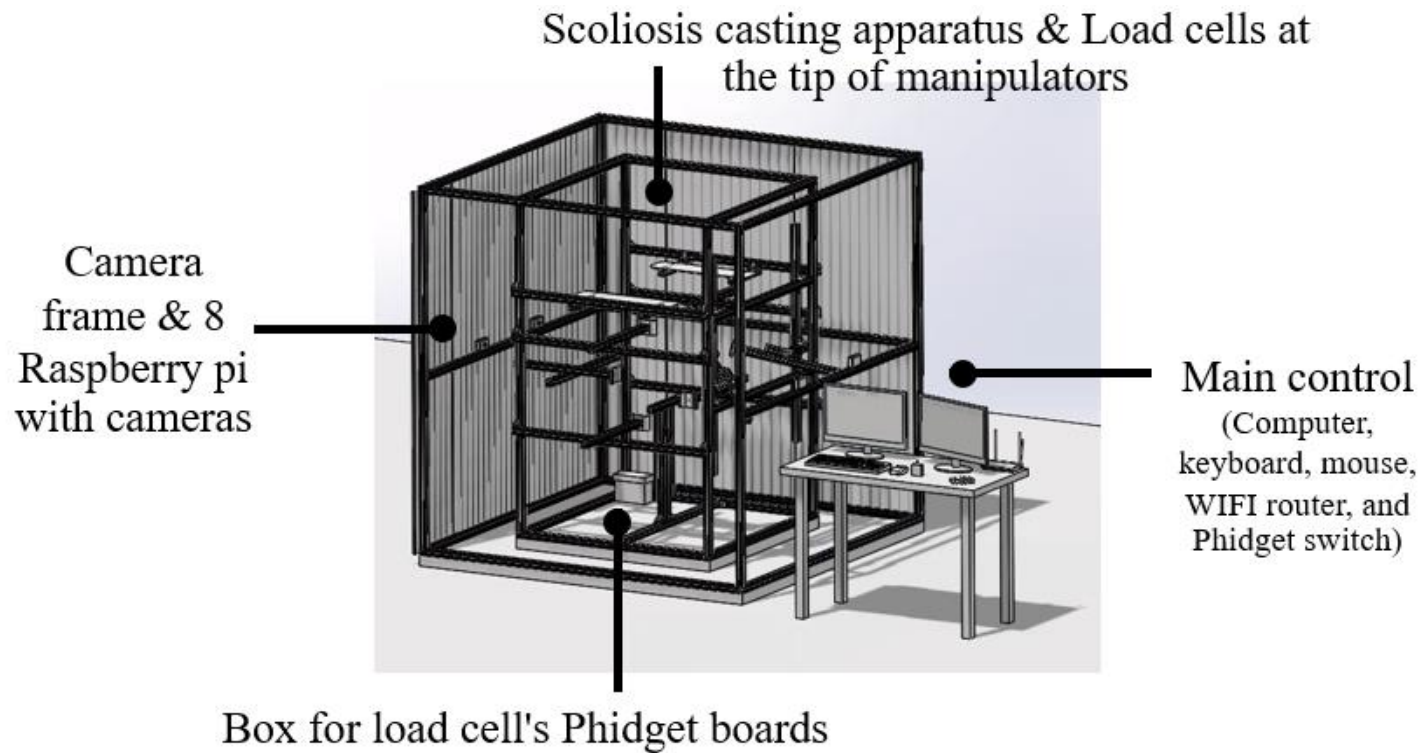


Figure 3.35 The conceptual design of a new development system for this thesis in oblique view.

## **4. Chapter 4 Development of Scoliosis Casting Apparatus**

## 4.1. Introduction

The previous chapter presented an analysis and evaluation of the relevant literature and described the thesis's research gap, aim, and objectives. Then, Chapter 4 discusses the first development, called a **Scoliosis casting apparatus**, to assist practitioners during the optimal correction and casting. For optimal correction and casting, clinicians use a 3-point force system concept to apply the forces to correct the 3-dimensional spinal deformity of scoliosis patients using their hands. Practitioners draw a blueprint on the coronal radiographs to identify the trimline of the spinal orthosis, allocate the apical, upper-end, and lower-end vertebrae, and identify the location of force to apply during casting by hand. Figure 4.1 illustrates a 3-point force system to correct the coronal plane's deformity, Figure 4.1a is a 3-point force system to correct the lumbar curve, and Figure 4.1b is for the thoracic curve. The force correction for sagittal and transverse deformities is also considered before casting. However, these do not commonly draw on the blueprint for the sagittal plane and cannot be drawn on the blueprint for the transverse plane.

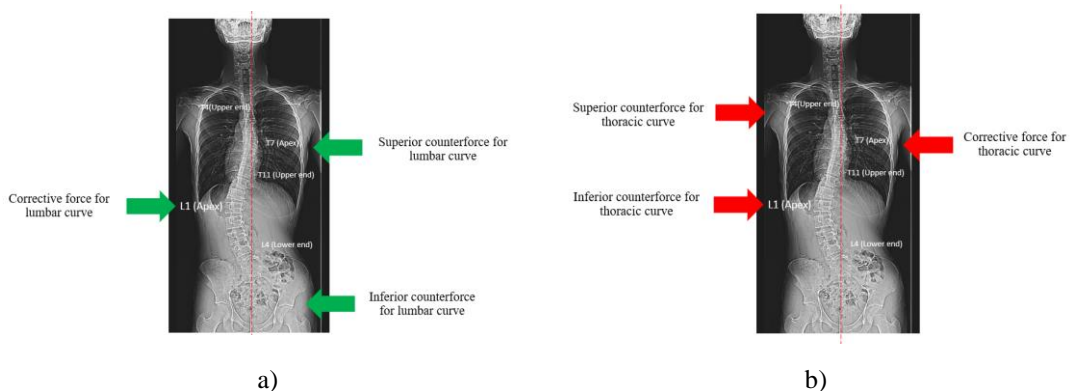


Figure 4.1 Consideration of a 3-point force system to correct spinal deformity in the coronal plane, a) to correct the lumbar curve, b) to correct the thoracic curve.

The casting process requires 2 to 4 practitioners to correct the spine (fig. 4.2) because scoliosis is a 3-dimensional deformity and is more complex than other deformities. This technique allows practitioners to get a touch feeling during casting and easily adjust the amount of force, force location, and force direction. However, the traditional method necessitates the participation of many people during casting, as well

as practitioner skills and teamwork. Furthermore, this method cannot control, maintain, or measure the force applied throughout the process.



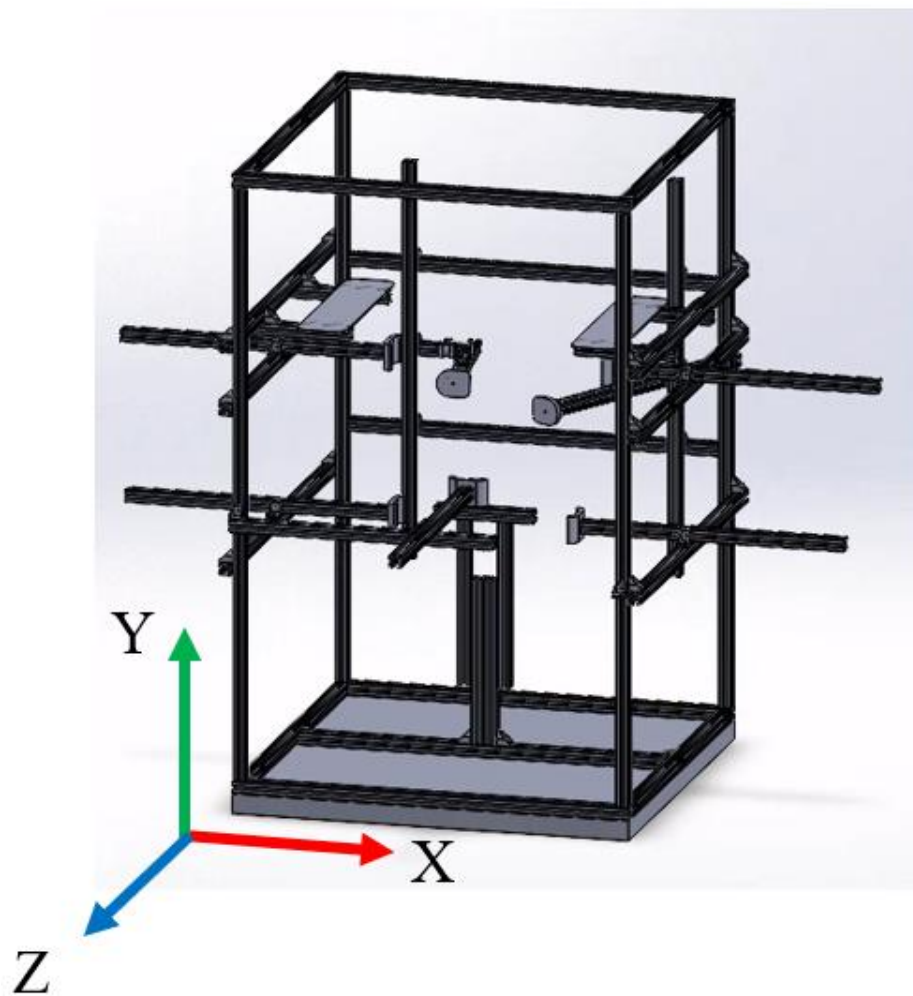
Figure 4.2 Traditional method to perform casting with a scoliosis patient.

Recently, a few casting frames were introduced to use during casting to overcome these problems as detailed in the literature review of this thesis. However, most can only apply the forces in one plane, and no one has attached a load cell to measure the force during casting.

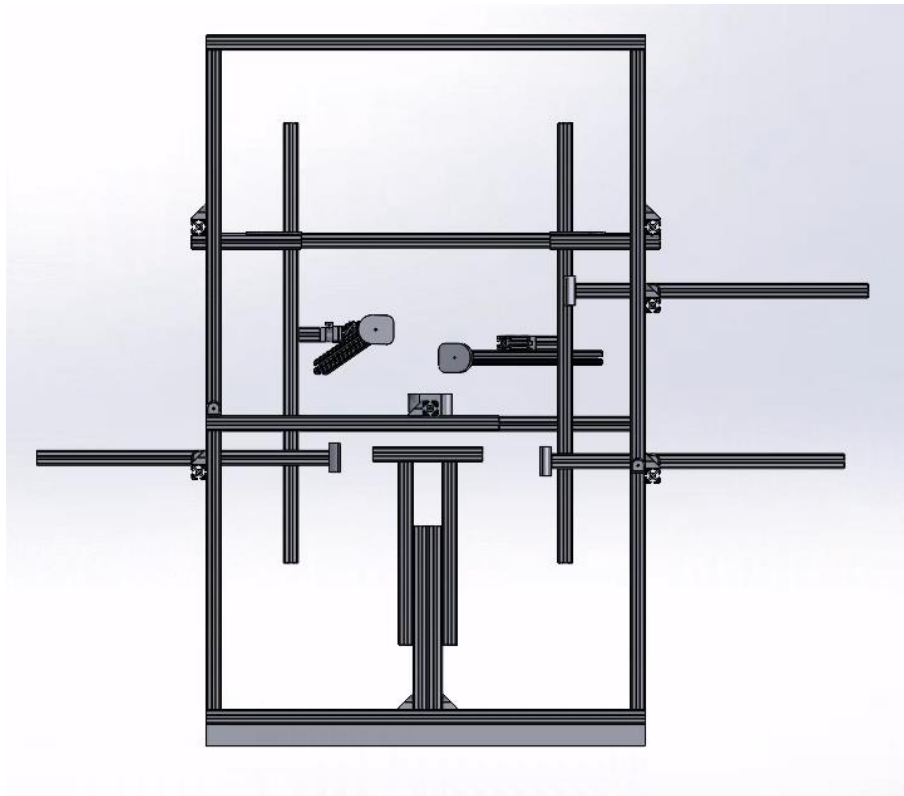
Therefore, the first objective of this chapter was to develop a scoliosis casting apparatus that could assist practitioners during an optimal correction and casting. The developed casting apparatus should be low-cost and of simple design. It should be able to apply the forces to correct the spine in 3 dimensions, reduce the number of practitioners during casting, improve the quality of the negative cast and the spinal orthosis, and measure the magnitude of force at each area to understand how the scoliosis was corrected using a 3-dimensional force system. The developed casting apparatus should be an appropriate size, allow casting and allow simply adjustment of the forces applied by manipulators.

## 4.2. Conceptual design of scoliosis casting apparatus

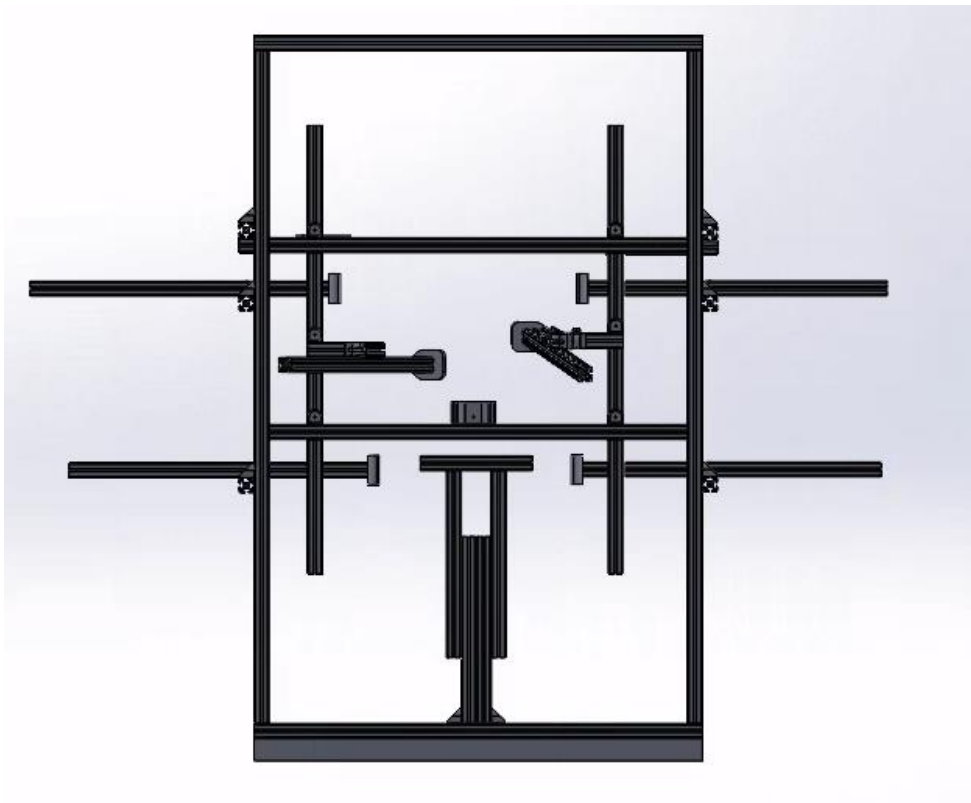
The first development of this research was a mechanical casting frame known as the "**Scoliosis Casting Apparatus**." The biomechanical principle of this apparatus was to allow a 3-point force system to be applied to the patient's trunk to correct the deformity in 3 dimensions. The scoliosis casting apparatus was designed using the computer software called SolidWorks. Several adjustments and re-designs were made in the program until the optimal version was produced. Figure 4.3 illustrates the scoliosis casting frame's conceptual design.



a)



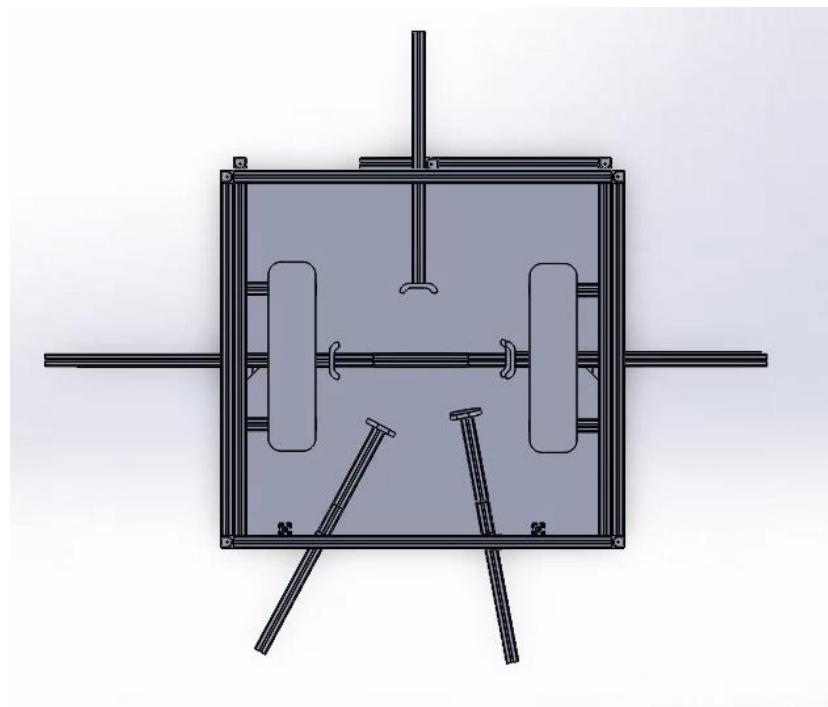
b)



c)



d)



e)

Figure 4.3 The conceptual design of a scoliosis casting apparatus, a). oblique view, b). front view, c). back view, d). side view, and e) top view.



### 4.3. Function of scoliosis casting apparatus

The apparatus was divided into eight parts:

1. Main structural element
2. Sitting bar
3. Arm-rest area (Left and right side)
4. Abdominal manipulator
5. Gluteus medius (GM) manipulators (Left and right side)
6. Axillary manipulators (Left and right side)
7. Left lumbar or thoracic-corrective manipulator
8. Right lumbar or thoracic-corrective manipulator

- **Main structural element**

The primary objective of the main structural element was to serve as the foundation of the casting apparatus. The structure should have enough space for a scoliosis patient to perch on the edge of the sitting bar. However, the apparatus's size should be manageable as greater size will increase the total weight and make movement of the frame difficult.

After consideration, the main structural element was designed in a rectangular shape with dimensions of 2 meters in height, 1 meter in width, and 1 meter in depth (fig. 4.4). The patient entered the apparatus from the front and perched on the edge of the sitting bar. Manipulators were attached to the main structural element at various locations depending on the forces required during the optimal correction and casting. The abdominal manipulator was attached to the main structural element on the anterior side. The gluteus medius manipulators were attached to both the left and right sides of the main structural element. The axillary manipulator was attached to the main structural element, either left or right, or both, depending on the patient's condition. The lumbar or thoracic-corrective manipulator was attached to the main structural element on the posterior side. The left-corrective manipulator was used if the patient presented the left curve and the right-corrective manipulator for the right curve, respectively. The main structural element was strongly fixed by connectors, bolts, and

nuts. However, the manipulators could be manually adjusted and fixed to desired locations by connectors, bolts, and nuts using the Allen key.



Figure 4.4 Design of main structural element in the oblique view.

- **Sitting bar**

The lower end of the sitting bar was permanently fixed to the main structural element in the centre. This part allowed the scoliosis patient to perch on the edge of the sitting bar during the optimal correction and casting process. The height of sitting bar could be adjusted using connectors, bolts, and nuts to match individual height. The height of sitting bar could be adjusted from 60 centimetres to 100 centimetres (fig. 4.5).

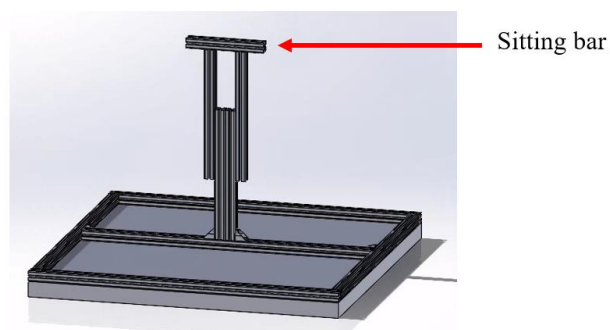


Figure 4.5 Design of sitting bar in the oblique view.

- **Arm-rest area (Left and right sides)**

During casting, forces will be applied around the axilla area as a counterforce or stabilizing force, depending on each clinical consideration. The patients must, therefore, elevate their arms. In the traditional method, they would rest their arms on the practitioner's shoulder, and this allows the practitioner's hand to compress the axilla area. In the scoliosis casting apparatus, we designed an arm-rest area for the patients to rest their arms on and allow the practitioners to work more freely while applying the axilla force. The arm-rest area was attached to the main structural element on both the left and right sides. The resting plate, a plastic sheet, was placed on the metal profile and covered with soft foam to provide comfort. There was a 50 centimetres length and a 30 centimetres width of the arm-resting structure. The function of this part was to rest the patient's forearm and, therefore, control and rest the entire upper limb throughout the process. The height of this structure could be adjusted to match the individual patient (fig. 4.6).

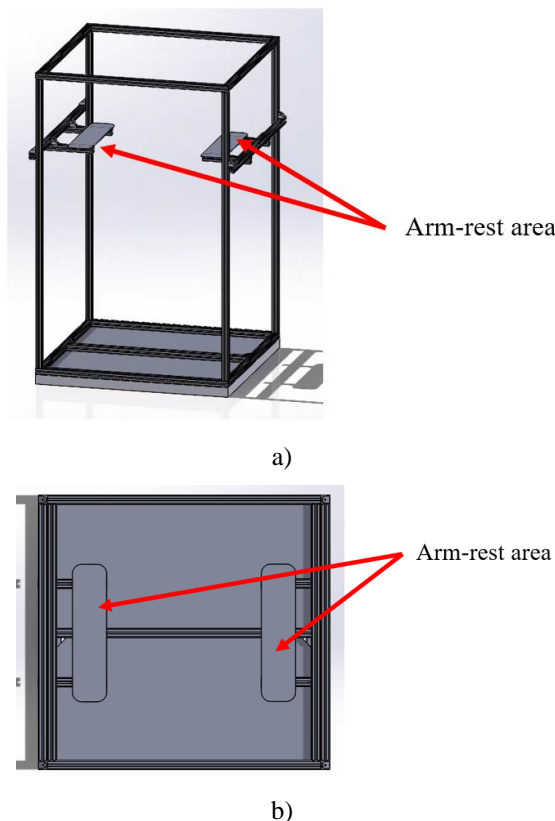


Figure 4.6 Design of arm-rest area, a). oblique view and b). top view

- **Abdominal manipulator**

Manipulators were attached to the main structural element of the scoliosis casting apparatus. However, the number of required manipulators depended on the individual patient's condition. The function of the manipulator was to apply the corrective force, counterforce, or stabilizing force to re-align the spine and correct the deformity in 3-dimensions during the optimal correction and casting.

The abdominal manipulator was attached to the front part of the main structural element and was 60 centimetres long (fig. 4.7). The manipulator could adjust the height (shifting up and down in the Y-axis) and depth (shifting to the centre of the volume in the Z-axis) at the connectors using the Allen key. The objective of this manipulator was to provide the stabilizing force at the abdominal area (posteriorly directed force), partially increase intra-abdominal pressure, and counteract the force from the posterior side (anteriorly directed force from lumbar- or thoracic-corrective manipulator).

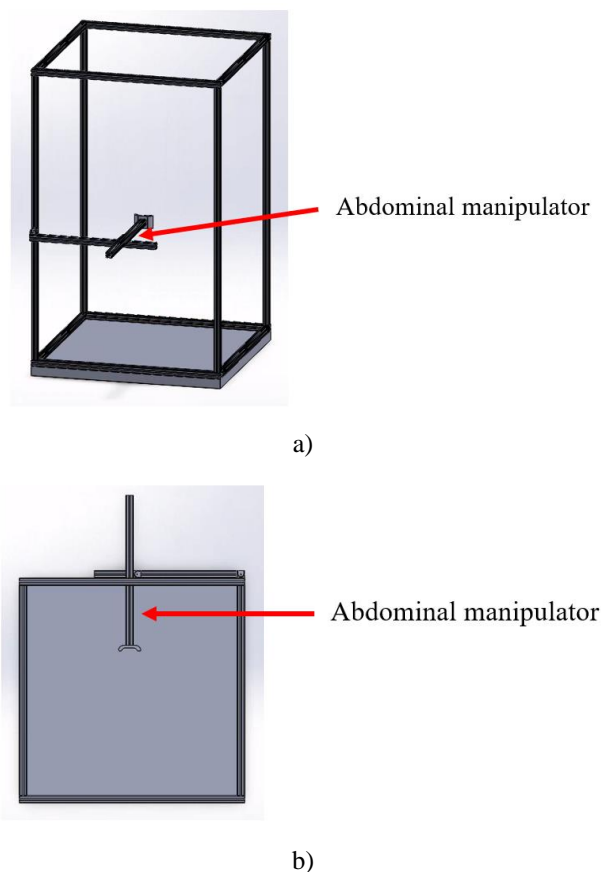


Figure 4.7 Design of abdominal manipulator, a) oblique view and b) top view.

- **Gluteus medius manipulators (Left and right side)**

Gluteus medius (GM) manipulators were attached to the main structural element on both the left and right sides. The function of these parts was to stabilize the pelvis and not allow the pelvis to move when applying other corrective forces. Another function of the GM manipulator, either on the left or right side, was to counteract the corrective force. For example, if a patient has a left lumbar curve with the apex at L2, upper- and lower-end at T7 and L5. As a result, the corrective force was applied at L2 to L4 on the left side. The superior and inferior counterforces were applied at T7 on the right side and right GM, respectively. The manipulators could be adjusted to compress at the GM area on both sides, and the force direction was generally a medial-directed force. The manipulators could adjust the height (shifting up and down in the Y-axis) and depth (shifting to the centre of the volume in the X-axis) at the connectors using the Allen key. The manipulator length was 60 centimetres (fig. 4.8).

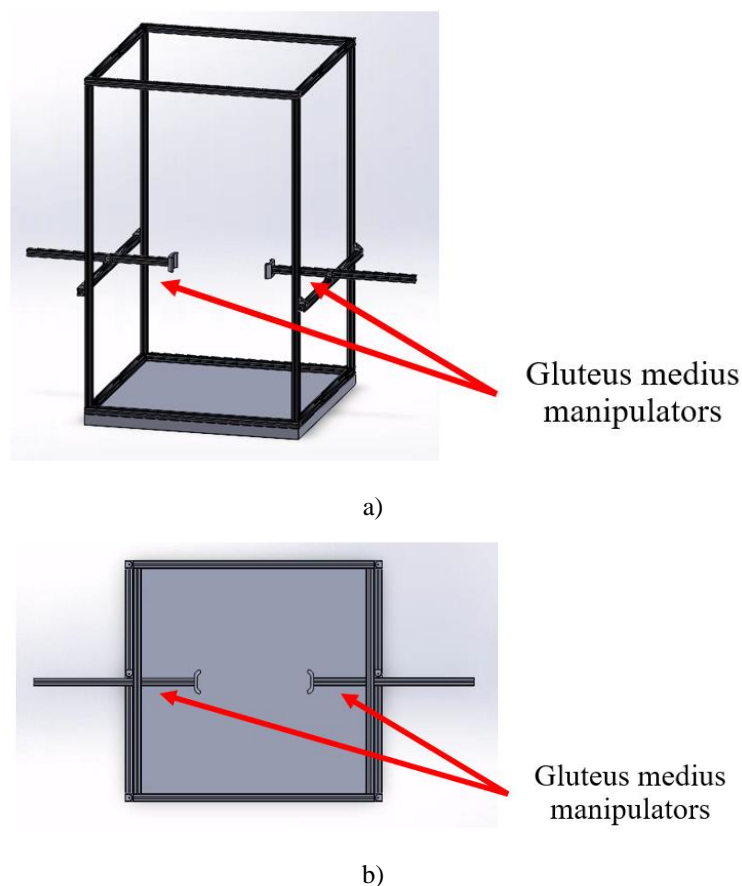


Figure 4.8 Design of gluteus medius manipulators, a) oblique view and b) top view.

- **Axillary manipulators (Left and right side)**

Axillary manipulators were attached to the main structural element on both the left and right sides. However, some cases used only one manipulator, and some used two, depending on the individual patient's condition. The function of these parts was to provide a superior counterforce to the corrective force. For example, if a patient has a left lumbar curve with the apex at L2, upper- and lower-end at T7 and L5. As a result, the corrective force was applied at L2 to L4 on the left side. The superior counterforce was applied at T7 on the right side or the right axilla area, and the interior counterforce was at the right GM. The force direction was a medial-directed force, similar to the force direction from the GM manipulators. The manipulators could adjust the height (shifting up and down in the Y-axis) and depth (shifting to the centre of the volume in the X-axis) at the connectors using the Allen key. The manipulator length was 60 centimetres (fig. 4.9).

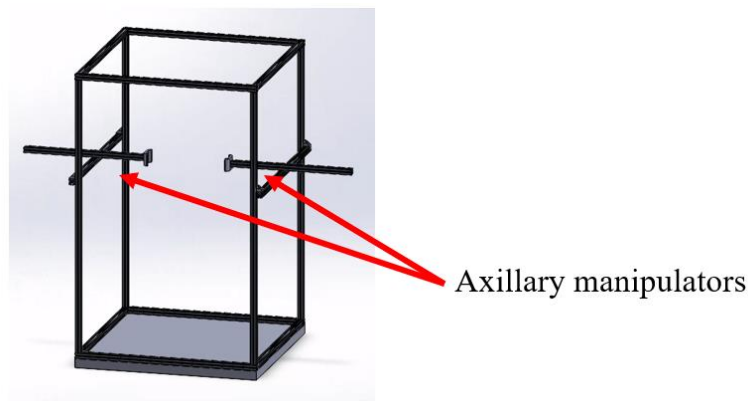


Figure 4.9 Design of axillary manipulators in oblique view.

- **Left lumbar or thoracic-corrective manipulator**

The spinal curve can be found on both the left and right sides and in the lumbar and thoracic regions. As a result, the new apparatus should be able to apply the force to correct the spine in various curve types. Therefore, we designed a lumbar or thoracic-corrective manipulator and attached it to the main structural element (fig. 4.10). The manipulator's height and depth can be adjusted to correspond with the curve

location. There were three force directions provided on this manipulator: anteriorly directed-force, medially directed force, and upward-directed force.

The force direction was determined by the type of curve. In the lumbar curve, the anteriorly directed force and medially directed force were applied. The anteriorly directed force was required to de-rotate the spine in the horizontal plane, and medially directed force was needed to centralize the spine in the coronal plane. Therefore, the manipulator to correct the lumbar curve was about  $45^\circ$  in the horizontal plane, combining both force directions. Moreover, in the thoracic curve, the anteriorly directed, medially directed, and upward-directed forces were applied. The objective of the first two forces was similar to the lumbar curve. Additionally, the upward-directed forces helped to de-rotate the spine in the horizontal plane following the direction of the rib cage. As a result, the manipulator used to correct the thoracic curve was  $45^\circ$  horizontal and  $45^\circ$  oblique upward to the same plane.

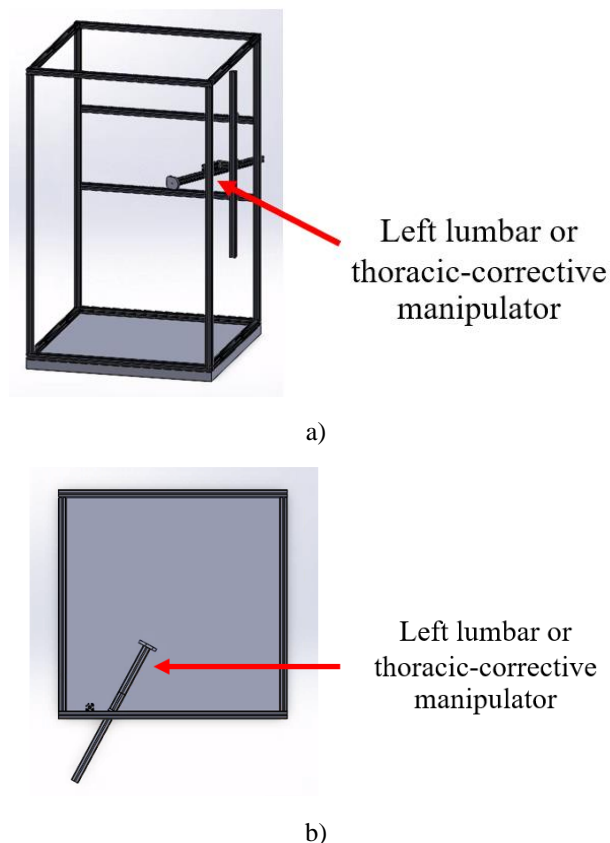


Figure 4.10 Design of left lumbar or thoracic-corrective manipulator, a) oblique view and b) top view.

- **Right lumbar or thoracic-corrective manipulator**

The right lumbar or thoracic-corrective manipulator was attached to the main structural element on the right side (fig. 4.11). The structure, function, adjustment, and force direction were similar to the left lumbar or thoracic-corrective manipulator.

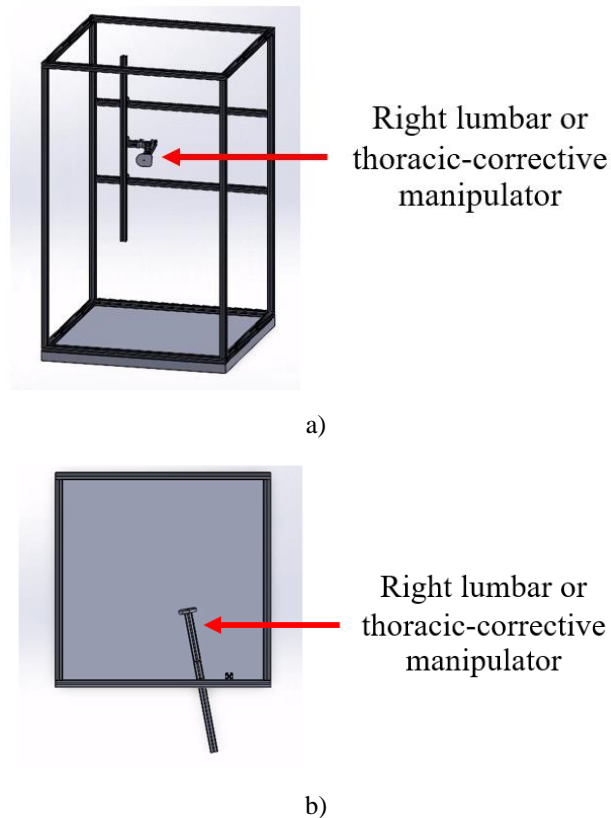


Figure 4.11 Design of right lumbar or thoracic-corrective manipulator, a) oblique view and b) top view.

Then, the commercial aluminium profile was ordered from the manufactures and cut into pieces according to the size specified in the program (An aluminium profile is a ready-made profile commercially available in the market that comes in various sizes and lengths). Finally, all aluminium profiles were assembled and connected by the connectors, bolts, and nuts. The design was based on biomechanical force application in all three planes, and the apparatus could apply, adjust, and realign the spine freely. Figure 4.8 illustrates the scoliosis casting apparatus used in the data collection process.



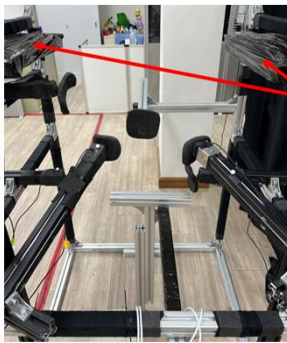


a)



b)

Figure 4.12 Scoliosis casting apparatus, a) posterior view and b) oblique view.



Arm-rest area

a)



Sitting bar

b)



Axillary manipulator

c)



Gluteus medius manipulators

d)

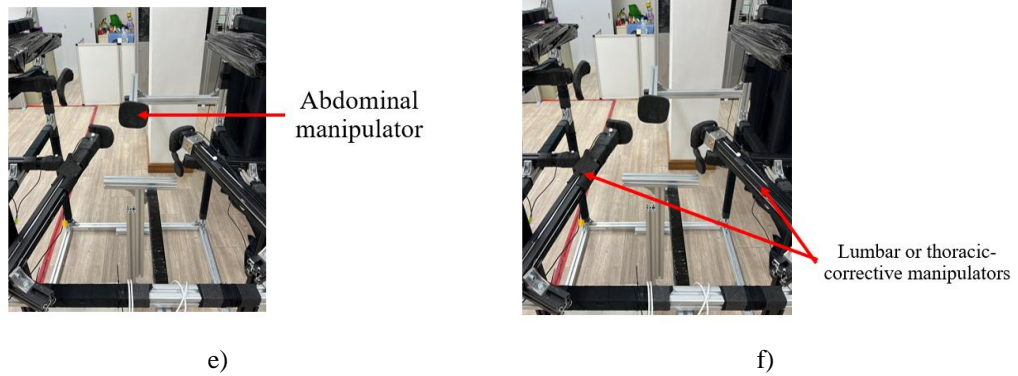


Figure 4.13 Scoliosis casting apparatus, a) arm-rest area, b) sitting bar, c) left axillary manipulator, d) gluteus medius manipulators, e) abdominal manipulator, and f) left and right lumbar or thoracic-corrective manipulators.

#### 4.4. How to adjust the scoliosis casting apparatus during optimal correction and casting process with scoliosis patient

Before adjusting the scoliosis casting apparatus, the clinician had to complete a physical assessment and radiographic evaluation to understand the patient's condition and create a treatment plan. The clinician had to identify the 3-point force system (location and direction) to correct the deformity before casting. During the data collection process, the patient was asked to change the clothes to a proper fit. Then, the researcher identified the anatomical bony landmarks and attached the markers to them (fig. 4.14a). The location of markers and spinal parameters for the clinical experiment will be described in Chapter 8. The process of adjusting the scoliosis casting apparatus is described below (fig. 4.14).

1. The patient was asked to perch on the sitting bar inside the scoliosis casting apparatus. The height was adjusted to match individual patients.
2. Then, the clinician adjusted the height and location of the arm-rest area. This part should not be too high or too low because it could affect the shoulder level and counterforce at the axillar area.
3. The abdominal manipulator was adjusted after the patient perched on the sitting bar. The location of this part was about the middle of the abdomen.

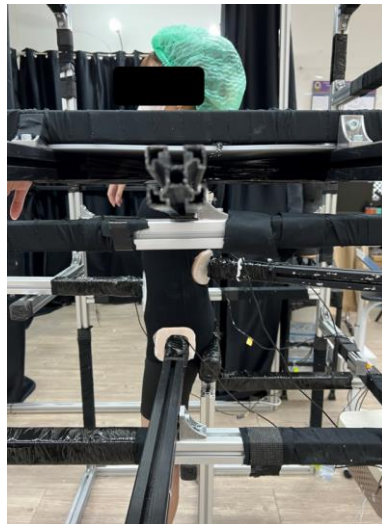
4. Then, the left and right GM manipulators were adjusted to compress at the GM area of both sides. The objective of this adjustment was to stabilize the pelvis and not allow the pelvis to move when applying other corrective forces. Another function was to counteract the corrective force.
5. The axillary manipulator was then adjusted. It depended on the curve pattern. Both manipulators were sometimes used to correct the deformity and decrease the coronal decompensation. However, some required only one side to counter or stabilize the spine.
6. Moving on to the most important manipulator, the lumbar or thoracic-corrective manipulator was applied to correct the deformity. If the patient has the right single thoracic curve, the right thoracic-corrective manipulator was used to apply the force. In addition, if the patient has a double curve with right thoracic and left lumbar, the right thoracic-corrective manipulator was used to apply the force to correct the thoracic curve, and the left lumbar-corrective manipulator was used to apply the force to correct the lumbar curve, respectively. The direction of the force also depended on the curve type. The lumbar curve required anterior and medial directed force. Moreover, the thoracic curve required anterior, medial, and upward-directed forces.
7. In general practice, we ask the patient to grade the level of corrective actions to the corrected area. The force level should be between 8 and 9. Zero level means no force acting on the spine, and ten means the patient cannot tolerate the force. If the force level is less than 8, the clinician will adjust the manipulator to increase the force, and if the level is 10, the force will be decreased.
8. A low-cost postural measurement system then recorded the marker's location and calculated the spinal parameters and alignment. Furthermore, the force-measuring system recorded and reported the magnitude of force applied at each area. A low-cost postural measurement system and a force-measuring system were described in detail in the following chapter.



a)



b)



c)



d)

Figure 4.14 Scoliosis patient in the optimal correction process, a) before applying the forces, b) while applying the forces in the posterior view, c) in the lateral view, and d) in the anterior view.

#### 4.5. Conclusion

This chapter detailed how the scoliosis casting apparatus was developed. The conceptual design of the apparatus was designed in the SolidWorks program. Several adjustments were made before getting the final version. Then, the researcher ordered the aluminium profile, connectors, and assembling tools from the manufactures. All structures were assembled following the program design. The apparatus was divided

into 8 parts, including the main structural element, sitting bar, arm-rest area (Left and right sides), abdominal manipulator, GM manipulators (Left and right sides), axillary manipulators (Left and right sides), left lumbar or thoracic-corrective manipulator, and right lumbar or thoracic-corrective manipulator. The main structural element was not allowed to be adjusted after assembly. In contrast, other parts and manipulators could be manually adjusted using the Allen key. There were 7 manipulators in total. However, 5 and 6 manipulators were commonly adjusted to correct the scoliosis deformity in 3 dimensions. Now, the scoliosis casting apparatus has been completely developed and is ready for the clinical experiment with AIS patients.

## **5. Chapter 5 Development of Force-Measuring System**

## 5.1. Introduction

The previous chapter presented the development of casting apparatus to assist practitioners during the optimal correction and casting for a spinal orthosis. The casting apparatus could apply a 3-dimensional force system to correct the 3-dimensional spinal deformity of the scoliosis patients. This chapter describes the second development, a force-measuring system, to measure the magnitude of force applied to correct the scoliosis deformity. Recently, there is still insufficient evidence in the literature to quantify the magnitude of the forces applied to correct the deformity in the 3-dimensional force system. No authors have attached load cells to the casting frame manipulators and measured the forces during the casting process. Therefore, we developed a force-measuring system that could measure the magnitude of force at multiple areas and in Chapters 6 and 7 we will explain how a low-cost postural measurement system was used to measure the direction of these forces in 3 dimensions. However, this chapter only focuses on the development of the force-measuring system.

During casting, the forces applied to the trunk should be applied with little or no shear force i.e. as a pressure over an area of the trunk. This is achieved with a pad at the end of the manipulator arm. The manipulator arm experiences a compressive force along the axis of the manipulator, which could be detected with a single axis, compression, force transducer or load cell. The load cell was attached at the end of the manipulators of the scoliosis casting apparatus between the manipulator rod and the casting pad.

The new load cell needs to be calibrated. During the calibration process, testing masses were measured by an accurate weighing scale and then used to compress the load cell. Two pieces of software were written, one for force transducer calibration and another for recording the forces during casting. The calibration software recorded the value from the load cell as a change of voltage. After that, the data were used to observe the linearity of the data, quantify errors, look for hysteresis and calculate the simple linear regression formula to estimate the magnitude of force applied from the voltage recorded. The formula obtained was then used in the casting program.



## 5.2. Components for load cell calibration

The force-measuring system was composed of hardware and software. The hardware components consisted of a load cell (ID: 3136\_0, a product from Phidgets Inc.) (fig 5.1), a PhidgetBridge 4-Input (ID: 1046\_1, a product from Phidgets Inc.) (fig 5.2), a USB cable, and a main computer. A load cell is a sensor widely used in many applications to measure the magnitude of force. In this research, we used the load cell ID3136\_0, which could measure the force up to 500 N (Phidgets, 2023a). This load cell was selected because the force applied to correct the spinal deformity in scoliosis from the previous research were lower than 500 N (van Den Hout and van Den Munckhof, 2002, Loukos *et al.*, 2011, Jang, 2018). The raw output from the load cell was a voltage ratio (V/V). Therefore, the calibration process required to convert the recorded voltage ratio to Newtons.

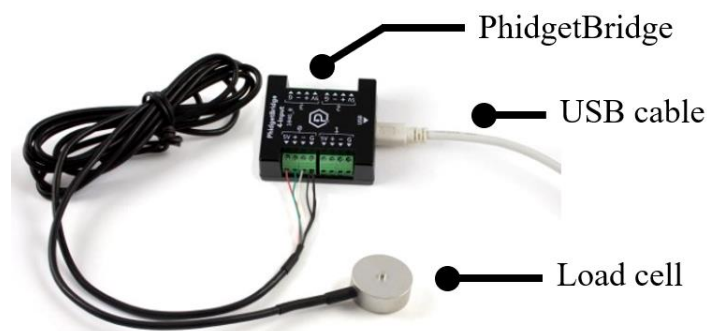


Figure 5.1 Load cell (ID: 3136\_0) connected with the PhidgetBridge

Figure 5.2 shows the PhidgetBridge 4-Input load cell interface (Phidgets, 2023b). It is a low-cost electronic component used to power up the load cell, amplify the load cell's output and convert it to a digital number. The PhidgetBridge provided a clean and stable digital signal. Each PhidgetBridge provided four inputs or channels, meaning four sensors could be connected to one PhidgetBridge. This research used six load cells to collect the data at different areas of the scoliosis trunk because the maximum force locations and directions to correct the scoliosis deformity were between 5 and 6 forces. Therefore, two PhidgetBridges were needed to collect the output from the load cells in this application.





Figure 5.2 PhidgetBridge (ID: 1046\_1)

A calibration program was created to read the incoming data from the sensor. This load cell calibration program was written by the author in the Python computer language using Visual Studio 2017 and the Microsoft Windows operating system (Windows 10). The program is provided in the electronic appendices of this thesis (Appendix 1 and 2).

Figure 5.3 illustrates the components used in the calibration process and the required components for load cell calibration are listed below.

- Calibration machine
- The testing masses were approximately 5 kg each, and there were 8 of them (the maximum testing mass was therefore 40 kg or 80% of the full scale range).
- Balance or weighing machine
- Load cell, PhidgetBridge, and a USB cable
- Support base of load cell
- Computer and program for load cell calibration
- Data collection sheet



Testing mass

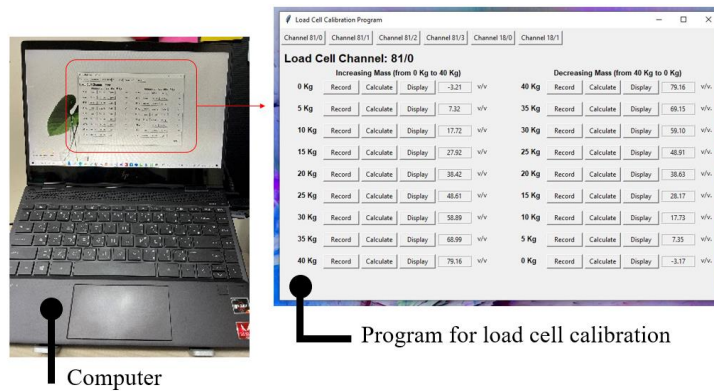
a)



Support base of load cell

Calibration machine

b)

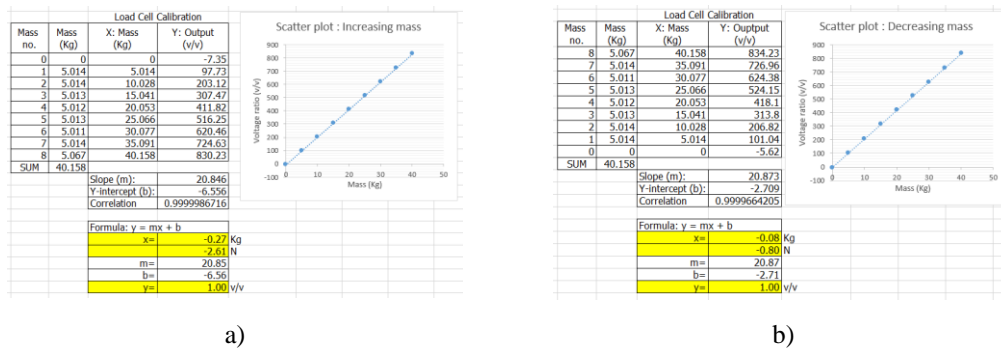


c)

Figure 5.3 Example of required components for load cell calibration, a) testing mass, b) calibration machine and support base of load cell, and c) computer and program for load cell calibration.

### 5.3. Procedure for load cell calibration

During the calibration process, the testing mass was placed on the top surface of the load cell, and the mass was gradually increased by 5 kg at a time, up to 40 kg. For every 5 kg, the software recorded the output from the load cell. After reaching the maximum, the mass was gradually reduced by 5 kg, and the program recorded the output. A similar process was performed for all load cells. After that, the output from increasing and decreasing mass was recorded in the data collection excel spreadsheet (fig 5.4), and a simple linear regression formula was then calculated. Therefore, each load cell had a different simple linear regression formula to estimate new results. Figure 5.5 shows the graphical user interface (GUI) of the program for load cell calibration, and this program could be used to calibrate all load cells.



a)

b)

Figure 5.4 Example of data collection sheet used for calculating a simple linear regression formula, a) increasing mass condition, and b) decreasing mass condition



Figure 5.5 Graphical user interface (GUI) of the program for load cell calibration

The procedure for calibrating a load cell is described further below.

- Measured the testing mass with a weighing machine. The testing mass was approximately 5 kg, and the total number of objects was 8 pieces. The actual mass was measured with three decimal points. For example, the first object had 5.226 kg. As each object was not precisely the same mass, the object was measured and labelled from numbers 1 to 8. Therefore, the testing object had to be placed on the load cell in the consequence order.
- Set up the testing equipment and program: the calibration machine, PhidgetBridge, load cell, USB cable, support base for load cell, computer, and program for load cell calibration.
- Set up the load cell and support base at the centre of the lower part of the calibration machine (fig 5.3b).
- Opened the program for load cell calibration (fig 5.5).
- Recorded the voltage output when no mass was applied (0 kg).
- Then, put a 5-kg mass on the load cell and clicked to record the voltage output on the program.
- Put another 5-kg mass and clicked to record the data. Recorded the data until it reached 40 kg.
- Then, removed a 5-kg mass and clicked to record the data on the program.
- Removed another 5-kg mass and clicked to record the data. Recorded the data until it reached 0 kg.

- Clicked another GUI page to calibrate another sensor.
- Calibrated the sensors until all sensors had been done.
- Recorded the voltage output in the data collection sheet (fig 5.4). The sheet then calculated the simple linear regression formula that could be used to predict the new value when compressing on the sensor in the following program.

#### 5.4. Calibration result

This section details the calibration result for one load cell (Serial Number 81 and Channel 0) to assess the quality of the load cell before using it in the data collection process with scoliosis patients. Figure 5.6 illustrates an example of the line graph of the calibration result while increasing the load by 5-kg of mass. The x-axis is the time (seconds), and the y-axis is the voltage ratio (V/V). Five variables are considered in this graph.

1. A step change is the steady magnitude of the voltage after testing mass is increased.
2. Rise time is the time it takes a signal to change from a low value to a high value.
3. Response time is the time it takes a signal from a high value in a rise time to the new steady-state value.
4. The mean of the steady state is an average voltage ratio calculated from about 10 seconds of data after reaching the new steady state.
5. The steady state's standard deviation (SD) is a measurement of data spread for 10 seconds after reaching the new steady state.

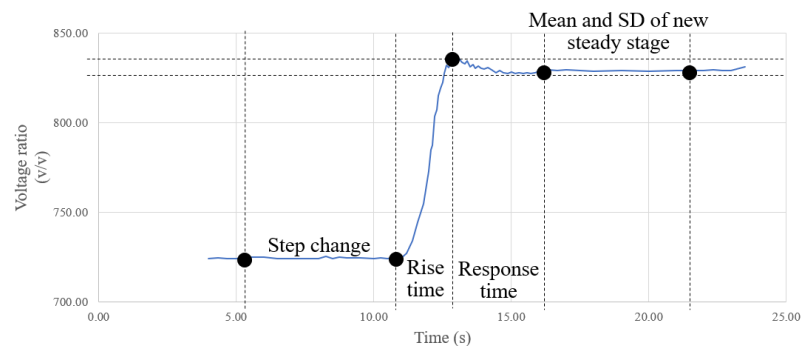


Figure 5.6 Example of the line graph of the calibration result.

The load cell calibration began with the testing condition of increasing mass followed by decreasing mass. Conditions 1 to 8 were the conditions while increasing every 5-kg mass, and conditions 9 to 16 were the conditions while decreasing every 5-kg mass. Table 5.1 summarizes the calibration result during increasing mass conditions.

Figure 5.7 illustrates the calibration result of the load cell in condition 1, between 0 kg and 5 kg (mass 1 = 5.014 kg). The initial result was -7.35 V/V. After applying a 5-kg mass, the result increased to 97.73 V/V within a rise time of 1.8 seconds and no response time. As a result, the sensor spent 1.8 seconds before reaching the new steady state. The new steady state's mean with a standard deviation (SD) was 97.73 (0.14) V/V. The voltage ratio increased by approximately 100 units for the application of 50 N. As a result, the system quantification level was about one voltage unit (v/v) equal to 0.5 N.

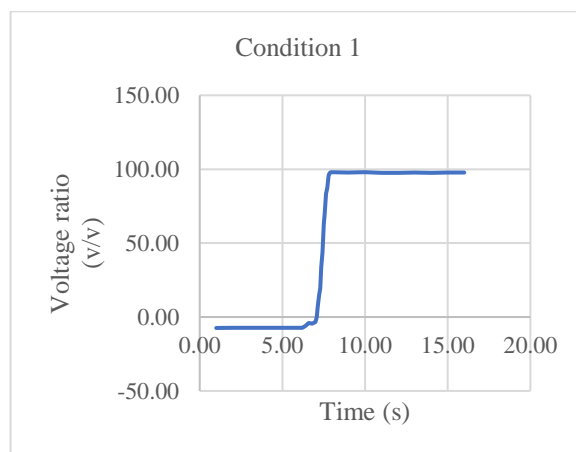


Figure 5.7 Calibration result of the load cell in condition 1.

Figure 5.8 illustrates the calibration result in condition 2, between 5 kg and 10 kg (mass 2 = 5.014 kg). The initial result was 97.73 V/V. After applying a 5-kg mass, the result increased to 203.12 V/V within a rise time of 2.80 seconds and a response time of 0.70 seconds. As a result, the sensor spent 3.50 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 203.12 (1.26) V/V.

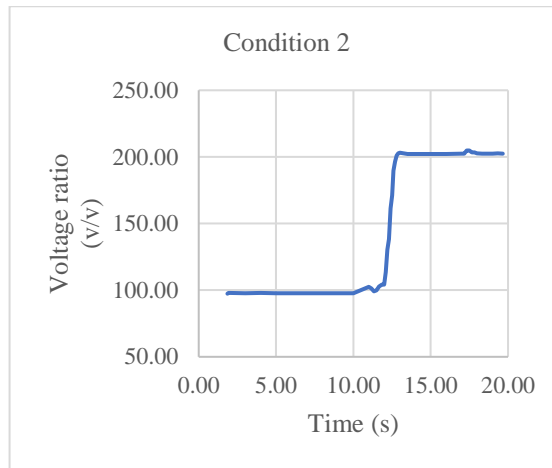


Figure 5.8 Calibration result of the load cell in condition 2

Figure 5.9 illustrates the calibration result in condition 3, between 10 kg and 15 kg (mass 3 = 5.013 kg). The initial result was 203.12 V/V. After applying a 5-kg mass, the result increased to 307.47 V/V within a rise time of 1.19 seconds and a response time of 0.61 seconds. As a result, the sensor spent 1.80 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 307.47 (0.52) V/V.

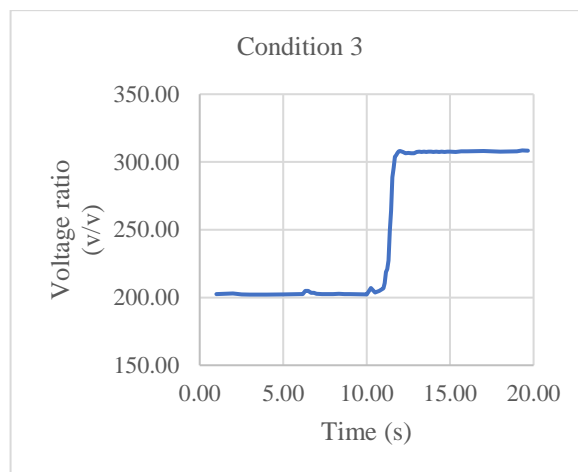


Figure 5.9 Calibration result of the load cell in condition 3

Figure 5.10 illustrates the calibration result in condition 4, between 15 kg and 20 kg (mass 4 = 5.012 kg). The initial result was 307.47 V/V. After applying a 5-kg mass, the result increased to 411.82 V/V within a rise time of 1.86 seconds and a response time of 0.80 seconds. As a result, the sensor spent 2.66 seconds (A + B) before reaching

the new steady state. The mean with SD of the new steady state was 411.82 (0.51) V/V.

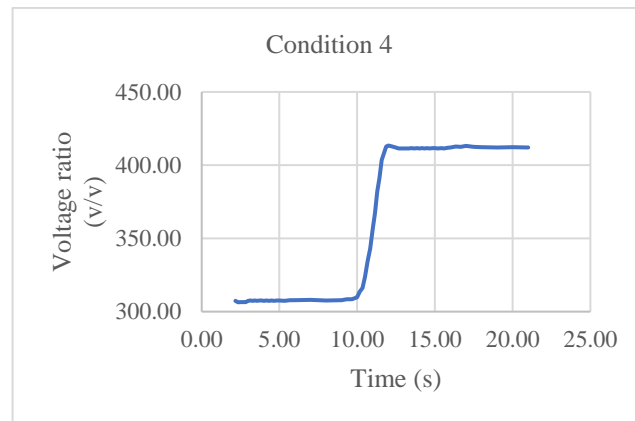


Figure 5.10 Calibration result of the load cell in condition 4.

Figure 5.11 illustrates the calibration result in condition 5, between 20 kg and 25 kg (mass 5 = 5.013 kg). The initial result was 411.82 V/V. After applying a 5-kg mass, the result increased to 516.25 V/V within a rise time of 1.86 seconds and a response time of 2.14 seconds. As a result, the sensor spent 4.00 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 516.25 (0.38) V/V.

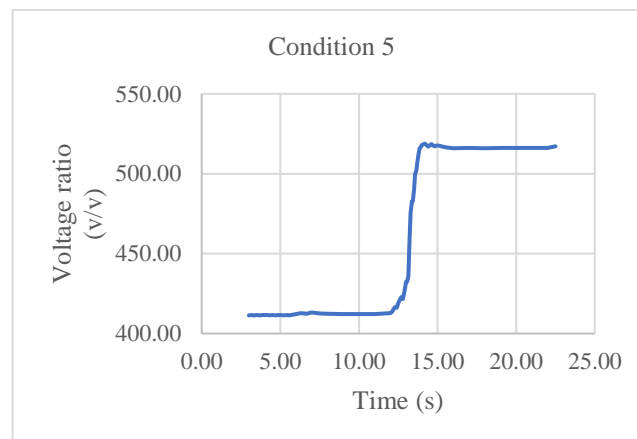


Figure 5.11 Calibration result of the load cell in condition 5.

Figure 5.12 illustrates the calibration result in condition 6, between 25 kg and 30 kg (mass 6 = 5.011 kg). The initial result was 516.25 V/V. After applying a 5-kg mass, the result increased to 620.46 V/V within a rise time of 1.85 seconds and a response

time of 2.15 seconds. As a result, the sensor spent 4.00 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 620.46 (0.22) V/V.

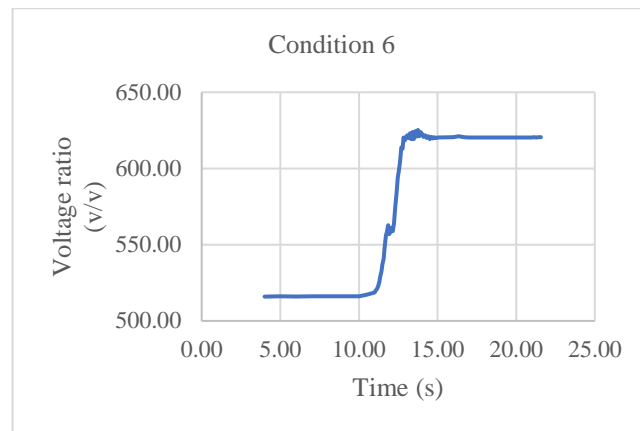


Figure 5.12 Calibration result of the load cell in condition 6.

Figure 5.13 illustrates the calibration result in condition 7, between 30 kg and 35 kg (mass 7 = 5.014 kg). The initial result was 620.46 V/V. After applying a 5-kg mass, the result increased to 724.63 V/V within a rise time of 1.58 seconds and a response time of 2.75 seconds. As a result, the sensor spent 4.33 seconds (A + B) before reaching the new steady state. The mean with SD of the steady state was 724.63 (0.66) V/V.

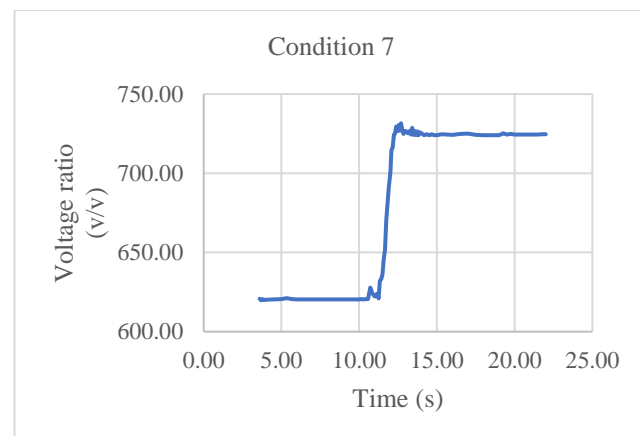


Figure 5.13 Calibration result of the load cell in condition 7

Figure 5.14 illustrates the calibration result in condition 8, between 35 kg and 40 kg (mass 8 = 5.067 kg). The initial result was 724.63 V/V. After applying a 5-kg mass, the result increased to 830.23 V/V within a rise time of 1.64 seconds and a response



time of 4.00 seconds. As a result, the sensor spent 5.64 seconds (A + B) before reaching a steady state. The mean with SD of the steady state was 830.23 (1.69) V/V.

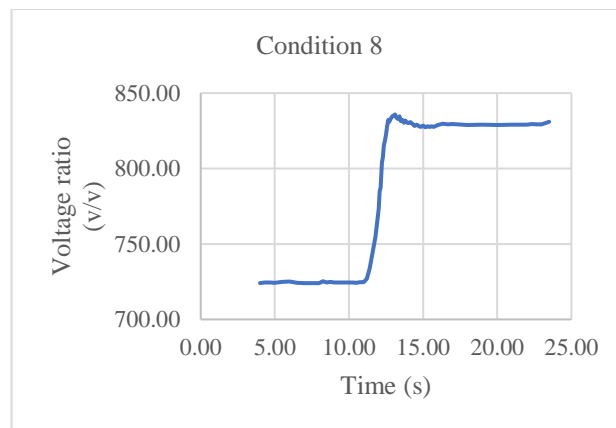


Figure 5.14 Calibration result of the load cell in condition 8

Table 5.1 Calibration result of the load cell in conditions 1 to 8 (Increasing mass condition).

Condition	Applied Mass (kg)	Applied Force (N)	Steady state (V/V)		Rise time (A) (s)	Response time (B) (s)	(A) + (B) (s)
			Mean	SD			
No mass	0.00	0.00	-7.35	0.05	n/a	n/a	n/a
Condition 1	5.014	49.19	97.73	0.14	1.80	0.00	1.80
Condition 2	10.028	98.37	203.12	1.26	2.80	0.70	3.50
Condition 3	15.041	147.55	307.47	0.52	1.19	0.61	1.80
Condition 4	20.053	196.72	411.82	0.51	1.86	0.80	2.66
Condition 5	25.066	245.90	516.25	0.38	1.86	2.14	4.00
Condition 6	30.077	295.06	620.46	0.22	1.85	2.15	4.00
Condition 7	35.091	344.24	724.63	0.66	1.58	2.75	4.33
Condition 8	40.158	393.95	830.23	1.69	1.64	4.00	5.64

Note: v/v = voltage ratio, (A) = Rise time, (B) = Response time, s = second, kg = Kilogram, N = Newton, SD = Standard Deviation, n/a = not applicable, Force = mass x g, where  $g = 9.81 \text{ m/s}^2$ .

The load cell calibration continued to be performed in the decreasing mass condition. Conditions 9 to 16 were the calibrating conditions while decreasing every 5-kg mass. Table 5.2 also summarizes the calibration results of these conditions.

In condition 9, figure 5.15 illustrates the calibration result of load cell when the mass changed from 40 kg to 35 kg. The initial result was 834.23 V/V. After decreasing a 5-kg mass, the result decreased to 726.96 V/V within a rise time of 0.66 seconds and a response time of 2.67 seconds. As a result, the sensor spent 3.33 seconds (A + B)

before reaching the new steady state. The mean with SD of the steady state was 726.96 (2.50) V/V.

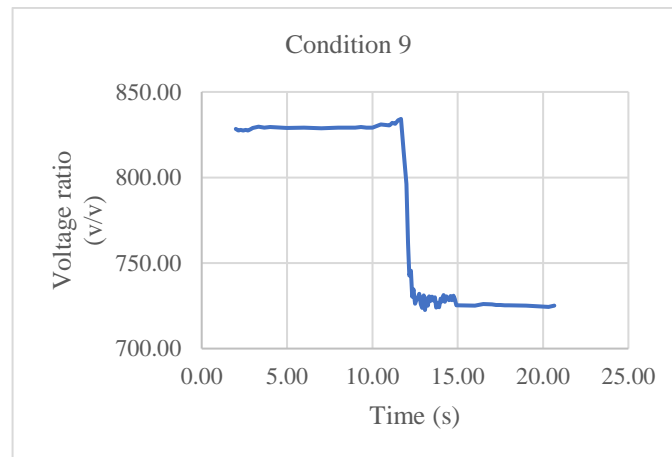


Figure 5.15 Calibration result of the load cell in condition 9

In condition 10, figure 5.16 illustrates the calibration result when the mass changed from 35 kg to 30 kg. The initial result was 726.96 V/V. After decreasing a 5-kg mass, the result decreased to 625.48 V/V within a rise time of 0.46 seconds and a response time of 3.75 seconds. As a result, the sensor spent 4.21 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 625.48 (0.65) V/V.

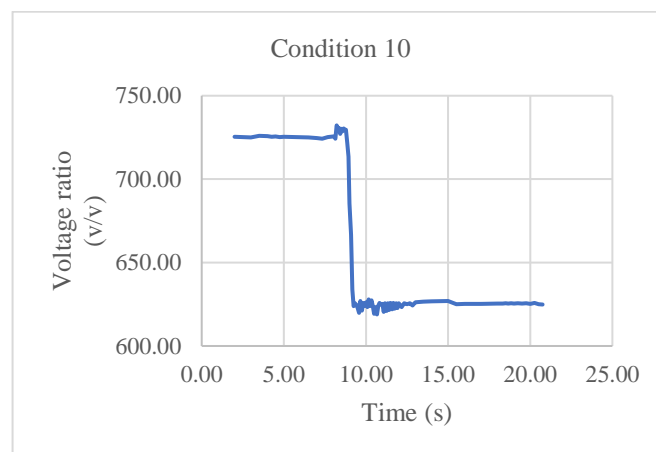


Figure 5.16 Calibration result of the load cell in condition 10

In condition 11, figure 5.17 illustrates the calibration result when the mass changed from 30 kg to 25 kg. The initial result was 624.38 V/V. After decreasing a 5-kg mass,

the result decreased to 524.11 V/V within a rise time of 0.54 seconds and a response time of 2.31 seconds. As a result, the sensor spent 2.85 seconds (A + B) before reaching the new steady state. The mean with SD of the steady state was 524.11 (0.65) V/V.

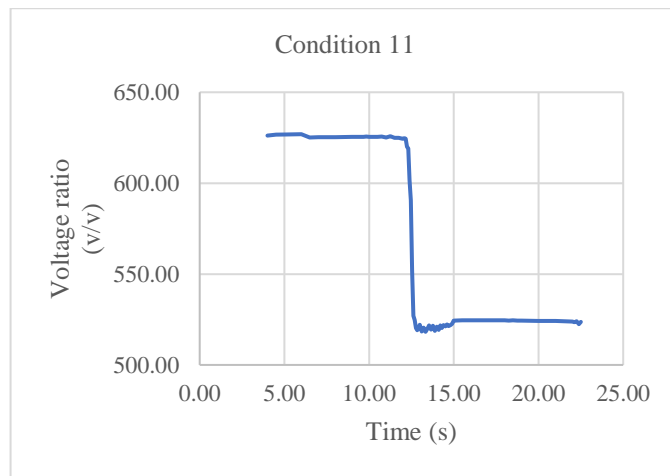


Figure 5.17 Calibration result of the load cell in condition 11

In condition 12, figure 5.18 illustrates the calibration result when the mass changed from 25 kg to 20 kg. The initial result was 524.15 V/V. After decreasing a 5-kg mass, the result decreased to 417.59 V/V within a rise time of 0.95 seconds and a response time of 0.30 seconds. As a result, the sensor spent 1.25 seconds (A + B) before reaching the new steady state. The mean with SD of the steady state was 417.59 (0.46) V/V.

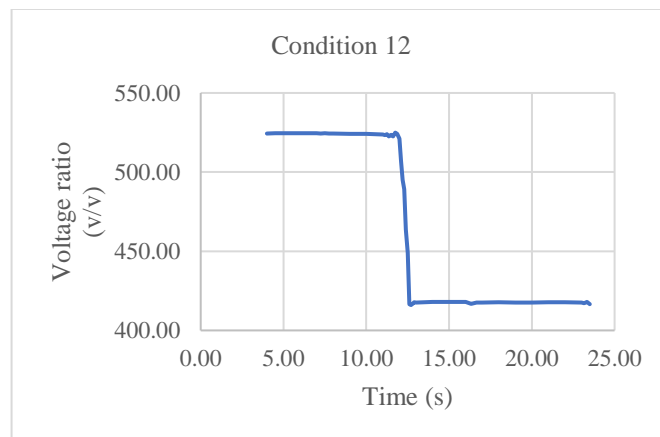


Figure 5.18 Calibration result of the load cell in condition 12.

In condition 13, figure 5.19 illustrates the calibration result when the mass changed from 20 kg to 15 kg. The initial result was 418.10 V/V. After decreasing a 5-kg mass,

the result decreased to 314.21 V/V within a rise time of 0.50 seconds and a response time of 1.50 seconds. As a result, the sensor spent 2.00 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 314.21 (0.56) V/V.



Figure 5.19 Calibration result of the load cell in condition 13.

In condition 14, figure 5.20 illustrates the calibration result when the mass changed from 15 kg to 10 kg. The initial result was 313.80 V/V. After decreasing a 5-kg mass, the result decreased to 206.95 V/V within a rise time of 0.60 seconds and a response time of 0.20 seconds. As a result, the sensor spent 0.80 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was 206.95 (0.53) V/V.

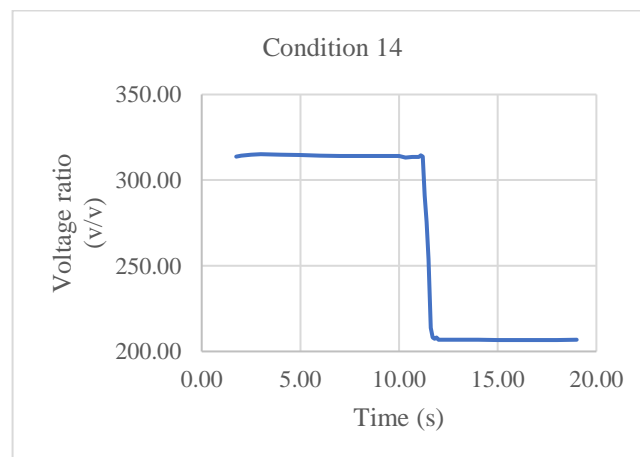


Figure 5.20 Calibration result of the load cell in condition 14.

In condition 15, figure 5.21 illustrates the calibration result when the mass changed from 10 kg to 5 kg. The initial result was 206.82 V/V. After decreasing a 5-kg mass, the result decreased to 101.11 V/V within a rise time of 1.16 seconds and a response time of 0.34 seconds. As a result, the sensor spent 1.5 seconds (A + B) before reaching the new steady state. The mean with SD of the steady state was 101.11 (0.07) V/V.

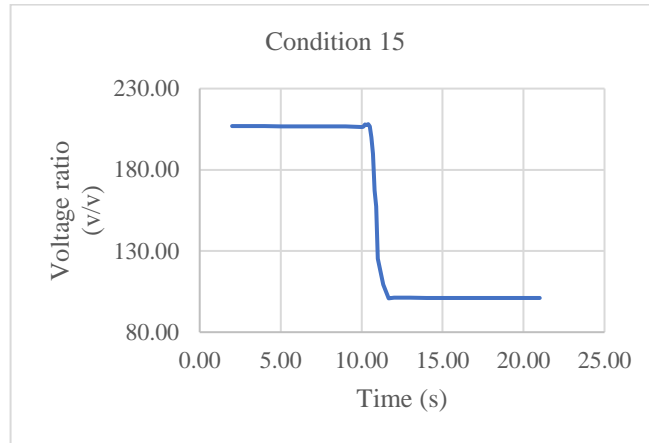


Figure 5.21 Calibration result of the load cell in condition 15.

The last condition (Condition 16), figure 5.22 illustrates the calibration result when the mass changed from 5 kg to zero. The initial result was 101.04 V/V. After decreasing a 5-kg mass, the result decreased to -5.62 V/V within a rise time of 0.86 seconds and a response time of 0.50 seconds. As a result, the sensor spent 1.36 seconds (A + B) before reaching the new steady state. The mean with SD of the new steady state was -5.62 (0.03) V/V.

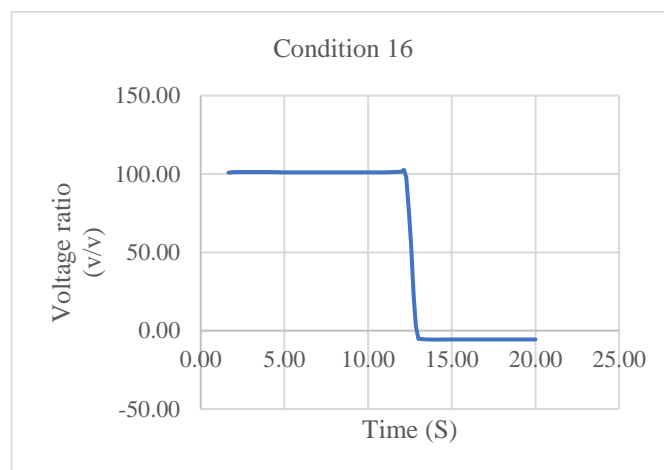


Figure 5.22 Calibration result of the load cell in condition 16.

Table 5.2 Calibration result of the load cell in conditions 9 to 16 (Decreasing mass condition).

Condition	Applied Mass (kg)	Applied Force (N)	Steady state (V/V)		Rise time (A) (s)	Response time (B) (s)	(A) + (B) (s)
			Mean	SD			
Condition 9	35.091	344.24	726.96	2.50	0.66	2.67	3.33
Condition 10	30.077	295.06	625.48	0.65	0.46	3.75	4.21
Condition 11	25.066	245.90	524.11	0.65	0.54	2.31	2.85
Condition 12	20.053	196.72	417.59	0.46	0.95	0.30	1.25
Condition 13	15.041	147.55	314.21	0.56	0.50	1.50	2.00
Condition 14	10.028	98.37	206.95	0.53	0.60	0.20	0.80
Condition 15	5.014	49.19	101.11	0.07	1.16	0.34	1.50
Condition 16	0.00	0.00	-5.62	0.03	0.86	0.50	1.36

Note: v/v = voltage ratio, (A) = Rise time, (B) = Response time, s = second, kg = Kilogram, N = Newton, SD = Standard Deviation, n/a = not applicable, Force = (mass) x (g), where  $g = 9.81 \text{ m/s}^2$ .

Table 5.1 reports the calibration result in conditions 1 to 8, or increasing the mass from 0 kg to 40 kg on the load cell. Furthermore, table 5.2 reports the calibration result in conditions 9 to 16, or decreasing the mass from 40 kg to 0 kg. Regarding the increasing mass condition, the mean and SD of the rise time (A), the response time (B), and (A) + (B) were 1.82 (0.46), 1.64 (1.34), and 3.47 (1.32) seconds, respectively. Furthermore, in the decreasing mass condition, the mean and SD of the rise time (A), the response time (B), and (A) + (B) were 0.72 (0.25), 1.45 (1.34), and 2.16 (1.18) seconds, respectively. As a result, both groups' rise and response times were quite similar. The rise and response time were slightly longer in the heavier accumulated mass than in the lighter accumulated mass. However, the load cell generally spent a short time of between 2 to 3 seconds to reach the new steady state while measuring the mass change. In conclusion, the sensor is fast enough to give the output after applying load and is suitable for clinical use, provided data is not collected within the first 3 seconds of applying the forces. In addition, the data shows the signal from the transducer to be drift-free and sufficiently noiseless for accurate forces to be measured.

Figure 5.23 shows the load cell's response when the mass increases (Condition 1 to 8) and decreases (Condition 9 to 16). Table 5.3 also compares the difference in voltage change at each testing point. The result showed that the voltage value between increasing and decreasing conditions at each testing point was quite similar. For example, the mean output of increasing and decreasing at the testing point of 5.014 kg was 97.73 v/v and 101.11 v/v, and the difference of mean output or (D) - (C) was 3.38

v/v or approximately 2 N. The average and SD of the mean difference between rising and falling data was 4.52 (2.02) v/v (approximately 2 N) with a maximum of 7.86 v/v (4 N) and a minimum of 1.73 v/v (1 N). The result showed that the mean output of each testing point of decreasing mass (D) was slightly higher than the point of increasing mass (C). However, this different value was only a couple of Newtons on average which would not affect the significance of the results obtained during the clinical experiments.

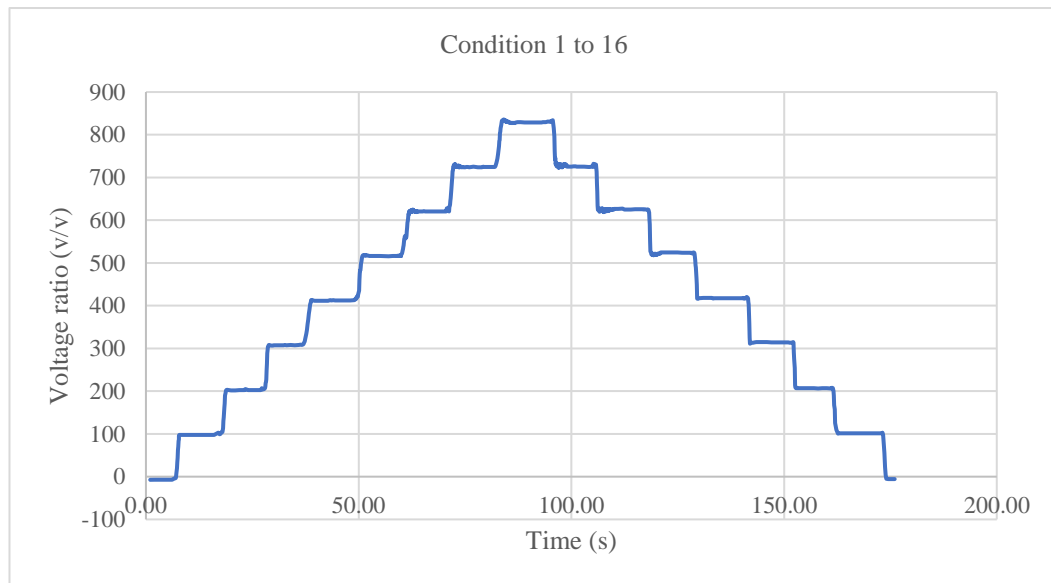


Figure 5.23 Calibration result of load cell when the mass increased and decreased between 0 kg and 40 kg.

Table 5.3 Comparison of the difference in voltage change at each testing point between increasing and decreasing mass conditions.

Applied Force	Increasing mass (v/v)		Decreasing mass (v/v)		Diff. of Mean Output (D - C) (v/v)
	Mean Output (C)	SD	Mean Output (D)	SD	
No force	-7.35	0.05	-5.62	0.03	1.73
At 49.19 N	97.73	0.14	101.11	0.07	3.38
At 98.37 N	203.12	1.26	206.95	0.53	3.83
At 147.55 N	307.47	0.52	314.21	0.56	6.74
At 196.72 N	411.82	0.51	417.59	0.46	5.77
At 245.90 N	516.25	0.38	524.11	0.65	7.86
At 295.06 N	620.46	0.22	625.48	0.65	5.02
At 344.24 N	724.63	0.66	726.96	2.50	2.33
At 393.95 N	830.23	1.69	834.23	n/a	4.00

Note: kg = Kilogram, v/v = Voltage ratio, SD = Standard deviation, N = newton, n/a = not applicable

## 5.5. Simple linear regression for load cell calibration

The calibration data from increasing and decreasing the mass were used to calculate the relationship between applied force and measured output which could then be reversed and used in the clinical testing program.

Simple linear regression is a common statistic used to predict the relationship between the independent input (x-value) and the dependent output (y-value). Once established, and provided there is little hysteresis, it can be reversed so as to convert the dependent variable (y value or measured output) to predict the independent variable (x value or applied force). In this study, the independent variable was force, and the dependent variable was a voltage change (v/v). The formula for a simple linear regression is described in Equation 5.1.

$$y = m(x) + b \quad (5.1)$$

where:

- **y** is the predicted value of the dependent variable or voltage ratio (v/v)
- **b** is Y-intercept or the predicted value of y when the x is 0
- **m** is the regression coefficient or slop
- **x** is the independent variable or force (Newton)

In the increasing force condition, the slope (m) was 2.13, the Y-intercept (b) was -6.56, as shown in the equation below. Therefore, 1 v/v was about 0.47 N. The graph (fig. 5.24) shows in a straight line, a positive direction, and no deviation of data. Moreover, the correlation efficiency was close to 1 (r-value = 0.999999) or an excellent relationship between force and voltage output.

$$y = 2.13x - 6.56$$

In decreasing mass condition, the slope (m) was 2.13, Y-intercept (b) was -2.71, and correlation efficiency was 0.999966, as shown in the equation below. Therefore, 1 v/v was about 0.47 N. The graph (fig. 5.24) also shows similar to the previous



condition, in a straight line, a positive direction, and no deviation of data. Moreover, the correlation efficiency was close to 1 (r-value = 0.999966), or an excellent relationship between force and voltage output.

$$y = 2.13x - 2.71$$

Finally, Figure 5.24 illustrates the scatter plot of increasing and decreasing force conditions at each testing point. The result showed that the value of each testing point was similar in magnitude and direction. As can be seen from the graph, there is no lag or deviation between both outputs or no hysteresis effect between increasing and decreasing the forces on this load cell.

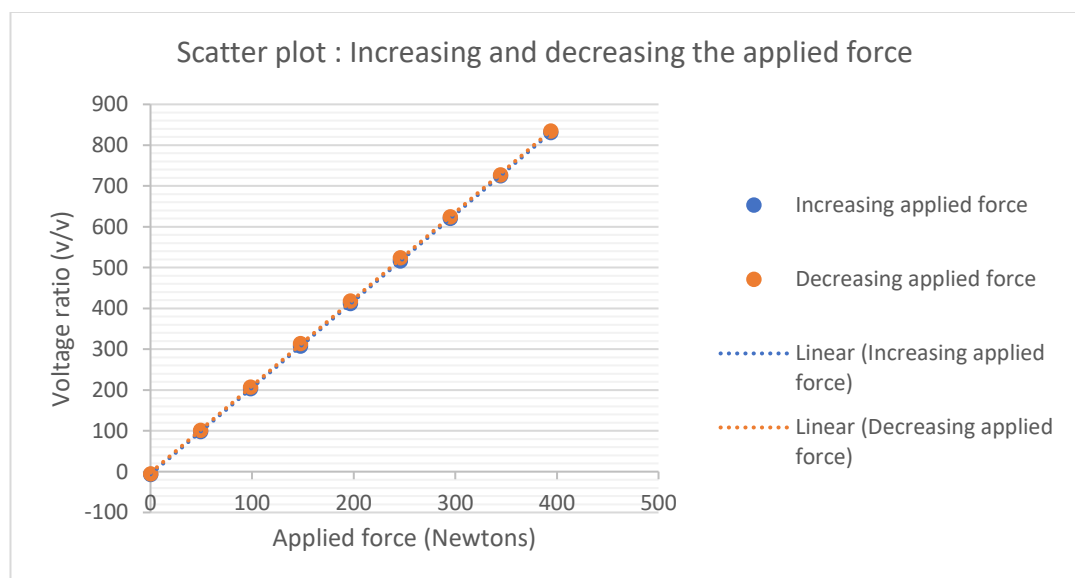


Figure 5.24 The scatter plot between applied force and voltage ratio during increasing and decreasing force conditions.

Then, the applied force and voltage output from both conditions were combined and used to calculate the final formula of the simple linear regression. After calculation, the slope (m) was 2.13, Y-intercept (b) was -4.63, and correlation efficiency was 0.99995, as shown in the equation below. Finally, another five load cells were calibrated, and the simple linear regression formula was then calculated following the previous description.

$$y = 2.13x - 4.63$$

## 5.6. Root-mean-square error in load cell calibration

The Root-mean-square error (RMSE) is a common statistic for analysing the errors. The residuals measure how far the data point is from the regression line, and an RMSE measures how spread out these residuals are. Equation 5.2 shows the formula of RMSE, where

- $t$  is variable or voltage ratio (v/v).
- $n$  is the number of non-missing data points.
- $y_i$  is the actual observation value or the voltage ratio result getting from the load cell at all testing points.
- $\hat{y}_i$  is an estimated value or the voltage ratio result calculated from the simple linear regression formula.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad 5.2.$$

The result showed that the RMSE for the regression was 2.72 (v/v). Referring to the previous calculation, 1 v/v was approximately 0.47 N. Therefore, the RMSE for this load cell was 1.28 N.

## 5.7. Simple linear regression results and errors for another five load cells

Another five load cells were calibrated using a similar method described above. Figure 5.25 illustrates the scatter plot between the applied force and voltage ratio for all load cells used in this development. Table 5.4 reports the simple linear regression results and the errors of voltage ratio and force for another five sensors. The result showed that the slope ( $m$ ) was similar on all load cells, ranging from 1.966 to 2.203. The Y-intercept results varied between load cells, ranging from -108.11 to 68.33. However, the trendline in the scatter plot went in a similar direction. For the RMSE of voltage ratio, the error ranged from 1.05 v/v to 1.74 v/v, similar to load cell 81\_0. One

voltage ratio ranged from 0.45 N to 0.51 N, and the error of force ranged from 0.50 N to 0.89 N.

In conclusion, the results showed that the error was considered small and would not affect the clinical experiment outcomes. Individual load cell calibration is essential. Even though the error was small and would not affect clinical use, the Y-intercept of each load cell varied and could affect the result if we did not perform the individual load cell calibration.

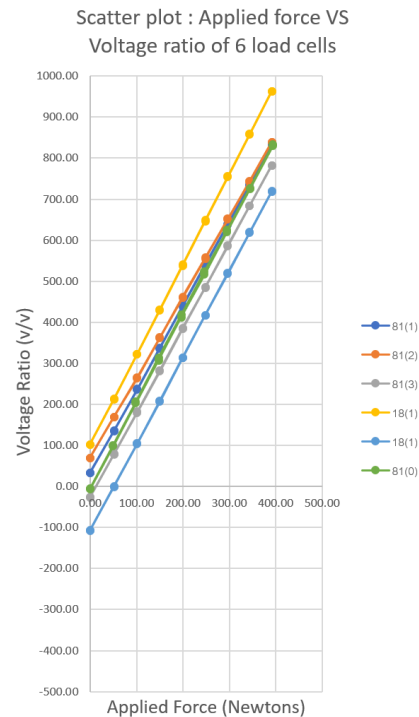


Figure 5.25 The scatter plot between applied force and voltage ratio for six load cells

Table 5.4 Simple linear regression results and RMSE of voltage and force for another 5 load cells.

Serial Number and Channel of Load cell	Slope	Y-intercept	RMSE of Voltage (v/v)	1 v/v unit (N)	Error of Force (N)
81_1	2.047	31.267	1.29	0.49	0.63
81_2	1.966	68.330	1.74	0.51	0.89
81_3	2.067	-27.197	1.21	0.48	0.59
18_0	2.203	100.734	1.20	0.45	0.54
18_1	2.114	-108.105	1.05	0.47	0.50

Note. N = Newtons, v/v = voltage ratio.

## 5.8. Development and procedure of a force-measuring system in clinical experiments

The previous section detailed a method to calibrate the load cells and assess the error. As a result, the errors of a few Newtons were relatively small, and the load cell was considered acceptable for clinical use. Subsequently a program was written to measure the magnitude of the axial forces between the manipulator and skin used to correct a scoliosis deformity. This program was again written by the author in Python using Visual Studio 2017 and the Microsoft Windows operating system (Windows 10) and ran on the testing computer. The program is provided in the electronic appendices of this thesis (Appendix 3).

The reversed simple linear regression formula from each load cell was added to the program to calculate the magnitude of the force for each individual sensor based on their individual calibration equations. The calibrated load cells were attached to the ends of manipulators and used to measure force in various areas of the scoliotic trunk, including the left and right gluteus medius, the left and right axilla, the left and right corrective, and the abdominal areas. However, the number of these seven manipulators used varied from case to case depending on the individual patient's condition. Five to six forces were commonly applied during the casting of scoliosis patients.

The program could record and display the results from all 7 transducers on the computer screen during the clinical experiments. Figure 5.26 illustrates the graphical user interface (GUI) of the force-measuring program.

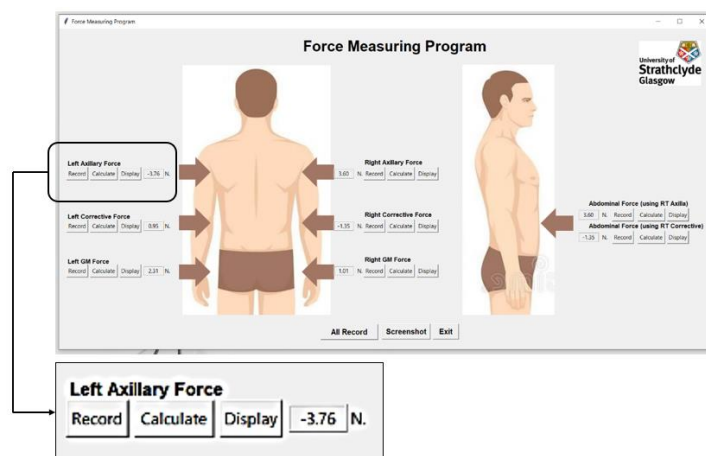


Figure 5.26 The graphical user interface (GUI) of the force-measuring program.

The procedure for measuring the force in the clinical experiment is described below.

- The patient was first asked to stand in the middle of the scoliosis casting apparatus and then perched on the edge of the sitting bar. Then, the practitioner adjusted the manipulators until reaching the optimal alignment of scoliosis deformity (fig 5.27). The method of apparatus adjustment is referred to in Chapter 4.
- The practitioner opened the program called the “**Force Measuring Program**” (fig 5.26).
- The practitioner clicked the “**Record**” button to record the voltage output, and the program then converted it to Newton.
- After that, the practitioner clicked the “**Calculate**” button to average the result from the previous click.
- Then, the practitioner clicked the “**Display**” button to display the result on the computer screen.
- This software allows the practitioner to record the result from an individual load cell or only one click to record all values simultaneously by clicking the “**All Record**” button.
- The result was then saved as a CSV file format in the C-drive. Alternatively, the practitioner clicked the “Screenshot” to capture the screen, rename the file, and save it in the desired folder.
- Finally, the practitioner clicked the “**Exit**” button to stop and exit the program.



Figure 5.27 The scoliosis casting apparatus applies multiple forces to correct the scoliosis deformity, and the program is measuring the magnitude of the force at each area.

## 5.9. Conclusion

This chapter detailed how the force-measuring system was developed, and the load cells were selected to measure the force magnitude. As a load cell result was a voltage ratio, not the Newton, a load cell calibration had to be done before use in the clinical experiment. The calibrated program was developed. The result of one load cell calibration (serial number 81\_0) showed that the load cell could measure the axial force applied to the object. The force difference between increasing and decreasing conditions was approximately 2 Newtons (with a maximum of 4 Newtons) after a delay of 3 seconds with the noise of the order of approximately 1 Newton.

The result from another five load cells showed that all load cells had a similar relationship between applied force and voltage ratio, the trendline on the scatter plot went in the same direction, the slope (m) was approximately 2, one voltage change was approximately 0.5 Newton with the noise of approximately 1 Newton, similar to sensor 81\_0. Even though the error was small and would not affect clinical use, the Y-intercept of each load cell varied and could affect the result if we did not perform the individual load cell calibration.

Another program, a force-measuring program, was developed for the clinical experiment. The reversed simple linear regression formula from each load cell was added to the program to calculate the magnitude of the force for each sensor based on their calibration equations. Now, the force-measuring system has been completely developed and is ready for the clinical experiment. However, we still need to know the force's orientation used to correct the spinal deformity in 3 dimensions, which was obtained from a low-cost postural measurement system developed in the next chapter.

**6. Chapter 6 Development of A Low-Cost Postural Measurement System: Using 2 and 3 Raspberry Pis and Cameras to Estimate Marker Positions**

## 6.1. Introduction

An X-ray of the whole spine of a person with scoliosis has been used as a standard to measure the severity of spinal deformity (Kotwicki, 2008), and this X-ray image repeatedly taken from the patient during the whole treatment process. As described in Chapter 3, the use of planar X-rays like this has significantly impacted the historical approach to diagnosing and treating scoliosis. Planar X-rays illustrated and defined the characteristics of scoliosis. Due to their widespread use in medicine, orthotists have also embraced them in orthotic treatment. However, X-ray exposure throughout the treatment and follow-up period may result in health complications in the future. Furthermore, since scoliosis is essentially a three-dimensional deformity, a three-dimensional force system must be applied during orthotic treatment in an effort to control and correct the deformity. However, planar X-ray image outcome measurement is limited to describing a two-dimensional deformity, which is not a sufficient representation of the outcome of scoliotic treatment.

Motion capture systems have been widely used to study 3-dimensional upper and lower limb movements but are rarely used in spinal evaluation, including in scoliosis research and practice. Chapter 3 previously discussed the limited use of motion capture technology in scoliosis. A previous study of 3D orthotic treatment of scoliosis (Jang, 2018) used a high-cost motion capture system to quantify the spinal parameters of scoliotic patients during the assessment process and to evaluate the spinal deformity in 3 dimensions in real-time. Interestingly, no low-cost motion capture system is currently available for quantifying spinal parameters in scoliosis applications. Although the Raspberry Pi has been widely used in computer vision applications, a low-cost motion capture system has yet to be developed to estimate the marker location and use multiple Raspberry Pis and cameras in human motion capture.

It is necessary to develop a low-cost motion capture system or a low-cost postural measurement system that can measure spinal parameters in scoliosis applications in order to close the research gaps. The new system should be affordable for clinicians and suitable for scoliosis application. The new system should also provide real-time biomechanical feedback to evaluate the immediate change in the outcome.



Additionally, the developed system's error should fall within the acceptable range in clinical practice, and its accuracy should be verified against the high-cost system.

Assessment, casting, and fitting evaluations mainly depend on decision-making by clinicians based on their experience, which is a subjective evaluation and not a quantitative evaluation method. In order to assess the change in spinal parameters while applying forces to realign the deformity, the developed system should be able to quantify spinal parameters during the assessment, optimal correction, and casting processes.

This chapter describes the development of a low-cost postural measurement system using a simple calculation technique (the right triangle similarity theorem and trigonometry) to calculate a marker position in 3-dimensional space. This development used 2 and then 3 Raspberry Pis and associated cameras to calculate marker positions and assess the error. This chapter begins with the details of the image processing technique to calculate the centre of a marker on an image and the method to calculate a marker's position in 3-dimensional space. The subsequent section will detail the method to assess the system's accuracy in 3 experiments. Experiment 1 used 2 Raspberry Pis and cameras to calculate marker positions. The cameras were attached along a wooden bar, facing toward the markers with the cameras parallel to each other and the wooden bar attached to a tripod. Experiment 2 also used 2 Raspberry Pis and cameras and the experiment setup was similar to experiment 1. However, the cameras were horizontally tilted  $45^\circ$  toward each other while remaining vertically aligned. Experiment 3 used 3 Raspberry Pis and cameras attached to the camera frame. Three cameras were attached along the same metal bar. The left and right cameras were horizontally tilted  $45^\circ$  to each other, while the middle camera was parallel to the metal bar.

## **6.2. Components of a low-cost postural measurement system**

The low-cost postural measurement system was composed of hardware and software. The hardware components consisted of Raspberry Pis and cameras, and a computer. A Raspberry Pi is a small computer developed in the United Kingdom by

the Raspberry Pi Foundation (Raspberry Pi, 2023). It can plug into a computer monitor, keyboard, mouse, electronic board, and Raspberry Pi camera. It is a capable little device that enables the user to write computer programming, connect to the internet and Bluetooth, play a game, and record video or pictures from a Raspberry Pi camera. Raspberry Pi has been used in many applications, such as robotics and weather monitoring, but not in human motion analysis (refer to Chapter 3). There are many generations of Raspberry Pi available in the market. However, Raspberry Pi 3 Model B+ was selected to use in this development (fig 6.1). This generation had a 1.4 GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT). The power supply of this Raspberry Pi was a 5V/2.5A DC micro USB (Raspberry Pi, 2023).



Figure 6.1 Raspberry Pi 3 Model B+ (Raspberry Pi, 2023).

The Raspberry Pi company also has a number of camera products. We used Raspberry Pi Camera Module 2 NoIR, the infrared camera, in this development. It has a Sony IMX219 8-megapixel sensor. It can take high-definition video and still photographs. Its camera function is similar to the standard camera but does not employ an infrared filter. This camera allows the user to see the object and take an image in a dark environment with infrared lighting. A Raspberry Pi camera was connected to each Raspberry Pi via a 15-cm ribbon cable (fig. 6.2).

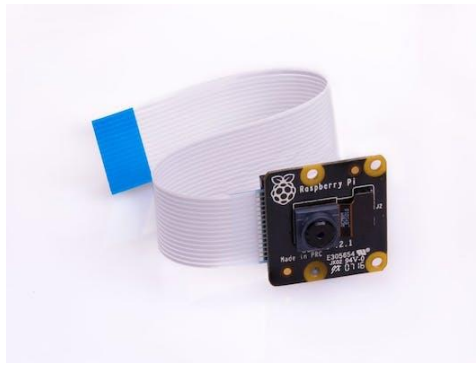


Figure 6.2 Raspberry Pi Camera Module 2 NoIR (Raspberry Pi, 2023).

The Raspberry Pi has numerous benefits in developing a low-cost measuring tool. It is a small computer that can perform data processing tasks like computer programming, load cell calculations, and image processing. The processed data from the Raspberry Pi can be sent to the computer for further calculations, such as a marker calculation from multiple Raspberry Pis. This technique can speed up the information processing process and improve computer performance. Additionally, each Raspberry Pi has its own IP address, so the Raspberry Pi and computer can interact wirelessly and independently.

A program to calculate the centre of a marker on an image was written by the author in the Python computer language using the Python 3 (IDLE) program and the Rasberian operating system on the Raspberry Pi (see appendix for full program, Appendix 4). An image was taken by the camera and then the program calculated the x and y values of a marker in the image. Those values were then manually recorded in an excel spreadsheet on the computer and excel used the data from 2 or more cameras to calculate the location in 3-dimensional space of the marker.

### **6.3. Image processing technique to calculate the centre of a marker on an image**

The previous section detailed the specifications of the Raspberry Pi and camera selected for this development and introduced a method of marker calculation. This section further describes a method to calculate a marker position on the Raspberry Pi

using the image processing technique. This technique's result was the marker's centre on an image (x and y values from each camera).

To begin this process, the author ran the program on the Raspberry Pi to open the camera. Then, the video (or image) was displayed on the monitor (fig 6.3).

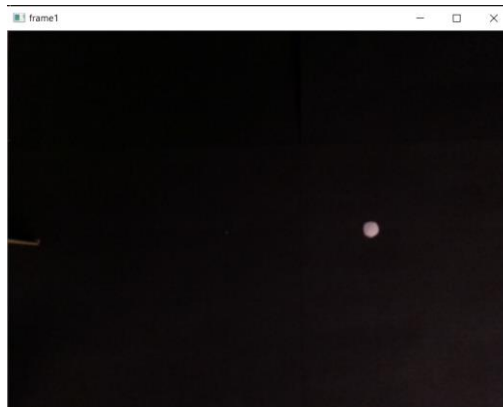


Figure 6.3 The original image after opening the Raspberry Pi camera.

The images were converted from a colour image to a grayscale image (fig. 6.4) using the “cv2.cvtColor function”. This is a method to change RGB (Red-green-blue) image, which contains 24 bits to be a shading of a black and white or gray which contains only 8 bits.

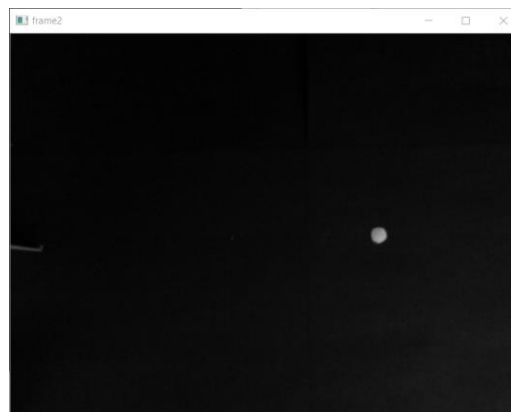


Figure 6.4 The grayscale image after applying cv2.cvtColor function.

Thresholding is the simplest image segmentation method for separating the object of interest from the environment. In this program, a specific thresholding value was manually set and could be adjusted. This number ranged from 0 to 255. The image

with zero intensity value was pure black, and the 255-intensity value was entirely white. The grayscale images were thresholded using the "cv2.threshold function" (fig 6.5) to give a black-and-white image.

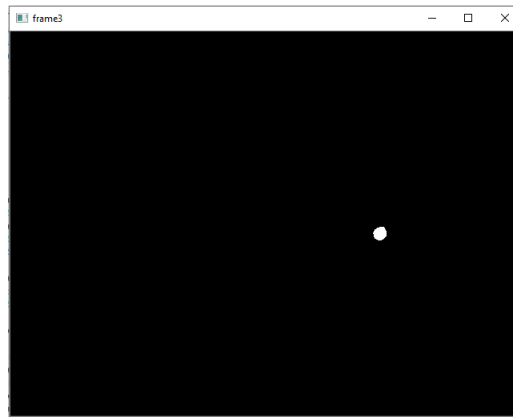


Figure 6.5 The thresholded image after applying cv2.threshold function.

After that, the region of interest or marker area on the thresholded image was extracted using the "cv2.bitwise\_and function". This function is a method to find the corresponding pixel from 2 images. The first image was the thresholded image, and the other was a maskframe image. A maskframe image was a pure black image. Then, the contour of the marker area was found and drawn using the "cv2.findContours and cv2.drawContours functions" (fig 6.6), respectively.

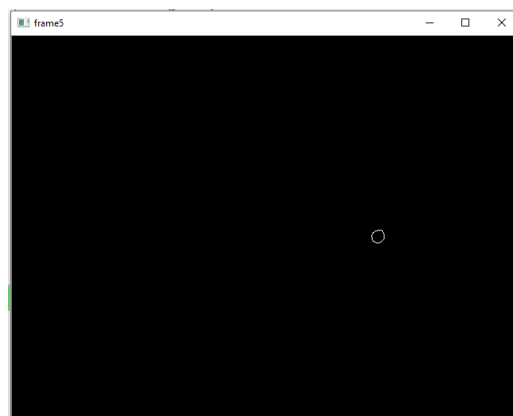


Figure 6.6 The image after applying cv2.bitwise\_and, cv2.findContours and cv2.drawContours functions, respectively.

Next, the program calculated the height and width of a detected area, calculated the centre of that area, and reported the x and y values on the monitor (fig 6.7). Finally, these x and y values were manually recorded in the excel spreadsheet on the computer. A marker position in 3-dimensional space was then calculated. The method for doing this varied between camera setups in the three experiments and the methods are given in each experimental section.

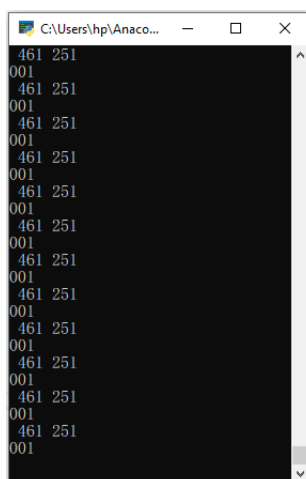


Figure 6.7 The result of the centre of the marker in x and y values

#### **6.4. Experiment 1: A low-cost postural measurement system using 2 Raspberry Pis with parallel cameras**

The previous section described the method to calculate a marker's x and y values on an image taken by a Raspberry Pi camera. This section further describes the method used to calculate a marker position in 3-dimensional space based on the right triangle similarity theorem and trigonometry.

This development began with a simple experimental setting of 2 Raspberry Pis and cameras. Experiment 1 initially set both cameras parallel to each other. Figure 6.8 illustrates the components used in Experiment 1.

- A tripod, wooden bar, flatboard, and white flat markers
- Two Raspberry Pis and cameras, monitors, keyboards, mouses, HDMI cable, and computer

- Pre-calculated excel spreadsheet



Figure 6.8 Example of required components for Experiment 1.

#### 6.4.1. Experimental setting for Experiment 1

Twenty-five markers were used in this experiment and were attached to the flatboard (fig 6.9). Each circular, flat, white marker had a diameter of 10 mm. To achieve a high contrast of colour with a white marker, the background was reversed by using black paper. The distance between markers was 100 mm on both the left and right, as well as the upper and lower positions. The order of markers is illustrated in Figure 6.9. There were five rows and five columns. The first marker was in the upper left corner, and the last marker was in the lower right corner. Furthermore, the middle marker was located at the 13th marker.

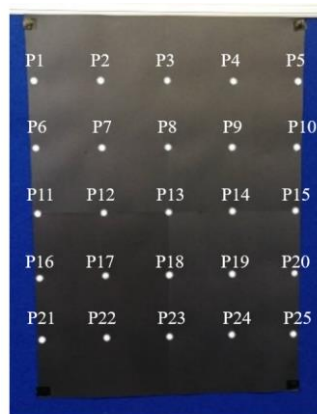


Figure 6.9 Twenty-five markers on the flatboard used in Experiment 1.

Figure 6.10 illustrates the component setting for Experiment 1. Two Raspberry Pis and cameras were attached along a wooden bar, and the distance between both cameras was 200 mm. Both cameras were set parallel and facing forward to the markers. The wooden bar was fixed to the tripod. The flatboard had to be perpendicular to the floor to maintain the same depth between markers.

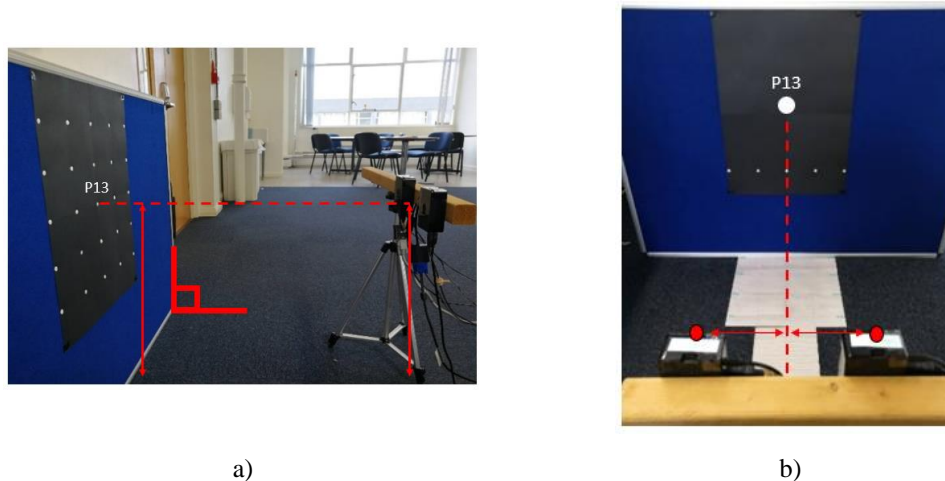


Figure 6.10 The component setting for Experiment 1, a) from the side view, and b) from the top view.

The position of the flatboard with markers attached was changed into five different depths. The distance between the cameras and the flatboard was measured in depth. Depth one was 565 mm between the cameras and the flatboard, depth two was 665 mm, depth three was 765 mm, depth four was 865 mm, and depth five was 965 mm. Therefore, 125 points were calculated in this experiment. Additionally, we considered the distance between markers and cameras as the z-value. The 125 points produced a capture volume in 3D which was 400 mm by 400 mm by 400 mm.

This experiment used a simple calculation technique (the right triangle similarity theorem and trigonometry) to calculate a marker position in 3-dimensional space. Several values had to be calculated and estimated before calculation. The values are as follows.

- The centre of the image in x and y axes ( $C_x$  and  $C_y$ ). In this experiment, we assumed that  $C_x = 320$  and  $C_y = 240$  (The camera image was 640 by 480 pixels).
- Focal length ( $f$ ). The following section will explain how to calculate this value.



- Camera orientation. In this experiment, we assumed zero-degrees of rotation for all 3 axes.
- Distance between cameras. In this experiment, the distance between both cameras was 200 mm.
- Distance between camera and marker 13.

#### 6.4.2. Focal length (f) calculation for Experiment 1

The previous section described the Experiment 1 component setup, marker locations, and relevant value estimation. On the other hand, the focal length calculation was not explained in the previous section. Therefore, this section explains how to calculate the focal length (f) for Experiment 1.

Figure 6.11 shows the geometry of the right triangle similarity between the camera plane, the image plane, the location of the marker 13, the distance between the cameras, and the distance between the cameras and marker 13. Furthermore, Equation 6.1 is a formula for the right triangle similarity to calculate the focal length (f).

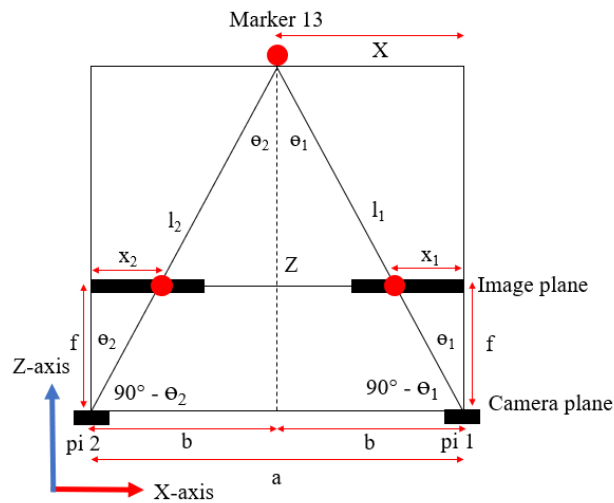


Figure 6.11 Geometry of right triangle similarity to calculate the focal length (f) for Experiment 1 in the top view

$$\tan \theta = \frac{(C_x - x_{pi})}{f} = \frac{b}{Z} \quad (6.1)$$

To be more specific, Figure 6.12 shows the geometry for a larger triangle, and Equation 6.2 shows how to find  $(\tan \theta_1)$  for that triangle.  $(b)$  is the distance between camera 1 and marker 13 on the x-axis and  $(b) = (X)$ , and  $(Z)$  is the distance between marker 13 and the camera plane on the z-axis.

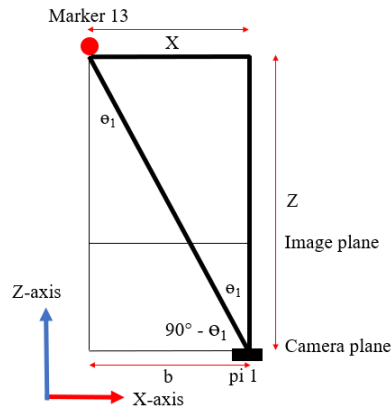


Figure 6.12 Geometry for a larger triangle for calculating a focal length ( $f$ ) in Experiment 1.

$$\tan \theta_1 = \frac{b}{Z} \quad (6.2)$$

Figure 6.13 shows the geometry for a smaller triangle, and Equation 6.3 shows how to find  $\tan \theta_1$  for that triangle.  $(x_1)$  is the distance or pixels between the middle of the image in the x-axis ( $C_x$ ) to marker 13 displayed on the image plane ( $x_1$ ). In Experiment 1, we assumed the  $(C_x)$  equals 320, and  $f$  is the distance between the image plane and the camera plane or the focal length.

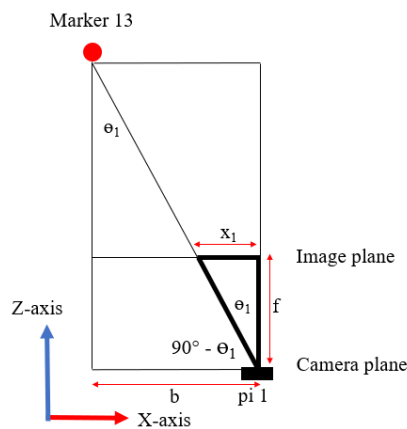


Figure 6.13 The geometry for a smaller triangle for calculating a focal length ( $f$ ) in Experiment 1.

$$\tan \theta_1 = \frac{(x_1)}{f} = \frac{(C_x - x_{pi1})}{f} = \frac{(320 - x_{pi1})}{f} \quad (6.3)$$

Because Equation 6.2 equals Equation 6.3, Equation 6.4 shows the final formula for calculating the focal length (f) in Experiment 1. We used 2 Raspberry Pi cameras in this experiment, and we assumed the component settings of both cameras would be similar. Rather than calculating the focal length for each camera, we calculated the  $(x_1)$  from the average of cameras 1 and 2 and used this value in Equation 6.4. Finally, Experiment 1's focal length (f) was 483.81.

$$\frac{b}{Z} = \frac{(320 - x_{pi1})}{f}$$

Therefore, 
$$f = \frac{(x_1)(Z)}{b} \quad (6.4)$$

### 6.4.3. A marker position calculation for Experiment 1

The previous section described a method for calculating a marker's x and y values within an image, as well as estimating and calculating all relevant values for further calculation. This section explains how to calculate a marker position in 3-dimensional space for Experiment 1. For the three-dimensional space global axes, the X axis was to the side, the Y axis was vertical, and the Z axis was forwards.

Equation 6.5 implements the “Sine rule” for three sides of the triangle shown for the geometry in Experiment 1 (refer to fig 6.11), where (a) is the distance between Raspberry Pi cameras 1 and 2, ( $l_1$ ) is the distance between camera 1 and marker 13, ( $l_2$ ) is the distance between camera 2 and marker 13, and ( $\theta_1$  and  $\theta_2$ ) are the horizontal angle of the marker displayed on the image plane away from the centre of image in x-axis.

$$\frac{a}{\sin(\theta_1 + \theta_2)} = \frac{l_1}{\sin(90^\circ - \theta_2)} = \frac{l_2}{\sin(90^\circ - \theta_1)} \quad (6.5)$$

Then, we calculated the  $(\theta_1)$  and  $(\theta_2)$  based on Equation 6.3. Then,  $(\theta_1)$  and  $(\theta_2)$  were entered into Equation 6.5 to calculate  $(l_1)$  and  $(l_2)$ .

Equation 6.6 illustrates how to calculate the component of the marker position along the x-axis using either the distance from camera 1 to the marker and the angle  $\theta_1$  ( $X_1$ ) or the distance from camera 2 to the marker and the angle  $\theta_2$  ( $X_2$ ). To get the best ( $X$ ) estimation, we averaged the  $(X_1)$  and  $(X_2)$  for the final result of ( $X$ ) (Equation 6.7).

$$X_1 = l_1 \sin \theta_1 \quad ; \quad X_2 = a - l_2 \sin \theta_2 \quad (6.6)$$

$$X = \frac{(X_1 + X_2)}{2} \quad (6.7)$$

Equation 6.8 illustrates how to calculate a marker position on the z-axis, the distance from the camera to the marker on the z-axis.  $(Z_1)$  is the result from camera 1, and  $(Z_2)$  is the result of camera 2 (Equation 6.8). To get the best ( $Z$ ) estimation, we again averaged the  $(Z_1)$  and  $(Z_2)$  for the final result of ( $Z$ ) (Equation 6.9).

$$Z_1 = l_1 \cos \theta_1 \quad ; \quad Z_2 = l_2 \cos \theta_2 \quad (6.8)$$

$$Z = \frac{(Z_1 + Z_2)}{2} \quad (6.9)$$

Figure 6.14 shows the geometry of the right triangle similarity in the side view, and Equation 6.10 illustrates how to calculate a marker position on the y-axis.  $(y_1)$  and  $(y_2)$  are the distance or pixels between the middle of the image on the y-axis ( $C_y$ ) to the marker displayed on the image plane of cameras 1 and 2. ( $Y$ ) is the distance (or height) from the camera to the marker on the y-axis. ( $Y_1$ ) is the result from camera 1, and  $(Y_2)$  is the result of camera 2 (Equation 6.10). To get the best ( $Y$ ) estimation, we again averaged the  $(Y_1)$  and  $(Y_2)$  for the final result of ( $Y$ ) (Equation 6.11).

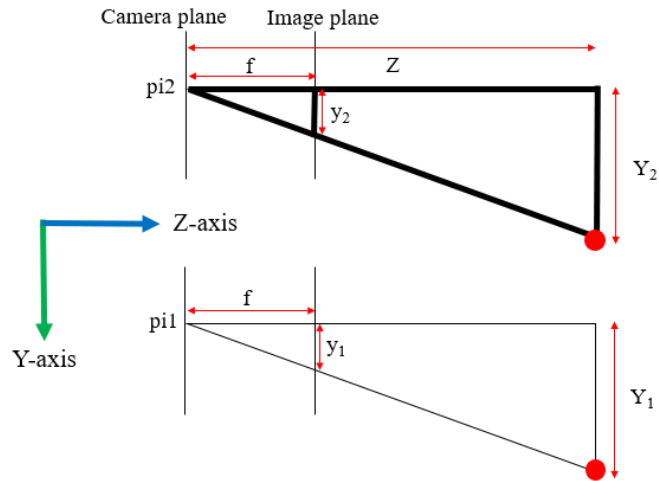


Figure 6.14 Geometry of right triangle similarity for Experiment 1 in the side view

$$\frac{y_1}{f} = \frac{Y_1}{Z} \quad ; \quad \frac{y_2}{f} = \frac{Y_2}{Z} \quad (6.10)$$

$$Y = \frac{(Y_1 + Y_2)}{2} \quad (6.11)$$

All 125 testing points were then calculated based on the calculation technique mentioned above. In the next step, we evaluate the experiment's accuracy by comparing it to the reference values.

#### 6.4.4. The error calculation for Experiment 1

The previous section described a method for calculating a marker position in three dimensions. The error must be evaluated in the following step to assess this experiment's accuracy.

The Root-mean-square error (RMSE) is a common statistic for analysing the errors. Equation 6.12 shows the formula of RMSE, where

- $n$  is the number of markers.
- $y_i$  is the marker positions calculated by the system.
- $\hat{y}_i$  is the reference marker positions.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad 6.12.$$

We divided the reference marker positions into 2 groups. In Reference 1, we used only one marker, marker 13 at depth 3 (765 mm), as a starting point because it was in the middle volume of this experiment. Then, in X, Y, and Z, we added 100 mm to calculate other reference points. In Reference 2, instead of using only one marker as a starting point, we used marker 13 at each depth to become the starting point of each depth. Then, we added 100 mm in X and Y to calculate other reference points.

Figure 6.15 illustrates the use of the red colour to represent the error range. White colour means an error between 0 and 5 mm; light red is more than 5mm but less than 10 mm; typical red is more than 10 mm but less than 20 mm; and dark red is more than 20 mm, respectively. Then, this colour range is used to represent the error of each marker in the following illustrations.

	0 – 5 mm error
	5 < x < 10 mm error
	11 < x < 20 mm error
	> 20 mm error

Figure 6.15 The range of red colour to represent the range of the error in Experiment 1.

Table 6.1 illustrates the RMSE comparing between Reference 1 and the system calculation. The RMSE of X, Y, Z, were 7.46 mm, 8.99 mm, 58.95 mm, and the RMSE of the three dimensional euclidean error was 8.78 mm, respectively.

Table 6.1 RMSE comparison between Reference 1 and the system in Experiment 1

	RMSE (mm)
X from all depths, n=125	7.46
Y from all depths, n=125	8.99
Z from all depths, n=125	58.95
3-dimensional euclidean error	8.78

Note: mm = millimetre, n = number of markers, RMSE = Root-mean-square error.

Figure 6.16 shows the X error across the volume, represented by the red colour range. Figures 6.17 and 6.18 also show the Y and Z errors, respectively. For the X error, the lowest error appeared at depth three, a middle depth of this experiment. The error then gradually increased as the markers moved away from depth three. Depth one had the greatest error, followed by depth five, both of which were the furthest away from depth three.

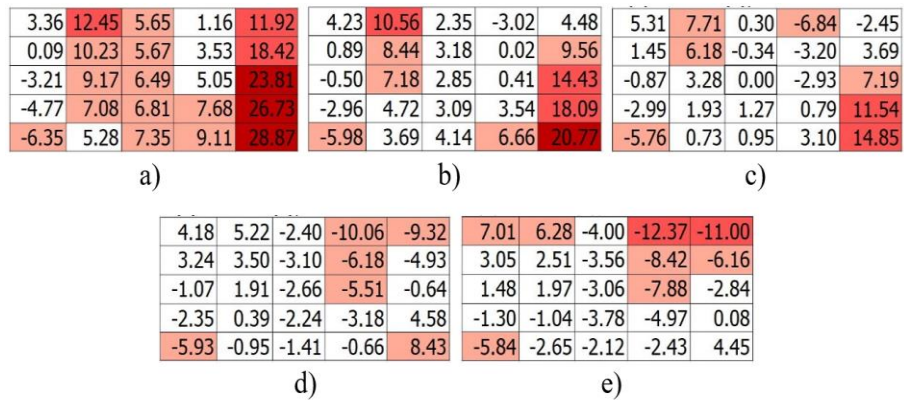


Figure 6.16 A different tone of red colour to represent the X error compared with Reference 1. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.

A similar error trend in X was presented in Y, as shown in Figure 6.17. However, the Y error was greater than in X. The smallest error appeared at depth three. The error gradually increased when the markers were moved away from depth three.

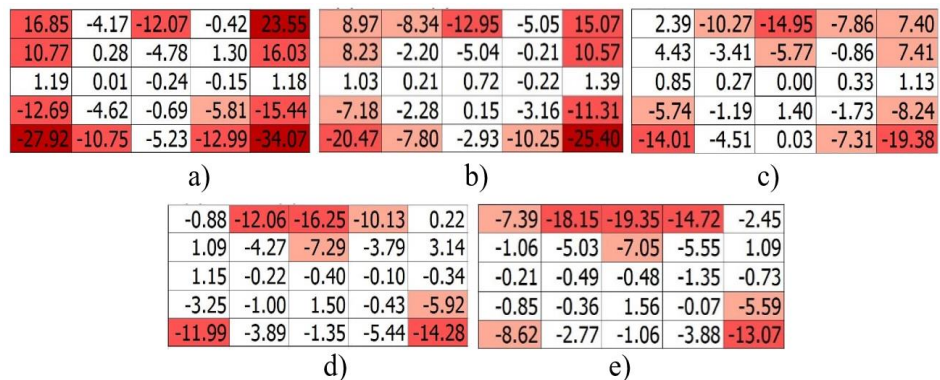


Figure 6.17 A different tone of red colour to represent the Y error compared with Reference 1. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.

A significant error was reported in Z. Errors greater than 20 mm were displayed in most areas and all depths, as shown in Figure 6.18. The lower row had a greater error than the upper row. Additionally, the first column had a greater error than the second column, and the fifth column also had a greater error than the fourth column.

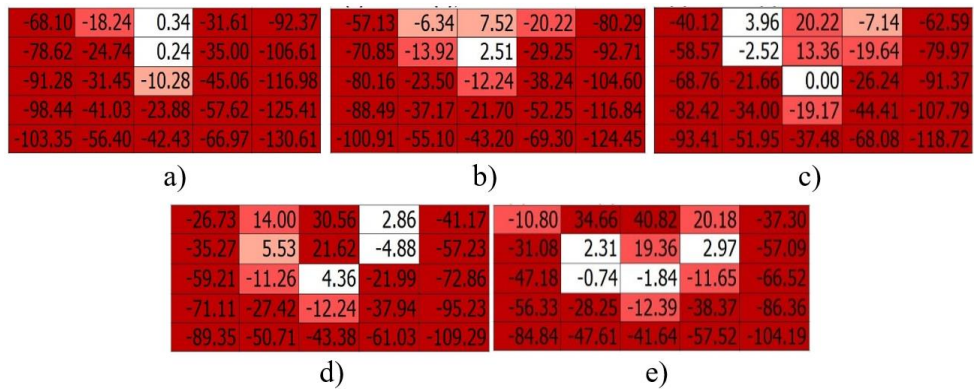


Figure 6.18 A different tone of red colour to represent the Z error compared with Reference 1. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.

Table 6.2 reports the RMSE comparing between Reference 2 and the system calculation. The RMSE of X, Y, and Z were 6.20 mm, 8.95 mm, 55.34 mm, and the RMSE of the 3-dimensional euclidean error was 8.75 mm, respectively.

Table 6.2 Root-mean-square error comparison between Reference 2 and the system in Experiment 1

	RMSE (mm)
X from all depths, n=125	6.20
Y from all depths, n=125	8.95
Z from all depths, n=125	55.34
3-dimensional euclidean error	8.75

Note: mm = millimetre, n = number of markers, RMSE = Root-mean-square error.

In the comparison between Reference 2 and the system calculation, Figure 6.19 shows the X error, represented by the red colour range. Figures 6.20 and 6.21 also show the Y and Z errors, respectively. Focusing on the X error, a higher error was presented at depth one. Then, the error gradually decreased when the depth increased.





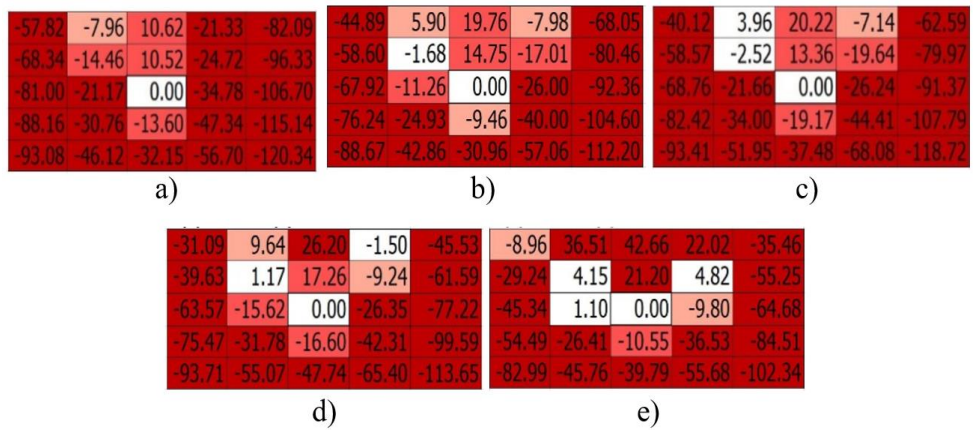


Figure 6.21 A different tone of red colour to represent the Z error compared with Reference 2. a). depth one, b). depth two, c). depth three, d). depth four, and e). depth five.

### 6.4.5. Discussion and conclusion of Experiment 1

Experiment 1 described the first and simplest concept to develop a low-cost postural measurement system using 2 Raspberry Pis and 2 cameras. This concept was based on a simple calculation technique using the right triangle similarity theorem and trigonometry to calculate a marker position in 3-dimensional space. This experiment used 2 Raspberry Pis and cameras and attached both cameras along a wooden bar. Cameras were set parallel and facing forward to the markers. Twenty-five markers were attached to a flatboard, and a flatboard was changed in 5 positions along the Z-axis. The system then calculated marker positions in all depths.

The results were compared with the reference to assess the system's accuracy, and there were 2 references for this comparison. Reference 1 was the best choice for this experiment because we could assess the error after calculating a cubic volume. Furthermore, in Reference 2, we would also like to see the error at each depth when using a middle marker of each depth as its reference.

According to the literature review in Chapter 3, current low-cost motion system errors range between 5 and 20 mm. The previous article reported the error of measurement of coronal Cobb angle in radiographs was about 5° (Wang *et al.*, 2018). As a result, our low-cost postural measurement system aims for less than 5 mm of error and it might be acceptable if the error is less than 10 mm.

The RMSE of X, Y, Z, and 3D Euclidean error when compared to Reference 1 were 7.46 mm, 8.99 mm, 58.95 mm, and 8.78 mm, respectively. When comparing Reference 2 and the system calculation, the RMSE were 6.20 mm, 8.95 mm, 55.34 mm, and 8.75 mm, respectively. Overall, the errors in both comparisons were similar, with comparison 1 slightly higher than comparison 2. All errors were greater than 5 mm, which exceeded our goal. In both comparisons, the X and Y errors were greater than 5 mm but less than 10 mm, which is within the acceptable error range. However, we should reduce the error in the following experiment. Unfortunately, the Z error in both comparisons was extremely high, and we could not accept it. Furthermore, the Z error caused the 3-dimensional Euclidean error to be high, which could not be accepted. All the Z errors were positive indicating a systematic error in the system which could be removed by simple linear regression if required.

After careful consideration, we determined that two reasons could have caused the errors. The first reason was the physical distance between both cameras. Because we needed to calculate all markers, the cameras should not be too far from one another. Otherwise, some markers along the volume's edge would not be seen in both cameras and could not be calculated. As a result, both cameras were placed as close together as possible (200 mm for this experiment, 50% of the width of the capture volume). Shortening the distance between cameras may result in smaller shaft angles of ( $\theta_1$ ) and ( $\theta_2$ ). A small change in the x and y values of the marker calculated on the image can significantly change the angle and Z value. The second reason could be due to the alignment of both cameras, which were parallel to each other. As a result, the next experiment should try increasing the distance between both cameras as well as horizontally tilting the cameras about  $45^\circ$  so that they could both still see all the markers and reassess the errors.

## **6.5. Experiment 2: A low-cost postural measurement system using 2 Raspberry Pi cameras with $45^\circ$ horizontally tilting cameras**

Experiment 1 demonstrated the practical possibility of developing and using a new low-cost postural measurement system to calculate marker positions in three dimensions. However, the errors were still high and needed to be reduced in the next

experiment. Therefore, Experiment 2 describes a method for calculating marker position when 2 cameras are tilted 45° horizontally.

This experiment used the same image processing technique to calculate the centre of a marker and the right triangle similarity theorem and trigonometry to calculate a marker's position. However, some calculation formulas had to be adjusted due to the change in camera position. Further to increase the practicality of the measurement technique, instead of reporting the x and y values on the Raspberry Pi monitor and then transposing those values manually to a spreadsheet, the values were wirelessly sent to the computer and saved in the CSV file (see the full program in Appendix 5). This meant that the computer and all the active Raspberry Pis communicated with each other via Wi-Fi. Furthermore, the computer installed a freeware program called VNC Viewer (RealVNC, 2023) (fig 6.22), which allowed us to access and adjust the Raspberry Pi via this program. This system did not require an additional monitor, keyboard, and mouse for the Raspberry Pis. The computer installed another freeware program called FileZilla (FileZilla, 2023) (fig 6.23), which enabled us to transfer files between the computer and the Raspberry Pi.

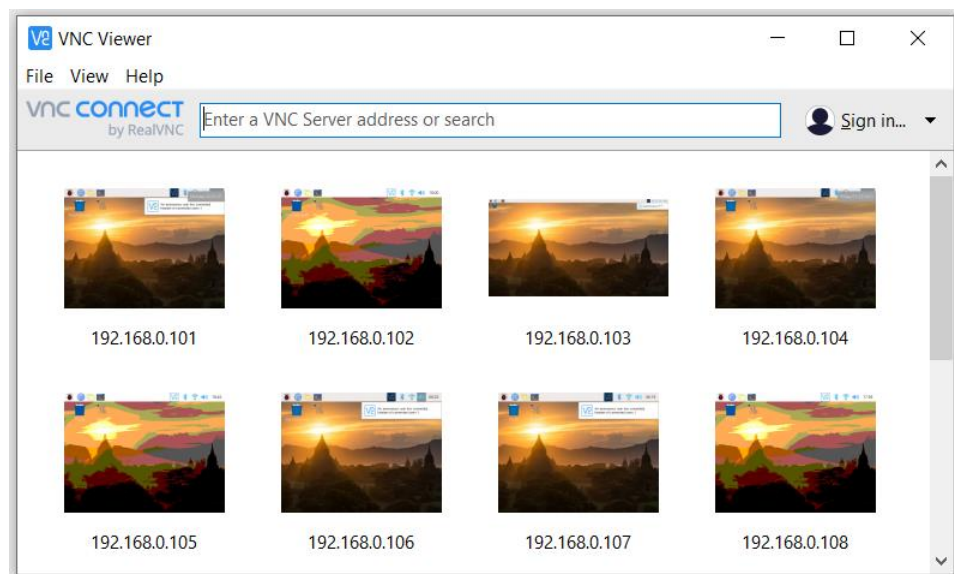


Figure 6.22 The VNC Viewer program installed on the computer.

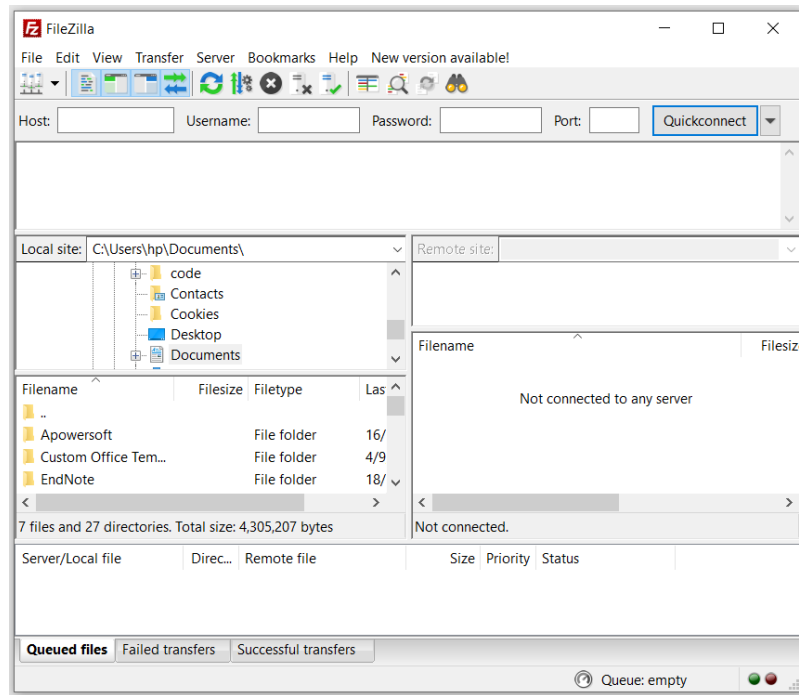


Figure 6.23 The FileZilla program installed on the computer.

### 6.5.1. Experimental setting for Experiment 2

Experiment 2 used the same experimental setup as Experiment 1. Twenty-five markers were attached to the flatboard (fig 6.24), and the distance between the markers on the X and Y axes was 100 mm. The marker's size was reduced from 10 mm in Experiment 1 to 6 mm in Experiment 2, and the shape was also changed from a circle to a square shape. We needed as small a marker as possible, but where the system could still detect a marker, and the square shape allowed us to fabricate it more easily in the trial phase. A flatboard with markers attached was moved into 5 depths along the Z-axis, similar to Experiment 1. However, there were some changes in this experiment. The two Raspberry Pis and cameras were rotated 45° horizontally (fig 6.25), and the distance between both cameras was increased to 1,235 mm.



Figure 6.24 Markers attached on the flatboard for Experiment 2

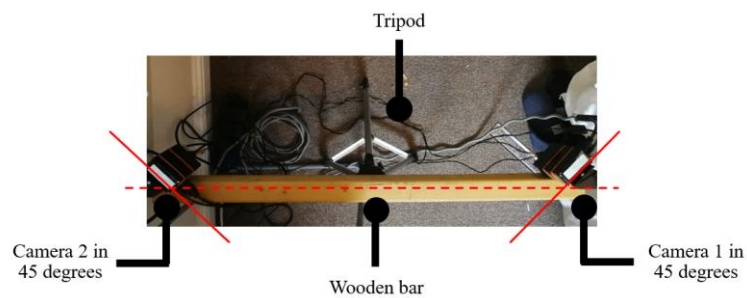


Figure 6.25 The camera position for Experiment 2.

### 6.5.2. A marker position calculation for Experiment 2

Experiment 2 continued the concept of the right triangle similarity theorem and trigonometry that were used in Experiment 1. For Experiment 2, a geometry is illustrated in Figure 6.26.

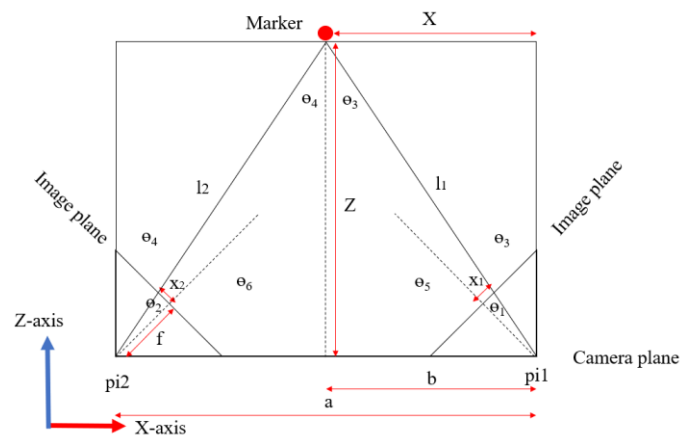


Figure 6.26 The geometry used to calculate the marker position of Experiment 2 in the top view.

Equation 6.13 implements the "Sine rule" from the geometry in Experiment 2, where (a) is the distance between Raspberry Pi cameras 1 and 2, ( $l_1$ ) is the distance between camera 1 and marker, ( $l_2$ ) is the distance between camera 2 and marker, ( $\theta_1$  and  $\theta_2$ ) are the horizontal angle of the marker displayed on the image plane away from the centre of the image in the x-axis. Additionally,  $\theta_3 = 45^\circ - \theta_1$ ,  $\theta_4 = 45^\circ - \theta_2$ ,  $\theta_5 = 45^\circ + \theta_1$ ,  $\theta_6 = 45^\circ + \theta_2$ .

Then, we calculated the ( $\theta_1$ ) and ( $\theta_2$ ) based on Equation 6.3. Then, we calculated ( $\theta_3$ ), ( $\theta_4$ ), ( $\theta_5$ ) and ( $\theta_6$ ). After that, ( $\theta_3$ ) + ( $\theta_4$ ), ( $\theta_5$ ) and ( $\theta_6$ ) were entered into Equation 6.13 to calculate ( $l_1$ ) and ( $l_2$ ).

$$\frac{a}{\sin(\theta_3 + \theta_4)} = \frac{l_1}{\sin(\theta_5)} = \frac{l_2}{\sin(\theta_6)} \quad (6.13)$$

Equation 6.14 illustrates how to calculate a marker position on the x-axis using the distance from camera 1 to the marker and the angle  $\theta_3$  ( $X_1$ ) or using the distance from camera 2 to the marker and the angle  $\theta_4$  ( $X_2$ ) (Equation 6.14). To get the best (X) estimation, we again averaged ( $X_1$ ) and ( $X_2$ ) for the final result of (X) (Equation 6.7).

$$X_1 = l_1 \sin \theta_3 ; X_2 = a - l_2 \sin \theta_4 \quad (6.14)$$

Equation 6.15 illustrates how to calculate a marker position on the Z-axis, using the distance from camera 1 to the marker and the angle  $\theta_3$  ( $Z_1$ ) or using the distance from camera 2 to the marker and the angle  $\theta_4$  ( $Z_2$ ) (Equation 6.15). To get the best (Z) estimation, we again the ( $Z_1$ ) and ( $Z_2$ ) for the final result of (Z) (Equation 6.9).

$$Z_1 = l_1 \cos \theta_3 ; Z_2 = l_2 \cos \theta_4 \quad (6.15)$$

Figure 6.27 illustrates the side view geometry of the right triangle similarity. The Y value was calculated using a similar concept to that described in Experiment 1. However, because the Raspberry Pis and cameras were tilted at  $45^\circ$  horizontally, the value of focal length (f) from the previous calculation could not be directly entered in

this calculation. As a result, trigonometry was used to compute this focal length using this formula:  $f_y = f \cos 45^\circ$ .

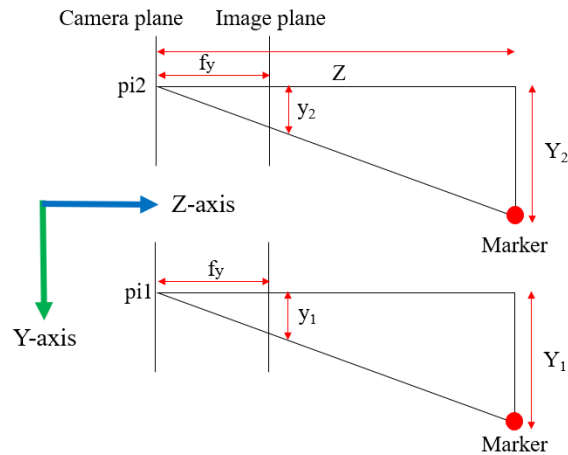


Figure 6.27 The geometry used to calculate the marker position of Experiment 2 in the side view.

All testing points were then calculated using the above-calculated technique. The next step is to compare Experiment 2's accuracy to the reference.

### 6.5.3. Introduction to simple linear regression and multiple linear regression

- **Simple linear regression**

Simple linear regression (SLR) is a common statistic used to predict the relationship between the independent and the dependent variables (refer to Equation 5.1). It can be used to evaluate and remove any systematic errors in the scaling or offset of results using a linear model. This study used the values calculated from the system and the reference values as the independent and dependent variables of the model. The method for calculating the reference was similar to that described in Reference 1 of Experiment 1. Then, we calculated a SLR formula for X, Y, and Z, as shown in Table 6.3. The system's X and the reference's X values were used to calculate the simple linear regression. This was also done for Y and Z. The system values (original values) were then entered into the simple linear regression formula to calculate new values.



Table 6.3 Simple linear regression formula used in Experiment 2.

	Formula
Simple linear regression of X	$y = 0.971263(x) + 16.293228$
Simple linear regression of Y	$y = 0.887673(x) - 6.080558$
Simple linear regression of Z	$y = 1.022173(x) - 27.024005$

- **Simple linear regression with multiple conditions**

This experiment attempted an alternative method to reduce the errors called a simple linear regression (SLR) with multiple conditions. A simple linear regression formula was calculated for each depth, and 5 depths were calculated. We used the distance of depth on the z-axis to set up a condition, and the regression was performed for X and Y. Table 6.4 reports a SLR formula from depth 1 to depth 5, respectively.

Table 6.4 Formula of SLR with multiple conditions used in Experiment 2.

	Formula
SLR with multiple condition for X at depth 1	$y = 0.996146(x) + 0.168278$
SLR with multiple condition for X at depth 2	$y = 0.996844(x) - 0.417202$
SLR with multiple condition for X at depth 3	$y = 0.982741(x) + 0.981659$
SLR with multiple condition for X at depth 4	$y = 0.947491(x) + 30.465277$
SLR with multiple condition for X at depth 5	$y = 0.939762(x) + 38.208009$
SLR with multiple condition for Y at depth 1	$y = 1.033098(x) - 23.736638$
SLR with multiple condition for Y at depth 2	$y = 0.952913(x) - 14.413025$
SLR with multiple condition for Y at depth 3	$y = 0.896042(x) - 5.449127$
SLR with multiple condition for Y at depth 4	$y = 0.857415(x) + 3.033053$
SLR with multiple condition for Y at depth 5	$y = 0.8125576(x) + 12.279573$

Note: SLR = Simple linear regression

- **Multiple linear regression**

Another method for reducing error is to estimate a new marker position using multiple linear regression (MLR), which we also tried in Experiment 2. MLR is a

statistic used to predict the dependent variable in relation to multiple independent variables. Equation 6.16 illustrates a formula for MLR, where (y) is a predicted dependent variable, (x<sub>1</sub>) to (x<sub>n</sub>) are the independent variables, (b<sub>0</sub>) is (y) when all independent variables are equal to zero, and (b<sub>1</sub>) to (b<sub>n</sub>) are the estimated regression coefficients. All variables were statistically analysed in the SPSS program (SPSS, Version 18.0.), where independent and dependent variables were X, Y, and Z values from the system and reference values. Then, we calculated MLR formula for X, Y, and Z, as shown in Table 6.5. This method of calculation may also improve accuracy. However, the following step is required to evaluate the error.

$$y = b_0 + b_1x_1 + b_2x_2 + \dots b_nx_n \quad (6.16)$$

Table 6.5 MLR formula used in Experiment 2.

	Formula
MLR of X	$y = 8.423 + 0.971(X) + 0.003(Y) + 0.01(Z)$
MLR of Y	$y = (-98.492) + (-0.002)(X) + 0.899(Y) + 0.113(Z)$
MLR of Z	$y = 12.098 + (-0.074)(X) + 0.011(Y) + 1.029(Z)$

Note: MLR = Multiple linear regression

#### 6.5.4. The error calculation for Experiment 2

The previous section described a method for calculating a marker position when 2 Raspberry Pis and cameras were horizontally tilted 45°, as well as a method for reducing error using SLR and MLR. This section describes the experiment's accuracy with the RMSE still used to assess the error.

Table 6.6 compares the RMSE for Experiment 2 before and after SLR. As shown in the table, all RMSEs were reduced after using SLR, representing 5.83 mm in X (from 7.86 mm), 24.49 mm in Y (from 33.25 mm), 15.70 mm in Z (from 18.20 mm), and 8.79 mm in 3D Euclidean error (from 9.42 mm). However, the Y and Z errors exceeded our acceptable (10 mm) error.

Table 6.6 also reports the RMSE after applying SLR with multiple conditions. Overall, all RMSEs decreased considerably, and the error was also slightly lower than

previously: 4.84 mm in X, 5.67 mm in Y, 15.70 mm in Z, and 8.00 mm in 3D Euclidean error.

Table 6.6 further reports the RMSE after applying MLR. Overall, all RMSEs decreased further and considerably compared to the original values, particularly the Y and Z values. The errors were 5.57 mm in X, 15.13 mm in Y, 7.69 mm in Z, and 8.54 mm in 3D Euclidean error, respectively.

Table 6.6 The results of RMSEs before and after applying SLR, SLR with multiple conditions, and MLR in Experiment 2 (n = 125).

	RMSE (mm)			
	X	Y	Z	3D Euclidean error
Before SLR	7.86	33.25	18.20	9.42
After SLR	5.83	24.49	15.70	8.79
After SLR with multiple conditions	4.84	5.67	15.70	8.00
After MLR	5.57	15.13	7.69	8.54

Note: RMSE = Root-mean-square error, RLR = simple linear regression, MLR = multiple linear regress, mm = millimetre, n = number of markers.

### 6.5.5. Discussion and conclusion of Experiment 2

Experiment 2 described the concept to develop a low-cost postural measurement system using 2 Raspberry Pis and cameras at greater separation and angled at 45°. This experiment was still based on a simple calculation technique using the right triangle similarity theorem and trigonometry to calculate a marker position in 3-dimensional space, similar to Experiment 1.

On the programming side, the calculated values from each Raspberry Pi were sent to the computer, and the whole system could communicate wirelessly via WI-FI, which improved the calculation process and convenience for the user when using this system.

All regressions reduced the X error by 2 - 3 mm. The error was within the acceptable (10 mm) value but did not reach the goal (5 mm). When comparing the X error before regression in Experiments 1 and 2, the error appeared similar even though the camera's position changed.

The Y error was extremely high when the camera position changed. When comparing the Y error before regression in Experiments 1 and 2, the error increased from 8.99 mm in Experiment 1 to 33.25 mm in Experiment 2. The SLR could reduce

the error approximately 10 mm (24.49 mm in Experiment 2), and MLR could reduce the error approximately half (15.13 mm in Experiment 2) of the total error. The lowest error was reported when the SLR with multiple conditions were applied. The error could be reduced to 5.67 mm, slightly higher than the goal.

The Z error was significantly decreased from this camera position. When comparing the Z error before regression in Experiments 1 and 2, the error decreased from 58.95 mm in Experiment 1 to 18.20 mm in Experiment 2. The SLR could reduce the error approximately 5 mm (15.70 mm in Experiment 2), and MLR could reduce the error approximately half (7.69 mm in Experiment 2) of the total error. The error after applying SLR with multiple conditions was quite similar to MLR. The error was within the acceptable (10 mm) value but did not reach the goal (5 mm).

Overall, SLR with multiple conditions resulted in the lowest error, followed by MLR and SLR. However, SLR with multiple conditions made calculating the new marker positions more complex because different depths required different formulas to calculate marker positions, which might not be applicable. Furthermore, MLR could provide a good result with an acceptable (10 mm) error for X, Z, and the 3D Euclidean error but not for Y.

After careful consideration, we determined that four factors could have caused the error. The first reason was the image's quality. Experiments 1 and 2 used raw images from the cameras to calculate the marker's location. However, the property of the camera lens may not be good quality, which can distort the image. As a result, the x and y values (Centre of marker on image) could be shifted, and this could cause the error. The second reason was the camera variables estimation, including  $C_x$ ,  $C_y$ , and focal length. We estimated and calculated these variables, which might not be accurate. The third reason was the position of the cameras. Two Raspberry Pis and cameras were mounted on the wooden bar. In Experiment 2, we added the adapter on the Raspberry Pi that could adjust the degrees of horizontal rotation. However, a slight rotation change of the camera might affect the calculation result, causing the error. The final reason was the number of cameras used to calculate the marker position. The study from Nagymáté and Kiss (Nagymáté and Kiss, 2018) showed that increasing the number of cameras in a high-cost motion capture system significantly reduced errors.

As a result, the next experiment should calibrate the camera to improve image quality prior to marker calculation and calculate each camera's  $C_x$ ,  $C_y$ , and focal length values. Furthermore, we should increase the number of cameras and position them more precisely on the tripod or metal frame.

## **6.6. Experiment 3: A low-cost postural measurement system using 3 Raspberry Pi cameras**

The previous section described a method for calculating marker position using 2 Raspberry Pis and cameras with different camera settings. Experiment 3 implemented Experiment 2's recommendations. This experiment describes a method for calibrating the camera, increasing the number of cameras to 3, positioning the cameras on a more precise metal frame, then calculating marker positions, and assessing errors.

### **6.6.1. Camera calibration**

Camera calibration is a process of determining the intrinsic and extrinsic parameters of a camera in order to rectify distortions and provide precise measurements in computer vision applications. The process calculates various factors, including focal length, principal point, and lens distortion coefficients. Camera calibration also relates the measurement from the camera (computer unit, pixels) to the size in 3-dimensional space (e.g., millimetre).

Camera calibration gives the "Intrinsic camera parameters", including the camera's geometry or matrix and camera distortion of the lenses or distortion coefficients. There is no perfect camera and lens in practice due to the properties of the lens and errors in the manufacturing process. As a result, errors distort the lens and also the image. There are 2 types of distortion, which are radial and tangential distortions (Brown, 1971, Fryer and Brown, 1986). The radial distortion causes straight lines on the image to appear curved, and tangential distortion occurs when the image plane and camera plane are not perfectly parallel, which causes some areas on the image to look nearer than expected. Figure 6.28 shows an image of a chessboard.

The chessboard is square in shape, the edges are straight and perpendicular to each other (fig 6.28a). However, the chessboard's edge appears curved due to distortion in the lens (fig 6.28b).



Figure 6.28 The image of the chessboard, a) original shape of chessboard and b) distorted image taken by distorted lens (Medium, 2023).

The intrinsic camera parameters are used to correct image distortion and make the final image accurate before marker calculation. Four results are obtained after camera calibration, including the camera matrix, distortion matrix, rotation matrix, and translation vector of the calibrated image with respect to the camera. For Experiment 3, however, we only used the camera and distortion matrices to undistort the image and calculate marker position.

The "Camera matrix" is a 3 x 3 matrix (Equation 6.17), which contains the focal length in the x-axis ( $f_x$ ), the focal length in the y-axis ( $f_y$ ), the principal point in the x-axis ( $C_x$ ), and the principal point in the y-axis ( $C_y$ ), respectively.

$$\text{Camera matrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (6.17)$$

The "Distortion matrix" or "Distortion coefficients" is a 5 x 1 matrix (Equation 6.18), which is used for undistorting the image. The parameter of  $k_1$ ,  $k_2$ , and  $k_3$  relates to radial distortion, and  $p_1$  and  $p_2$  relates to tangential distortion, respectively.

$$\text{Distortion Coefficients} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \quad (6.18)$$

A program to capture camera images and perform the camera calibration was written in the Python computer language using the Python 3 (IDLE) program and the Rasberian operating system on the Raspberry Pi (see the full program at Appendix 6). The steps of camera calibration are described below.

- The program to capture the images was opened and run to capture the chessboard images. We selected a 9x6 pattern for this experiment. Figures 6.29 and 6.30 show the chessboard pattern and the program running to capture the images in this experiment.

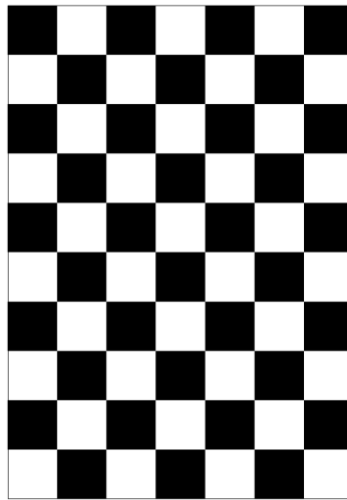


Figure 6.29 The chessboard pattern used in Experiment 3.

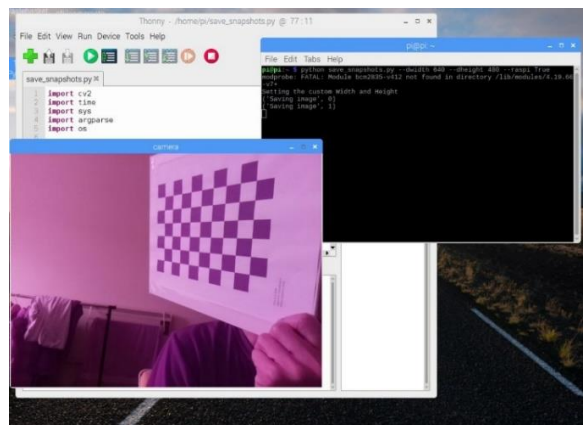


Figure 6.30 The program to capture the image in Experiment 3.

- The images of calibrated pattern had to be taken in various positions and orientations, and it is recommended to take at least 30 images per camera for

calibration. To take the new image, the user pressed the space bar on the keyboard and then pressed Esc when 30 different images had been taken. The program then completed. All recorded images were saved in a Raspberry Pi folder. Figure 6.31 shows an example of images used for camera calibration.

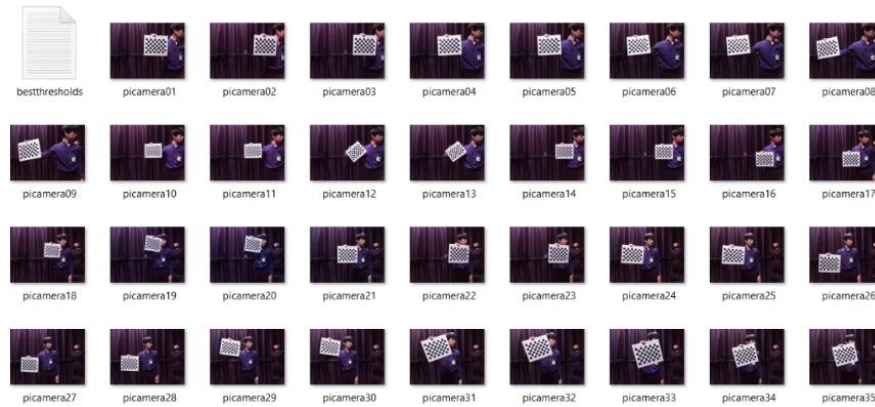


Figure 6.31 An example of images used for camera calibration.

- The camera calibration program was opened and run to calibrate the camera. The program first checked the chessboard pattern on the image. If the pattern matches the program setting and a complete chessboard was found, the image was used in the subsequent calculation. If not, the program rejected the image and checked the next image.
- Once the pattern matched the setting, the program used the “cv2.findChessboardCorners function” to find the chessboard's corners and saved them in the list of good images for further calculation. This procedure was applied to all images.
- After that, the “cv2.cornerSubPix function” was applied to the image to find more exact corner positions.
- Then, the “cv2.drawChessboardCorners function” drew or rendered the corners of the chessboard on the image. Figure 6.32 shows the result after drawing the chessboard corners.



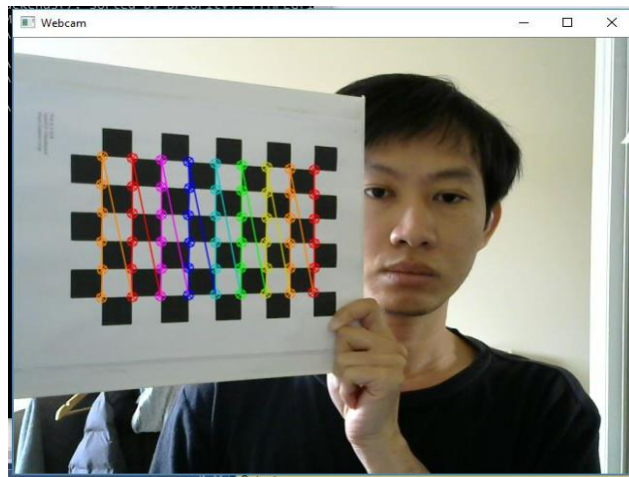


Figure 6.32 The result after drawing the chessboard corners.

- Finally, the “cv2.calibrateCamera function” was applied to all the good images (n = 30+) to calculate the camera matrices. The calibration results were saved in the .txt file and were used in the next step. Figure 6.33 illustrates the example of calibration results saved in the Raspberry Pi.

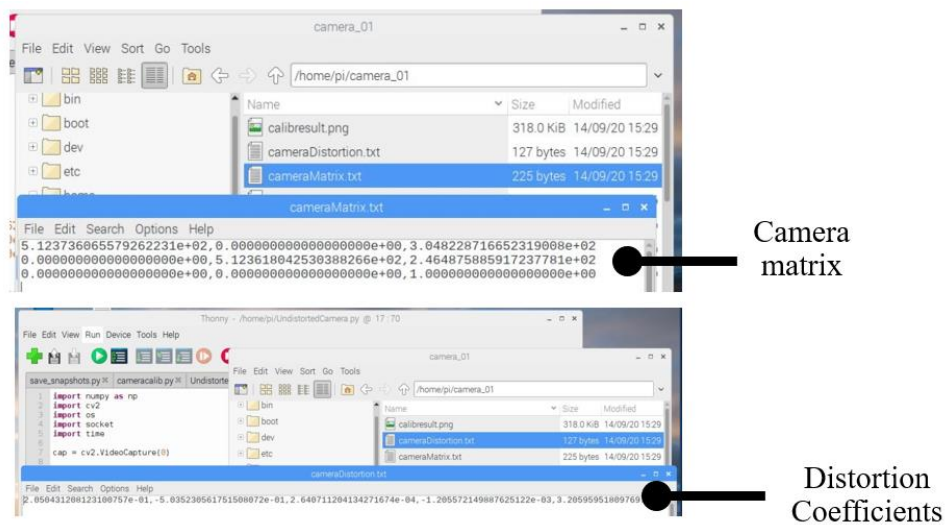


Figure 6.33 The example of calibration results saved in the Raspberry Pi.

The program was then modified for calculating a marker's centre on an image when the camera had been calibrated by adding the distortion coefficients to undistort the image using the undistort command.

### 6.6.2. A marker position calculation for Experiment 3

Experiment 3 extended the right triangle similarity theorem and trigonometry concepts described in Experiment 2 to be suitable for 3 cameras. Figure 6.34 illustrates the geometry for this experiment. Three camera pairs were used to calculate a marker position: cameras 1 paired with 2, 1 paired with 3, and 2 paired with 3. Marker calculation in this experiment referred to Experiment 2's calculation technique. However, some variables were changed, such as the degrees of camera 2 (zero degree) and  $C_x$ ,  $C_y$ ,  $f_x$ , and  $f_y$  (using the camera calibration results).

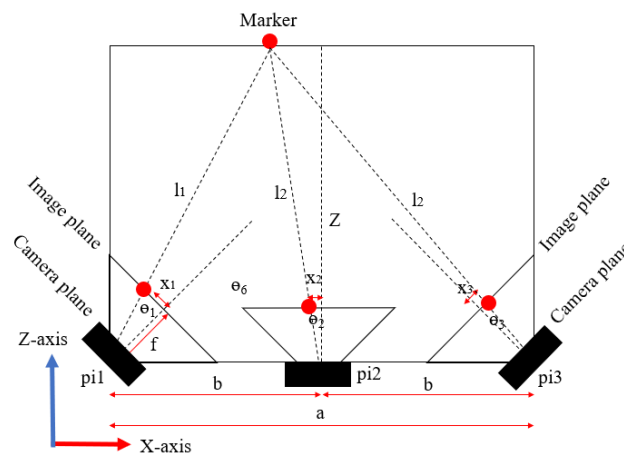


Figure 6.34 The geometry used to calculate the marker position of Experiment 3 in the top view.

### 6.6.3. Experimental setting for Experiment 3

We built a frame using aluminium profiled bar. The frame dimensions were 1,000 mm in length, width, and height. Connectors, bolts, and nuts were used to assemble the camera frame. To get the rectangular shape, each profile was perpendicular to each other, which would help improve the camera position. Figure 6.35 illustrates the camera frame used in Experiment 3. To position the cameras, we added a metal bar on one side of the camera frame, and the height of this bar was similar to the distance between marker 13 and the floor. Cameras 1 and 3 were positioned on the bar's left and right corners with  $45^\circ$  horizontal rotation, respectively, while camera 2 was

positioned in the middle between cameras 1 and 3 with no rotation. All cameras were facing toward the markers.

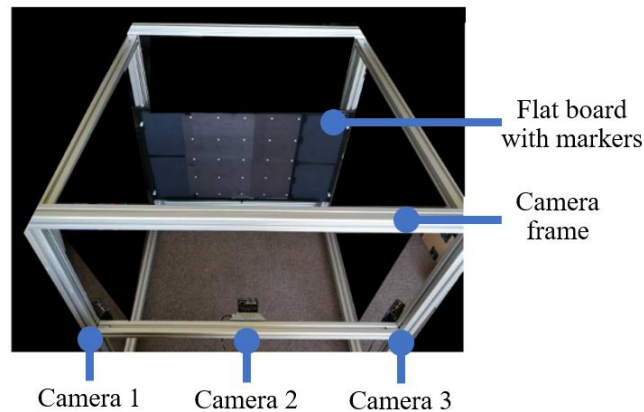


Figure 6.35 The camera frame and volume experiment in Experiment 3.

Two data collections were done in Experiment 3. The first data collection was a Volume experiment (fig 6.35) and was similar to Experiment 2. A flatboard was attached to the camera frame by connectors, bolts, and nuts. It could be shifted along the Z-axis, remaining perpendicular to the floor and maintaining the Z-value of the markers located at the same depth. Twenty-five markers were attached to the flatboard, and the board was shifted in 5 depths along the Z-axis. The marker size and shape and the distance between markers were also similar to Experiment 2. The second data collection in Experiment 3 was a surface experiment (fig 6.36). The spherical shape of a yoga ball was chosen for this experiment. This ball was 650 mm in diameter. We placed numerous markers on the surface of the ball.

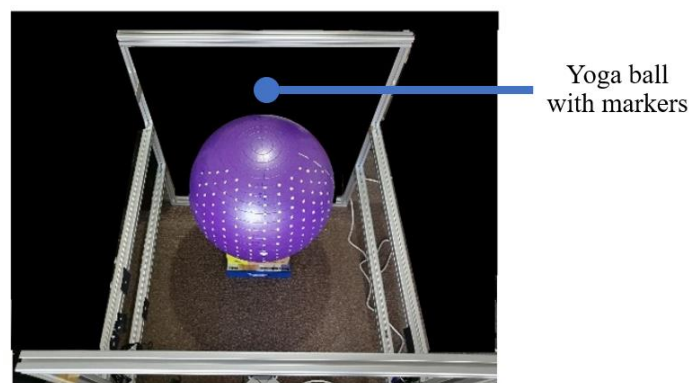


Figure 6.36 The surface experiment in Experiment 3.

#### 6.6.4. The error calculation for Experiment 3

The previous section described a method for camera calibration, a concept of marker calculation using 3 Raspberry Pi cameras, and Experiment 3's data collection. Then, this section describes the experiment's accuracy.

The reference markers were similar to Experiment 2, which used marker 13 as a starting point. Then, in X, Y, and Z, we added 100 mm to calculate other reference points. The MLR formula was then calculated using values from both groups, references and systems. To calculate a new marker position, the original values were re-calculated with a regression formula. The marker position of both experiments and the references were used to calculate the error using the RMSE.

In the volume experiment, Table 6.7 reports the errors of X, Y, Z and 3D Euclidean error before and after MLR. Overall, all errors were reduced considerably. The X errors before and after regression were 24.40 mm and 7.27 mm. The Y errors before and after regression were 14.39 mm and 11.22 mm. The Z errors before and after regression were 22.38 mm and 9.64 mm. The 3D Euclidean errors before and after regression were 22.24 mm and 10.83 mm.

Table 6.7 RMSE of the volume experiment in Experiment 3 compared before and after MLR.

	X (n=87)	Y (n=87)	Z (n=87)	3D Euclidean error
Before regression	24.40	14.39	22.38	22.24
After regression	7.27	11.22	9.64	10.83

Note: MLR = Multiple linear regression.

In the surface experiment, Table 6.8 reports the errors of X, Y, Z and 3D Euclidean error before and after MLR. Overall, all errors were reduced considerably. The X errors before and after regression were 10.87 mm and 2.33 mm. The Y errors before and after regression were 9.15 mm and 5.19 mm. The Z errors before and after regression were 12.26 mm and 2.51 mm. The 3D Euclidean error before and after regression were 6.51 mm and 4.99 mm.

Table 6.8 The RMSE of the surface experiment in Experiment 3 compared before and after MLR

	X (n=116)	Y (n=116)	Z (n=116)	3D Euclidean error
Before Regression	10.87	9.15	12.26	6.51
After regression	2.33	5.19	2.51	4.99

Note: MLR = Multiple linear regression.

### 6.6.5. Discussion and conclusion of Experiment 3

Experiment 3 described the concept to develop a low-cost postural measurement system using 3 Raspberry Pis and cameras. This experiment was still based on the right triangle similarity theorem and trigonometry, similar to Experiments 1 and 2. However, the cameras' positions were adjusted. Cameras 1 and 3 were 45° horizontally rotated, while camera 2 was not rotated. The formulas for calculating X, Y, and Z values were slightly modified. Camera calibration was performed on the programming side to undistort the image and to calculate a camera matrix that included the essential variables ( $C_x$ ,  $C_y$ ,  $f_x$ , and  $f_y$ ).

The camera frame was built in this experiment to position the cameras more precisely and attach the flatboard more firmly. Two data collections were performed in this experiment, including the volume and surface experiments. Similar to Experiment 2, MLR was used to reduce error in this experiment. The system's accuracy was also evaluated using the RMSE.

When Experiments 3 and 2 were compared in the volume experiment before regression, the X error in Experiment 3 (22.40 mm) was considerably higher than in Experiment 2 (7.86 mm). The Y error in Experiment 3 (14.39 mm) was considerably lower than in Experiment 2 (33.25 mm). The Z error in Experiment 3 (22.38 mm) was slightly higher than in Experiment 2 (18.20 mm). After regression, the X error in Experiment 3 (7.27 mm) was slightly higher than in Experiment 2 (5.57 mm). The Y error in Experiment 3 (11.22 mm) was lower than in Experiment 2 (15.13 mm). The Z error in Experiment 3 (9.64 mm) was slightly higher than in Experiment 2 (7.69 mm). As a result, three Raspberry Pis and cameras in Experiment 3 could improve the error of Y compared to the use of 2 Raspberry Pis and cameras in Experiment 2. In contrast, the X error was increased, and the Z error was similar.

Comparing the volume and surface experiments before regression, the surface experiment had smaller errors than the volume experiment. The error in X narrowed from 24.40 mm to 10.87 mm. The error in Y declined from 14.39 mm to 9.15 mm. The error in Z lowered from 22.38 mm to 12.26 mm. After regression, the surface experiment had lower errors than the volume experiment. The error in X lowered from 7.27 mm to 2.33 mm. The error in Y lowered from 11.22 mm to 5.19 mm. The error in Z lowered from 9.64 mm to 2.51 mm. As a result, the error in the surface experiment was considerably lower than in the volume experiment. The system presented better accuracy when calculating markers on the surface (the shorter distance between markers) than on volume (the longer distance between markers).

In conclusion, Experiment 3 showed that increasing the number of cameras could not significantly decrease the error. Interestingly, the errors considerably went down after re-calculating the new marker position with MLR. The error in the volume experiment after regression presented within an acceptable (10 mm) value, and the error in the surface experiment was within our goal (5 mm). Although we increased the number of cameras to 3 and positioned them on the metal frame more precisely, the errors remained relatively high.

After careful consideration, the possible cause of the error might come from the camera position and orientation. In the calculation, we entered only a 45-degree horizontal rotation to cameras 1 and 3 and assumed no rotation of the cameras in other axes. However, a slight change in camera position and orientation may result in a significant increase in the errors. Therefore, the subsequent development of a low-cost postural measurement system should focus on how to know the exact position and orientation of cameras and a method to improve the accuracy of the system.

## **6.7. Discussion and conclusion of Experiments 1, 2, and 3**

This chapter described the possibility of developing a low-cost postural measurement system using 2 or 3 Raspberry Pis and cameras to calculate a marker position in 3-dimensional space. We used the image processing technique to calculate the centre of a marker on an image. We used the right triangle similarity theorem and

trigonometry to calculate a marker position in 3-dimensional space. The Raspberry Pis and cameras, and the computer could send and receive data wirelessly via Wi-Fi. Three experiments were carried out in this development, and the RMSE was calculated to assess the system's accuracy.

The development began with 2 Raspberry Pis and cameras. The cameras were attached to the bar, facing toward the markers and no tilting (Experiment 1). We collected 125 marker positions; 25 markers were attached to the flat board at the same depth (distance between cameras and markers), and the data was collected in 5 depths. After the program calculated the centre of the marker on the image, the values were entered into the pre-calculated Excel spreadsheet to calculate the marker positions.  $C_x$ ,  $C_y$ , and focal length were all estimated in this experiment. Overall, the X and Y errors were within our acceptable (10 mm) range. However, the Z error was extremely high, which increased the 3D Euclidean error. After determining the possible cause of the error, this experiment recommended the following experiment to try positioning the cameras in a different position and re-evaluating the error.

Experiment 2 improved the system based on the recommendation from Experiment 1. Two Raspberry Pis and cameras were continued to be used in this experiment. The cameras were attached to the bar, facing toward the markers with  $45^\circ$  horizontally rotating. Experiment 2 had the same number of testing markers and marker positions as Experiment 1. Furthermore, three regressions (SLR, SLR with multiple conditions, and MLR) were used in this experiment to recalculate the new marker positions and to improve the system's accuracy. Before regression, the Z error highly improved, and the X error was similar. In contrast, the Y error extremely increased. Regressions helped to decrease the errors considerably. SLR with multiple conditions resulted in the lowest error, followed by MLR and SLR. However, SLR with multiple conditions was more complex to recalculate the new marker positions than other regressions. MLR was simpler and recommended for use in the following development. After determining the possible cause of the error, the following experiment had to improve the image quality or undistort the image before further calculation. The following experiment required the development of calculating the values of  $C_x$ ,  $C_y$ , and focal length of cameras. The following experiment had to increase the number of cameras and position them more precisely on the tripod or metal frame.

Experiment 3 improved the system based on Experiment 2's recommendations. In this experiment, three Raspberry Pis and cameras were used. The camera frame was constructed, and cameras were attached to a metal bar on one side of the frame. Cameras 1 and 3 were positioned on the left and right corners of the bar with a 45-degree horizontal rotation, respectively, while camera 2 was positioned in the middle between cameras 1 and 3 with no rotation. This experiment included 2 data collection methods (volume and surface experiments). The new marker positions were recalculated using MLR. In the volume experiment before regression, the Y error was improved or lower than in Experiment 2. The Z error, on the other hand, was slightly increased, while the X error was highly increased. After regressions, the errors were highly decreased and were within our acceptable (10 mm) range. In the surface experiment before regression, the errors were lower than in the volume experiment. However, the errors were still high, especially the Z error. After regressions, the errors were highly decreased and were within our goal (5 mm). Additionally, the Y error in the volume experiment before regression was better or lower than in Experiment 2. The Z error was slightly increased, whereas the X error was greatly increased. The errors were greatly decreased after the regression and were within our acceptable range. Before the regression, the errors were lower in the surface experiment than in the volume experiment. The errors, however, remained high, particularly the Z error. The errors were greatly decreased after the regressions and were well within our goal (5 mm).

From Experiment 1 to Experiment 3, there were several improvements and learnings. We learned how to calculate the marker's centre on an image, how to calculate a marker position in 3-dimensional space, how to reduce an error using regressions, how to communicate between Raspberry Pi and computer wirelessly via Wi-Fi, how to perform camera calibration, and how to assess the accuracy of the system. The developed system showed a high possibility of calculating marker positions in 3-dimensional space and using it in clinical practice. Although, the errors after regression were highly decreased and ranged between 5 and 10 mm. However, the errors before regression were significantly high, which might refer to inappropriate camera settings or insufficient programming capability.



After realizing the possible cause of the error might come from the camera position and orientation. In the calculation, we entered only a 45-degree horizontal rotation to cameras 1 and 3 and assumed no rotation of the cameras in other axes. However, a slight change in camera position and orientation may result in a significant increase in errors. Therefore, the subsequent development of a low-cost postural measurement system should focus on how to know the exact position and orientation of cameras and a method to improve the accuracy of the system. The camera's quality could also influence the error. We used the Raspberry Pi camera in this study because it comes as a set with a specific port, and the image quality was appropriate for this application. However, a higher-quality camera may be another option to consider in order to improve image quality in the following development. Finally, the errors could be caused by using incorrect reference values to compare with the system values. In these experiments, we used a middle marker (marker 13) as a starting point and added the distances to estimate the positions of the surrounding markers. The calculated reference may not represent the exact position of reference markers. The following development should have valid reference markers to compare the system, or the system should be compared to a high-cost motion capture system.

In this chapter, we showed the practicality and potential of a Pi camera system to calculate marker positions, but the errors remained too high. We used elements of the OpenCV computer vision programming library in our pi-based programs to calibrate the cameras with chessboards, threshold images, etc. This technique had helped reduce the errors, but simple trigonometry and camera calibration had not proved sufficiently accurate for our application. We determined to see if other elements of the OpenCV package could help reduce the errors further.

**7. Chapter 7 Development of A Low-Cost Postural Measurement System based on OpenCV and Stereo Camera Calibration methods**

## 7.1. Introduction

The previous chapter described the implementation of a new low-cost postural measurement system using 2 and 3 Raspberry Pis with cameras and a developed program to calculate marker positions in 3 dimensions using simple trigonometry concepts. The previous chapter's concept of calculating marker positions was based on the geometry of the right triangle similarity theorem. To assess the system's accuracy, the calculated results were compared to the reference values. Simple and multiple linear regressions were used to reduce systematic errors, and the errors were reduced from more than 10 mm to less than 10 mm. However, the errors remained greater than 5 mm. After careful consideration, the primary inaccuracy issue were likely to come from the location of the cameras and the non-linearity of the camera lens. A minor change in camera position and orientation may significantly impact the system's error, as would distortion of the camera lens during manufacture. As a result, the next step should undistort the camera lens and then determine the best camera position and orientation before calculating marker positions.

OpenCV is an open-source computer vision software package widely used in robotics and other engineering applications where stereo vision is required. It was decided to determine if OpenCV methods could further increase the system's accuracy to meet the clinical requirements. Chapter 7 describes a new low-cost postural measurement system based on OpenCV and the stereo camera concept. This system undistorts two cameras separately. It then calibrates them as a pair of cameras, called stereo camera calibration, and calculates marker positions in 3 dimensions using the combined data of both cameras.

The first section of this chapter discusses how to perform stereo camera calibration using a pair of cameras. As a result of the process, these cameras know their position and orientation in relation to one another, and this must be maintained; otherwise, the pair need to be recalibrated. The following program calculates marker positions in 3-dimensional space and compares the results to a reference. The chapter then describes how to use multiple Raspberry Pis with cameras to estimate marker positions in a 3D volume and reassess the system's accuracy.

## 7.2. General concept of stereo camera to calculate marker positions

In general, vision is the process of seeing an object in space using human eyes. There are 3 components involved in this process (Kaehler and Bradski, 2016):

1. The imagers or lenses of eye
2. A light source (e.g., a light bulb or the sun)
3. An object

To begin this process, the ray of light emitted from the light source passes through space and hits the object's surface. Some light will be absorbed by the object's surface, but some will not and will reflect in our eyes, and the image is created at the retina. It can simply explain this process in Computer vision. Instead of using our eye as an image producer, the camera can be used to create the image. Computer vision is widely used in many applications, such as medical imaging, security, robotics, and human motion analysis (Kaehler and Bradski, 2016).

A simple computer vision model to create an image is a "Pinhole camera model", as shown in Figure 7.1 (Kaehler and Bradski, 2016). The first part, the pinhole plane, is a plane of the camera with a small hole located about the centre of that plane, and the ray of light passes through it. The second part is the image plane, which illustrates the image of an object and originally presents it upside down. Lastly is the object in space. The reflecting light from the object travels through a small hole of the pinhole plane and hits the image plane to create the image. Additionally, the optical axis is the axis from the centre of the image plane to the centre of the pinhole plane, and the focal length ( $f$ ) is the distance between these two planes. " $Z$ " is the distance from the pinhole plane to the object in 3-dimensional space parallel to the optical axis. " $X$ " is the distance from the optical axis to the object in 3-dimensional space (in the vertical direction), and " $x$ " is the distance in the computer unit from the centre of the image plane to the image created on the screen. Regarding the similarity of the right triangle theory, we can use equation 7.1 to calculate the unknown value.

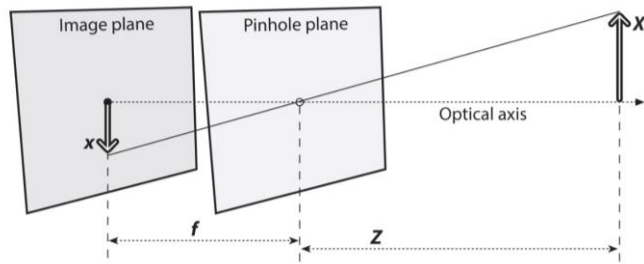


Figure 7.1 The pinhole camera model to create an image.

$$-x = f \times \frac{X}{Z} \quad (7.1)$$

In practice, we can switch between the image plane and pinhole plane to re-upside down the image for simpler calculation and illustration, and the calculation result is still equivalent. Figure 7.2 illustrates the switching between the image plane and the pinhole plane (Kaehler and Bradski, 2016). The centre point at the pinhole plane is reinterpreted as the centre of projection or the camera's origin. The point at the intersection between the image plane and the optical axis is called the principal point,  $C_x$  and  $C_y$ . Equation 7.2 illustrates the formula to calculate the unknown value. For the idealized pinhole camera model, the principal point is equivalent to the centre of the image plane. Unfortunately, this does not perfectly happen on every camera due to the error in the manufacturing process of the camera lens. Therefore, an individual camera calibration is needed to re-adjust the image and calculate the exact value of the principal point ( $C_x$  and  $C_y$ ) and the focal length ( $f$ ).

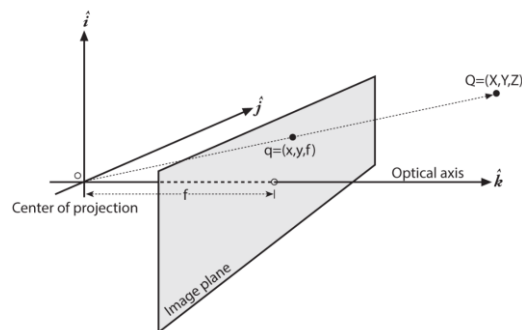


Figure 7.2 The pinhole camera model after switching between the image plane and pinhole plane.

$$x = f \times \frac{X}{Z} \quad (7.2)$$

One pinhole camera can only calculate an object in two dimensions. As a result, at least two cameras are required to calculate an object in 3 dimensions, called the stereo camera concept. The simplest way to explain the "**Stereo camera concept**" is to use both eyes to see an object in space and determine its depth. Like computer vision, it performs this task by calculating the correspondence between points or objects seen by both cameras (Kaehler and Bradski, 2016). The stereo camera concept is a process of comparing 2 images looking at the same scene and calculating the object's position in 3-dimensional space.

OpenCV is a computer vision library with an open-source or free-source license. It can be accessed from its official website (<http://opencv.org>). More than 500 OpenCV library functions can be used in many applications, such as factory product inspection, medical imaging, security, robotics, camera calibration, and stereo vision. In our postural measurement system, we integrate the OpenCV concept into the project, use some library features, and further develop the program to match our application.

This system developed 2 components. The hardware consisted of a computer (Windows 10), Raspberry Pis with cameras, a chessboard or calibrated pattern (Zhang *et al.*, 1999, Sturm and Maybank, 1999, Zhang, 2000), a Phidget board, a Phidget switch, camera tripod, camera frame, a flatboard, imitated scoliosis model, and markers. For the software, all programs running on the computer were written in C++ programming language on the Visual Studio 2019 software and using the OpenCV library (see appendix for full program, Appendix 7). All associated programs running on Raspberry Pis were written in Python language (see appendix for full program, Appendix 7). All Raspberry Pis were wirelessly connected to the computer via a WIFI router. There were 4 processes involved in calculating marker positions for this development.

1. Camera calibration
2. Stereo camera calibration and image rectification
3. Correspondence
4. Reprojection

## 1. Camera calibration

Camera calibration is a process of determining the intrinsic and extrinsic parameters of a camera in order to rectify distortions and provide precise measurements in computer vision applications. The concept and programming for camera calibration for this development were similar to those described in Chapter 6. However, while the previous chapter calibrated the camera using the Raspberry Pis, the new program performed the calibration on the computer. Therefore, Raspberry Pis with cameras were used for recording the images, and the recorded images were sent and saved on the computer.

The program called "getpicameraimage" was run on the computer to open the Raspberry Pi cameras and take the images. The program then requested the user to specify the camera number from 1 to 8. The images could be taken independently and did not need to be in order of camera number. A Phidget switch was used to detect the input voltage change by clicking on the button. The user had to click on this switch, one-click for one image taken. All images were recorded and kept in the folder in C-drive. The user could exit the program by pressing the "Esc" button on the keyboard. Figure 7.3 illustrates the recorded images of the chessboard saved in C-drive for camera calibration.

Moving on to the next step, the "calibpicamera" program was run on the computer to calibrate the camera. The program then requested the user specify the camera number, and this process was performed camera-by-camera, similar to the previous step. After setting the camera number, the program reviewed the images in the folder in C-drive, following the list of images in the "list.txt" file format (fig. 7.3). Then, the program calibrated the camera. The image processing method and the intrinsic and extrinsic parameters calculation were similar to the information described in Chapter 6. The camera calibration results were saved in ".xml" file format in C-drive for use in the following process (fig. 7.13).

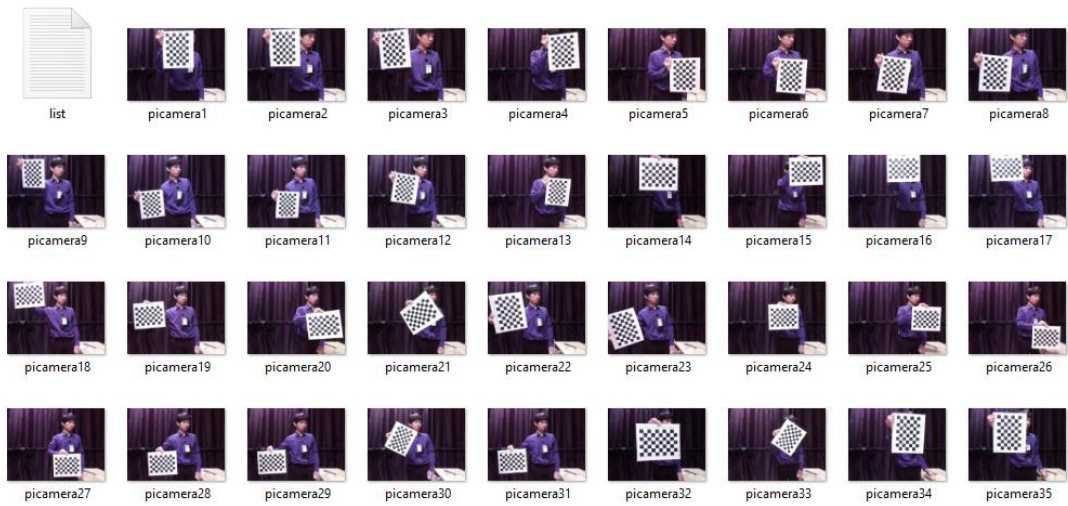


Figure 7.3 Recorded images for camera calibration using the “getpicameraimage” program.

Figure 7.4 illustrates the image of the chessboard taken by the camera (Medium, 2023). Figure 7.4a shows the original image before applying the intrinsic and extrinsic camera parameters. The chessboard's edge appears curved but should be straight for the actual chessboard. Furthermore, the intrinsic and extrinsic camera parameters obtained from camera calibration are applied to the image, and the result of the undistorted image is shown in Figure 7.4b. The edge of the chessboard appears straight, the chessboard image appears the exact size and shape it should be, and this image is ready for further image processing in the following step.

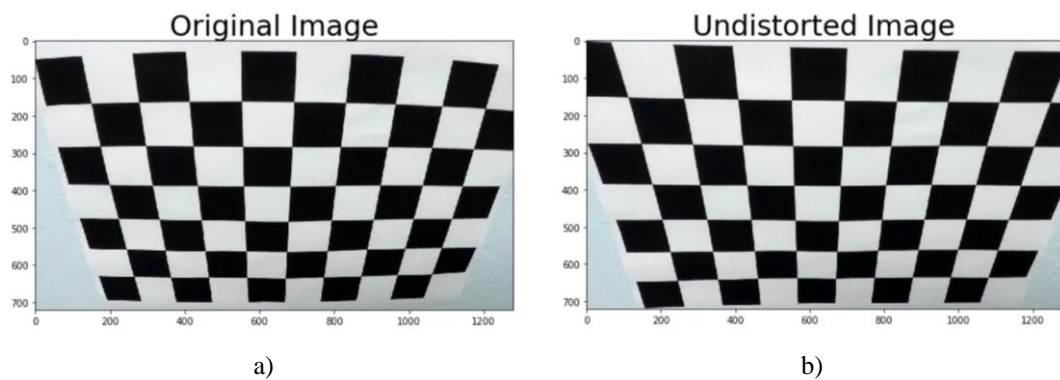


Figure 7.4 Example of image, a) the original image and b) the undistorted image.



## 2. Stereo camera calibration and image rectification

Stereo camera calibration and image rectification are the processes to rectify and row-align the image plane between the left and right cameras. As a result, two image planes are coplanar and horizontally aligned, or the corresponding image rows on the two image planes are collinear relative to each other. This process also calculates the position and orientation of the camera pair relative to one another. Figure 7.11 illustrates the stereo vision concept after the stereo camera calibration and image rectification process finishes.

To perform this process, the program called "getpistereimages" was run on the computer to open both cameras and take pictures simultaneously. The program requested the user specify the camera number, similar to the previous program. The first camera number was the left camera, and the second was the right camera, respectively. The process of taking chessboard images with various positions and orientations was also similar to the previous process. More than 30 pairs of images were recommended for stereo camera calibration, and all images were recorded and kept in the folder in C-drive (fig 7.5).

! Two images were taken by left and right cameras simultaneously.

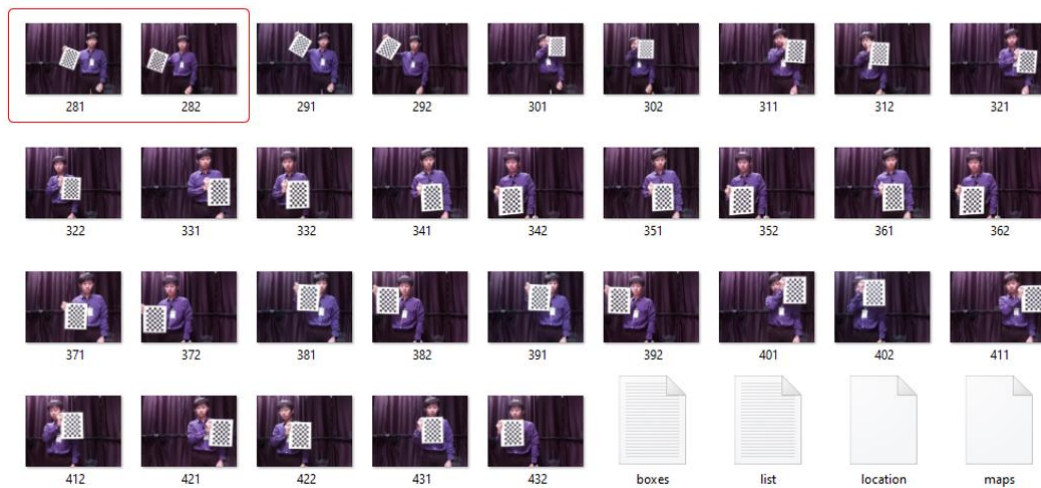


Figure 7.5 Recorded images for stereo camera calibration using the "getpistereimages" program.

The program called "calibpistereocamera" was run on the computer to perform stereo camera calibration. The program requested the user specify the camera number. The first camera number was the left camera, and the second was the right camera, respectively. After setting the camera number, the program reviewed the images in the folder and the list of images in the "list.txt" file format. The camera matrix and distortion coefficient from the individual camera calibration were applied to undistort images in the stereo camera image folder. Then, the undistorted images were converted from colour to grayscale images. Next, the program checked the chessboard pattern on each image. The pattern and size of the chessboard had to be the same as the camera calibration process. If the images matched the pattern, the program found and drew the corner of the chessboard pattern. The program then created and stored vectors of 2D and 3D points from the selected images. After that, the program performed stereo camera calibration on these images. The results from stereo camera calibration were the essential matrix (E) and the fundamental matrix (F). The essential matrix consisted of translation vectors and the rotation matrix that related two cameras in physical space. The fundamental matrix contained the same information as both cameras' essential matrix and intrinsic parameters, which helped to relate 2 cameras in pixel coordinates. The result of the stereo camera calibration was saved in "location.xml" file format and kept on the C-drive (fig. 7.5). Furthermore, the left camera became the origin point for the stereo camera concept, with the x-axis going from the left to the right camera, the y-axis downward from the left camera, and the z-axis moving forward from the left camera (fig. 7.11).

### **3. Correspondence**

Correspondence is the process of finding the same features or points between the left and right images. The output of this step is a "Disparity map", where the disparity is the difference in distance in x-coordinates on the image planes of the same feature viewed from the left and right cameras. The result of the disparity map was saved in "maps.xml" file format and kept on the C-drive (fig. 7.5) for calculating marker positions in the following process.

## 4. Reprojection

Once stereo calibration has been completed a process called Reprojection could be used for calculating the depth of an object or markers in 3-dimensional space using the disparity map from the previous process and a triangulation calculation between marker and cameras. For measurement to be conducted, a pair of images of the object were taken to be quantified again using the "getimagepairs" program. The process for taking the images was similar to the previous step using the Phidget switch, and all images were recorded and kept in a specific folder in C-drive. Figure 7.6 illustrates the recorded images for marker calculation using the "getimagepairs" program.

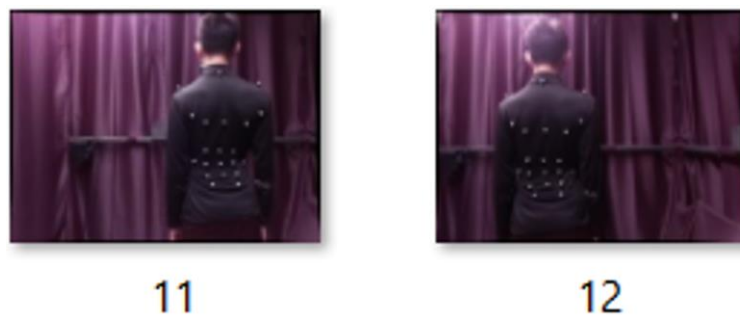


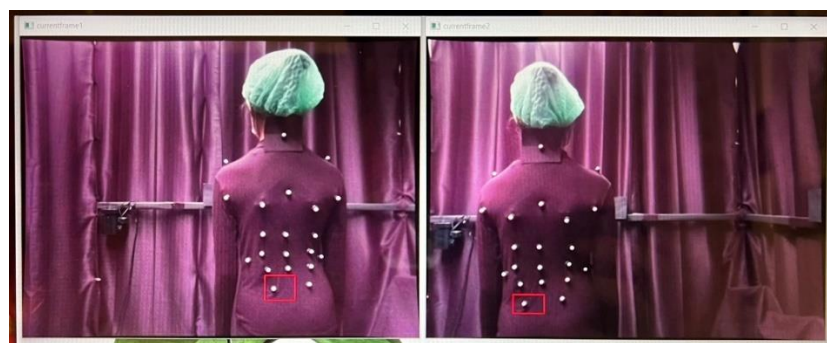
Figure 7.6 Recorded images for reprojection process using the "getimagepairs" program.

Finally, another program called "reconstructfromimages" was run on the computer to perform the reprojection on the frames and calculate the marker positions in 3-dimensional space for the pair of images. The result of this process was an XYZ value for each marker. The program requested the user specify the camera number. The first camera number was the left camera, and the second was the right camera, respectively (figure 7.7).

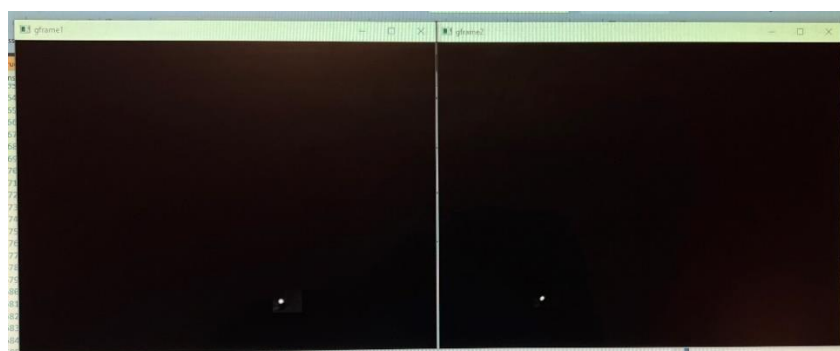
```
C:\tumprogs\x64\Debug\reconstructfromimages.exe
enter the first camera number
1
enter the second camera number
2
Done - getting location and rotation data
reading maps file
```

Figure 7.7 User specify the camera number for running the "reconstructfromimages" program

After setting the camera number, the program reviewed the pairs of images in the folder (fig. 7.6). The images were converted from colour to grayscale images before thresholding. Figure 7.8a shows the undistorted image of a scoliosis patient with markers attached to anatomical landmarks obtained from cameras 1 and 2. Figure 7.8b shows the images after converting from colour to grayscale images. The current program allowed the user to select an individual marker to calculate the position by adjusting the selected area, as seen in Figure 7.8a. A small red box shows the cropped area where the user could select a specific area inside the box to calculate a marker and exclude others outside the box, which could be manually adjusted by pressing the function on the keyboard. If the user wants to increase or decrease the left edge of the box, they have to press the keyboard at F1 or F2. If the user wants to adjust the right edge of the box, they have to press F3 or F4. If the user wants to adjust the top edge of the box, they have to press F5 or F6. If the user wants to adjust the bottom edge of the box, they have to press F7 or F8, respectively.



a)



b)

Figure 7.8 Image of a scoliosis patient with markers attached to anatomical landmarks obtained from cameras 1 and 2, a) undistorted image and b) converted image from colour to grayscale.

After that, the program found the object or marker that appeared inside the selected area. The user could manually adjust the image threshold in the program from 0 to 255 to decrease or increase the contrast between the background and the detected marker area. Figure 7.9 illustrates the images with an appropriate thresholding.

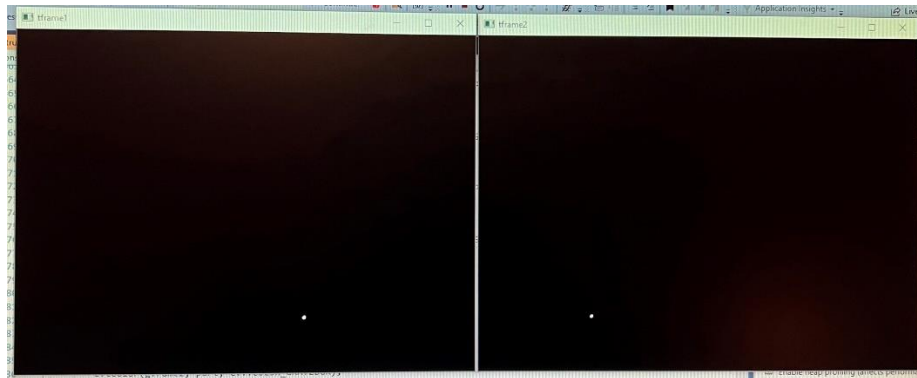


Figure 7.9 Thresholding the images

The program found and drew the contour of the marker (fig. 7.10). The program then calculated the geometric centre of that marker and gave x and y values from each image.

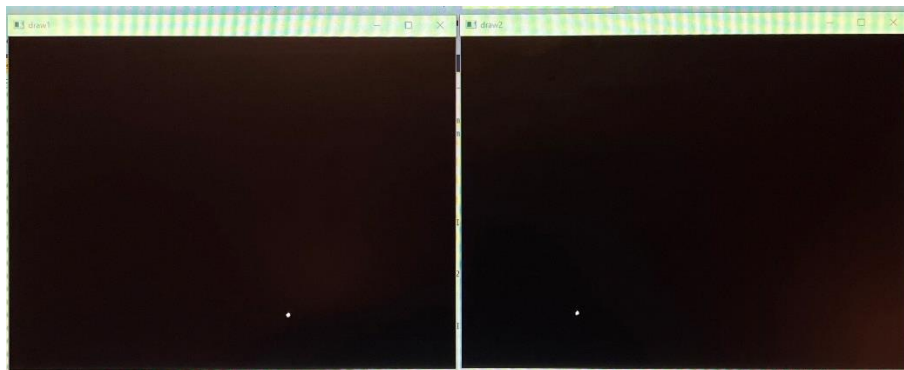


Figure 7.10 Find and draw contour of marker in images

Finally, the program re-loaded the disparity map for the pair of cameras from the previous process and calculated a marker position in a 3-dimensional space based on triangulation theory. The marker position, XYZ value, was saved in "reconstructdatafromimages.txt" file format and kept on the C-drive. Figure 7.11 summarises the overall concept of stereo vision, including camera calibration, stereo

camera calibration and image rectification, correspondence, and reprojection, to calculate a marker position in a 3-dimensional space (Kaehler and Bradski, 2016).

In summary, Figure 7.11 illustrates the overall concept of stereo vision after completing all processes, including camera calibration, stereo camera calibration, image rectification, correspondence, and reprojection. The left and right image planes are coplanar and horizontally aligned, the left camera becomes the origin point, with the x-axis going from the left to the right camera, the y-axis going downward from the left camera, and the z-axis moving forward from the left camera. When a marker is present in front of the two image planes, the program calculates the x and y values from each image, reloads the mapping file, and calculates the XYZ value based on trigonometry theory.

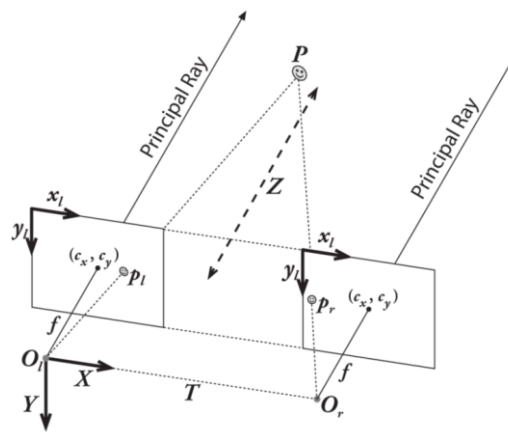


Figure 7.11 Stereo vision concept after completing all processes.

In addition, Figure 7.12 illustrates all programs used to calibrate single and stereo cameras and calculate marker positions. Figure 7.13 illustrates all folders on the C-drive used to store the image for camera calibration, stereo camera calibration, and the camera calibration results (distortion coefficient and camera matrix). The computer's C-drive contained the programs, images, and calibration results.

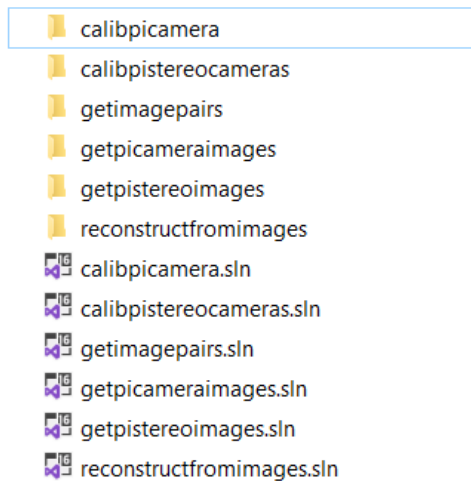


Figure 7.12 All programs for calibrating the cameras and calculating marker positions. The programs were kept on the C-drive of the computer.



Figure 7.13 Folders on C-drive that contain image folders for camera calibration, the image folders for stereo camera calibration, and the camera calibration result.

### 7.3. Experiment 4: A low-cost postural measurement system based on stereo camera concept

The previous section described the stereo camera concept to calculate marker positions in 3-dimensional space and introduced the developed software used to perform camera calibration, stereo camera calibration, and marker calculation. The

following section describes a series of experiments to calculate marker positions using the stereo camera concept. The experiment begins by comparing camera calibration results when the chessboard images are in various conditions. Then, the following experiments assess the error when using single and multiple stereo camera pairs to calculate marker positions. Lastly, the experiments also assess the errors when comparing the result to high-cost motion and 3D scanning systems.

### **7.3.1. Experiment 4.1: To compare the camera calibration results with different image conditions**

One factor that can cause the error shown in the previous chapters experiments was the inaccurate values of focal length ( $f$ ) and principle points ( $C_x$  and  $C_y$ ). The previous experiment took the chessboard images in various positions and orientations with more than 30 images. However, no information was found in the literature that explains the effects on accuracy of chessboard image positions and orientations for camera calibration and how many images should be used. As a result, the following experiment was performed to compare the camera calibration results if the chessboard images' positions and orientations change in various conditions.

At the beginning of this development, the cameras were attached to the same camera frame as those used in the previous experiment. All cameras were attached to the servo motors, which could rotate the camera from leftward to rightward and upward and downward. Moreover, the images could be taken every  $5^\circ$  or  $10^\circ$  of rotation, depending on the program setting. The cameras were attached to the Raspberry Pis with a long cable, and all Raspberry Pis were kept in the same case. The chessboard pattern was attached to each side of the box, and the box was positioned at the centre of the camera frame. Figure 7.14 illustrates all cameras on the frame and the chessboard pattern attached to the box. The "getpicameraimages" program was run to take dynamic and static images of the chessboard. Dynamic images are those taken while the servo motors are rotating. In contrast, static images are the images of chessboard taken while the servo motors are not rotating.



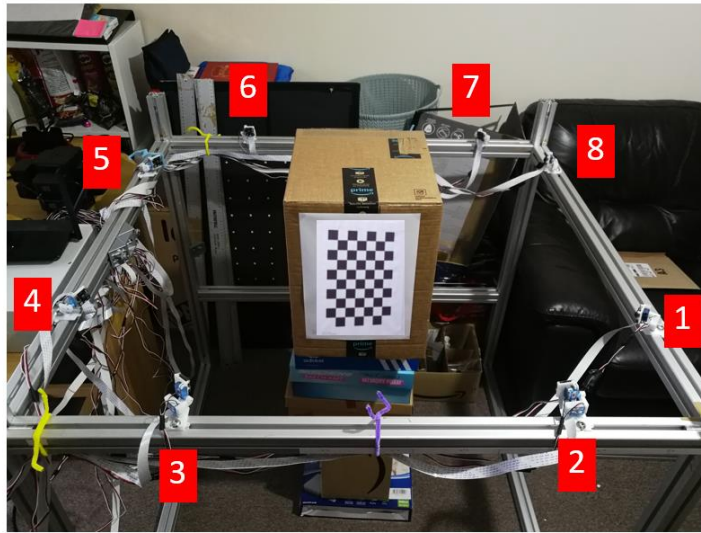


Figure 7.14 Cameras setting on the frame with the chessboard in the middle of camera frame.

The previous articles did not recommend how to obtain a good image for the camera calibration process. To prove this issue, we divided the images into 2 conditions based on the chessboard position and orientation and the number of static images.

- **Condition 1:** Take images by servo motor. It consisted of 50 dynamic images and different amounts of static images, ranging from 0, 10, 20, 30, 40, and 50 images.
- **Condition 2:** Take images by user. It consisted of 50 dynamic images and between 0 and 50 static images.

Table 7.1 reports the camera calibration result of Raspberry Pi with camera 1, including the focal length in X ( $f_x$ ) and Y ( $f_y$ ), the principal point in X ( $C_x$ ) and Y ( $C_y$ ). All conditions were compared to the result from the previous method, which had previously been run on the Raspberry Pi to calibrate the camera with 30 images. The images were taken by the user changing the chessboard in various positions and orientations.

In condition 1, the  $f_x$ ,  $f_y$ , and  $C_y$  gradually increased, and  $C_x$  gradually decreased from no static image to 50 static images. All values were highly different compared to the result from the previous method. In condition 2, the  $f_x$ ,  $f_y$ ,  $C_x$ , and  $C_y$  had a small difference between no static and 50 static images, and all values were similar to the result from the previous method.

This experiment showed that the position and orientation of the chessboard did affect the results of the camera matrix (focal length and principle point), which could also affect the marker calculation in the following process. When comparing condition 1 with the result from the previous method, there was a high difference between both comparisons. A difference when increasing static images and compared to the previous method might come from the accumulated errors when increasing the static images. In contrast, condition 2 with no static images and condition 2 with 50 static images showed a small difference compared to the previous method.

Although the images taken by the servo motor were dynamic, the position and orientation of the chessboard still did not vary enough compared to the images taken by the user, causing a big difference in results. On the other hand, the images taken by the user were more dynamic and varied in positions and orientations of the chessboard. Whether static images were included or not would not affect the calibration result. As a result, we recommend that users take images of the chessboard in various positions and orientations, and more than 30 images were suggested. We abandoned the idea of automated calibration using motorized cameras as inaccurate.

Table 7.1 The camera calibration results of camera 1 with different image conditions. Condition 1 is when the images are taken by servo motors, and condition 2 is by the user.

Conditions	Focal length in X ( $f_x$ )	Focal length in Y ( $f_y$ )	Principle point in X ( $C_x$ )	Principle point in Y ( $C_y$ )
Result from previous method with 30 image	501.50	501.24	325.75	245.45
Condition 1 plus no static image	548.33	541.05	297.48	272.04
Condition 1 plus 10 static images	552.31	543.51	293.37	274.61
Condition 1 plus 20 static images	558.85	548.01	288.68	277.48
Condition 1 plus 30 static images	564.53	551.86	284.78	279.84
Condition 1 plus 40 static images	570.44	555.89	281.20	281.63
Condition 1 plus 50 static images	576.89	560.39	277.93	283.04
Condition 2 plus no static image	502.29	501.75	325.07	245.37
Condition 2 plus 50 static images	502.85	502.40	325.69	243.32

Note:  $f_x$  = Focal length in X,  $f_y$  = Focal length in Y,  $C_x$  = Principle point in X,  $C_y$  = Principle point in Y.

### 7.3.2. Experiment 4.2: To calculate marker positions using a stereo camera pair

The previous section described the concept of stereo cameras, the programs used to calibrate single and stereo cameras, calculate marker position, and a method for taking the best images for camera calibration. This section further describes the method used the stereo camera concept to calculate markers in 3-dimensional space.

This development begins with marker calculation using 2 Raspberry Pis with cameras. Figure 7.15 illustrates the experimental setting to calculate marker positions. Sixty white flat markers with a diameter of 10 mm were attached to a flat blackboard, with 20 markers for each depth and 3 depths in total. Each marker was about 100 mm apart in the X, Y, and Z axes. After calculating all marker positions from the program, the mean and standard deviation (SD) of marker distance were compared to the reference.

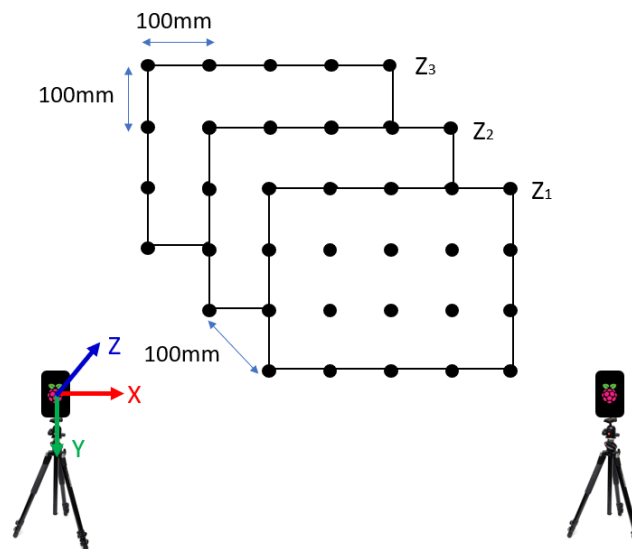


Figure 7.15 The experimental setting to calculate marker positions using a stereo camera pair.

Table 7.2 shows the different distances between markers in X, Y, Z, and Euclidean distance. For the different distances in columns, the mean difference distances in X, Y, and Z were 98.92 mm (SD = 1.50), -0.70 mm (SD = 1.43), and 3.52 mm (SD = 2.73), respectively. The mean Euclidean distance between markers for this condition was 99.03 mm (SD = 1.52). Similar results are presented in the different distances in rows. The mean difference distances in X, Y, and Z were 0.06 mm (SD = 1.11), 98.96

mm (SD = 1.76), and 4.26 mm (SD = 2.48), respectively. The mean Euclidean distance between markers for this condition was 99.09 mm (SD = 1.77). As a result, dX in the columns, dY in the rows, and Euclidean distances reported a good result because the calculating values were close to the actual distance, and the SD was less than 5 mm. Furthermore, the mean and SD of dX in the rows, dY in the columns, and dZ in both conditions were lower than 5 mm.

Table 7.2 also shows the different distances between markers in X, Y, Z, and Euclidean distances for depths 1, 2 and 3. The mean and SD reported a similar trend among dX, dY, dZ, and Euclidean distance. Overall, the different distances were close to the actual distance, and all SD were less than 5 mm. These results represent a significant improvement in accuracy compared to those in the previous chapter and show the benefit of introducing OpenCV.

Table 7.2 The mean and SD of the difference distance in X, Y, Z, and Euclidean distance for Experiment 4.2.

Conditions	dX (mm)		dY (mm)		dZ (mm)		Euclidean distance (mm)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Diff. distance in columns	98.92	1.50	-0.70	1.43	3.52	2.73	99.03	1.52
Diff. distance in rows	0.06	1.11	98.96	1.76	4.26	2.48	99.09	1.77
Z <sub>1</sub> in columns	99.62	1.25	-0.97	1.87	3.10	3.29	99.74	1.31
Z <sub>2</sub> in columns	98.70	1.44	-0.69	1.16	4.36	2.89	98.84	1.44
Z <sub>3</sub> in columns	98.51	1.62	-0.46	1.26	3.10	1.96	98.59	1.63
Z <sub>1</sub> in rows	0.14	1.13	99.32	1.93	3.65	3.03	99.44	1.93
Z <sub>2</sub> in rows	0.18	1.15	98.89	1.88	3.83	1.69	98.98	1.86
Z <sub>3</sub> in rows	-0.12	1.11	98.72	1.56	4.90	2.47	98.89	1.60

Note: d = Difference distance, Diff. = Difference distance, SD = standard deviation, mm = millimetre.

### 7.3.3. Experiment 4.3: To calculate marker positions using a stereo camera pair with different camera distances

The previous experiment revealed the great potential of using a stereo camera concept to calculate marker positions in 3 dimensions with SD of less than 5 mm. To confirm this advantage the following experiment was performed in which the distance

between cameras was changed. The cameras were positioned at 320 mm, 420 mm, 520 mm, 620 mm, and 720 mm separation. For these experiments, the number of markers and marker positions were similar to the previous experiment. Again, the mean and SD of the different in distance between markers in X, Y, Z, and Euclidean distance were calculated and reported.

Table 7.3 shows the different distances between markers in X, Y, Z, and Euclidean distance. The mean and SD of different distances reported a similar trend among X, Y, Z, and Euclidean distance. Overall, the different distances in X, Y, and Euclidean distance were close to the actual distance. Furthermore, all SD were also lower than 5 mm.

Table 7.3 The mean and SD of the difference distance in X, Y, Z, and Euclidean distance for Experiment 4.3.

Conditions	dX (mm)		dY (mm)		dZ (mm)		Euclidean distance (mm)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
720 mm, diff. in columns	98.92	1.50	-0.70	1.43	3.52	2.73	99.03	1.52
620 mm, diff. in columns	101.26	1.77	-0.19	3.03	-5.10	2.02	101.40	1.80
520 mm, diff. in columns	99.04	3.32	0.26	2.42	-4.90	3.64	99.30	3.30
420 mm, diff. in columns	99.12	1.22	-0.30	1.63	-2.10	3.09	99.20	1.20
320 mm, diff. in columns	99.68	2.24	-0.29	1.67	-1.70	4.22	99.80	2.30
720 mm, diff. in rows	0.06	1.11	98.96	1.76	4.26	2.48	99.09	1.77
620 mm, diff. in rows	0.43	3.06	99.69	4.87	-0.78	3.56	99.8	4.8
520 mm, diff. in rows	0.4	1.53	99.71	2.77	-1.64	3.04	99.8	2.7
420 mm, diff. in rows	-0.29	1.10	100.11	2.87	-0.60	2.57	100.1	2.8
320 mm, diff. in rows	-0.89	1.82	99.97	3.04	0.02	3.82	100.1	3.0

**Note:** diff = different distance, SD = standard deviation, mm = millimetre.

This experiment proved that the stereo camera concept could calculate marker positions in 3 dimensions with reasonable accuracy, and the standard deviation was lower than 5 mm. Although the camera distance was changed (from 320 mm to 720 mm or two times the distance increased) and the camera orientation was not perfectly aligned (position the cameras on tripods), the system could still calculate marker positions with acceptable accuracy. When using this stereo camera concept, the user can choose or adjust the camera distance to cover their application's volume or working space.

#### **7.3.4. Experiment 4.4: To locate a new global reference frame and calculate marker positions using multiple stereo camera pairs**

The previous experiment proved the stereo camera concept by calculating marker positions in 3 dimensions with an error of less than 5 mm using a stereo camera pair. In the next step to be able to quantify a volume and where an object is present in the volume, marker positions must be calculated using multiple stereo camera pairs looking at the scene from multiple directions. As a result, this section discusses a method for linking several stereo camera pairs together to make a system that can estimate marker positions from different viewpoints but uses the same global coordinates.

The new camera frame was built to position the cameras, and the dimension of the frame was 2,000 mm in width, length, and height. The camera bar was set at approximately the same height as the human abdomen or about the middle height of the trunk. Black curtains were hung around the camera frame to create a high-contrast background for the white markers. Eight Raspberry Pis with cameras were used in this development, which means that 2 Raspberry Pis with cameras were attached to each side of the camera frame. Figure 7.16 illustrates a camera frame and 8 Raspberry Pis with cameras attached to the camera bar. Numbers 1 to 8, illustrated on Figure 7.16, represent the camera numbers. Each camera was rotated slightly to face the centre of volume. The distance between the left and right cameras was set to approximately 800 mm.

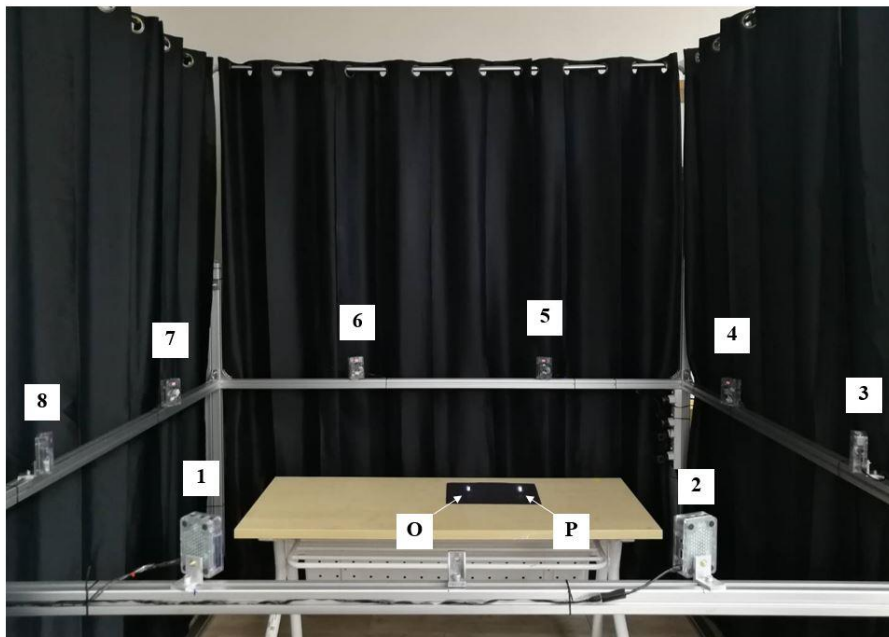


Figure 7.16 The camera frame, Raspberry Pis with cameras positioned on the frame, and experimental setting for Experiment 4.4.

After the hardware and software were completely set, the stereo camera calibration had to be performed between nearby cameras, including cameras 1&2, 2&3, 3&4, 4&5, 5&6, 6&7, 7&8, and 8&1, respectively. However, each camera pair had its origin at the left camera. As a result, the next step was to shift the origin from each stereo camera pair to be the same origin at the centre of the volume or at some point inside the camera frame.

One marker was positioned at the centre of the volume, and all cameras could clearly see this marker. This marker was set as the new origin or global reference frame for all cameras, called point "O". Another marker was positioned somewhere inside the volume, called point "P". Figure 7.16 also illustrates a new origin at point O and another marker at point P.

After completing stereo camera calibrations and marker calculations for points O and P, each stereo camera pair's translation vector and rotation matrix were applied to those marker position to shift the reference frame and re-calculate point P's new position with respect to the new origin at point O. Equation 7.3 describes how to calculate the new point P with respect to the new origin. Vector  $(\vec{OP})$  is a vector from the new origin to point P. XYZ value of point O is the vector  $u(\vec{u})$ , and the XYZ

value of point P is the vector  $v \left( \begin{matrix} \rightarrow \\ v \end{matrix} \right)$ . Furthermore, the rotation matrix is the stereo camera orientation with respect to the new global reference frame at point O. Equation 7.4 describes the formula of rotation matrix.

$$\begin{matrix} \rightarrow \\ OP \end{matrix} = \left( \begin{matrix} \rightarrow \\ (-u) \end{matrix} + \begin{matrix} \rightarrow \\ (v) \end{matrix} \right) \times [Rotation\ matrix] \quad (7.3)$$

$$Rotation\ matrix_{(Y-axis)} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \quad (7.4)$$

To shift the reference frame from the stereo cameras 1&2 to the new global reference frame and to calculate new point P, the transport vector  $u \left( \begin{matrix} \rightarrow \\ -u \end{matrix} \right)$  and vector  $v \left( \begin{matrix} \rightarrow \\ v \end{matrix} \right)$  were put in the Equation 7.3 with no rotation matrix. As a result, the new global reference frame at point O had the same XYZ axes as the stereo cameras 1&2, but had a different position. The new XYZ of point P or vector  $\left( \begin{matrix} \rightarrow \\ OP \end{matrix} \right)$  was respected to new global reference frame.

Additionally, the rotation matrix was required to calculate the new point P in other stereo camera pairs. The distance between the left and right cameras was set to approximately 800 mm. As a result, every stereo camera pair was rotated approximately 45° on the Y-axis. Figure 7.17 illustrates the camera positions and orientations, and points O and P in the top view. To be clarified, the new origin and cameras 1&2 had the same axes, -45° was the rotation in the Y-axis for cameras 2&3, -90° for cameras 3&4, -135° for cameras 4&5, -180° for cameras 5&6, -225° for cameras 6&7, -270° for cameras 7&8, and -315° for camera 8&1, respectively. Moreover, we assumed no rotation in the X and Z axes. Because these axes had already been modified and made perfectly parallel since the stereo camera calibration and image rectification process, and the camera pairs were precisely positioned on the camera frame.



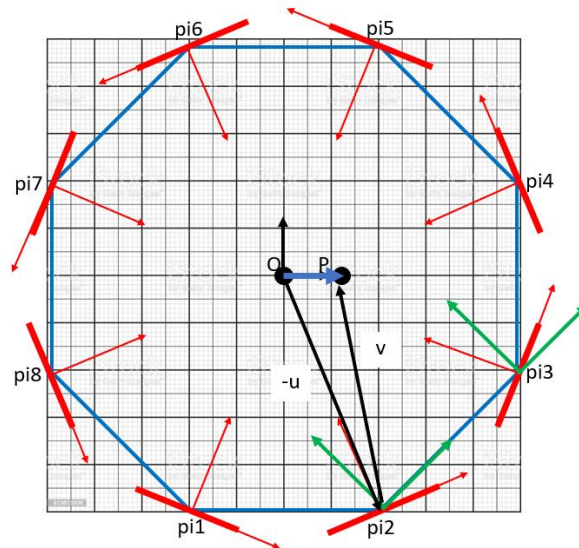


Figure 7.17 The camera positions and orientations on the camera frame, and points O and P in the top view.

After calculating the new point P with respect to the new global reference frame, marker positions from all camera pairs were averaged. This value was then used as a reference and compared to the value from each camera pair. The result reported that the RMSE in X, Y, Z, and 3D Euclidean distances were 2.59 mm, 2.97 mm, 3.87 mm, and 3.19 mm, respectively. As a result, this experiment showed a high possibility of using multiple stereo camera pairs to estimate marker positions in 3-dimensional space with errors lower than 5 mm.

### 7.3.5. Experiment 4.5: To calculate 3 marker positions with respect to the new global reference frame using multiple stereo camera pairs

The previous section described how to link multiple stereo camera pairs to respect the new global reference frame at point O and how to calculate a marker position on point P with respect to the new origin. This section describes another experiment that placed 3 markers around the new origin and used multiple stereo camera pairs to estimate its position. The first marker was attached along the X-axis, followed by the second and third markers along the Y and Z axes. The marker positions were then calculated and compared to the reference to determine the error. Figure 7.18 illustrates the experimental setting for Experiment 4.5. The distance between the new origin and

each marker was 150 mm. Then, the RMSE was calculated to assess the system's accuracy.

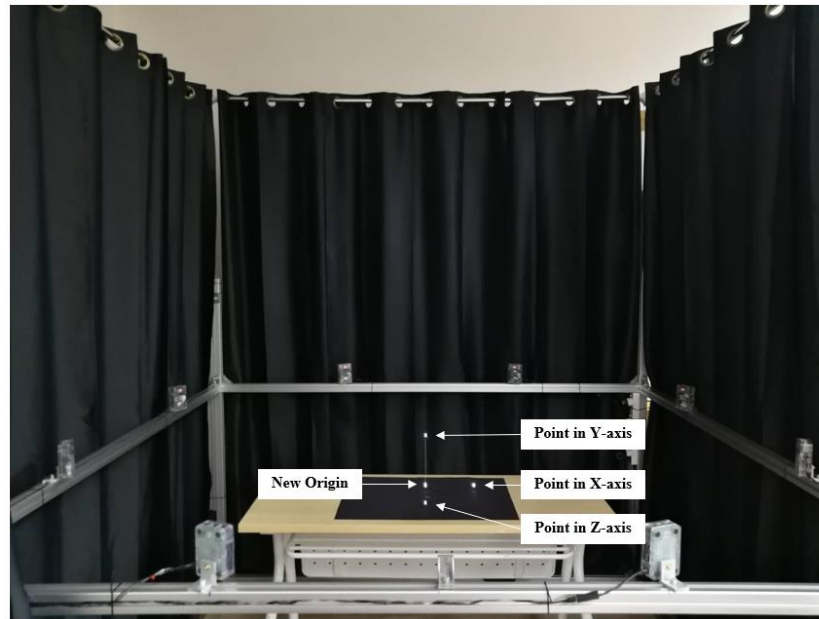


Figure 7.18 Experimental setting for Experiment 4.5.

Table 7.4 shows the RMSE in X, Y, Z, and Euclidean distances for Experiment 4.5. For a marker along the X-axis, the errors of X, Y, Z, and Euclidean distances were 3.60 mm, 3.44 mm, 4.43 mm, and 3.85 mm, respectively. Similar results show that the errors of X, Y, Z, and Euclidean distances at a marker along the Y-axis were 3.71 mm, 2.64 mm, 3.74 mm, and 3.40 mm, respectively. Similar results also show that the errors of X, Y, Z, and Euclidean distances at a marker along the Z-axis were 2.25 mm, 3.56 mm, 4.40 mm, and 3.52 mm, respectively. Finally, the RMSE of Euclidean distances for 3 markers was 3.59 mm on average.

Table 7.4 The RMSE in X, Y, Z, and Euclidean distance for Experiment 4.5.

	RMSE (mm)			
	X	Y	Z	Euclidean distance
Marker along the X-axis	3.60	3.44	4.43	3.85
Marker along the Y-axis	3.71	2.64	3.74	3.40
Marker along the Z-axis	2.25	3.56	4.40	3.52
Mean from all makers				3.59

**Note:** RMSE = root-mean-square-error, mm = millimetre.

As a result, all errors were lower than 5 mm, which still follows our goal. Experiments 4.4 and 4.5 showed that multiple stereo camera pairs could be used to estimate marker positions in 3-dimensional space with an error lower than 5 mm. The following experiment should calculate more markers and assess the system's accuracy.

#### **7.3.6. Experiment 4.6: To compare the 3D Euclidean distances between markers using a low-cost postural measurement system, a high-cost motion capture system, and a 3-dimensional scanning system**

The previous section described the errors after calculating 3 marker positions using multiple stereo camera pairs, with an error of less than 5 mm. This section extends on the prior experiment by adding more markers in the space, estimating their positions, and evaluating accuracy.

The imitated scoliosis model was created to mimic the shape of the scoliosis trunk. Thirty-six white passive markers with a diameter of 8 mm were attached to the model. Nine markers were attached to each side of the body, including the front, back, left, and right. During data collection, the marker positions were calculated using 3 systems, including a low-cost postural measurement system, a high-cost motion capture system (Motion analysis with Cortex software), and a 3D scanning system (Structure sensor with 3D Builder software). A high-cost motion capture system used circle reflective markers, while a low-cost postural measurement and 3-dimensional scanning system used circle white passive markers created by 3D printing. All systems' marker sizes and locations remained the same throughout the data collection process.

After estimating all marker positions, the Euclidian distances between markers at the same level were calculated. For example, the Euclidian distance was calculated between front-to-back markers at the lowest level, between left-to-right markers at the lowest level, and so on. The absolute maximum, minimum, and mean errors with standard deviation were calculated to evaluate the system's accuracy. Figure 7.19 illustrates the experimental setting for Experiment 4.6.

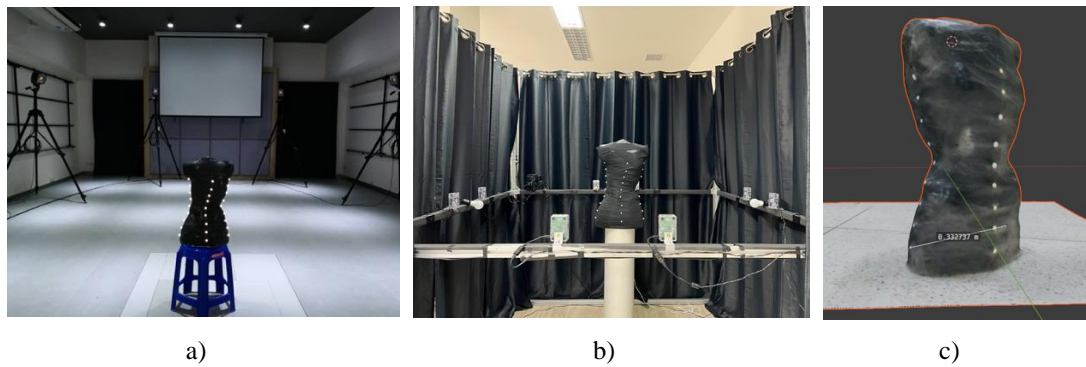


Figure 7.19 Experimental setting for Experiment 4.6, a) a high-cost motion system, b) a low-cost postural measurement system, and c) a 3D scanning system.

Table 7.5 shows the absolute maximum, minimum, and mean errors with standard deviation by comparing the Euclidian distance between low-cost postural measurement, high-cost motion capture, and 3D scanning systems. When comparing low-cost and high-cost motion capture systems, the absolute maximum and minimum errors were 22.05 mm and 0.08 mm, and the absolute mean error was 8.52 mm (SD = 6.44 mm). A similar result was presented when comparing low-cost postural measurement and 3D scanning systems. The absolute maximum and minimum errors were 18.67 mm and 0.41 mm, and the absolute mean error was 7.67 mm (SD = 5.62 mm). Furthermore, the errors were slightly lower when comparing 3D scanning and high-cost motion capture systems. The absolute maximum and minimum errors were 9.86 mm and 1.40 mm, and the absolute mean error was 5.54 mm (SD = 2.18 mm).

Table 7.5 The absolute maximum, minimum, and mean errors with SD by comparing the Euclidian distance between a low-cost postural measurement system, a high-cost motion capture system, and a 3-dimensional scanning system.

Conditions	Absolute Max. Error (mm)	Absolute Min. Error (mm)	Absolute Mean Error (mm)	SD (mm)
Low-cost VS High-cost motion	22.05	0.08	8.52	6.44
Low-cost VS 3D scanning	18.67	0.41	7.67	5.62
3D scanning VS High-cost motion	9.86	1.40	5.54	2.18

Note: SD = standard deviation, VS = versus, mm = millimetre.

Experiment 4.6 shows that errors increased highly when more marker positions were calculated using multiple stereo camera pairs. Although the absolute mean errors

were lower than 10 mm, the absolute maximum error and standard deviation were high. The following experiment should investigate the cause of the errors and find a method to reduce them.

### **7.3.7. Experiment 4.7: To assess the errors when moving the imitated scoliosis model in various positions**

The previous experiment's errors were reported to be notable, and the cause must be investigated in the following experiment. As a result, Experiment 4.7 continued using the imitated scoliosis model with markers attached to assess the errors. The model was moved to 9 positions, and a low-cost postural measurement system calculated marker positions. Euclidian distances between markers and errors were then calculated. The absolute maximum, minimum, mean, and RMSE errors was calculated to evaluate the system's accuracy. The position of the imitated scoliosis model is described below.

- Position 1: the model at the level of the camera with no rotation
- Position 2: the model at the level of the camera with 90° rotation
- Position 3: the model at the level of the camera with 180° rotation
- Position 4: the model is lower than the level of the camera with no rotation
- Position 5: the model is lower than the level of the camera with 90° rotation
- Position 6: the model is lower than the level of the camera with 180° rotation
- Position 7: the model is higher than the level of the camera with no rotation
- Position 8: the model is higher than the level of the camera with 90° rotation
- Position 9: the model is higher than the level of the camera with 180° rotation

Table 7.6 shows the errors of Experiment 4.7. In position 1, the absolute maximum, minimum, and means errors were 29.2 mm, 4.7 mm, and 15.0 mm, and the RMSE was 16.4 mm. In position 2, the absolute maximum, minimum, and means errors were 29.5 mm, 4.4 mm, and 15.0 mm, and the RMSE was 16.5 mm. In position 3, the absolute maximum, minimum, and means errors were 29.4 mm, 5.9 mm, and 15.5 mm, and the RMSE was 16.9 mm. In position 4, the absolute maximum, minimum, and means errors were 19.8 mm, 1.6 mm, and 8.4 mm, and the RMSE was 9.9 mm. In position 5, the absolute maximum, minimum, and means errors were 20.2 mm, 2.7 mm, and 8.4

mm, while the RMSE was 9.7 mm. In position 6, the absolute maximum, minimum, and means errors were 20.2 mm, 1.1 mm, and 8.5 mm, and the RMSE was 10.2 mm. In position 7, the absolute maximum, minimum, and means errors were 50.4 mm, 16.7 mm, and 30.5 mm, and the RMSE was 32.0 mm. In position 8, the absolute maximum, minimum, and means errors were 50.0 mm, 15.0 mm, and 31.1 mm, and the RMSE was 32.6 mm. In position 9, the absolute maximum, minimum, and means errors were 47.6 mm, 16.5 mm, and 30.4 mm, while the RMSE was 31.7 mm.

Overall, the absolute maximum error, absolute mean errors, and RMSE were higher than 10 mm in most positions, except for absolute mean errors and RMSE of positions 4 to 6. The errors showed similar when the model was positioned at the same height. The errors in position 1 were similar to positions 2 and 3. The errors in position 4 were similar to positions 5 and 6. The errors in position 7 were similar to positions 8 and 9. Furthermore, the errors decreased when the model moved downward. In contrast, the errors increased when the model moved upward.

#### **7.3.8. Experiment 4.8: To assess the errors when moving the imitated scoliosis model in various positions and estimate marker positions using 3 camera pairs versus 1 camera pair**

The marker positions from the previous experiment were estimated from the average values of 3 camera pairs. For instance, camera pairs 8&1, 1&2, and 2&3 averaged the marker location on the back of the model. On further investigation, we found a considerable discrepancy between the marker location estimated by these pairs of cameras with the camera pairs close together (cameras 8&1 and 2&3) and across corners of the frame showing different values to the pair straight onto the model (cameras 1&2). Hence, Experiment 4.8 excluded marker positions calculated by the left and right camera pairs and included only one pair from the middle directly facing the markers. For instance, the markers located on the back side of the model were located by the camera pairs 1&2. After that, the absolute maximum, minimum, mean, and RMSE were calculated to evaluate the system's accuracy and compare it to Experiment 4.7.

Table 7.6 also shows the errors of Experiment 4.8. In position 1, the absolute maximum, minimum, and means errors were 34.1 mm, 1.2 mm, and 15.7 mm, and the RMSE was 18.5mm. In position 2, the absolute maximum, minimum, and means errors were 32.7 mm, 1.5 mm, and 15.6 mm, and the RMSE was 18.1 mm. In position 3, the absolute maximum, minimum, and means errors were 34.7 mm, 4.1 mm, and 15.9 mm, and the RMSE was 18.6 mm. In position 4, the absolute maximum, minimum, and means errors were 23.6 mm, 0.2 mm, and 7.8 mm, and the RMSE was 10.4 mm. In position 5, the absolute maximum, minimum, and means errors were 20.7 mm, 0.2 mm, and 8.4 mm, and the RMSE was 10.5 mm. In position 6, the absolute maximum, minimum, and means errors were 20.8 mm, 0.8 mm, and 8.4 mm, and the RMSE was 10.7 mm. In position 7, the absolute maximum, minimum, and means errors were 54.0 mm, 15.3 mm, and 33.5 mm, and the RMSE was 35.4 mm. In position 8, the absolute maximum, minimum, and means errors were 53.1 mm, 12.9 mm, and 34.0 mm, and the RMSE was 36.1 mm. In position 9, the absolute maximum, minimum, and means errors were 56.7 mm, 17.4 mm, and 34.0 mm, and the RMSE was 36.0 mm.

Table 7.6 The absolute maximum, minimum, mean, and RMSE for Experiments 4.7 and 4.8.

Scoliosis model position	Absolute max. error (mm)		Absolute min. error (mm)		Absolute mean error (mm)		RMSE (mm)	
	Exp.4.7	Exp.4.8	Exp.4.7	Exp.4.8	Exp.4.7	Exp.4.8	Exp.4.7	Exp.4.8
Position 1	29.2	34.1	4.7	1.2	15.0	15.7	16.4	18.5
Position 2	29.5	32.7	4.4	1.5	15.0	15.6	16.5	18.1
Position 3	29.4	34.7	5.9	4.1	15.5	15.9	16.9	18.6
Position 4	19.8	23.6	1.6	0.2	8.4	7.8	9.9	10.4
Position 5	20.2	20.7	2.7	0.2	8.4	8.4	9.7	10.5
Position 6	20.2	20.8	1.1	0.8	8.5	8.4	10.2	10.7
Position 7	50.4	54.0	16.7	15.3	30.5	33.5	32.0	35.4
Position 8	50.0	53.1	15.0	12.9	31.1	34.0	32.6	36.1
Position 9	47.6	56.7	16.5	17.4	30.4	34.0	31.7	36.0

Note: Max. = maximum, Min. = minimum, mm = millimetre, Exp. = Experiment.

Overall, the errors in Experiment 4.8 were slightly higher than in Experiment 4.7. The absolute maximum error, absolute mean errors, and RMSE were higher than 10 mm in most positions, except for absolute mean errors and RMSE of positions 4 to 6. Like the previous experiment, the errors showed similar when the model was positioned at the same height. The errors decreased when the model moved downward. In contrast, the errors increased when the model moved upward. This experiment showed that using one camera pair to estimate marker positions gave a higher error than using three. However, both experiments' errors were still high and changed when the model and marker positions changed the height. The following experiment should investigate the causes of the errors and find a method to resolve them.

#### **7.3.9. Experiment 4.9: To assess the errors when calculating the Euclidean distances between upper and lower markers on each side of the scoliosis model**

Experiments 4.7 and 4.8 showed that the errors varied and depended on the height of the scoliosis model when multiple stereo cameras were used to estimate marker positions. Both experiments reported high errors, which was the opposite result when using one stereo camera pair in Experiment 4.2. After careful consideration, we determined that the problem may come from the estimated values of the rotation matrix for each stereo camera pair used to calculate the new marker positions with respect to the new global reference frame. If the position and orientation of the camera used to construct the rotation matrix did not exactly match the actual position and orientation, then errors would be introduced. Hence, it was decided to repeat the experiment with the cameras carefully rotated and when attached directly to the frame bars so that they were parallel to the bars.

As a result, Experiment 4.9 assessed the errors again when calculating the Euclidean distances between upper and lower markers on each side of the scoliosis model using one stereo camera pair directly facing the markers. Furthermore, this experiment assessed the error when both cameras of each stereo camera pair were in horizontal rotation and no rotation (parallel camera). The absolute maximum, minimum, mean, and RMSE were calculated to assess the system's accuracy.



Table 7.7 shows the errors of Experiment 4.9. The results showed that the absolute maximum, minimum, and mean errors for the horizontal rotation of the camera were 5.63 mm, 0.42 mm, and 2.32 mm, and the RMSE was 2.93 mm. The absolute maximum, minimum, and mean errors for a parallel camera were 4.18 mm, 0.43 mm, and 2.06 mm, and the RMSE was 2.42 mm.

Table 7.7 The absolute maximum, minimum, mean, and RMSE for Experiments 4.9.

Camera Condition	Absolute Max. Error (mm)	Absolute Min. Error (mm)	Absolute Mean Error (mm)	RMSE (mm)
Horizontal rotating cameras	5.63	0.42	2.32	2.93
Parallel cameras	4.18	0.43	2.06	2.42

Note: Max. = maximum, Min. = minimum, mm = millimetre, RMSE = root-mean-square-error.

This set of experiments proved that the stereo camera concept for calculating marker positions was still applicable to the application in orthotics, with an error lower than 5 mm. However, errors significantly increased when marker positions were estimated using multiple stereo camera pairs. The estimation of the rotation matrix to link all stereo camera pairs to respect the new global reference frame and estimate new marker position was the major problem that caused high errors. The errors were similar when the camera were horizontally rotated or parallel. However, the parallel condition showed slightly lower errors. Therefore, it was decided to position the stereo camera in parallel to each other and perpendicular to the support bar so as to minimize errors.

#### 7.4. Discussion and conclusion for Experiment 4

In this chapter, we took the previous chapter's recommendations and found the method to position 2 cameras in perfect alignment before calculating marker positions. The previous chapter attached the cameras precisely on the frame and calculated marker positions using a simple right triangle similarity theorem and trigonometry. In contrast, this chapter used the stereo camera concept from OpenCV and further developed the program to relate 2 cameras before calculating marker positions. As a

result, this chapter showed the practicality and potential of using single and multiple stereo camera pairs to estimate marker positions in 3-dimensional space with reasonable errors.

The literature did not describe how to obtain good chessboard images for calibrating the camera. Most mentioned that the chessboard had to be moved in front of the camera, but none mentioned the exact position and orientation of the chessboard or how many images were needed for this process. **Experiment 4.1** proved the method for getting the best camera calibration results. This experiment had 2 sets of images: the dynamic and static images taken by the servo motors (Condition 1) and the dynamic and static images taken by the user (Condition 2). The camera calibration results were compared with the results from the previous method. The experiment proved that taking the dynamic and static images from the user in Condition 2 provided the best result of camera calibration, and the results were similar to the previous method. The reason for the significant difference in Condition 1 might be the position and orientation of the chessboard pattern. The images from Condition 1 did not vary the positions and orientations of the chessboard compared to Condition 2. As a result, insufficient change of the positions and orientations of the chessboard pattern could cause a high error accumulation. We finally decided to use the manual method to calibrate all Raspberry Pis and cameras. We recommended taking more than 30 chessboard images in various positions and orientations to calibrate the camera.

**Experiments 4.2 and 4.3** showed that the stereo camera concept could be used to calculate marker positions in 3D space with errors lower than 5 mm. Sixty markers were used to calculate and assess the errors for both experiments. Even when the camera distances were changed, the errors remained under 5 mm. We were satisfied with this concept and continued using it in the following development.

The previous experiment reported reasonable errors. However, it calculated marker positions using two cameras or one stereo camera pair. In our application, we want to quantify the spinal parameters, and the marker positions should be calculated by more than one stereo camera pair. As a result, **Experiments 4.4 and 4.5** described a method to create a new global reference frame, link multiple stereo camera pairs with respect to the new origin, and estimate new marker positions with respect to the new origin. Each stereo camera pair's rotation matrix and translation vector were essential

parameters for this process. The translation vector could be calculated from each stereo camera pair. The rotation matrix could not be calculated but was estimated by the camera orientation on the frame. These experiments initially estimated one and three marker positions. The result showed that the errors of both experiments were lower than 5 mm. This experiment proved that multiple stereo camera pairs could be used to estimate marker positions in 3D space with errors lower than 5 mm.

The imitated scoliosis model was built to mimic the shape of a scoliosis patient. Thirty-six circle white markers were attached to the model. **Experiment 4.6** calculated marker positions and Euclidian distances between markers using low-cost postural measurement, high-cost-motion capture, and 3D scanning systems. When comparing errors between low-cost and high-cost motion capture systems, the absolute mean error was larger than 5 mm but less than 10 mm. However, the absolute maximum error and standard deviation were high. **Experiment 4.7** continued to assess the errors when the scoliosis model was moved in 9 positions. The result showed that the errors were still high, greater than 10 mm. The errors slightly decreased when the model was moved downward, and the error slightly increased when the model was moved upward. **Experiment 4.8** continued to assess the errors by comparing the accuracy of using one stereo camera pair and three stereo camera pairs to estimate marker positions on the scoliosis model. The result showed that using three stereo camera pairs gave lower errors than one pair. However, the errors from both groups were still high and greater than 10 mm.

Experiments 4.6 to 4.8 reported high errors, which is the opposite when using one stereo camera pair to estimate marker positions in Experiment 4.2. After careful consideration, the problem may come from the estimated values of the rotation matrix for each stereo camera pair to estimate the new marker positions with respect to the new global reference frame. **Experiment 4.9** assessed the errors when calculating the Euclidean distances between upper and lower markers using one stereo camera pair. The results showed that the errors were lower than 5 mm again, and the parallel camera showed slightly lower errors when compared to horizontally rotating the camera.

In Chapter 3, the literature review, we proposed the error of a motion capture system at approximately 5 mm, and the RMSE in Experiment 4.9 was 2.42 mm, which was lower than the proposed error. We could conclude that our low-cost postural

measurement system had a suitable accuracy to quantify spinal parameters in scoliosis patients. We used a similar geometry in Figure 3.28 (refer to Chapter 3) to re-calculate the angle deviation from the RMSE of 2.42 mm. Point A to point B (X-value) is the distance of spinal segments. Y-value is the half of RMSE from Experiment 4.9 (1.21 mm) because the error deviates up and down. Theta ( $\Theta$ ) is the deviation angle we expected to calculate.

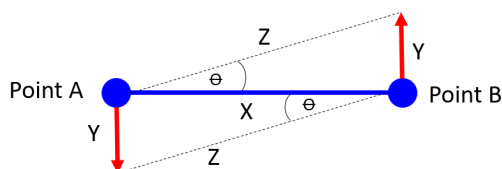


Figure 3.28 (Refer to Chapter 3) Geometry to calculate the error for a postural measurement system in a scoliosis application.

Table 7.8 The deviation angle when the RMSE of the postural measurement system was 2.42 mm.

Spinal Segments	Deviation Angle, $2\Theta$ (Degree)
CUTA	0.78
CLTA	1.24
CULA	2.42
CLLA	2.65

Table 7.8 shows the angle deviation result from four spinal segment lengths when the RMSE from Experiment 4.9 was used for calculation. At the shortest spinal segment (CLLA), the angle was  $2.65^\circ$ , and the angle decreased when the length increased. This calculation confirmed that our developed postural measurement system had enough accuracy to quantify spinal parameters in scoliosis patients. The RMSE of developed postural measurement system was 2.42 mm, which could make the angle deviation  $2.65^\circ$ , which was lower than a maximum of  $5^\circ$  of curve progression. Even though the developed system calculated the shortest distance of the spinal segment, the angle deviation was still small, and the system was still accurate in calculating the spinal parameters in scoliosis application.

## 7.5. Limitations and recommendations

These experiments showed the practicality and potential of using single and multiple stereo camera pairs to estimate marker positions in 3-dimensional space. Overall, the errors were lower than 5 mm when using a single stereo camera pair. However, the errors increased between 15 mm and 30 mm when using multiple stereo camera pairs due to inaccuracies in relating pairs of cameras. In our developing application, most markers will be attached to the scoliosis trunk on the back side, where they are used to quantify the spinal parameters in 3 dimensions. These calculations require high system accuracy. Furthermore, some markers will be attached to the front and lateral sides but will only be used to illustrate the force direction and will not need high system accuracy. As a result, all spinal parameters will be calculated by one stereo camera pair (stereo camera pair 1&2), while any stereo camera pairs (1&2, 3&4, 5&6, or 7&8) can be used to locate other anatomical or force related locations for the visualisation.

The above is a pragmatic solution for this project because we found errors significantly increased when marker positions were estimated using multiple stereo camera pairs and camera pairs not planar to the spine. Estimating the rotation matrix for each stereo camera pair was the major problem causing these errors. We chose to end the development of a low-cost postural measurement system at this stage as we had a system which could be used to accurately quantify spinal parameters in scoliosis patients. However, a generalised system for 3-dimensional measurement could be produced by further research if a method could be found to calculate the rotation matrix for each stereo camera pair relative to the other pairs and, hence, accurately set them in a global measurement frame.

The system developed using the Raspberry Pi with cameras could be used to record images of an object and send these images to the computer for 3D marker location. The marker position and spinal parameter calculations were performed after the data collection. This system was sufficient for the application envisaged in this thesis for orthotic treatment. Further research should find a method to calculate marker positions on the Raspberry Pi, send the result to the computer, report the results in real-time and

illustrate spinal parameters on a graphical user interface in real time to assist practitioners during scoliosis treatment.

**8. Chapter 8 Clinical research methodology and Outcome measurements for Adolescent Idiopathic Scoliosis**

## 8.1. Introduction

The previous sections described the successful development of a force-measuring system, a scoliosis casting apparatus, and a low-cost postural measurement system. The force-measuring system could measure the magnitude of forces applied to correct the spinal deformity in various locations. The scoliosis casting apparatus could be adjusted to apply the forces needed to correct the spinal deformity. The low-cost postural measurement system could calculate marker positions in 3-dimensional space with reasonable errors and is ready to quantify spinal parameters in 3 dimensions. The system has, therefore, the potential to quantify the three-dimensional biomechanical force system used during orthotic treatment and measure the change in spinal posture caused by the treatment.

Scoliosis is a 3-dimensional deformity of the spine. However, most research has focused on spinal deformity change in the coronal and sagittal planes, which is still a lack of understanding of spinal deformity in three dimensions. Radiographs have become a standard tool for quantifying spinal deformity because they are relatively low-cost and most available in the hospital. However, this method can only describe the deformity in the coronal and sagittal planes. The use of spinal parameters calculated from planar radiographs to describe two-dimensional deformity has been in use for a long time, and as a result, the nature of 3-dimensional scoliotic deformity is still unclear. Some research has studied vertebral rotation in the transverse plane using two planar radiographs as an indirect measurement. However, this method could only describe the rotation of vertebrae-by-vertebrae, not the whole spinal column in three dimensions. A spinal orthosis is a standard treatment provided for AIS patients to stop curve progression and delay spinal surgery. The orthosis provides 3-dimensional force correction to re-align the spine to the optimal position. However, the current methods used to quantify the spinal outcome from orthotic treatment remain two-dimensional. There needs to be more evaluation in the transverse plane and in three dimensions. Furthermore, casting is an essential process to capture the shape of the scoliotic trunk under the forces applied by the clinician's hands. It requires skill and teamwork from clinicians to achieve a good negative cast. The use of a casting frame is rare in clinical practice because it is relatively expensive and complicated to use. There is still a lack



of information about the effectiveness and benefits of using a casting frame to assist clinicians in obtaining a good negative cast. Furthermore, there is still a lack of information about the magnitude of forces applied to correct the scoliosis deformity and the locations and directions of forces in 3 dimensions during the casting process. The result from this clinical experiment will help clearly understand how the 3-dimensional biomechanical force correction from the orthosis re-aligns the spine to the optimal position in 3 dimensions and whether such technology can help progress practice in this complicated field.

## **8.2. The objectives of the clinical study**

The next step was to apply all systems to treat scoliosis patients and evaluate the performance of the developed system. The main purposes of this thesis were to develop a system that could quantify the spinal deformity of AIS patients in 3 dimensions, a system that could apply forces to correct the spinal deformity in 3 dimensions, and a system that could measure the magnitude of forces and illustrate those force directions in 3 dimensions. The second purpose of this thesis was to demonstrate the feasibility of using the developed system to treat AIS patients in clinical practice. As a whole system, this thesis aims to demonstrate, evaluate, and improve the understanding of the spinal deformity change in 3 dimensions during the orthotic treatment of AIS patients. The detailed objectives are listed below.

- To verify if the developed postural measurement system could quantify the spinal parameters in coronal and sagittal planes and to study the validity of the spinal parameters using the low-cost postural measurement system compared to the radiographic evaluation. The spinal parameters for this objective consisted of the spinal column in 4 segments on coronal and sagittal planes and the trunk balance in the coronal plane. Jang (Jang, 2018) previously introduced new spinal parameters based on motion capture data by dividing the spinal column into four segments (referred to in Chapter 3), and we adopted these for this thesis. These spinal parameters divided the spinal column into four

sections, including the upper thoracic (between C7 and T7), lower thoracic (between T7 and T12), upper lumbar (between T12 and L2), and lower lumbar (between L2 and L5). To clarify, the spinal angles in the coronal plane consisted of CUTA (coronal-upper-thoracic angle), CLTA (coronal-lower-thoracic angle), CULA (coronal-upper-lumbar angle), and CLLA (coronal-lower-lumbar angle). Then, the spinal angles in the sagittal plane consisted of SUTA (sagittal-upper-thoracic angle), SLTA (sagittal-lower-thoracic angle), SULA (sagittal-upper-lumbar angle), and SLLA (sagittal-lower-lumbar angle). The advantage of this technique was that it was a convenient and practical method for clinicians to identify the location of the spinous process of vertebrae and quantify the spinal angles without considering the curve's type and location. The spinal angle could be considered at individual segments, and the change of an individual segment could be compared and evaluated. The trunk balance in the coronal plane was a primary parameter clinicians used to observe the trunk as a whole spine. It was the relationship of the vertical line between C7 and L5 (referred to in Chapter 3). If these points were in the same line, the trunk was balanced or had no coronal decompensation or trunk leaning. In contrast, if these points were not in relation, the trunk was not balanced, or the coronal decompensation presented to either the left or right side. Clinicians had to measure the distance between the vertical lines of C7 and L5. While the patients are wearing the spinal orthosis, clinicians expect that the patients should not present coronal decompensation, or this distance should be close to zero as much as possible to keep the whole spine in balance.

- To examine if the developed postural measurement system could quantify the angles of the spinal column in 3 dimensions. The spinal parameter for this objective consisted of the spinal column in 4 segments similar to the parameters in coronal and sagittal planes, which Jang had previously introduced (Jang, 2018). This parameter was a 3-dimensional angle respective to the horizontal plane (referred to in Chapter 3 and in Topic 8.5). To clarify, the spinal angles in 3 dimensions consisted of 3D-UTA (3-dimensional-upper-thoracic angle), 3D-LTA (3-dimensional-lower-thoracic angle), 3D-ULA (3-dimensional-

upper-lumbar angle), and 3D-LLA (3-dimensional-lower-lumbar angle). Like the previous parameters, the advantage of this technique was the convenience and practical method for the clinicians to identify the location of the spinous process and quantify the spinal angle without considering the curve's type and location. The spinal angle could be considered at individual segments, and the change of an individual segment could be compared and evaluated. Furthermore, this parameter might help clarify the unclear 3-dimensional deformity of AIS patients.

- To examine if the developed postural measurement system could quantify the angle of trunk rotation in the horizontal plane. The spinal parameter for this objective consisted of the angle of horizontal trunk rotation at the T7 (Interior angle of the scapula), T12, L2, and L5 levels in relation to the PSIS level (see the illustration and calculation in Topic 8.5). Previous research from Jang (Jang, 2018) studied horizontal rotation at interior angle of the scapula (T7 level) and paraspinal muscles around the L3/L4 area (referred to in Chapter 3). This thesis continued to use this parameter at T7 (Interior angle of the scapula) and added more levels to cover the trunk shape. The trunk typically presents the horizontal rotation following the deformity in the transverse plane and presents the rib hump for the thoracic curve and lumbar prominence for the lumbar curve. Clinicians only report this deformity as appearing or disappearing but do not measure the exact value. There is still a lack of information on this angle to represent the asymmetry of the trunk in the horizontal plane. Therefore, this spinal parameter was a new parameter to evaluate the trunk asymmetry in the horizontal plane, which was a convenient and practical method for clinicians to evaluate the shape of trunk change in the horizontal plane. This parameter might also help clarify the unclear 3-dimensional deformity of AIS patients.
- To examine if the developed postural measurement system could quantify the trunk asymmetry in the coronal plane. This objective included 5 spinal parameters, including the shoulder, axilla and waist levels, and the distance

between the CSL line to the left and right at the axilla and waist (see the illustration and calculation in Topic 8.5). These spinal parameters are commonly observed during physical examination to evaluate the body asymmetry. Clinicians report the outcome with the level or not level at these areas, and what side is longer or higher than another side, which does not measure the outcome with the exact value. Therefore, this system might help to quantify these parameters with the exact values. Clinicians could evaluate the change of these parameters before and when applying the forces to correct the spinal deformity. However, these parameters were not the key parameters to evaluate the improvement or failure of AIS treatment, and they could only describe how the shape of the trunk changes during the treatment.

- To examine if the scoliosis casting apparatus could be adjusted to correct the spinal deformity in 3 dimensions. The casting is an essential process to capture the shape of the scoliosis trunk under the forces applied by the clinician's hands. It requires skill and teamwork to achieve a good negative cast. As a result, the scoliosis casting apparatus was designed in this thesis to assist clinicians during casting process (referred to in Chapter 4). This apparatus provided 7 force locations and directions to correct the spinal deformity in 3 dimensions, including left and right gluteus medius, left and right axilla, left and right corrective at thoracic or lumbar, and abdominal area. However, the number of required forces to correct deformity could differ depending on the curve type, curve location, and other clinical considerations. Clinicians could choose the force locations and directions most suitable to correct the deformity for individual patients. The spinal parameters were used to examine this objective, consisting of the angles of the spinal column in 3 dimensions, the angles of trunk rotation in the horizontal plane, and the trunk asymmetry in the coronal plane, to evaluate and compare the change of the spinal deformity before and when applying the forces to correct the spine.
- To examine if the force measuring system could measure the force magnitude and the postural measurement system could illustrate the force directions when

the forces were applied to correct the spinal deformity in 3 dimensions. Recently, there has yet to be a research study about the magnitude of forces applied to correct the scoliosis deformity during casting process, and also a lack of a clear illustration of force locations and directions to describe the spinal correction in 3 dimensions. Two to three clinicians apply the forces with their hands to the patient's trunk and never measure how much force is applied to correct the deformity. The judgement is based on their clinical experience. As a result, the force measuring system was developed in this thesis to measure the magnitude of forces (referred to in Chapter 5). The magnitude of forces and illustration of force directions from the developed system might help to clarify how 3-dimensional biomechanical forces could correct scoliosis deformity in 3 dimensions.

In the following step, all systems were applied to treat AIS patients, and the performances of the developed system were evaluated following the objectives mentioned above.

This chapter describes the research methodology, including participant inclusion and exclusion criteria, sample size selection, study design, research protocol and procedure, data collection, and data analysis. This chapter details the methods used to calculate the spinal parameters. The clinical research methodology and outcome measurement presented in this thesis should allow us to better understand the orthotic treatment for AIS patients in 3 dimensions and examine the developed system's usefulness before use in clinical practice in the future.

### **8.3. Clinical research methodology**

The study was interested in applying the developed system to treat scoliosis patients and assist practitioners during the assessment, optimal correction, and casting processes. The study design was a pilot study. Ten AIS patients were recruited for the study, and the number of participants was referred from previous articles (Pennella *et al.*, 2013, Khanali *et al.*, 2015, Wang *et al.*, 2022). A sample of 10 case studies should

be sufficient to determine the success or otherwise of the system and provide sufficient feedback to evaluate its usefulness and any future modifications that would be beneficial. Participants' inclusion and exclusion criteria are listed below.

### **Inclusion criteria**

- Male or female who was between 10 - 18 years old
- Person who was diagnosed with adolescent idiopathic scoliosis (AIS) by rehabilitation or orthopaedic doctor
- Coronal Cobb angle was between 25° and 40°, and Risser's sign was between 0 and 5
- All spinal curve patterns
- Flexibility of the curve was between flexible to semi-rigid curves
- Apical vertebra from T8 and below
- Body mass index (BMI) was between 18.5 - 24.9 kg/m<sup>2</sup>

### **Exclusion criteria**

- Any history of spinal surgery
- Any underlying disease that affects the lung and heart function
- Not be able to walk
- Psychological and communication problems

### **Recruitment process**

The data collection was conducted at the Sirindhorn School of Prosthetics and Orthotics (SSPO), under the Faculty of Medicine Siriraj Hospital, Mahidol University, Thailand. Participants were recruited by 3 methods, poster invitation (see the Appendix 10), researcher invitation, and clinician invitation. All participants were AIS patients attending for treatment at SSPO.

Ethical approval was obtained from the University of Strathclyde and Mahidol University. Posters were placed in the treatment areas of the SSPO. Patients who were interested in the project could contact the researcher by using the telephone number provided on the poster. Furthermore, the researcher reviewed the SSPO patient database and contacted patients who met the inclusion criteria who were due to attend.

Clinicians who regularly meet patients in the P&O clinic also introduced the project to patients.

The researcher informed prospective participants about the research purposes, procedures, and schedule (see the Appendix 10). Patients were allowed to decline or accept the proposal with all conditions. If the patients were willing to join the project, they were asked to complete the consent form. The participants could withdraw from the study at any time, and the withdrawal did not affect their current or future treatment. If the candidates were interested in joining this research study later, within the research time frame, they could contact us again.

### **Appointment schedule and Procedures**

Participants attended the research project two times, including the recruitment and data collection processes.

- **At the first appointment**

Researcher gave detailed information about the research objectives, procedures, and schedule. Once the participants agreed to participate in the study, they signed the consent form (see the consent form in the Appendix 11). Then, demographic data was collected in the case record form (see the case record form in the Appendix 12). They also gave access to out-of-brace radiographs of the whole spine, which were taken within the last three months. If no such radiographs existed participants were referred to take radiographs at Siriraj Hospital.

- **At the second appointment**

To minimize the bias, only one researcher performed radiographic evaluation, physical examination, and data collection processes as a single assessor. Furthermore, practitioner was required to assist the main researcher during the casting process and general requests.

The researcher performed a radiographic evaluation and a physical assessment of the participants, including trunk asymmetry, trunk balance, rotational asymmetry of the spine, flexibility of the spinal curves, and abnormal sagittal curves of the spine. Then, the researcher carefully identified the anatomical landmarks on the participant's trunk before attaching the marker bases. The marker bases (fig. 8.1a), consisted of a piece of magnet and were attached to anatomical landmarks to identify the marker

locations. The bases were secured to the skin using adhesive tape (fig. 8.1c). Then, participants wore black bodysuits to turn the background black.

In the following step, **the assessment process**, the markers were attached to the bases previously attached to the participant's skin (fig. 8.1d). The markers used in this thesis were a white spherical marker with an 8 mm diameter fabricated by 3D printing. The marker was permanently attached to a second piece of magnet (fig. 8.1b) with the opposite pole to the first one the base. As a result, one magnet from the base and the other magnet on the marker would align well when attached and did not move during data collection. The reason for having two magnets was that it was hard to perform casting while the markers remained on the skin. Therefore, the markers had to be removed during the casting process and could be replaced if needed. However, the bases remained in place, allowing us to refer to anatomical landmarks again on the negative cast and when quantifying the spinal parameters on the positive cast.

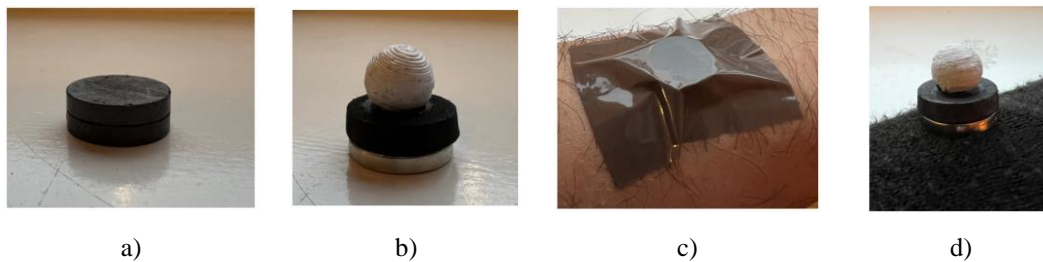


Figure 8.1 The base and marker used for data collection, where a) a base (a piece of magnet), b) a marker (a white circle marker attached with another piece of magnet), c) place the base on the landmark and secure with tape, and d) attach the marker to the base.

After attaching all markers on the bases, participants were asked to stand in the middle of the camera frame for 5 minutes (fig. 8.2). The researcher ran the low-cost postural measurement program (getimagepairs program) to collect the marker positions. In the current software version, the operator had to manually identify single markers in each camera image to quantify them, which was a rather slow process, and the resulting spinal quantification took up to 5 minutes. After completing this process, participants rested for 10 to 15 minutes.



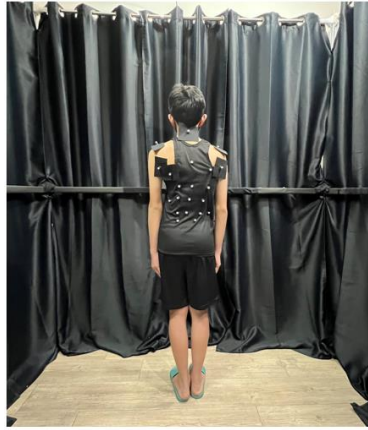


Figure 8.2 Participant stand in the middle of the camera frame and the low-cost postural measurement program collects the data during assessment process.

**In the optimal correction process**, the researcher moved the scoliosis casting apparatus inside the camera frame. The researcher then removed the non-spinal markers, but the bases remained in place. Some markers representing the spinal column (C7, T7, T12, L2, and L5) we kept in place to aid optimal correction. When the set-up was ready, participants were asked to stand in the middle of the apparatus and then perch on the edge of the sitting bar. The researcher adjusted the manipulators to correct the deformity. Once the operator deemed the manipulator adjustment had obtained the maximum achievable or optimum correction, the researcher ran the program to collect the magnitude of forces for three trials using the “force measuring program”. The researcher then ran the low-cost postural measurement program, “getimagepairs program”, to collect the marker positions (fig. 8.3).



Figure 8.3 The spinal deformity is corrected by scoliosis casting apparatus and the programs (force measuring and postural measurement systems) collect the data during optimal correction process.

The researcher created a line mark on the manipulators using a permanent marker before releasing them. The orientation of the manipulators to the frame was preserved when they were released. Using this mark and by preserving the manipulator orientation, the operator could return the manipulator to the same position and hence apply the same forces as previously to correct the spine. The mark was created on three manipulators (the abdominal, left axilla, and left gluteus medius). Releasing these manipulators was enough to allow the participants to move outside the apparatus before casting. After that, the researcher removed the 5 spinal markers before casting, but the bases remained in place. The researcher wrapped plastic wrap around the participant's trunk to prevent plaster contact with the skin. The researcher then performed casting on the participant's trunk. Afterwards, participants were asked to move back inside the apparatus again and perch on the sitting bar. The researcher then adjusted the three released manipulators back to the same position and re-checked the magnitude of forces using the force-measuring program. A wait of 10 - 15 minutes until the plaster was completely set ensued while the manipulators applied the forces to correct the deformity (fig. 8.4) . The result of this process was a cast of spinal shape after correcting the deformity. The negative cast with the bases incorporated was removed from the participants' trunk. The participants were cleaned immediately, and they could then change their clothes and return home.



Figure 8.4 The spinal deformity is corrected by scoliosis casting apparatus and researcher performs casting.

In the described procedure, we could not directly collect the data with the participants during the casting process due to the use of the plaster of Paris wrap to preserve the trunk shape. Based on the literature review, no research described a method of casting the prosthesis and orthosis when markers were attached to the anatomical landmarks. It is hard to perform casting while the markers remain on the skin. As a result, we quantified the spinal parameters from the positive cast instead. The negative cast was filled by plaster mixed with water, and the result of this process was called a positive cast. The location of the bases on the positive cast were carefully identified. When the plaster bandage was wrapped over the base attached to the participant's skin during the casting process, it presented as a small dimple on the surface of the plaster cast, which was mirrored on the positive cast. Therefore, we could use this area to recreate the marker locations. New bases were attached to these areas on the positive cast. The cast was then wrapped in elastic black tape to turn it into a black model. The researcher then attached markers to the bases and ran the low-cost postural measurement system to collect marker positions using the “getimagepairs program” (fig. 8.5).



Figure 8.5 Re-attach markers to the positive model, and researcher run program to collect marker positions.

The magnitude of the forces applied could be recorded during the casting process with participants. However, the locations and directions of forces could not. We, therefore, quantified these alongside the spinal parameters using the positive cast, not the participants. To achieve this, long wooden sticks with 2 markers attached were created to represent the manipulator arms. The sticks fitted tightly into the sockets of

the removable pads (fig. 8.6b) were previously used to apply pressure to the trunk using the manipulators. Figure 8.6 illustrates how the removable pad can be removed from the load cell at the end of the manipulator and attached to the wooden stick. The removable pads were re-attached to the positive cast at the same place as during the casting, and the markers on the wooden sticks were used to determine the location and orientation of each force by again running the low-cost postural measurement system to collect marker positions using the “getimagepairs program”.



Figure 8.6 a) a removable pad that can be removed from the load cell, and b) a long wooden stick that can fit along the tip of the removable pad to represent the force direction.

During post-data processing, all collected data (images of markers) were used to calculate their positions in 3-dimensional space using the “reconstructfromimages program”, and the result of this process was an XYZ of each marker. All marker positions were put into an Excel spreadsheet with suitable formulae to calculate the spinal parameters. The next section further describes the parameters collected during the data collection process and describes the method used to calculate the value of each parameter.

#### 8.4. Clinical outcome measurement and Data analysis

Several clinical parameters had to be measured in this thesis. For the out-brace radiographs (referred to in Chapter 3), the clinical parameters are as follows.

- Apical vertebrae with their upper- and lower-end (fig. 8.7a)
- Trunk balance or coronal decompensation, and Risser sign (fig. 8.7a)
- Magnitude of coronal Cobb angle, curve pattern, and curve location (fig. 8.7a)

- Spinal angles at the sagittal plane: SUTA, SLTA, SULA, and SLLA (fig. 8.7b)
- Spinal angles at the coronal plane: CUTA, CLTA, CULA, and CLLA (fig. 8.7c)

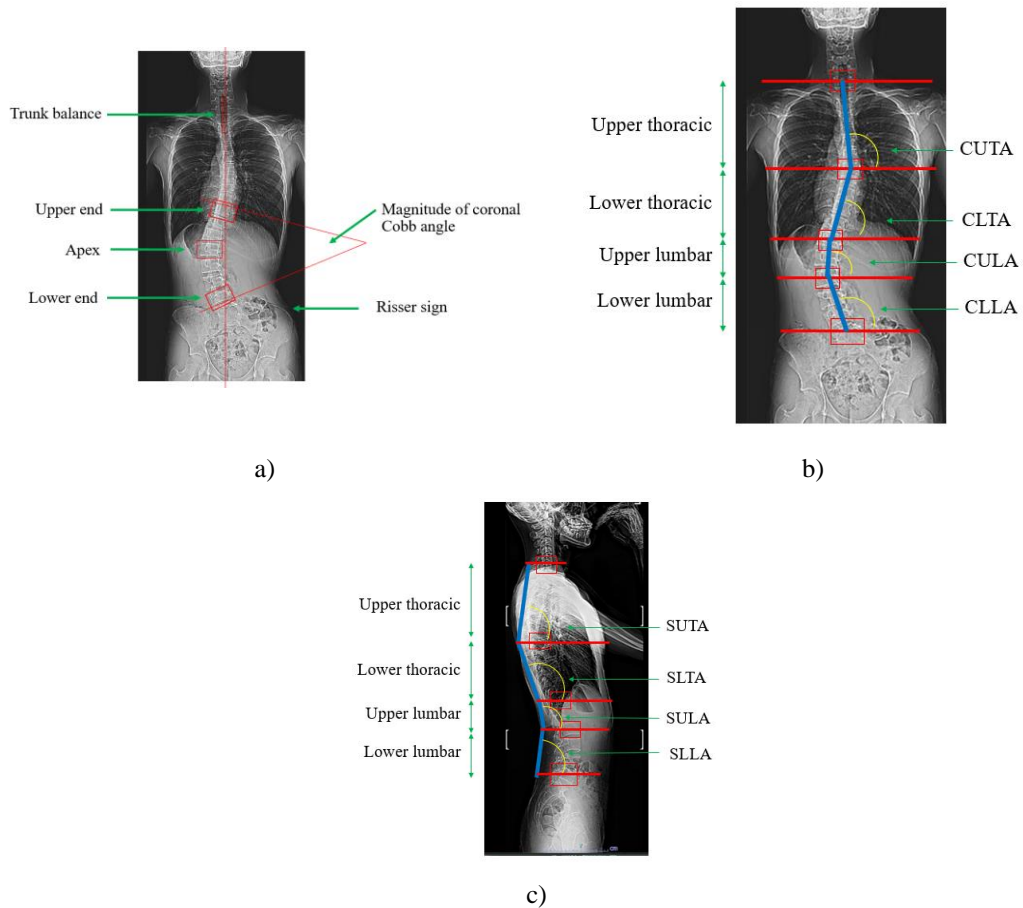


Figure 8.7 Spinal parameters from radiographic evaluation.

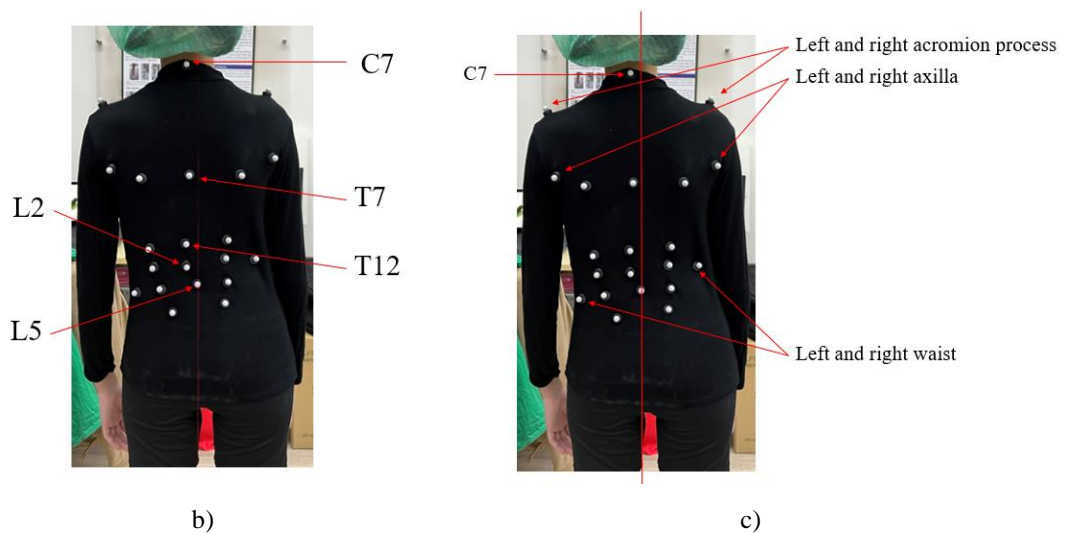
For the physical examination (referred to in Chapter 3), the clinical parameters are as follows.

- Trunk asymmetry: uneven shoulders, uneven axilla, uneven waist
- Trunk balance or coronal decompensation
- Abnormal sagittal curves of the spine: thoracic kyphosis and lumbar lordosis
- Rotational asymmetry of the spine, rib hump and lumbar prominent
- Flexibility of the spinal curves

When using the low-cost postural measurement system, the researcher had to carefully identify the anatomical landmarks before attaching the markers to quantify the spinal parameters. There were 25 markers attached to the participant's trunk. Figure 8.8 illustrates the list and the location of markers. Furthermore, another 14 markers were attached to the wooden stick to represent the force directions (2 markers for each stick).

Marker No.	Anatomical Landmarks	Marker No.	Anatomical Landmarks
1	C7	14	Left to T12
2	T7	15	Right to T12
3	T12	16	Left to L2
4	L2	17	Right to L2
5	L5	18	Left to L5
6	Left acromion process	19	Right to L5
7	Right acromion process	20	Left PSIS
8	Left axilla	21	Right PSIS
9	Right axilla	22	Xyphoid process
10	Left waist	23	Middle distance between Xyphoid and umbilicus
11	Right waist	24	Umbilicus
12	Left inferior angle of scapula	25	Middle distance between ASIS
13	Right inferior angle of scapula		

a)



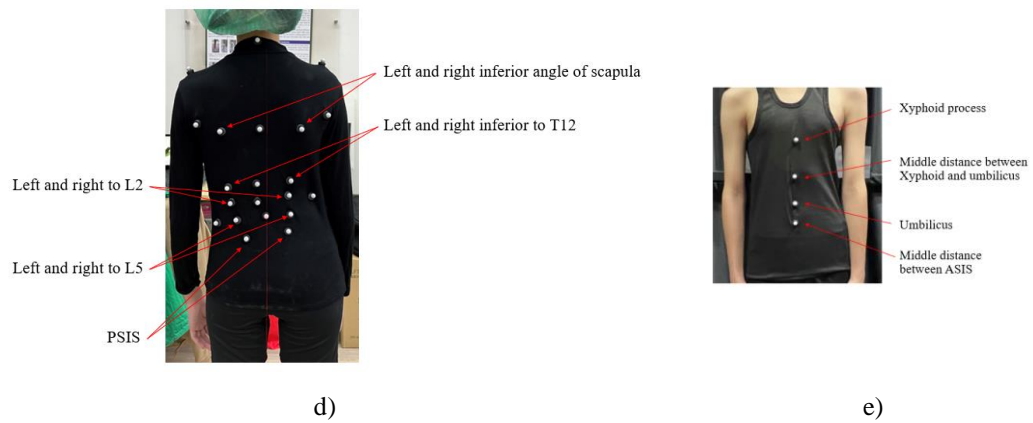


Figure 8.8 Location of markers, where a) the list of marker positions, b) markers for spinal column, c) markers for trunk asymmetry, d) markers for horizontal rotation, and e) markers illustration with the horizontal rotation.

The clinical parameters for the low-cost postural measurement system in the **assessment process** are as follows (spinal parameter calculation referred to in Chapter 3 and Topic 8.5).

- Spinal angles at the sagittal plane: SUTA, SLTA, SULA, and SLLA (fig. 8.7c and 8.8b)
- Spinal angles at the coronal plane: CUTA, CLTA, CULA, and CLLA (fig. 8.7b and 8.8b)
- Spinal angles at 3-dimensions: 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA (fig. 8.8b)
- Coronal decompensation (fig. 8.8c)
- Trunk asymmetry: uneven shoulders, uneven axilla, uneven waist, the distance between left and right sides at the axilla and waist levels (fig. 8.8c)
- Trunk horizontal rotation at T7, T12, L2, and L5 with respect to PSIS (fig. 8.8d and 8.8e)

The clinical parameters for the low-cost postural measurement system in the **optimal correction process** are as follows (spinal parameter calculation referred to in Chapter 3 and Topic 8.5).

- Spinal angles at the sagittal plane: SUTA, SLTA, SULA, and SLLA (fig. 8.7c and 8.8b)



- Spinal angles at the coronal plane: CUTA, CLTA, CULA, and CLLA (fig. 8.7b and 8.8b)
- Spinal angles at 3-dimensions: 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA (fig. 8.8b)
- Coronal decompensation (fig. 8.8c)
- Magnitude of forces to correct the deformity (referred to in Chapter 5)

For the **casting process** result, the clinical parameters for the low-cost postural measurement system were done on the positive cast, and the spinal parameters are listed below (spinal parameter calculation referred to in Chapter 3 and Topic 8.5).

- Spinal angles at the sagittal plane: SUTA, SLTA, SULA, and SLLA (fig. 8.7c and 8.8b)
- Spinal angles at the coronal plane: CUTA, CLTA, CULA, and CLLA (fig. 8.7b and 8.8b)
- Spinal angles at 3-dimensions: 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA (fig. 8.8b)
- Coronal decompensation (fig. 8.8c)
- Trunk asymmetry: uneven shoulders, uneven axilla, uneven waist, the distance between left and right sides at the axilla and waist levels (fig. 8.8c)
- Trunk horizontal rotation at T7, T12, L2, and L5 with respect to PSIS (fig. 8.8d and 8.8e)

Descriptive statistics was used to describe participant and clinical characteristics. Number and percentage were used for qualitative data (e.g., sex, Risser sign, flexibility of curve, and type of curve). Furthermore, quantitative data (e.g., age, spinal angles, menarche status, and BMI) were summarized by the mean and standard deviation (SD) or median and range (min, max), as appropriate. Furthermore, the scatter plot was used to display the relationship between two dependent quantitative data. Pearson correlation coefficient or Spearman rank correlation coefficient analysis was performed to analyse the relationship between two dependent quantitative data, as appropriate. A paired t-test or Wilcoxon signed rank test was also performed to compare the dependent quantitative data. A two-sided test with p-value <0.05 was



considered statistically significant. A paired t-test for equivalence using the Two One-Sided Test (TOST) was also performed to assess the equivalence of the dependent quantitative data. Some statistical analyses were performed using SPSS program (Version 18.0.), and some were performed using NCSS program (Version 2023).

### 8.5. Formula to calculate spinal parameters

The previous section described the spinal outcomes for this thesis. The spinal parameters were measured using radiographs and a postural measurement system. The motion system calculated the spinal parameters using the markers attached to the anatomical landmarks on the participant's trunk. However, the radiographs were used to calculate the spinal parameters by looking at the anatomical bone structure that appeared in the images (referred to in Chapter 3). Therefore, this section further describes a method to calculate spinal parameters in 3 dimensions using marker positions.

Trigonometry theory was the basis for calculating spinal parameters, and each parameter required at least three marker positions. Figure 8.9 illustrates a formula for calculating the unknown value using trigonometry theory, where A, B, and C represent marker points A, B, and C, (a) is the distance between points B and C, (b) is the distance between points A and C, and (c) is the distance between points A and B, respectively. Furthermore, the distances (a) and (b) must be 90° from each other.

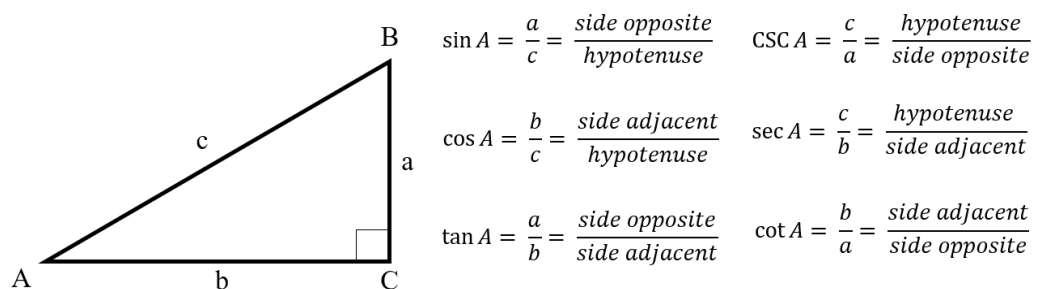


Figure 8.9 Geometry and formula for calculating the trigonometry.

### 8.5.1. 3-Dimensional spinal angles

Begins with the calculation for the **3D spinal angles (3DSA)**. Figure 8.10 illustrates the geometry and marker locations used to calculate these parameters with respect to the horizontal plane. This spinal parameters was previously introduced by Jang (Jang, 2018) by dividing the spinal column into four segments, and we adopted it for this thesis. The 3DSA consisted of the 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA, and 5 markers were used to calculate these parameters. The marker positions consisted of the spinous process of C7, T7, T12, L2, and L5. To clarify, markers C7 and T7 were used for calculating the 3D-UTA, markers T7 and T12 for the 3D-LTA, markers T12 and L2 for the 3D-ULA, and markers L2 and L5 for the 3D-LLA, respectively.

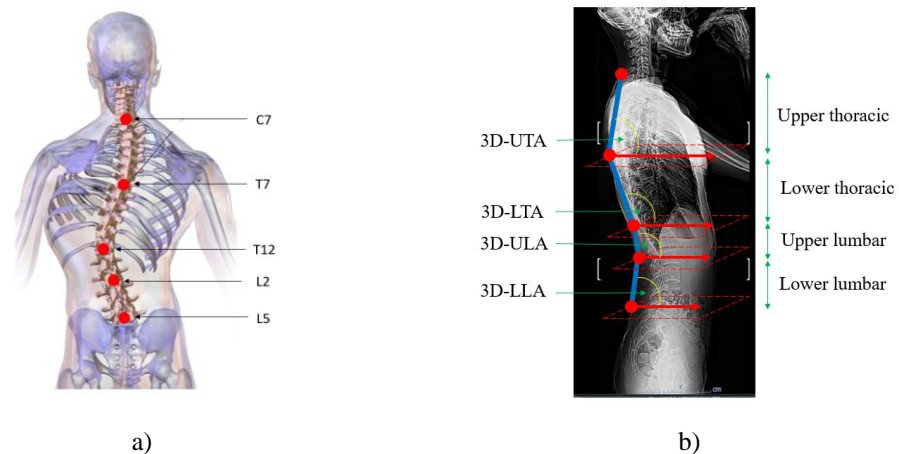


Figure 8.10 Marker locations and the 3D spinal angles, including 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA. These angles are relative to the horizontal plane.

Figure 8.11 illustrates the marker positions and geometry to calculate **3D-UTA** using markers at C7 and T7. Equation 8.4 is a formula to calculate the angle ( $\theta_1$ ) with respect to the horizontal plane calculated from markers 1, 2, and new marker 1. Markers 1 and 2 were the marker positions from the postural measurement system, while the new marker 1 was a new marker position estimating from markers 1 and 2. To clarify, marker 1 was X, Y, and Z of marker 1 or C7. Marker 2 was X, Y, and Z of marker 2 or T7. The new marker 1 was X and Z values from marker 1, and Y value from marker 2. Furthermore, ( $d_{3UT1}$ ) was the distance between markers 1 and 2, ( $d_{3UT2}$ ) was the

distance between marker 2 and the new marker 1, and ( $d_{3UT3}$ ) was the distance between markers 1 and the new marker 1. The distances ( $d_{3UT2}$ ) and ( $d_{3UT3}$ ) had to be  $90^\circ$  from each other. Furthermore, Equations 8.1, 8.2, and 8.3 are formulas to calculate the distances of ( $d_{3UT1}$ ), ( $d_{3UT2}$ ), and ( $d_{3UT3}$ ), respectively.

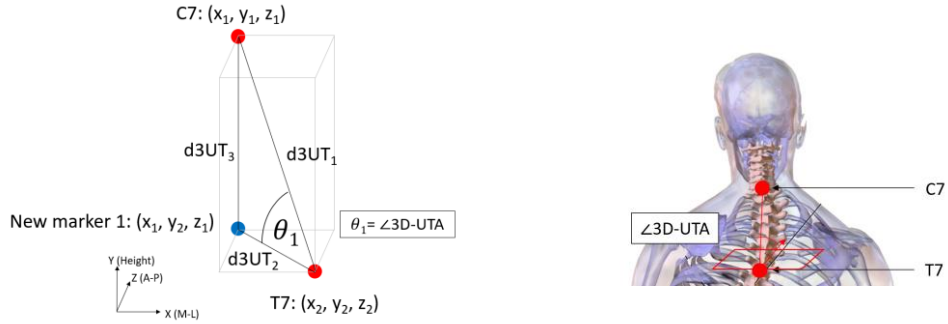


Figure 8.11 Marker positions and geometry to calculate 3D-UTA.

$$d_{3UT_1} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (8.1)$$

$$d_{3UT_2} = \sqrt{(x_1 - x_2)^2 + (z_1 - z_2)^2} \quad (8.2)$$

$$d_{3UT_3} = |y_1 - y_2| \quad (8.3)$$

$$\theta_1 = \cos^{-1} \frac{d_{3UT_2}}{d_{3UT_1}} \quad (8.4)$$

$$\text{if } z_1 < z_2, \angle 3DUTA = 180^\circ - \theta_1$$

Figure 8.12 illustrates the marker positions and geometry to calculate **3D-LTA** using markers at T7 and T12. The method to calculate this angle was similar to the calculation described above. Equation 8.5 is a formula to calculate this angle.



Figure 8.12 Marker positions and geometry to calculate 3D-LTA.

$$\begin{aligned}
d3LT_1 &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2} \\
d3LT_2 &= \sqrt{(x_2 - x_3)^2 + (z_2 - z_3)^2} \\
d3LT_3 &= |y_2 - y_3| \\
\theta_2 &= \cos^{-1} \frac{d3LT_2}{d3LT_1} \\
\text{if } z_2 < z_3, \angle 3DLTA &= 180^\circ - \theta_2
\end{aligned}
\tag{8.5}$$

Figure 8.13 illustrates the marker positions and geometry to calculate **3D-ULA** using markers at T12 and L2, and Equation 8.6 is a formula to calculate this angle.

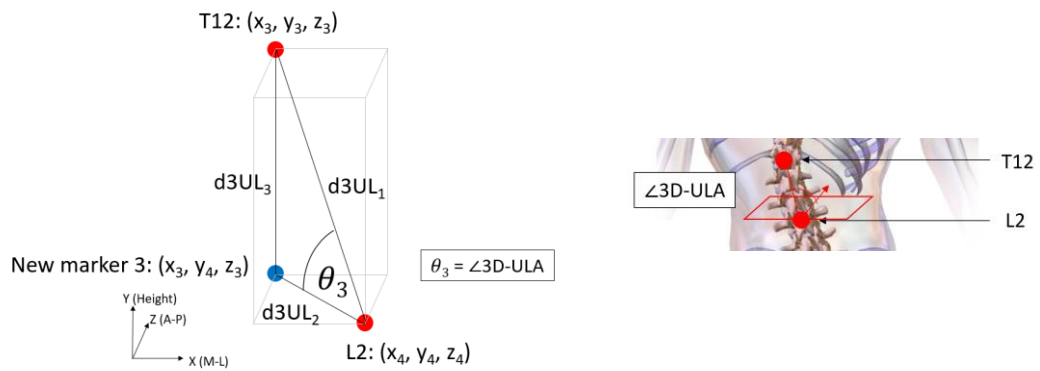


Figure 8.13 Marker positions and geometry to calculate 3D-ULA.

$$\begin{aligned}
d3UL_1 &= \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2 + (z_3 - z_4)^2} \\
d3UL_2 &= \sqrt{(x_3 - x_4)^2 + (z_3 - z_4)^2} \\
d3UL_3 &= |y_3 - y_4| \\
\theta_3 &= \cos^{-1} \frac{d3UL_2}{d3UL_1} \\
\text{if } z_3 < z_4, \angle 3DULA &= 180^\circ - \theta_3
\end{aligned}
\tag{8.6}$$

Figure 8.14 illustrates the marker positions and geometry to calculate **3D-LLA** using markers at L2 and L5, and Equation 8.7 is a formula to calculate this angle.

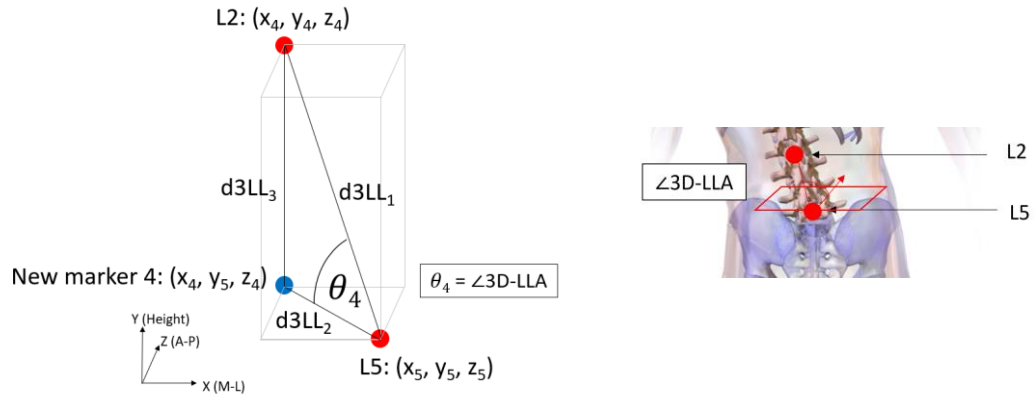


Figure 8.14 Marker positions and geometry to calculate 3D-LLA.

$$\begin{aligned}
 d3LL_1 &= \sqrt{(x_4 - x_5)^2 + (y_4 - y_5)^2 + (z_4 - z_5)^2} \\
 d3LL_2 &= \sqrt{(x_4 - x_5)^2 + (z_4 - z_5)^2} \\
 d3LL_3 &= |y_4 - y_5| \\
 \theta_4 &= \cos^{-1} \frac{d3LL_2}{d3LL_1} \\
 \text{if } z_4 < z_5, \angle 3D-LLA &= 180^\circ - \theta_4
 \end{aligned} \tag{8.7}$$

### 8.5.2. Coronal spinal angles

For the coronal spinal angle (Márkus *et al.*), there were 4 angles needed to be calculated, including CUTA, CLTA, CULA, and CLLA. This spinal parameters was also introduced by Jang (Jang, 2018), and we adopted it for this thesis. Figure 8.15 illustrates the marker positions and geometry to calculate **CUTA** using markers at C7 and T7. Equation 8.8 is a formula to calculate the angle ( $\theta_{1c}$ ) with respect to the horizontal plane calculating from makers 1, 2, and new marker 1c. The new marker 1c was a new marker position estimated from markers 1 and 2, the Z value from marker 1, and the X and Y values from marker 2. Furthermore, ( $d_{CUT1}$ ) was the distance between markers 1 and new marker 1c, ( $d_{CUT2}$ ) was the distance between new marker 1 and the new marker 1c, and ( $d_{CUT3}$ ) was the distance between marker 1 and the new marker 1.

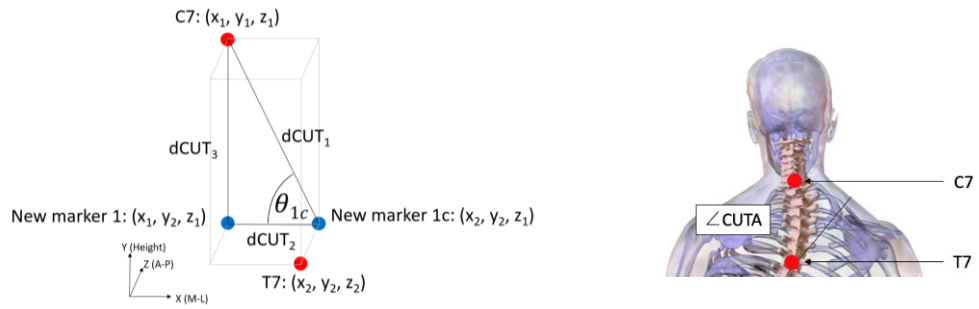


Figure 8.15 Marker positions and geometry to calculate CUTA.

$$\begin{aligned}
 dCUT_1 &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\
 dCUT_2 &= |x_1 - x_2| \\
 dCUT_3 &= |y_1 - y_2| \\
 \theta_{1c} &= \cos^{-1} \frac{dCUT_2}{dCUT_1} \\
 \text{if } x_1 < x_2, \angle CUTA &= 180^\circ - \theta_{1c}
 \end{aligned} \tag{8.8}$$

Figure 8.16 illustrates the marker positions and geometry to calculate **CLTA** using markers at T7 and T12. The method to calculate this angle was similar to the calculation described above. Equation 8.9 is a formula to calculate this angle.

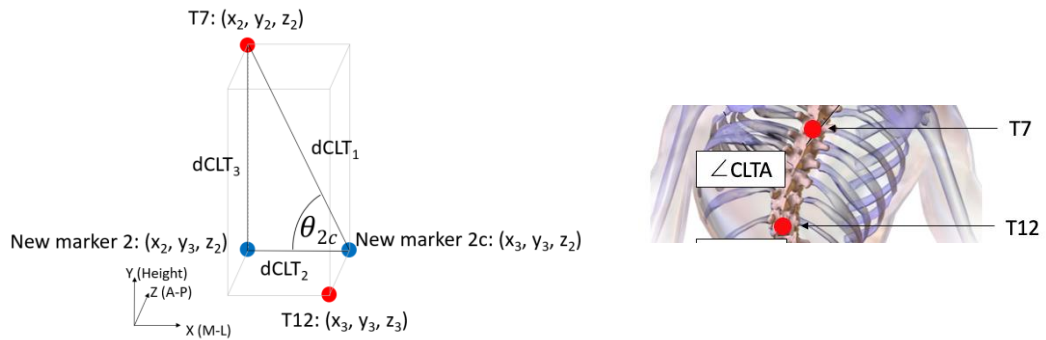


Figure 8.16 Marker positions and geometry to calculate CLTA.

$$\begin{aligned}
 dCLT_1 &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\
 dCLT_2 &= |x_2 - x_3| \\
 dCLT_3 &= |y_2 - y_3| \\
 \theta_{2c} &= \cos^{-1} \frac{dCLT_2}{dCLT_1} \\
 \text{if } x_2 < x_3, \angle CLTA &= 180^\circ - \theta_{2c}
 \end{aligned} \tag{8.9}$$

Figure 8.17 illustrates the marker positions and geometry to calculate **CULA** using markers at T12 and L2, and Equation 8.10 is a formula to calculate this angle.

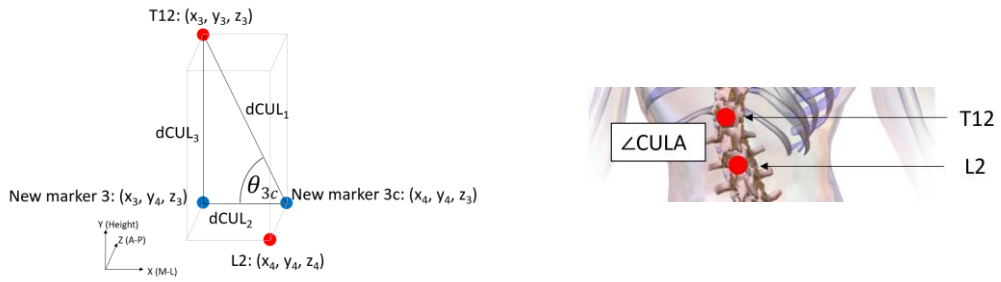


Figure 8.17 Marker positions and geometry to calculate CULA.

$$\begin{aligned}
 dCUL_1 &= \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2} \\
 dCUL_2 &= |x_3 - x_4| \\
 dCUL_3 &= |y_3 - y_4| \\
 \theta_{3c} &= \cos^{-1} \frac{dCUL_2}{dCUL_1} \\
 \text{if } x_3 < x_4, \angle CULA &= 180^\circ - \theta_{3c}
 \end{aligned}
 \tag{8.10}$$

Figure 8.18 illustrates the marker positions and geometry to calculate **CLLA** using markers at L2 and L5, and Equation 8.11 is a formula to calculate this angle.

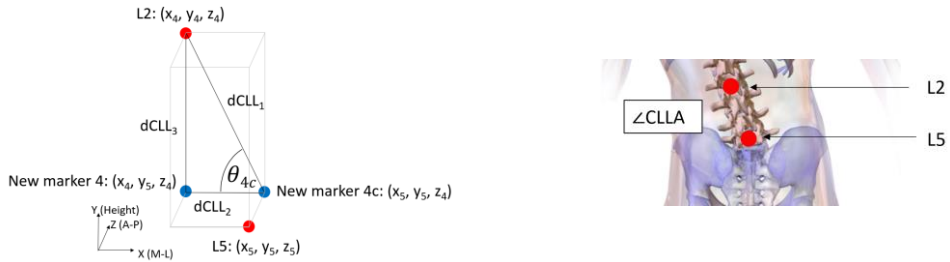


Figure 8.18 Marker positions and geometry to calculate CLLA.

$$\begin{aligned}
 dCLL_1 &= \sqrt{(x_4 - x_5)^2 + (y_4 - y_5)^2} \\
 dCLL_2 &= |x_4 - x_5| \\
 dCLL_3 &= |y_4 - y_5| \\
 \theta_{4c} &= \cos^{-1} \frac{dCLL_2}{dCLL_1} \\
 \text{if } x_4 < x_5, \angle CLLA &= 180^\circ - \theta_{4c}
 \end{aligned}
 \tag{8.11}$$

### 8.5.3. Sagittal spinal angles

For the sagittal spinal angles (SSA), there were 4 angles needed to be calculated, including SUTA, SLTA, SULA, and SLLA. This spinal parameters was also introduced by Jang (Jang, 2018), and we adopted it for this thesis. Figure 8.19 illustrates the marker positions and geometry to calculate **SUTA** using markers at C7 and T7. Equation 8.12 is a formula to calculate the angle ( $\theta_{1s}$ ) with respect to the horizontal plane calculating from makers 1, new marker 1, and new marker 1s. The new marker 1s was a new marker position estimated from markers 1 and 2, the X value from marker 1, and Y and Z values from marker 2. Furthermore, ( $d_{SUT1}$ ) was the distance between markers 1 and new marker 1s, ( $d_{SUT2}$ ) was the distance between new marker 1 and the new marker 1s, and ( $d_{SUT3}$ ) was the distance between marker 1 and the new marker 1, respectively.

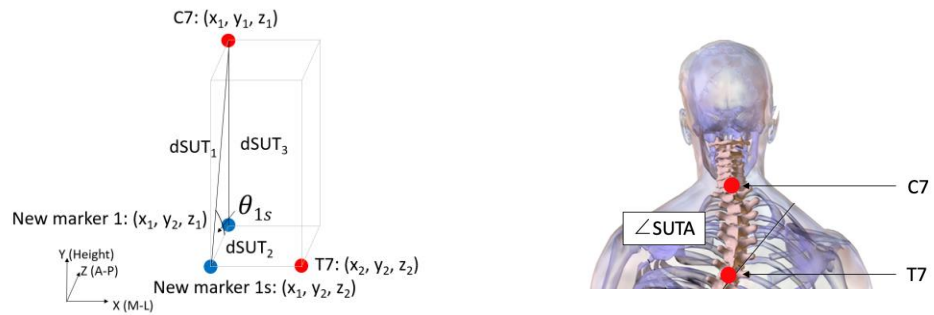


Figure 8.19 Marker positions and geometry to calculate SUTA.

$$\begin{aligned}
 dSUT_1 &= \sqrt{(y_1 - y_2)^2 + (z_1 - z_2)^2} \\
 dSUT_2 &= |z_1 - z_2| \\
 dSUT_3 &= |y_1 - y_2| \\
 \theta_{1s} &= \cos^{-1} \frac{dSUT_2}{dSUT_1} \\
 \text{if } z_1 < z_2, \angle SUTA &= 180^\circ - \theta_{1s}
 \end{aligned}
 \tag{8.12}$$

Figure 8.20 illustrates the marker positions and geometry to calculate **SLTA** using markers at T7 and T12. The method to calculate this angle was similar to the calculation described above. Equation 8.13 is a formula to calculate this angle.



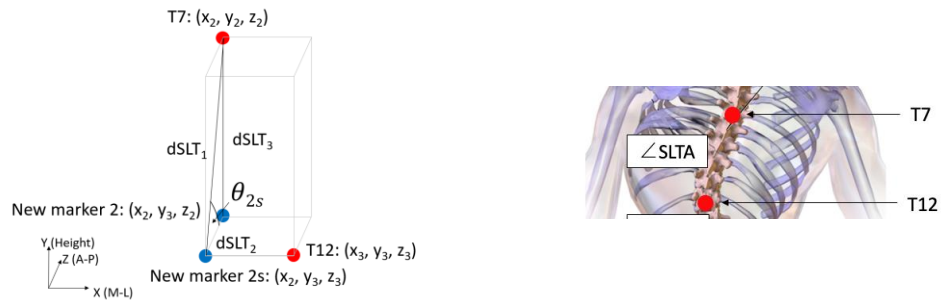


Figure 8.20 Marker positions and geometry to calculate SLTA.

$$\begin{aligned}
 dSLT_1 &= \sqrt{(y_2 - y_3)^2 + (z_2 - z_3)^2} \\
 dSLT_2 &= |z_2 - z_3| \\
 dSLT_3 &= |y_2 - y_3| \\
 \theta_{2s} &= \cos^{-1} \frac{dSLT_2}{dSLT_1} \\
 \text{if } z_2 < z_3, \angle SLTA &= 180^\circ - \theta_{2s}
 \end{aligned} \tag{8.13}$$

Figure 8.21 illustrates the marker positions and geometry to calculate **SULA** using markers at T12 and L2, and Equation 8.14 is a formula to calculate this angle.

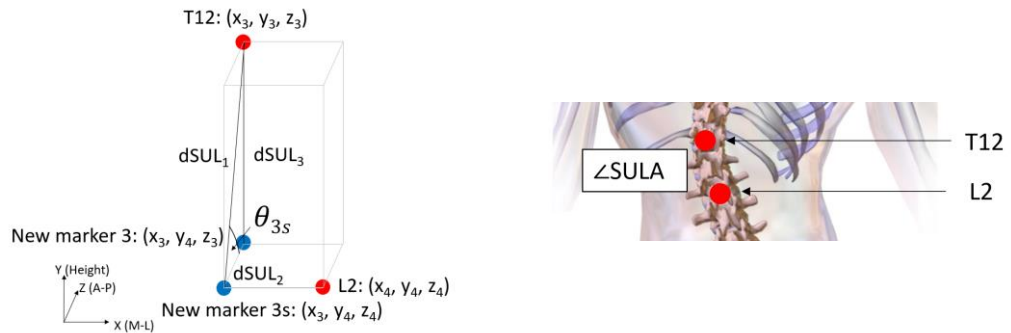


Figure 8.21 Marker positions and geometry to calculate SULA.

$$\begin{aligned}
 dSUL_1 &= \sqrt{(y_3 - y_4)^2 + (z_3 - z_4)^2} \\
 dSUL_2 &= |z_3 - z_4| \\
 dSUL_3 &= |y_3 - y_4| \\
 \theta_{3s} &= \cos^{-1} \frac{dSUL_2}{dSUL_1} \\
 \text{if } z_3 < z_4, \angle SULA &= 180^\circ - \theta_{3s}
 \end{aligned} \tag{8.14}$$

Figure 8.22 illustrates the marker positions and geometry to calculate **SLLA** using markers at L2 and L5, and Equation 8.15 is a formula to calculate this angle.

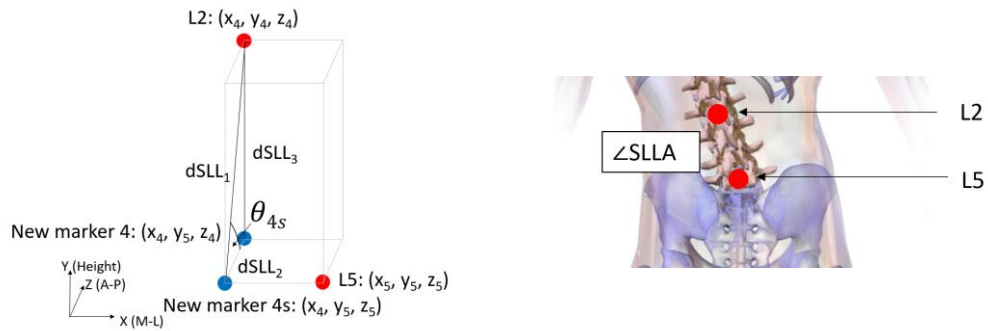


Figure 8.22 Marker positions and geometry to calculate SLLA.

$$\begin{aligned}
 dSLL_1 &= \sqrt{(y_4 - y_5)^2 + (z_4 - z_5)^2} \\
 dSLL_2 &= |z_4 - z_5| \\
 dSLL_3 &= |y_4 - y_5| \\
 \theta_{4s} &= \cos^{-1} \frac{dSLL_2}{dSLL_1} \\
 \text{if } z_4 < z_5, \angle SLLA &= 180^\circ - \theta_{4s}
 \end{aligned} \tag{8.15}$$

#### 8.5.4. Reduction percentage of the apical translation

In the radiographs, the reduction percentage of the coronal spinal angles was calculated to see the change in the spinal curve during the outcome measurement. The method used the coronal Cobb angle magnitude of the with-out brace compared with the coronal Cobb angle magnitude of the with-in brace. The coronal Cobb angle reduction percentage can be calculated using Equation 8.16, where (A) is the coronal Cobb angle magnitude of the with-out brace and (B) is the coronal Cobb angle magnitude of the with-in brace.

$$\text{Reduction \% of Cobb angle} = \left[ \frac{(A) - (B)}{(A)} \right] \times 100\% \tag{8.16}$$

However, we could not use this method to calculate the result of this experiment because it used a different method to quantify the spinal parameter. Therefore, we

proposed another method to calculate “**the reduction percentage of the apical translation**” by looking at the difference in the distance in the X-direction between the L5 marker and C7, T7, T12, and L2 markers. Figure 8.23 illustrates the geometry to calculate the reduction percentage of the apical translation. Equations 8.17 and 8.18 are formulas to calculate this parameter during assessment, optimal correction, or casting, where (d1), (d2), (d3), and (d4) are the X-distances from C7 to L5, T7 to L5, T12 to L5, and L2 to L5, respectively. The longer distance indicated the coronal spinal column's greater deviation from the centre. In contrast, the shorter distance indicated that the coronal spinal column was closer to the centre. Furthermore, the summation of distance could be used to compare the deviation and calculate the reduction percentage of the apical translation. Equation 8.19 is a formula to calculate this parameter, where (A) is the summary of X-distance in the assessment process and (B) is the summary of X-distance in optimal correction or casting processes.

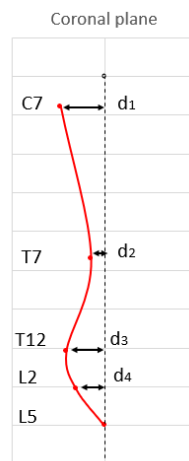


Figure 8.23 Geometry to calculate the reduction percentage of the apical translation.

$$(A) = |d_1| + |d_2| + |d_3| + |d_4| \text{ of Assessment} \quad 8.17$$

$$(B) = |d_1| + |d_2| + |d_3| + |d_4| \text{ of Optimal correction or Casting} \quad 8.18$$

$$\% \text{ reduction of the apical translation} = \left[ \frac{(A) - (B)}{(A)} \right] \times 100\% \quad 8.19$$

### 8.5.5. Trunk balance and trunk asymmetry

For the asymmetry of the trunk, there were 6 parameters calculated, including trunk balance, shoulder level, axilla level, waist level, distance of left and right side at axilla and distance of left and right side at waist. Clinicians usually evaluate these spinal parameters during the physical assessment to observe trunk balance and trunk asymmetry.

Figure 8.24 illustrates the marker positions and geometry to calculate **trunk balance or coronal decompensation** using markers at C7 and L5, and Equation 8.20 is a formula to calculate this parameter. ( $x_1$ ) is the X value from marker C7, and ( $x_5$ ) is the X value from marker L5. If the result equalled zero, the trunk was in balance or had no coronal decompensation. If the result was negative (-), the trunk was leaning to the left side. In contrast, the positive (+) result means the trunk was leaning to the right side. The trunk balance in the coronal plane is a primary parameter clinicians used to observe the trunk as a whole spine. While the patients are wearing the spinal orthosis, we expect that the patients should not present coronal decompensation, or this distance should be close to zero as much as possible to keep the whole spine in balance.

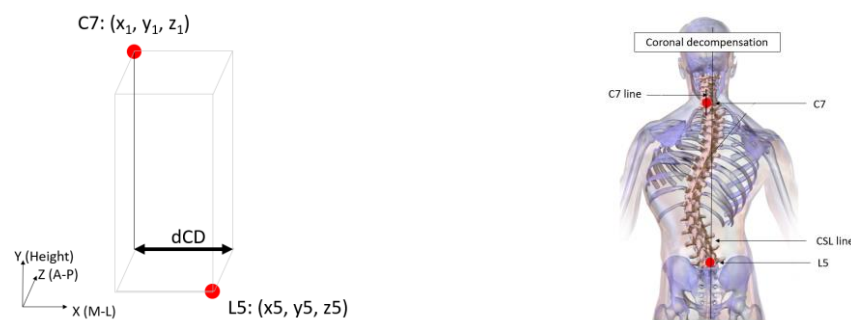


Figure 8.24 Marker positions and geometry to calculate trunk balance.

$$dCD = x_1 - x_5 \quad (8.20)$$

Figure 8.25 illustrates the marker positions and geometry to calculate **shoulder level** using markers at left and right acromion processes, and Equation 8.21 is a formula to calculate this parameter. ( $y_6$ ) was the Y value from marker at left acromion process, and ( $y_7$ ) was the Y value from marker at right acromion process. If the result equalled

zero, the shoulder was at the same level. If the result was negative (-), the left shoulder was higher than the right side. In contrast, the positive (+) result means the right shoulder was higher than the left side.

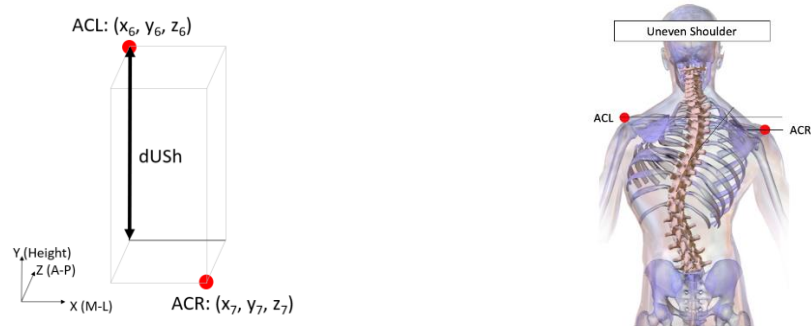


Figure 8.25 Marker positions and geometry to calculate shoulder level.

$$dUSh = y_7 - y_6 \quad (8.21)$$

Figure 8.26 illustrates the marker positions and geometry to calculate **axilla level** using markers at left and right axilla, and Equation 8.22 is a formula to calculate this parameter. ( $y_8$ ) was the Y value from marker at left axilla, and ( $y_9$ ) was the Y value from marker at right axilla. If the result equalled zero, the axilla was at the same level. If the result was negative (-), the left axilla was higher than the right side. In contrast, the positive (+) result means the right axilla was higher than the left side.

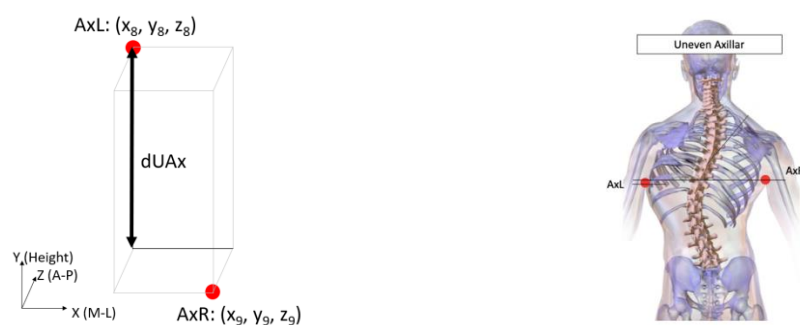


Figure 8.26 Marker positions and geometry to calculate axilla level.

$$dUAx = y_9 - y_8 \quad (8.22)$$

Figure 8.27 illustrates the marker positions and geometry to calculate **waist level** using markers at left and right waist, and Equation 8.23 is a formula to calculate this angle. ( $y_{10}$ ) was the Y value from marker at left waist, and ( $y_{11}$ ) was the Y value from marker at right waist. If the result equalled zero, the waist was at the same level. If the result was negative (-), the left waist was higher than the right side. In contrast, the positive (+) result means the right waist was higher than the left side.

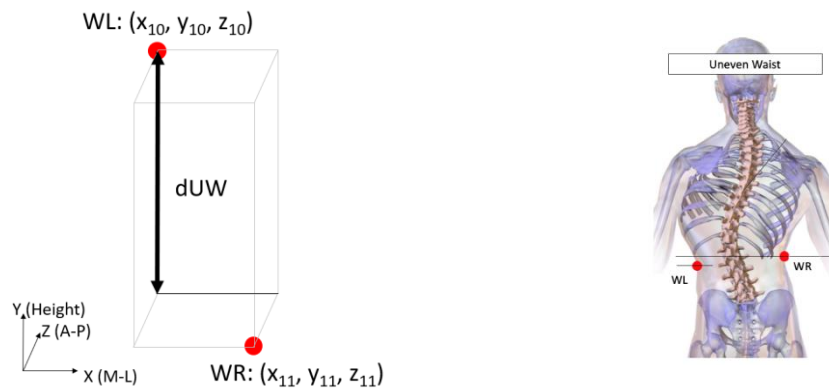


Figure 8.27 Marker positions and geometry to calculate waist level.

$$dUW = y_{11} - y_{10} \quad (8.23)$$

Figure 8.28 illustrates the marker positions and geometry to calculate **the distance between the left and right axilla** using markers at left axilla, right axilla, and L5. Equation 8.24 is a formula to calculate this parameter. Firstly, the distance of the right axilla was calculated ( $d_{DAxR}$ ), where ( $x_9$ ) was the X value from the marker at the right axilla, and ( $x_5$ ) was the X value from the marker at L5. The distance of the left axilla was then calculated ( $d_{DAxL}$ ), where ( $x_8$ ) was the X value from the marker at the left axilla, and ( $x_5$ ) was the X value from the marker at L5. Then, the distance difference ( $dDAx$ ) was calculated. If the result equalled zero, the distance from the left axilla to the CSL line and the right axilla to the CSL line would be the same. If the result was negative (-), the left axilla distance was longer than the right side. In contrast, the positive (+) result means the right axilla distance was longer than the left side.

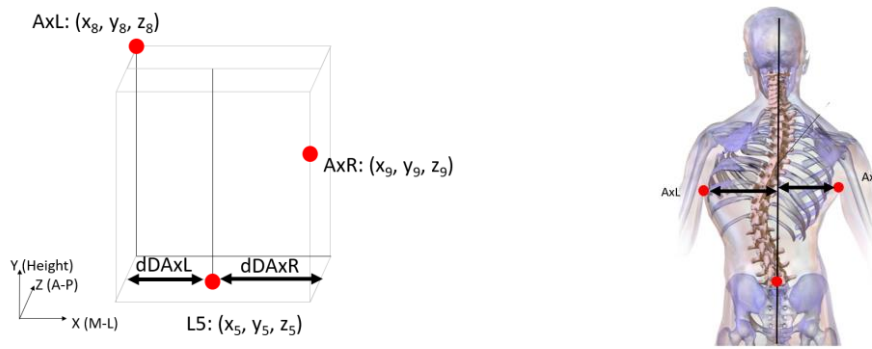


Figure 8.28 Marker positions and geometry to calculate the distance of left and right side at axilla.

$$\begin{aligned}
 d_{DAxR} &= |x_9 - x_5| \\
 d_{DAxL} &= |x_8 - x_5| \\
 dDAx &= d_{DAxR} - d_{DAxL}
 \end{aligned}
 \tag{8.24}$$

Figure 8.29 illustrates the marker positions and geometry to calculate the **distance between the left and right waist** using markers at the left waist, right waist, and L5. Equation 8.25 is a formula to calculate this parameter. Like the previous calculation, the distance of the right waist was calculated ( $d_{DWR}$ ), where ( $x_{11}$ ) was the X value from the marker at the right waist, and ( $x_5$ ) was the X value from the marker at L5. The distance of the left waist was then calculated ( $d_{DWL}$ ), where ( $x_{10}$ ) was the X value from the marker at the left waist, and ( $x_5$ ) was the X value from the marker at L5. Then, the distance difference ( $dDW$ ) was calculated. If the result equalled zero, the distance from the left waist to the CSL line and the right waist to the CSL line would be the same. If the result was negative (-), the left waist distance was longer than the right side. In contrast, the positive (+) result means the right waist distance was longer than the left side.

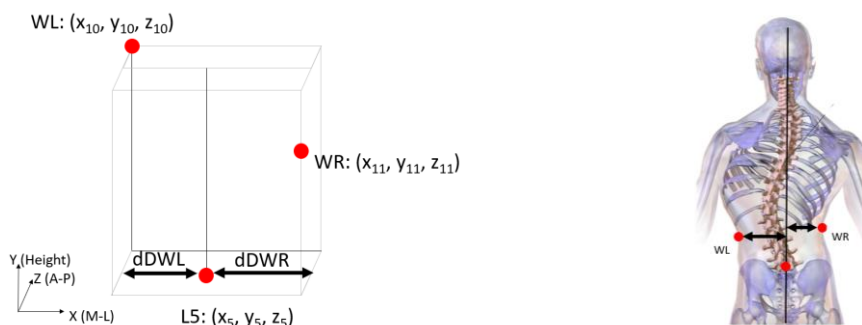


Figure 8.29 Marker positions and geometry to calculate the distance of left and right side at waist.

$$\begin{aligned}
d_{DWR} &= |x_{11} - x_5| \\
d_{DWL} &= |x_5 - x_{10}| \\
dDW &= d_{DWR} - d_{DWL}
\end{aligned}
\tag{8.25}$$

### 8.5.6. The POsterior Trunk Symmetry Index (POTSI)

Clinicians usually assess the physical appearance of scoliosis patients and evaluate the asymmetry of the trunk in the coronal plane by looking at the trunk balance, the shoulder level, the axillar level and the waist level. Only the trunk balance commonly measures the actual value using a steel ruler or tape measure. In contrast, other parameters only describe the characteristic, such as the change in the same level or not the same level and which side is higher than the other.

Suzuki and colleagues introduced a method to calculate the asymmetry of the trunk in the coronal plane, called the “**POsterior Trunk Symmetry Index (POTSI)**” (Suzuki *et al.*, 1999, Inami *et al.*, 1999). Figure 8.30 illustrates how to calculate the POTSI index (Suzuki *et al.*, 1999, Inami *et al.*, 1999). Nine values had to be measured on the patient's trunk, including the X-distance between C7 and L5 (i), the X-distance between the left (c) and right (d) at the axilla level, the left (a) and right (b) at the waist level, the Y-distance between the left and right shoulder (h), axilla (g), and waist (f), and spine length (e). The method to measure nine values was already described in the previous topic, Topic 8.5.5.

After that, the values were subscribed into the formula to calculate 6 indices (the formula described in fig, 8.30 and Equation 8.26). The summary of 6 indices was the final result of POTSI. If the POTSI index was equal to or less than 10, it indicated normal or trunk symmetry. In contrast, if the POTSI index was greater than 10, it indicated the spinal pathology or asymmetry of the trunk in the coronal plane. The POTSI were calculated and compared between assessment and casting processes.



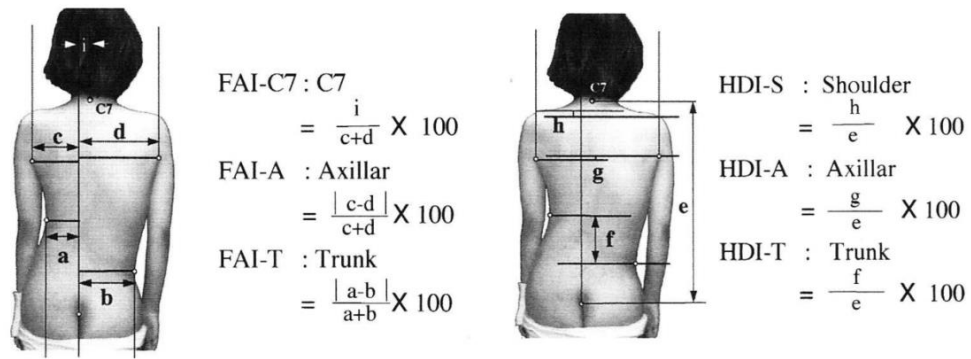


Figure 8.30 Geometry and formula to calculate the POsterior Trunk Symmetry Index (POTSI).

$$POTSI = (FAI - C7) + (FAI - A) + (FAI - T) + (HDI - S) + (HDI - A) + (HDI - T) \quad 8.26$$

### 8.5.7. Angles of the horizontal trunk rotation

For the horizontal rotation of trunk, there were 4 parameters needed to be calculated, including horizontal rotation of trunk at T7, T12, L2, and L5 levels with respect to PSIS level. Previous research from Jang (Jang, 2018) studied horizontal rotation at interior angle of the scapula (T7 level) and paraspinal muscles around the L3/L4 area. This thesis continued to use this parameter at T7 (Interior angle of the scapula) and added more levels to cover the trunk shape. Figures 8.31 and 8.32 illustrate the marker positions and geometry to calculate **horizontal trunk rotation at T7 with respect to PSIS level** using markers at left and right inferior angle of scapula and PSIS. Equations 8.27 and 8.28 are the formulas to calculate this angle.

Firstly, the PSIS rotation in the horizontal plane was calculated. Vector  $\vec{c}$  was a vector from the left ( $x_1, z_1$ ) to the right ( $x_2, z_2$ ) PSIS, and vector  $\vec{a}$  was a vector from the left PSIS ( $x_1, z_1$ ) to point ( $x_2, z_1$ ). Equation 8.27 is a formula to calculate the angle ( $\theta_1$ ) from 2 vectors. It notes that if the z-value of the right PSIS was lesser than the left PSIS, the angle was rotated in a clockwise direction. In contrast, if the z-value of the right PSIS was greater than the left PSIS, the angle was rotated in an anticlockwise direction.

In the following step, the horizontal trunk rotations at other levels were calculated similarly to the calculation from the PSIS level. For example, the vector  $\vec{c}$  of the T7 level was a vector from the left ( $x_6, z_6$ ) to the right ( $x_7, z_7$ ) inferior angle of the scapula,

and the vector  $\vec{a}$  was a vector from the left inferior angle of scapula ( $x_6, z_6$ ) to point ( $x_7, z_6$ ). The angle ( $\theta_2$ ) was the calculated from 2 vectors (Equation 8.27).

Then, the angle difference ( $\Delta\theta$ ) was calculated using Equation 8.28, where ( $\theta_1$ ) was the horizontal angle from PSIS and the ( $\theta_2$ ) from T7 level, and so on. If the final angle ( $\Delta\theta$ ) was in the positive (+) value, that level was rotated in a clockwise direction. In contrast, if the angle was in the negative (-) value, that level was rotated in an anticlockwise direction.



Left and right markers at T7

Figure 8.31 Marker locations at PSIS and left and right markers at T7

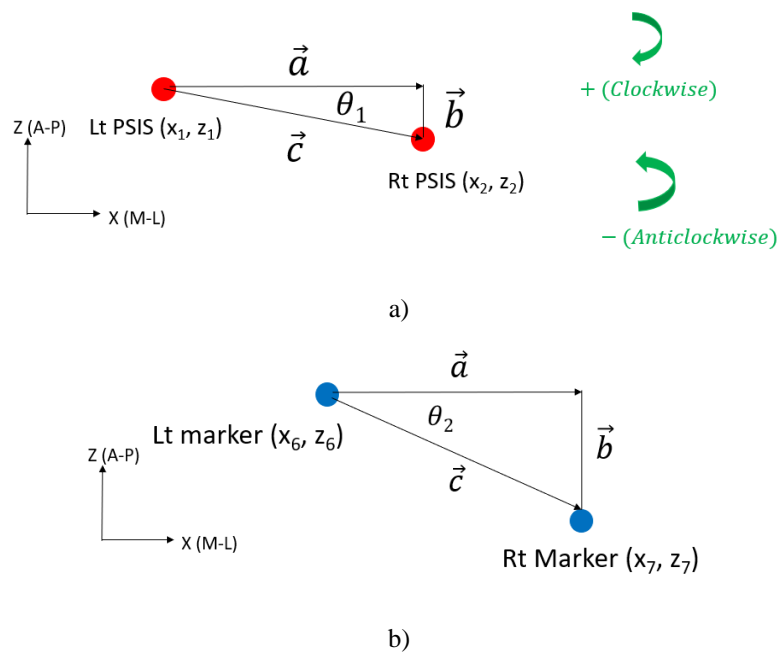


Figure 8.32 Geometry to calculate the horizontal rotation of the trunk at T7 with respect to the PSIS level, where a) is the geometry for PSIS level and b) is the geometry for markers at T7 level.

$$\theta_1 = \cos^{-1} \frac{\vec{a}}{c} \qquad \theta_2 = \cos^{-1} \frac{\vec{a}}{c} \qquad (8.27)$$

$$\Delta\theta = \theta_2 - \theta_1 \qquad (8.28)$$

### 8.5.8. Total of the horizontal trunk rotation and reduction angle of the horizontal trunk rotation

We proposed another method to calculate “**the total horizontal trunk rotation**” using the summary of the rotation angle at all levels. The calculation was performed and compared between the assessment and casting processes. Equation 8.29 is a formula to calculate this parameter, where  $(\theta_1)$ ,  $(\theta_2)$ ,  $(\theta_3)$ , and  $(\theta_4)$ , are the angle of horizontal trunk rotation at T7, T12, L2, and L5, respectively. It notes that if the total angle was in the positive (+) value, the overall trunk rotated in a clockwise direction. In contrast, if the total angle was negative (-), the overall trunk rotated in an anticlockwise direction.

Furthermore, “**the reduction angle of horizontal trunk rotation**” could be calculated by looking the change from assessment to casting process. Equation 8.30 is used to calculate this parameter, where  $(\theta_{assessment})$  is the total of horizontal trunk rotation angle at assessment process and  $(\theta_{casting})$  is the total of horizontal trunk rotation angle at casting process.

$$Total\ of\ horizontal\ trunk\ rotation\ angle = (\theta_1) + (\theta_2) + (\theta_3) + (\theta_4) \qquad 8.29$$

$$|Reduction\ angle| = |(\theta_{assessment}) - (\theta_{casting})| \qquad 8.30$$

## 8.6. Conclusion

This chapter described how to apply the developed system to treat scoliosis patients in three dimensions and study its effectiveness following clinical scientific methods.

The project had already been approved by the ethical committees from the University of Strathclyde and Mahidol University. The developed system was installed in the research room at Sirindhorn School of Prosthetics and Orthotics (SSPO), and the data collection was conducted there. Participants were the outpatients who usually came to get treatment at SSPO. Patients were recruited by 3 methods, including poster invitation, researcher invitation, and clinician invitation. The scoliosis patients interested in the project and following the inclusion criteria had to sign the consent form before starting the project. Participants had to come to SSPO for two appointments. Participants did not need to take any responsibility for payment. However, they got 800 bahts for transportation for each visitation.

In the first appointment, the researcher explained the details of the research project to the participants again. Participants could decide to join the project on that day or the following day or reject it without affecting the current or future treatment. If the participants agreed to join the project, they had to sign the consent form. The researcher checked the current out-brace radiographs. If the radiographs were over three months old, the participants had to go to Siriraj Hospital to obtain the X-rays. Before the participants returned home, the second appointment date had to be confirmed with the participants.

In the second appointment, the researcher again explained the data collection process to the participants. The researcher then performed a radiographic evaluation and a physical assessment. Participants were asked to change into a black bodysuit. The researcher carefully identified the anatomical landmarks and then attached the marker base to the landmarks and the markers to the base. Participants then stood inside the camera frame for 5 minutes, and the researcher ran the "getimagepairs program" to collect the data (Images of markers). Participants rested for 10 to 15 minutes outside the camera frame. At the same time, the researcher moved the scoliosis casting apparatus inside the camera frame. The researcher then removed markers but kept only those representing the spinal column. If participants were ready, they were asked to stand inside the casting frame and then perch on the sitting bar. The researcher adjusted manipulators to apply the 3-dimensional forces to correct the spine. After reaching the optimal alignment, the researcher ran the programs to collect the data, including the "force-measuring program" and the "getimagepairs program". After that,

the researcher performed casting when the participants were under forced correction. The researcher created a line mark on the manipulators before releasing them, allowing participants to move outside the frame. The researcher then wrapped the plastic on the trunk before applying the plaster bandages during the casting process. After the researcher wrapped the plaster bandages, participants were asked to return to the casting apparatus and perch the sitting bar, similar to the previous step. The researcher re-adjusted the manipulators to the previous position and rechecked the force using the "force-measuring program". The negative cast was then removed after the plaster had completely set. The participants had to be cleaned immediately, and they could change their clothes and return home. The negative cast was filled by the plaster. Finally, the researcher attached markers to the positive cast and ran the "getimagepairs program" to calculate the data.

The spinal parameters were then calculated from all marker positions. The researcher entered the marker positions in the precalculated excel spreadsheet, which had already embedded the formula in the file. All parameters were calculated and summarised in the file, and some parameters were illustrated as a graph to see the spine change, including the spinal column, trunk asymmetry, and direction of forces. For the data analysis, the normal distribution of data was checked before analysis. The appropriate statistics type was then selected to match the results. The data interpretation, discussion, limitation, recommendation, and conclusion were reported at the end of the study and will be described in the following chapter.

**9. Chapter 9 Evaluation of The Developed System to Treat Adolescent Idiopathic Scoliosis During Assessment, Optimal Correction, and Casting**

## 9.1. Introduction

The successfully developed system in the previous step needs to be evaluated for its effectiveness in treating adolescent idiopathic scoliosis (AIS) patients. Before the project began, the related documents were submitted to the ethics committees at the University of Strathclyde and Mahidol University for review before approval of the project. Both universities approved the project (see the approval documents in the Appendix 8 and 9).

The data collection was performed in the research room at SSPO. Researchers provided detailed project information to participants, and participants who agreed to join the project had to sign the consent form before collecting the data. Ten scoliosis patients were included in the study following the inclusion criteria.

Descriptive statistics were used to describe participant characteristics. For qualitative data (Gender, Risser sign grading, curve flexibility, and curve type), the results were reported by the number of participants with a percentage. For quantitative data (age, coronal Cobb angle, BMI, and menarche status), the results were reported by mean and standard deviation or median and range, as appropriate.

Table 9.1 shows the participant characteristics of the ten AIS patients. The mean age of participants was 16.04 years (SD = 1.49). Most of them were female, representing seven patients (70%). The mean coronal Cobb angle at the thoracic curve was 31.67° (SD = 8.50), and the lumbar curve was 32.86° (SD = 5.79), respectively. Two participants had the Risser sign with grade 2 (20%), six participants had grade 4 (60%), and another two had grade 5 (20%). Most had a semi-rigid curve type, accounting for eight patients (80%). Seven participants were a single curve pattern (70%). The mean BMI was 19.29 kg/m<sup>2</sup> (SD = 0.68). For female participants, the median of Menarche status was 24 months, ranging from 2 months to 36 months.

Table 9.1 Participant characteristics

	n = 10
Age, mean (SD), years	16.04 (1.49)
Female sex, n (%)	7 (70)
Coronal Cobb angle of thoracic curve, mean (SD), degrees	31.67 (8.50)

	n = 10
Coronal Cobb angle of lumbar curve, mean (SD), degrees	32.86 (5.79)
Risser sign grading, n (%)	
Grade 3	2 (20)
Grade 4	6 (60)
Grade 5	2 (20)
Flexibility of curve, n (%)	
Simi-rigid curve	8 (80)
Flexible curve	2 (20)
Type of curve, n (%)	
Single curve	7 (70)
Double curve	3 (30)
Menarche status, median(min, max), month	24 (2, 36)
Body mass index, mean (SD), kg/m <sup>2</sup>	19.29 (0.68)

Note. n = number of participants, SD = standard deviation.

The study design of this research was a pilot study with ten AIS patients. A sample of ten participants was adequate to establish the system's success and provide sufficient feedback to evaluate its usefulness and any future improvements. Furthermore, the clinical experimental result would help to clarify the unclear 3-dimensional spinal deformity of AIS patients before and when applying the 3-dimensional biomechanical forces to treat 3-dimensional deformity.

Two chapters will be used to describe the results of the clinical experiments in this thesis (Chapters 9 and 10). This chapter (Chapter 9) focuses on the results from individual participants to demonstrate the feasibility of using the developed system to treat AIS patients and describe the posture of the spinal before and after applying the 3-dimensional biomechanical force correction system to treat the 3-dimensional spinal deformity. The next chapter (Chapter 10) further describes the group of ten participants and the changes in spinal posture using descriptive and inferential statistics to validate the developed system and highlight the clinical findings.

Jang previously studied the angle of spinal parameters in four spinal segments in 3 dimensions (CSA, SSA, and 3DSA) with 20 non-scoliosis persons (Jang, 2018). The study divided the values of this angle into three groups, including the neutral alignment, the potentially abnormal alignment, and the mal-alignment, as shown in



Table 9.2. To clarify, we added the colour in the table to indicate the zone of the alignment. The green box in the table indicates the neutral alignment, the yellow box indicates the potentially abnormal alignment, and the red box indicates the mal-alignment. We then add the colour zone to our result to evaluate the spine change.

Table 9.2 Spinal angles in 3 dimensions of non-scoliosis persons proposed by Jang (n = 20)

SP	Mal-alignment	Potentially abnormal alignment	Neutral alignment			Potentially abnormal alignment	Mal-alignment
	< -2 SD	-2 SD	-1 SD	Mean	+ 1 SD	+ 2 SD	> +2 SD
CUTA	<86.15	86.15	88.12	90.08	92.05	94.02	>94.02
CLTA	<85.21	85.21	87.36	89.51	91.66	93.81	>93.81
CULA	<85.07	85.07	87.23	89.39	91.54	93.70	>93.70
CLLA	<85.35	85.35	87.25	89.16	91.06	92.96	>92.96
SUTA	<66.96	66.96	72.37	77.79	83.20	88.61	>88.61
SLTA	<86.48	86.48	93.09	99.70	106.31	112.92	>112.92
SULA	<88.37	88.37	93.05	97.74	102.42	107.10	>107.10
SLLA	<59.90	59.90	68.11	76.32	84.53	92.73	>92.73
3D-UTA	<66.96	66.96	72.29	77.61	82.93	88.26	>88.26
3D-LTA	<89.01	89.01	93.5	100	106.49	112.98	>112.98
3D-ULA	<89.01	89.01	93.56	98.1	102.65	107.2	>107.20
3D-LLA	<59.97	59.97	68.03	76.1	84.17	92.24	>92.24

Note. SP = spinal parameter, SD = standard deviation.

## 9.2. Experimental results for individual participants

### Participant 1

Participant 1 was a 17-year-three-month-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a single curve to the left side with an apex at L1 and the upper- and lower-end vertebra at T10 and L4, respectively. The coronal Cobb angle at the lumbar curve was 37°, and the Risser sign was grade 4. She had her period for three years, or since she was fourteen. The trunk balance was slight to the left side by 20 mm. The physical examination revealed that the trunk balance was slight to the left side, similar to the radiographs. The shoulder level was the same on both sides. However, the left waist was higher than the right side, and the right axilla was higher than the left. She had no

true or apparent leg length discrepancy (LLD). A spinal curve's flexibility was semi-rigid. The BMI was 18.55 kg/m<sup>2</sup>. She had no history of spinal surgery, no underlying condition that affects lung or heart function, and no psychological or communication problems. She could stand and walk normally.

Participant 1's curve pattern was a single left lumbar curve. After clinical consideration, six forces were applied to correct the deformity. Firstly, the corrective force was at the curve's apex from L1 to L3 on the left side. This force was applied in the anterior and medial directions to centralize the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T7 to T9 on the right side and the inferior counterforce at the gluteus medius on the right side. The left gluteus medius force was then applied to stabilize the pelvis. The left axilla was applied to reduce trunk leaning to the left side. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.1 and 9.2 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs and a low-cost postural measurement system during the assessment, optimal correction, and casting. Table 9.3 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and the postural measurement system. The following table, Table 9.4, further reports the results from the postural measurement system for other spinal parameters, including the angles of spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

Comparing the spinal alignment in the coronal and sagittal planes between radiographs and the postural measurement system, the graphs from the postural measurement system could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column shifted from the left to the centre and was straighter. The coronal decompensation was reduced and close to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement

system also gave the results of the spinal angle in 3D. The following section analysed the coronal, sagittal, and 3D spinal angles in detail using descriptive and inferential statistics.

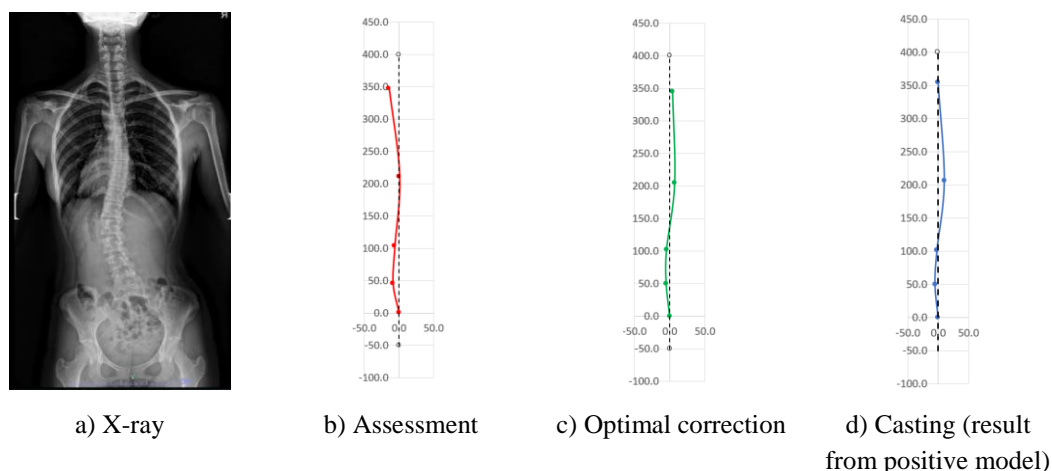


Figure 9.1 Spinal column in the coronal plane of participant 1, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

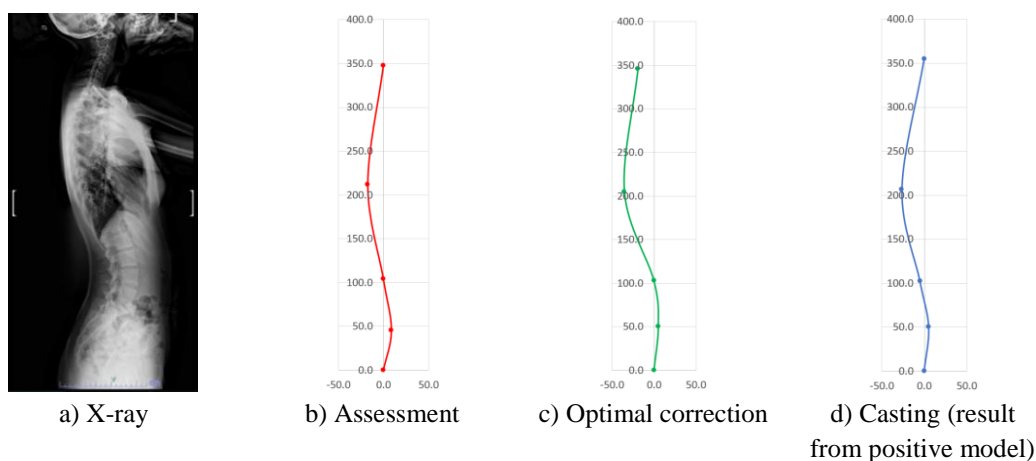


Figure 9.2 Spinal column in the sagittal plane of participant 1, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.3 Results of spinal parameters in the sagittal and coronal planes from an X-ray and a low-cost postural measurement system for participant 1.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	95.00	96.17	92.02	93.72
CLTA, degree	85.00	86.41	82.58	83.85
CULA, degree	91.00	87.56	86.18	90.24
CLLA, degree	104.00	100.44	98.86	93.01
SUTA, degree	83.00	82.73	88.03	80.23
SLTA, degree	102.00	99.12	109.06	99.06

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
SULA, degree	100.00	98.68	95.68	104.23
SLLA, degree	90.00	78.96	84.03	82.40
Trunk balance, mm	-20.00	-13.81	4.04	0.00

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity was between normal and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same zone. Some improved the deformity, and some did not, but increased the deformity. The deformity varied between normal and mal-alignment zones after applying the force to correct the deformity.

SSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal spinal deformity was in the normal alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same zone, and some increased deformity and changed to a lower zone. The deformity varied between normal and mal-alignment zones after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone, and some improved the deformity and changed to a better zone. The deformity was mostly in normal alignment after applying the force to correct the deformity.

Table 9.4 Results of a low-cost postural measurement system for participant 1 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	80.51	83.11	79.57
3D-LTA, degree	99.78	110.27	100.89
3D-ULA, degree	99.01	96.83	104.23
3D-LLA, degree	74.98	79.37	81.84

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
Trunk rotation at T7, degree	4.38		1.81
Trunk rotation at T12, degree	-5.60		5.25
Trunk rotation at L2, degree	-4.69		1.56
Trunk rotation at L5, degree	-0.52		-4.82
Shoulder level, mm	-0.42		-3.44
Axilla level, mm	4.60		-8.72
Waist level, mm	-20.65		-12.67
Distance at axilla level, mm	-7.52		23.38
Distance at waist level, mm	-9.43		-12.30
POTSI for trunk balance, percentage	4.49		0.00
POTSI for the left and right distance at axilla, percentage	2.44		9.39
POTSI for the left and right distance at waist, percentage	4.16		5.16
POTSI for shoulder level, percentage	0.12		0.96
POTSI for axilla level, percentage	1.32		2.42
POTSI for waist level, percentage	5.94		3.52
POTSI in total, percentage	18.48		21.41

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre. POTSI = POsterior Trunk Symmetry Index.

Figure 9.3 illustrates a graph representing the spinal parameters in the transverse plane (horizontal trunk rotation). The graphs show the change in trunk rotation compared between the assessment process (without force applied) and the casting process (with force applied). Table 9.4 also reports the angles of horizontal trunk rotation at each level. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T12 and L2 levels or around the level of apical vertebrae.

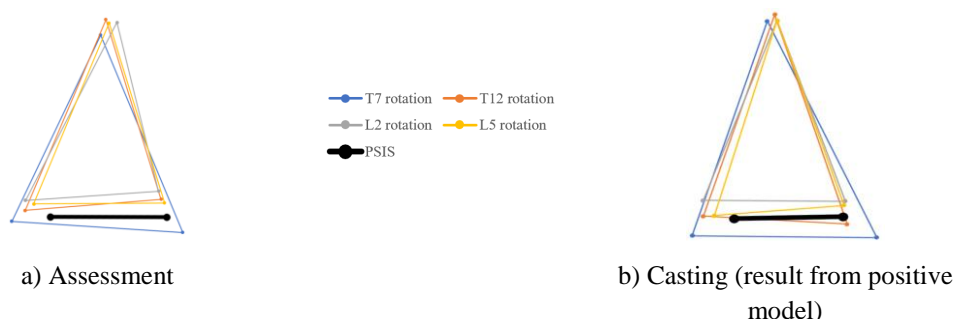


Figure 9.3 The trunk horizontal rotation of participant 1, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.4 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.4 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

The result showed that the shoulder level remained the same, the waist level improved, but the axilla level went to the opposite side (from right to left sides). The left and right distances at the waist level remained similar, but the distances went opposite sides at the axilla level (from left to right sides). The total POTSI index between assessment and casting was not different (from 18.48% to 21.41%). As a result, the shape of the trunk in the coronal plane slightly changed after applying the forces to correct the spine, and the asymmetry of the trunk seemed to be increased.

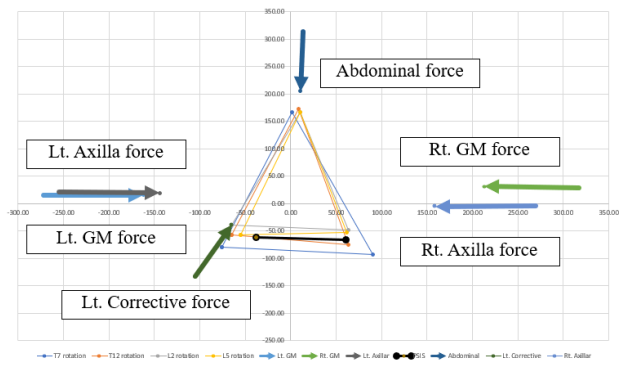


Figure 9.4 The trunk asymmetry of participant 1, where a) and b) the results from a postural measurement system during the assessment and casting.

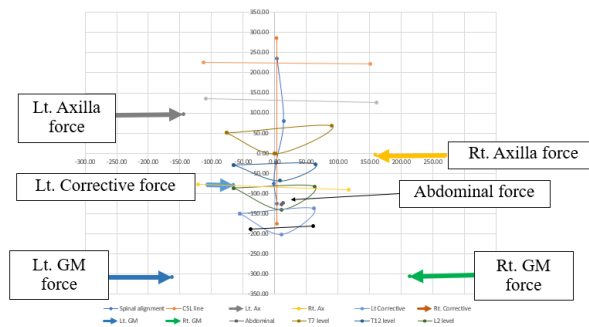
Figure 9.5 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 1 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.5 reports the magnitude of forces applied to Participant 1's trunk to correct the deformity.

The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 13.55 N to 31.92 N, and the total force was 130.96

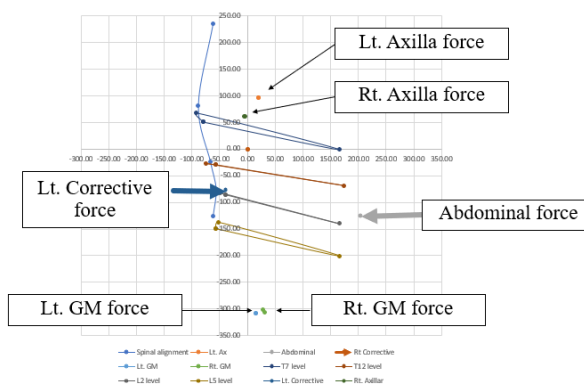
N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the spine more towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.5 Forces applied to participant 1's trunk in, a) the transverse, b) coronal, and c) sagittal planes.

Table 9.5 Results of the forces that were applied to participant 1's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	19.38
Left axilla force	17.35
Right axilla force	22.37
Left gluteus medius force	26.40
Right gluteus medius force	13.55
Abdominal force	31.92
Total	130.96

## Participant 2

Participant 2 was a 15-years-five-month-old boy who was diagnosed with AIS by an orthopaedic doctor. His parents first noticed his scoliosis when he was 12 years old. The radiographic evaluation revealed that his spinal curve pattern was a single curve to the right side with an apex at T11 and the upper- and lower-end vertebra at T6 and L2, respectively. The coronal Cobb angle at the thoracic curve was 28°, and the Risser sign was grade 4. He had no coronal decompensation. The physical examination revealed that he still had no coronal decompensation, similar to the X-ray result. The left shoulder and waist levels were higher than the right side. However, the right axilla was higher than the left side. He had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 19.10 kg/m<sup>2</sup>. He had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 2's curve pattern was a single right thoracic curve. After clinical consideration, five forces were applied to correct the deformity. Firstly, the corrective force was at the curve's apex from T11 to L1 on the right side. This force was applied in the anterior, medial, and upward directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T6 to T8 on the left side and the inferior counterforce at the gluteus medius on the left side. The right gluteus medius force was then applied to stabilise the pelvis. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.



Figures 9.6 and 9.7 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs and the postural measurement system during the assessment, optimal correction, and casting processes. Table 9.6 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.7 reports the results of the postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column shifted from the right to the centre and was straighter. The coronal decompensation was slightly increased and went to the opposite side. However, it was lower than 5 mm and considered a small change. In the sagittal plane, the graph from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

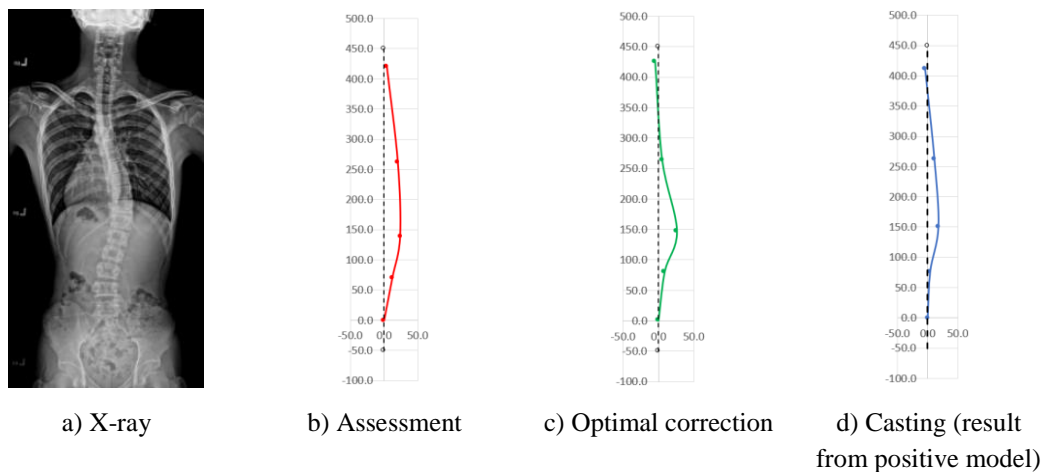


Figure 9.6 Spinal column in the coronal plane of participant 2, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

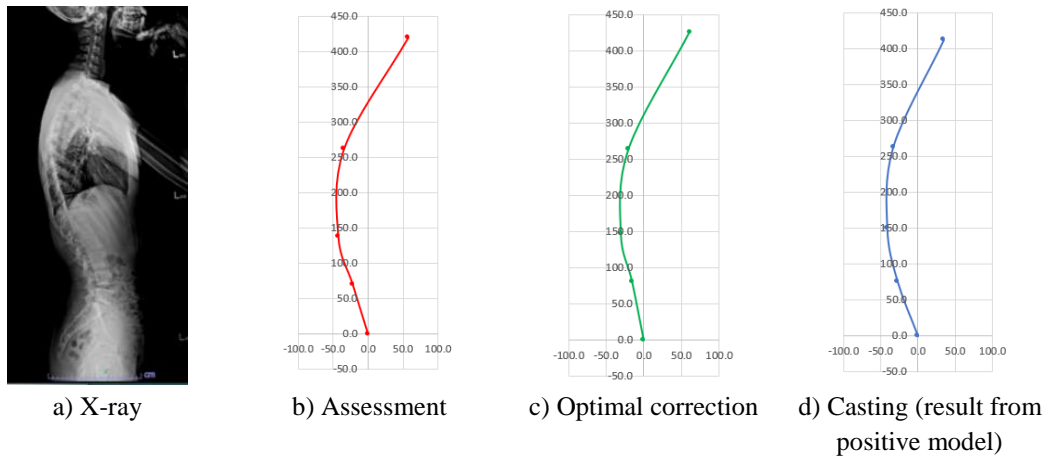


Figure 9.7 Spinal column in the sagittal plane of participant 2, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.6 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 2.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	96.00	95.95	93.46	95.77
CLTA, degree	95.00	91.62	100.06	93.36
CULA, degree	77.00	80.35	75.72	79.86
CLLA, degree	82.00	80.10	83.78	86.97
SUTA, degree	72.00	59.82	63.23	65.73
SLTA, degree	95.00	86.20	85.07	85.92
SULA, degree	108.00	106.81	102.79	100.47
SLLA, degree	106.00	107.24	100.88	109.64
Trunk balance, mm	0.00	3.99	-4.70	-4.28

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity was mostly in the mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same zone. Some improved the deformity and changed to a better zone. The deformity was improved after applying the force to correct the spine to be potentially abnormal and mal-alignment.

SSA between radiographs in assessment and the postural measurement system in assessment showed that some spinal parameters were in a similar colour zone, some changed to a better zone, and some changed to a lower zone. The sagittal spinal deformity varied between normal and mal-alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same zone. The deformity was slightly changed but still presented in the same zone. The deformity was between normal and mal-alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same colour zone, and some improved the deformity and changed to a better zone. The deformity was improved but still presented in the mal-alignment after applying the force to correct the deformity.

Table 9.7 Results of a low-cost postural measurement system for participant 2 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	59.15	63.06	65.20
3D-LTA, degree	85.87	78.84	84.72
3D-ULA, degree	109.12	108.83	104.42
3D-LLA, degree	109.59	102.46	109.84
Trunk rotation at T7, degree	7.49		5.32
Trunk rotation at T12, degree	7.16		1.60
Trunk rotation at L2, degree	4.55		-1.97
Trunk rotation at L5, degree	1.31		2.22
Shoulder level, mm	-17.81		0.64
Axilla level, mm	21.41		20.86
Waist level, mm	-59.31		-60.54
Distance at axilla level, mm	43.84		45.74
Distance at waist level, mm	42.49		28.56
POTSI for trunk balance, percentage	1.18		1.27
POTSI for the left and right distance at axilla, percentage	13.01		13.58
POTSI for the left and right distance at waist, percentage	19.00		13.66
POTSI for shoulder level, percentage	4.23		0.15
POTSI for axilla level, percentage	5.09		5.05
POTSI for waist level, percentage	14.10		14.67
POTSI in total, percentage	56.62		48.38

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.

Figure 9.8 illustrates a graph representing the trunk's horizontal rotation. Table 9.7 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T12 and L2 levels or around the level of apical vertebrae.

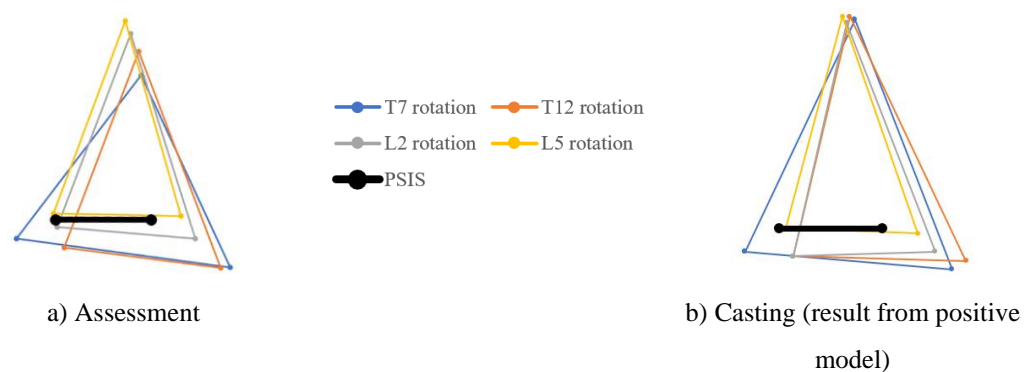


Figure 9.8 Spinal parameters in the transverse plane of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.9 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.7 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

The result showed that the axilla and waist levels looked the same, and the shoulder level improved. The left and right distances at the axilla level also remained similar, but the distances at the waist level improved. The total POTSI index between assessment and casting was slightly reduced (from 56.62% to 48.38%). As a result, the

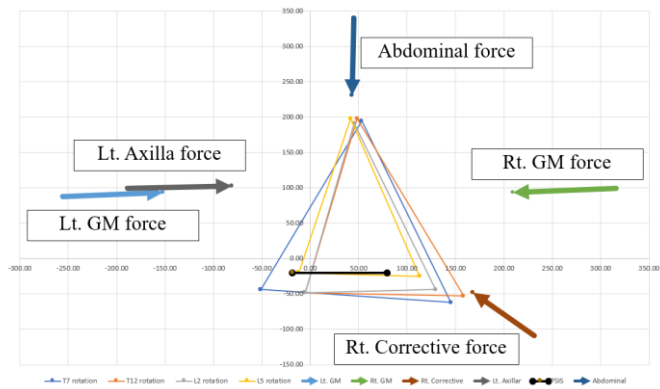
shape of the trunk in the coronal plane slightly changed, or the trunk asymmetry was improved after applying the forces to correct the spine.



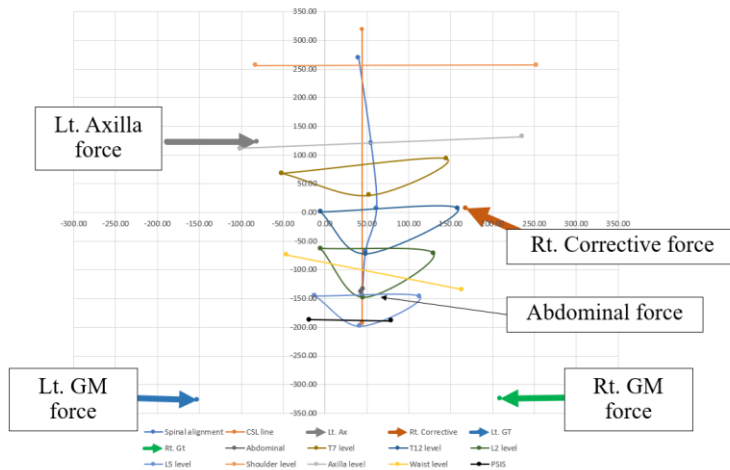
Figure 9.9 Spinal parameters in the trunk asymmetry of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.10 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 2 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.8 reports the magnitude of forces applied to Participant 2's trunk to correct the deformity.

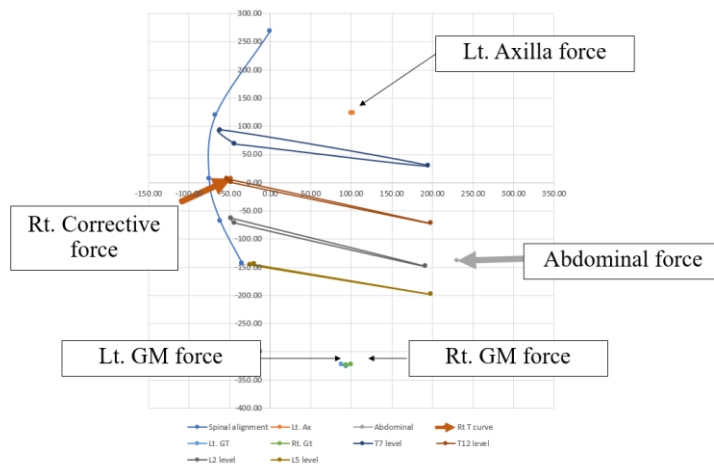
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 19.92 N to 35.17 N, and the total force was 145.96 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The thoracic corrective force, providing medial, anterior, and upward-directed force, could shift the spine towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.10 Forces applied to participant 2's trunk in a) the transverse, b) coronal, and c) sagittal planes.

Table 9.8 Result of the forces that were applied to participant 2's trunk.

Area of Forces	Amount of Force (N)
Right corrective force	31.05
Left axilla force	26.98
Left gluteus medius force	32.84
Right gluteus medius force	19.92
Abdominal force	35.17
Total	145.96

### Participant 3

Participant 3 was a 15-year-eight-month-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a single curve to the right side with an apex at T10, the upper- and lower-end vertebra at T6 and L1, respectively. The coronal Cobb angle at the thoracic curve was 40°, and the Risser sign was grade 4. She had her period for two years. The coronal decompensation was slight to the left side by 5 mm. The physical examination revealed slight coronal decompensation to the left side, similar to the X-ray result. The left shoulder and waist levels were higher than the right side. However, the axilla seemed level. She had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 19.47 kg/m<sup>2</sup>. She had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 3's curve pattern was a single right thoracic curve. After clinical consideration, five forces were applied to correct the deformity. Firstly, the corrective force was at the curve's apex from T10 to T12 on the right side. This force was applied in the anterior, medial, and upward directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T6 to T8 on the left side and the inferior counterforce at the gluteus medius on the left side. The right gluteus medius force was then applied to stabilise the pelvis. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.11 and 9.12 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between

radiographs and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.9 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.10 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column shifted from the right to the centre and was straighter. The coronal decompensation improved and closed to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

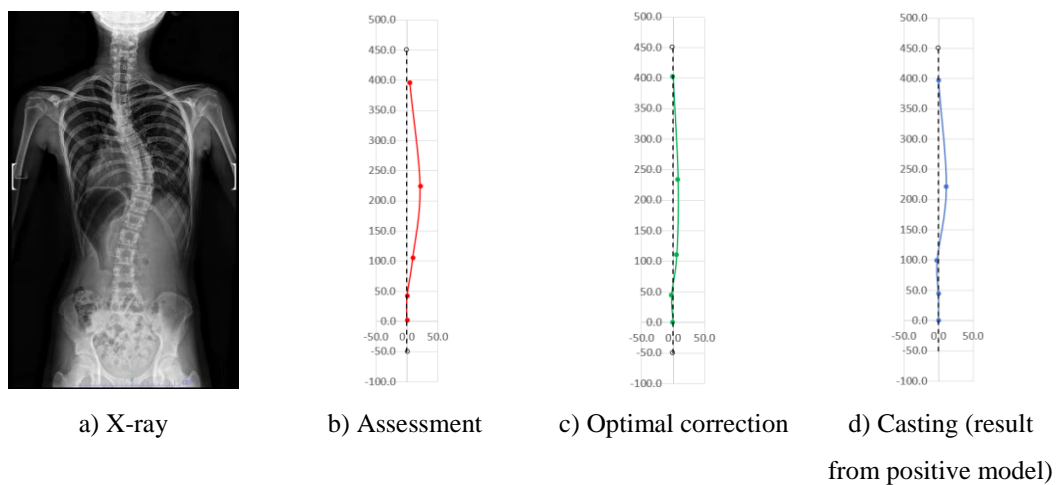


Figure 9.11 Spinal column in the coronal plane of participant 3, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.



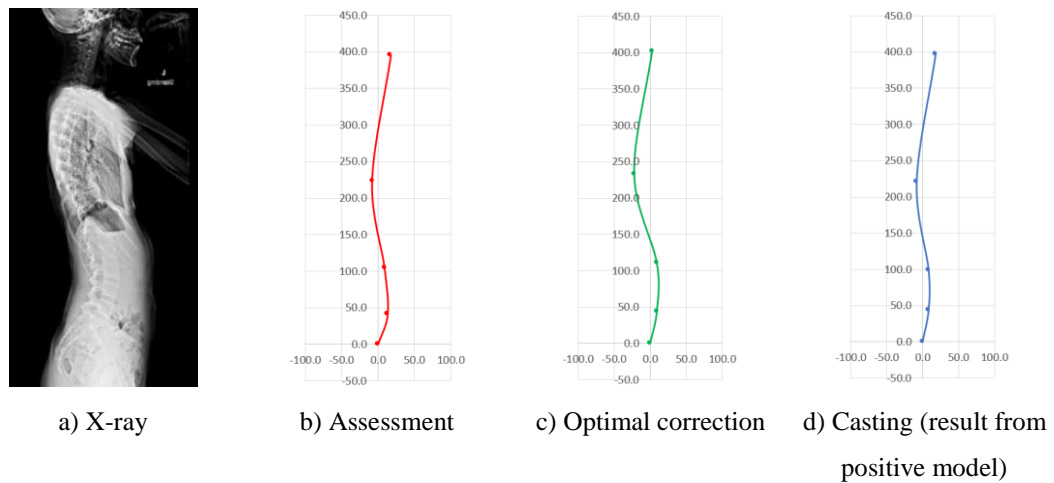


Figure 9.12 Spinal column in the sagittal plane of participant 3, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.9 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 3.

Spinal Parameters	Result from X-ray	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	96.00	95.49	92.66	93.66
CLTA, degree	92.00	84.59	88.99	83.79
CULA, degree	78.00	81.56	82.99	92.00
CLLA, degree	90.00	90.49	93.14	90.04
SUTA, degree	86.00	81.49	81.59	81.46
SLTA, degree	100.00	97.58	104.44	97.75
SULA, degree	100.00	94.08	90.00	90.00
SLLA, degree	83.00	69.26	78.24	79.10
Trunk balance, mm	-5.00	4.90	0.00	0.07

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity varied between normal and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same zone. Some improved the deformity and changed to a better zone, but some increased the deformity. The deformity was improved, but the

deformity varied between normal and mal-alignment after applying the force to correct the spine.

SSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and some changed to a better zone. The sagittal spinal deformity was mostly in normal alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, all spinal parameters remained in the same zone. The deformity was in the normal alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same colour zone. The deformity mostly presented in the normal alignment after applying the force to correct the deformity.

Table 9.10 Results of a low-cost postural measurement system for participant 3 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	79.92	81.19	80.73
3D-LTA, degree	99.27	104.47	99.88
3D-ULA, degree	99.35	97.01	92.00
3D-LLA, degree	69.25	77.85	79.10
Trunk rotation at T7, degree	10.03		2.51
Trunk rotation at T12, degree	9.86		5.49
Trunk rotation at L2, degree	4.80		0.20
Trunk rotation at L5, degree	5.08		-0.12
Shoulder level, mm	-21.10		-17.35
Axilla level, mm	-0.06		-0.86
Waist level, mm	-32.01		-38.28
Distance at axilla level, mm	35.14		41.03
Distance at waist level, mm	13.42		10.35
POTSI for trunk balance, percentage	1.73		0.03
POTSI for the left and right distance at axilla, percentage	12.48		18.51
POTSI for the left and right distance at waist, percentage	7.00		5.33
POTSI for shoulder level, percentage	5.34		4.37
POTSI for axilla level, percentage	0.02		0.22
POTSI for waist level, percentage	8.10		9.64
POTSI in total, percentage	34.66		38.11

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.

Figure 9.13 illustrates a graph representing the trunk's horizontal rotation. Table 9.10 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T7 and T12 levels or around the level of apical vertebrae.

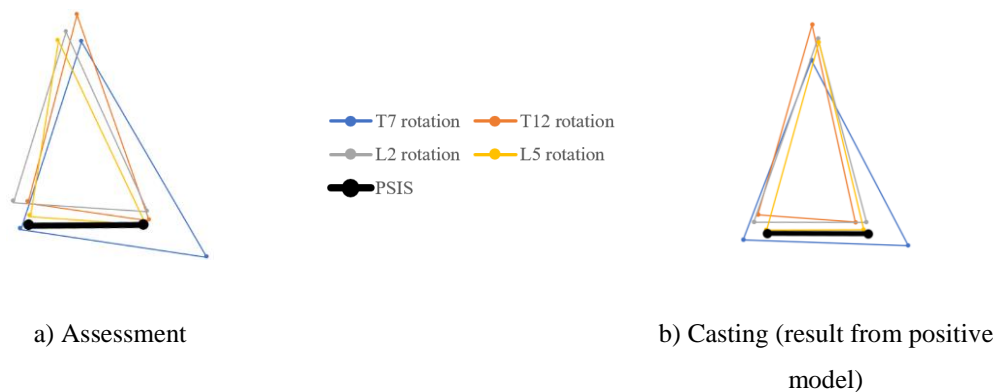
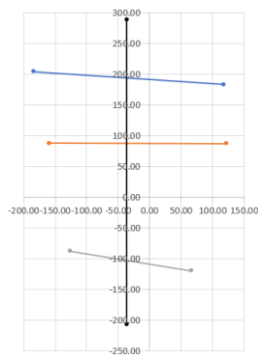


Figure 9.13 Spinal parameters in the transverse plane of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting.

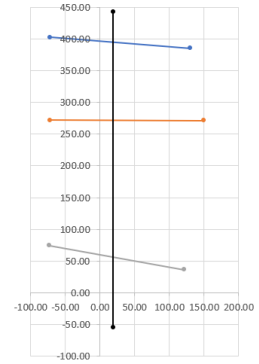
Figure 9.14 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.10 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

The result showed that the shoulder, axilla, and waist levels looked the same. The left and right distances at the axilla and waist levels also remained similar. The total POTSI index between assessment and casting was slightly increased (from 34.66% to 38.11%). As a result, the shape of the trunk in the coronal plane slightly changed after applying the forces to correct the spine, and the asymmetry of the trunk seemed to be increased.



a) Assessment

—●— Shoulder level    —■— Axilla level  
 —▲— Waist level    —●— CSL

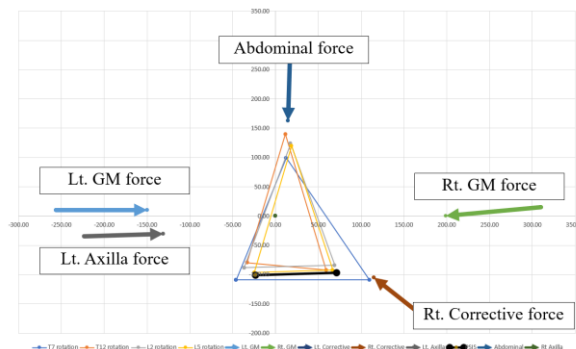


b) Casting (result from positive model)

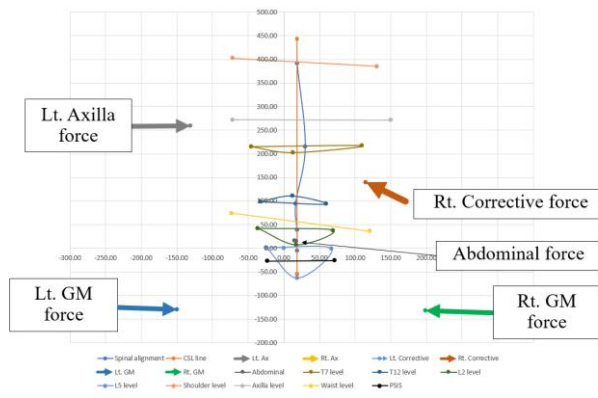
Figure 9.14 Spinal parameters in the trunk asymmetry of participant 2, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.15 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 3 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.11 reports the magnitude of forces applied to Participant 3's trunk to correct the deformity.

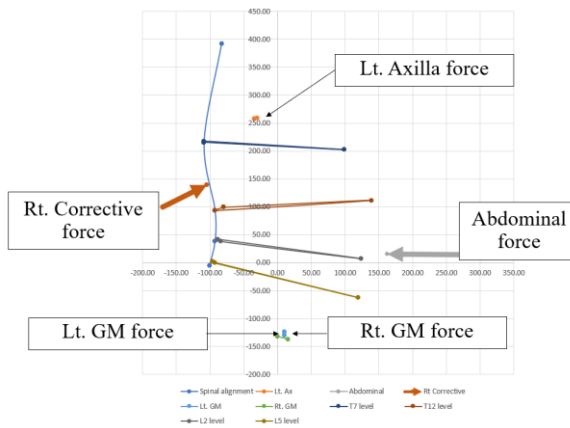
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 21.57 N to 47.35 N, and the total force was 174.95 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The thoracic corrective force, providing medial, anterior, and upward-directed force, could shift the spine towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.15 Forces applied to participant 3's trunk in a) the transverse, b) coronal, and c) sagittal planes.

Table 9.11 Result of the forces that were applied to participant 3's trunk.

Area of Forces	Amount of Force (N)
Right corrective force	21.57
Left axilla force	33.86
Left gluteus medius force	47.35
Right gluteus medius force	27.80
Abdominal force	44.36
Total	174.95

### Participant 4

Participant 4 was an 18-year-old boy who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that his spinal curve pattern was a single curve to the left side with an apex at L1, the upper- and lower-end vertebra

at T11 and L4, respectively. The coronal Cobb angle at the lumbar curve was 25°, and the Risser sign was grade 5. The coronal decompensation was not present. The physical examination revealed that the left shoulder and axilla levels were higher than the right side. However, the right waist level was higher than the left side. He had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 20.76 kg/m<sup>2</sup>. He had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 4's curve pattern was a single left lumbar curve. After clinical consideration, five forces were applied to correct the deformity. Firstly, the corrective force was at the curve's apex from L1 to L3 on the left side. This force was applied in the anterior and medial directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T7 to T9 on the right side and the inferior counterforce at the gluteus medius on the right side. The left gluteus medius force was then applied to stabilise the pelvis. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.16 and 9.17 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.12 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.13 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graph from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column shifted from the left to the centre and was straighter. The coronal decompensation slightly increased towards the right side. However, it was considered a small change because it was less than 3 mm from the centre. In the sagittal

plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

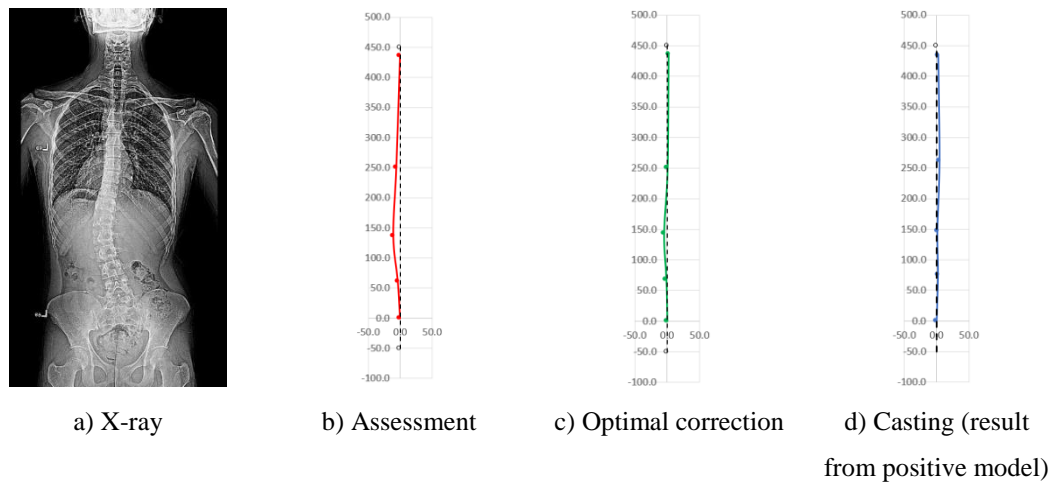


Figure 9.16 Spinal column in the coronal plane of participant 4, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

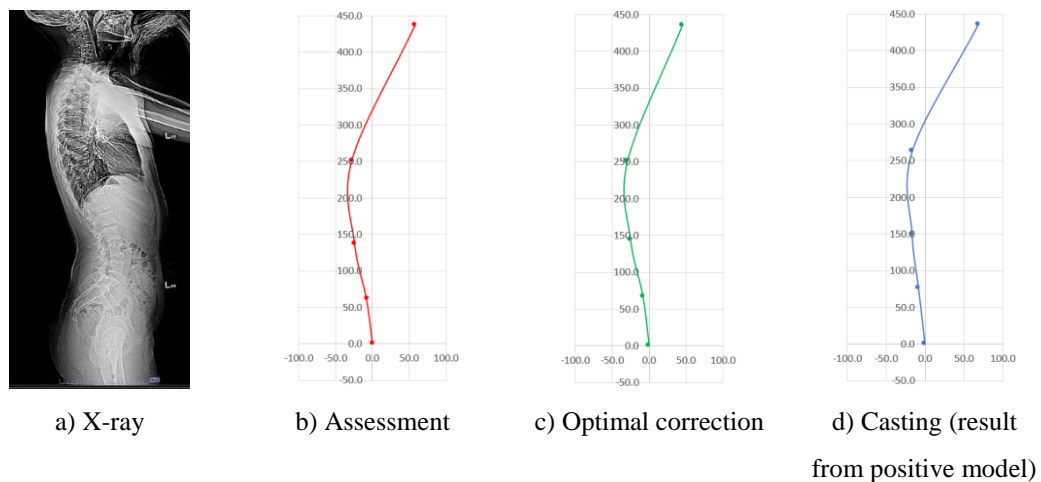


Figure 9.17 Spinal column in the sagittal plane of participant 4, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.12 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 4.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	87.00	88.14	90.49	90.66
CLTA, degree	84.00	87.61	87.06	88.28
CULA, degree	94.00	95.37	92.68	90.70

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CLLA, degree	98.00	93.38	91.22	88.43
SUTA, degree	84.00	65.22	68.00	63.80
SLTA, degree	100.00	91.97	92.21	90.05
SULA, degree	106.00	101.96	102.43	96.45
SLLA, degree	100.00	97.59	97.15	96.31
Trunk balance, mm	0.00	0.00	2.10	2.61

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity varied between normal and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters improved the deformity and changed to a better zone. The deformity was improved after applying the force to correct the spine to be the normal and potentially abnormal alignment zones.

SSA between radiographs in assessment and the postural measurement system in assessment showed that some spinal parameters were in a similar colour zone, and some changed the zone. The sagittal spinal deformity varied between normal and mal-alignment zones. For SSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same zone, and the deformity varied between normal and mal-alignment zones after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same zone. Some improved to a better zone, but some changed to a lower zone. The deformity varied between normal and mal-alignment zones after applying the force to correct the deformity.



Table 9.13 Results of a low-cost postural measurement system for participant 4 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	65.17	68.00	63.79
3D-LTA, degree	93.10	93.67	91.72
3D-ULA, degree	103.05	102.70	96.49
3D-LLA, degree	98.29	97.25	96.50
Trunk rotation at T7, degree	-1.17		-1.07
Trunk rotation at T12, degree	-8.10		-3.93
Trunk rotation at L2, degree	-5.86		-4.42
Trunk rotation at L5, degree	-2.23		-2.37
Shoulder level, mm	-4.28		-25.20
Axilla level, mm	-12.82		-5.78
Waist level, mm	33.21		15.97
Distance at axilla level, mm	-12.42		19.89
Distance at waist level, mm	-15.24		5.35
POTSI for trunk balance, percentage	0.00		0.88
POTSI for the left and right distance at axilla, percentage	3.71		6.71
POTSI for the left and right distance at waist, percentage	6.43		2.29
POTSI for shoulder level, percentage	0.98		5.79
POTSI for axilla level, percentage	2.94		1.33
POTSI for waist level, percentage	7.62		3.67
POTSI in total, percentage	21.68		20.68

Note: 3D-UTA = 3-dimesional upper thoracic angle. 3D-LTA = 3-dimesional lower thoracic angle. 3D-ULA = 3-dimesional upper lumbar angle. 3D-LLA = 3-dimesional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.

Figure 9.18 illustrates a graph representing the trunk's horizontal rotation. Table 9.13 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T12 level or around the level of the apical vertebrae.

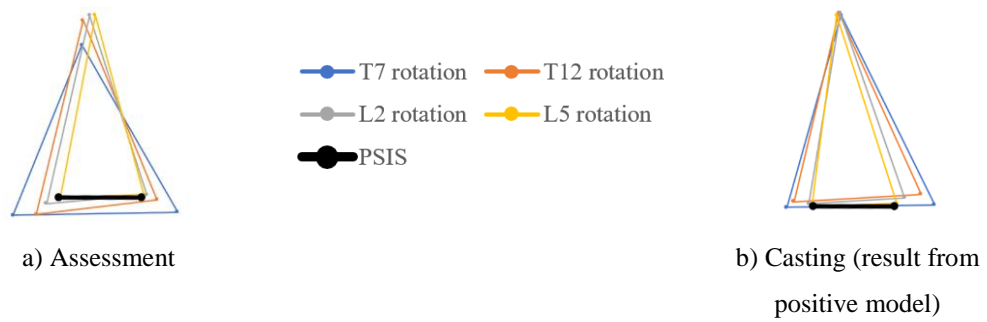


Figure 9.18 Spinal parameters in the transverse plane of participant 4, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.19 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.13 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

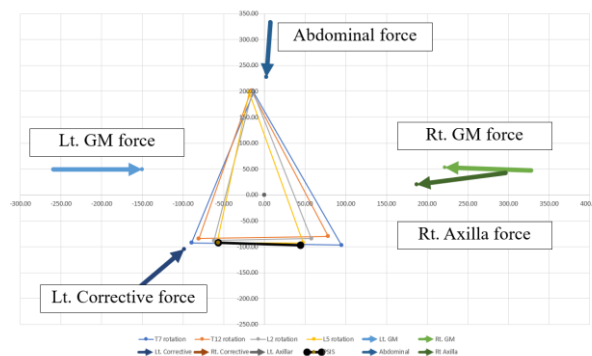
The result showed that the axilla levels looked the same, and the waist level improved. The shoulder level did not improve, but it increased. The left and right distances at the axilla and waist levels were improved and slightly shifted from the left to the right. The total POTSI index between assessment and casting was not different (from 21.68% to 20.68%). As a result, the shape of the trunk in the coronal plane slightly changed after applying the forces to correct the spine, and the asymmetry was slightly improved.



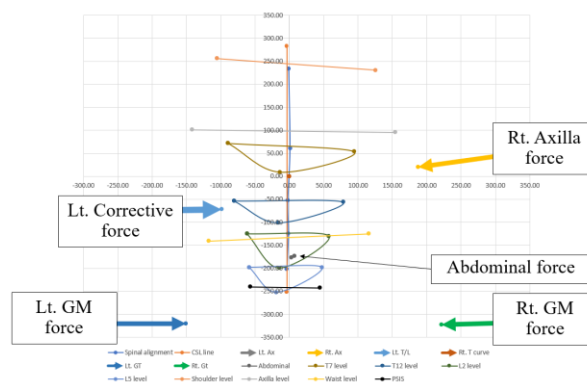
Figure 9.19 Spinal parameters in the trunk asymmetry of participant 4, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.20 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 4 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.14 reports the magnitude of forces applied to Participant 4's trunk to correct the deformity.

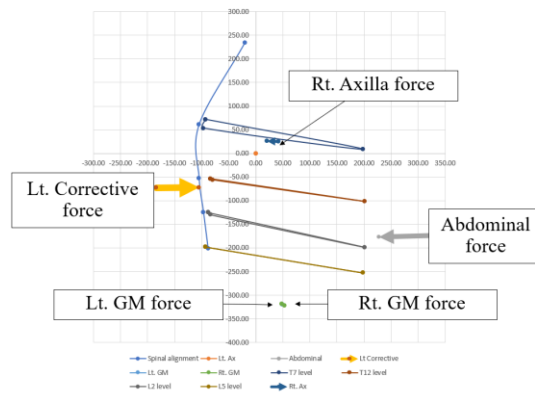
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 23.32 N to 28.43 N, and the total force was 129.80 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the spine towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.20 Forces applied to participant 4's trunk in a) the transverse, b) coronal, and c) sagittal planes.

Table 9.14 Result of the forces that were applied to participant 4's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	25.49
Right axilla force	28.43
Left gluteus medius force	25.67
Right gluteus medius force	26.89
Abdominal force	23.32
Total	129.80

## Participant 5

Participant 5 was a 16-year-two-month-old boy who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that his spinal curve pattern was a double-curve: right thoracic and left lumbar curves. At the thoracic curve, the apical vertebra was at T7, the upper- and lower-end vertebra at T4 and T11, respectively. The coronal Cobb angle at the thoracic curve was 25°. At the lumbar curve, the apical vertebra was at L1, and the upper- and lower-end vertebra at T11 and L4, respectively. The Risser sign was grade 4. The coronal decompensation was slight to the left side by 20 mm. The physical examination revealed slight coronal decompensation to the left side, similar to the X-ray result. The left waist level was higher than the right side. However, the right shoulder and axilla levels were higher

than the left side. He had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 19.49 kg/m<sup>2</sup>. He had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 5's curve pattern was a double-curve with right thoracic and left lumbar curves. After clinical consideration, six forces were applied to correct the deformity. Firstly, the lumbar corrective force was at the curve's apex from L1 to L3 on the left side. This force was applied in the anterior and medial directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Then, the corrective force for the thoracic curve was located on the right side at the curve's apex from T7 to T9. This force was applied in the medial direction and had to be applied cautiously because it would shift the spine to the left as the apical vertebra was now located at the CSL line. This corrective force also acted as the counter force for the lumbar corrective force. The inferior counterforce at the right gluteus medius was another counterforce required to counteract the lumbar corrective force. The left gluteus medius force was then applied to stabilise the pelvis. The superior counterforce at the axilla on the left side was another counterforce required to counteract the thoracic corrective force. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.21 and 9.21 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.15 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.16 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column shifted from the left to the centre and looked was straighter. The coronal decompensation was improved and closed to the centre. In the sagittal

plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

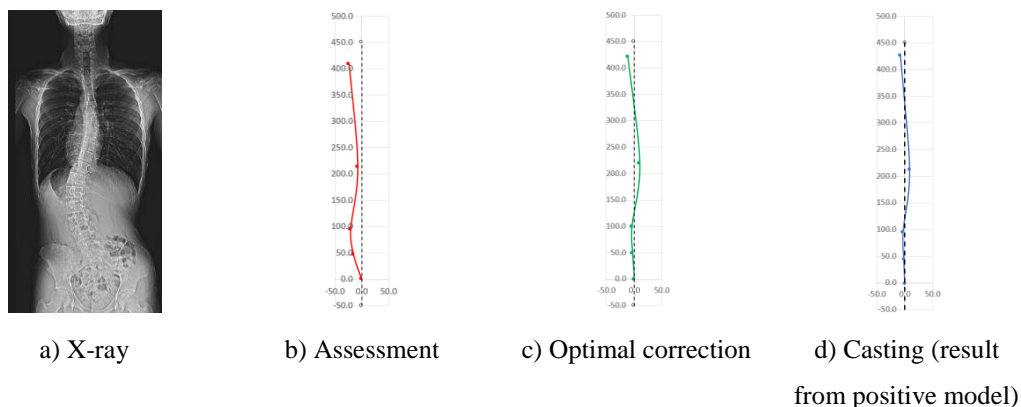


Figure 9.21 Spinal column in the coronal plane of participant 5, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

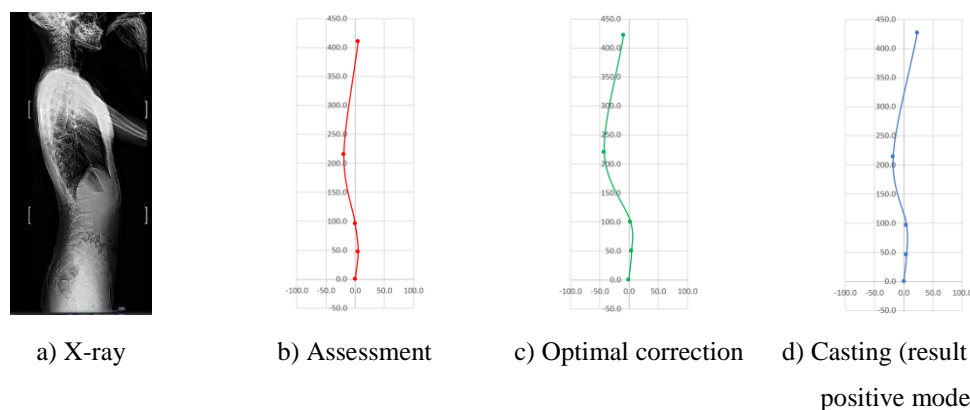


Figure 9.22 Spinal column in the sagittal plane of participant 5, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.15 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 5.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	94.00	94.82	95.99	93.65
CLTA, degree	75.00	83.53	83.31	86.56
CULA, degree	86.00	96.35	88.56	90.00
CLLA, degree	115.00	108.16	93.78	91.30
SUTA, degree	83.00	82.98	80.70	78.98

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
SLTA, degree	104.00	99.11	108.50	101.04
SULA, degree	100.00	95.77	88.10	90.00
SLLA, degree	90.00	84.03	84.62	84.15
Trunk balance, mm	-20.00	-23.70	-11.60	-7.54

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that some spinal parameters were in a similar colour zone, but some changed from potential abnormal to mal-alignment in the postural measurement system. Overall, the coronal spinal deformity was between potential and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters improved and changed to a better zone. Overall, the deformity improved, but the deformity was still in the mal-alignment zone in optimal correction and potential abnormal zone on casting after applying the force to correct the deformity.

SSA between radiographs and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal spinal deformity was in the normal alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone, and some increased deformity and changed to a lower zone. The deformity was between normal and potential abnormal alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone. Some increased the deformity and changed to the lower zone. The deformity was mostly in normal and potentially abnormal alignment after applying the force to correct the deformity.

Table 9.16 Results of a low-cost postural measurement system for participant 5 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	81.51	78.99	78.42
3D-LTA, degree	101.11	111.59	101.54
3D-ULA, degree	98.55	87.63	90.00
3D-LLA, degree	71.00	83.44	84.01
Trunk rotation at T7, degree	7.67		4.78
Trunk rotation at T12, degree	-3.38		0.00
Trunk rotation at L2, degree	-3.70		2.39
Trunk rotation at L5, degree	2.90		-0.27
Shoulder level, mm	8.35		0.90
Axilla level, mm	5.90		20.54
Waist level, mm	-6.50		-0.98
Distance at axilla level, mm	0.63		11.09
Distance at waist level, mm	-25.75		1.20
POTSI for trunk balance, percentage	8.51		2.85
POTSI for the left and right distance at axilla, percentage	0.23		4.20
POTSI for the left and right distance at waist, percentage	11.56		0.52
POTSI for shoulder level, percentage	2.04		0.21
POTSI for axilla level, percentage	1.44		4.81
POTSI for waist level, percentage	1.59		0.23
POTSI in total, percentage	25.35		12.83

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.

Figure 9.23 illustrates a graph representing the trunk's horizontal rotation. Table 9.16 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T7, T12, and L2 levels or around the level of the apical vertebrae.



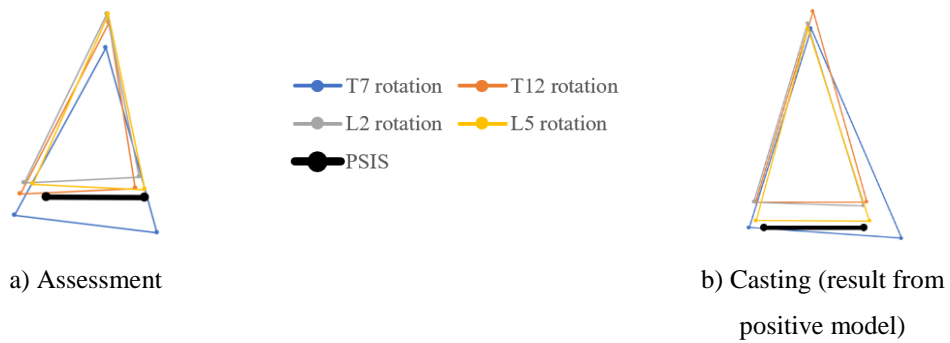


Figure 9.23 Spinal parameters in the transverse plane of participant 5, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.24 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.16 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

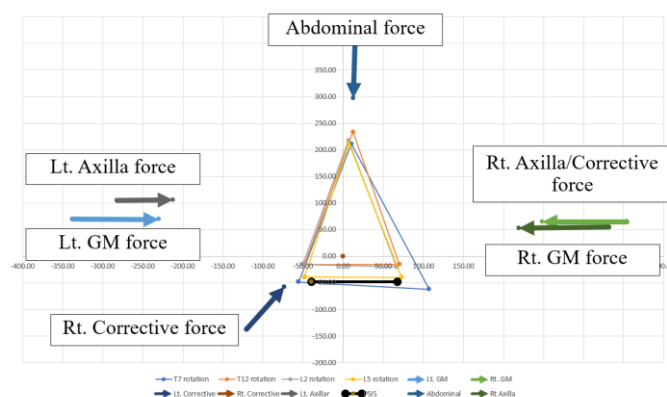
The result showed that the shoulder and waist levels were improved. The axilla level did not improve, but it increased. The left and right distances at the waist level was improved. The left and right distances at the axilla level did not improve, but it increased. The total POTSI index between assessment and casting was considerably decreased (from 25.35% to 12.83%). As a result, the shape of the trunk in the coronal plane obviously changed or improved after applying the forces to correct the spine.



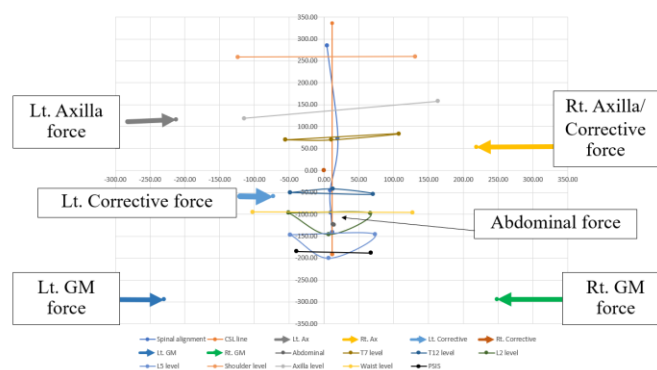
Figure 9.24 Spinal parameters in the trunk asymmetry of participant 5, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.25 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 5 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.17 reports the magnitude of forces applied to Participant 5's trunk to correct the deformity.

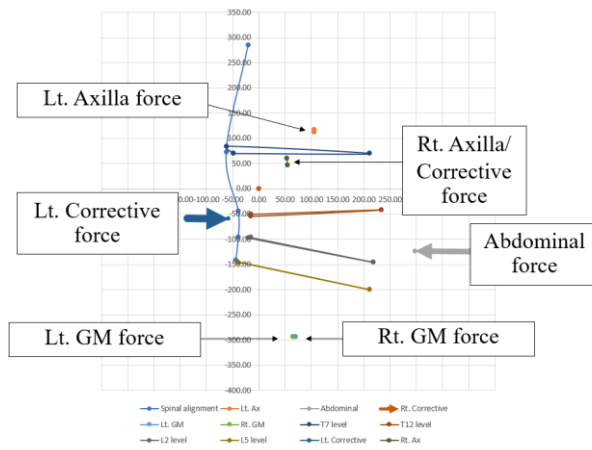
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 21.02 N to 42.61 N, and the total force was 172.65 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the spine towards the centre in the coronal plane and derotate the spine in the transverse plane. The thoracic corrective force, providing medial-directed force, could reduce the curve magnitude and stabilise the spine. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.25 Forces applied to participant 5's trunk in a) the transverse, b) coronal, and c) sagittal planes.

Table 9.17 Result of the forces that were applied to participant 5's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	42.61
Left axilla force	21.38
Right axilla force	21.02
Left gluteus medius force	33.39
Right gluteus medius force	27.05
Abdominal force	27.21
Total	172.65

## Participant 6

Participant 6 was a 17-year-seven-month-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a double-curve: right thoracic and left lumbar curves. At the thoracic curve, the apical vertebra was at T9, and the upper- and lower-end vertebra were at T5 and T12, respectively. The coronal Cobb angle at the thoracic curve was 37°. At the lumbar curve, the apical vertebra was at L2, and the upper- and lower-end vertebra were at T12 and L4, respectively. The Risser sign was grade 4. She had her period for three years, or since she was fourteen. The coronal decompensation was slight to the left side by 11 mm. The physical examination revealed slight coronal decompensation to the left side, similar to the X-ray result. The physical examination revealed that the

shoulder level appeared to be equal. The left waist level was higher than the right, but the right axilla level was higher than the left. She had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 19.95 kg/m<sup>2</sup>. She had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 6's curve pattern was a double-curve with right thoracic and left lumbar curves. After clinical consideration, six forces were applied to correct the deformity. Firstly, the lumbar corrective force was at the curve's apex from L2 to L3 on the left side. This force was applied in the anterior and medial directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Then, the corrective force for the thoracic curve was located on the right side at the curve's apex from T9 to T11. This force was applied in the anterior, medial, and upward directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. This corrective force also acted as the counter force for the lumbar corrective force. The inferior counterforce at the right gluteus medius was another counterforce required to counteract the lumbar corrective force. The left gluteus medius force was then applied to stabilise the pelvis. The superior counterforce at the axilla on the left side was another counterforce required to counteract the thoracic corrective force. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.26 and 9.27 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs in assessment and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.18 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.19 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal

plane. The curve magnitudes were reduced, and the spinal column looked was straighter. The coronal decompensation was improved and closed to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

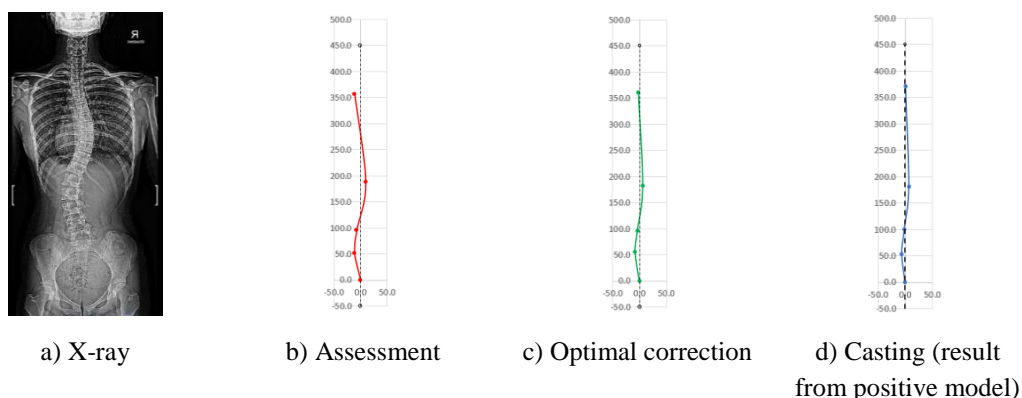


Figure 9.26 Spinal column in the coronal plane of participant 6, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

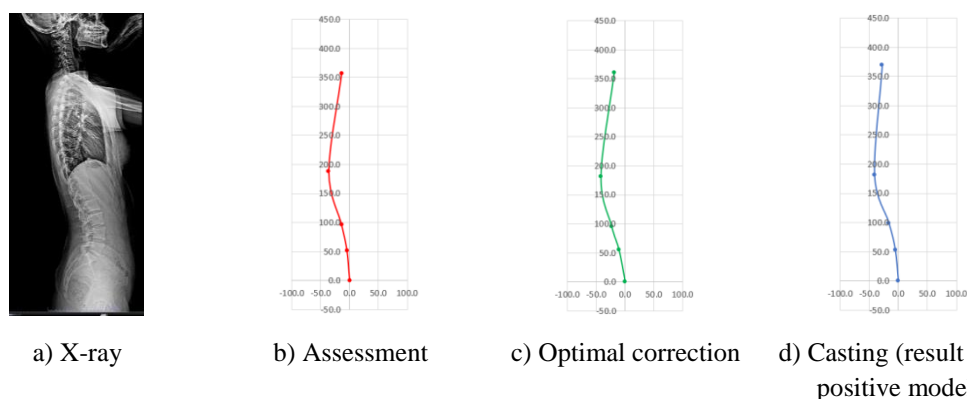


Figure 9.27 Spinal column in the sagittal plane of participant 6, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.18 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 6.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	97.00	97.02	92.55	91.80
CLTA, degree	80.00	79.62	83.38	84.02
CULA, degree	77.00	84.67	84.42	84.34
CLLA, degree	105.00	102.19	97.82	96.49

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
SUTA, degree	87.00	82.49	82.80	85.94
SLTA, degree	102.00	103.64	102.00	106.93
SULA, degree	106.00	101.78	107.11	103.47
SLLA, degree	97.00	95.17	100.84	95.70
Trunk balance, mm	-11.00	-11.05	-1.56	1.19

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that all spinal parameters were in a similar colour zone, and the coronal spinal deformity was mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same zone, and some improved to a better zone. Overall, the deformity improved, but the deformity was still in the mal-alignment zone after applying the force to correct the deformity.

SSA between radiographs and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal spinal deformity varied between normal and mal-alignment. For SSA from the motion system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone, and some were improved to a better zone. Overall, the deformity varied between normal and mal-alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone. Some improved the deformity, and some did not, but increased deformity. The deformity was mostly in potentially abnormal alignment after applying the force to correct the deformity.

Table 9.19 Results of a low-cost postural measurement system for participant 6 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	79.77	82.37	85.57
3D-LTA, degree	106.91	103.61	107.84
3D-ULA, degree	102.87	107.90	104.53
3D-LLA, degree	103.18	103.26	98.61
Trunk rotation at T7, degree	4.18		3.36
Trunk rotation at T12, degree	2.19		-7.37
Trunk rotation at L2, degree	-1.19		0.00
Trunk rotation at L5, degree	-3.05		-2.93
Shoulder level, mm	-2.81		-1.13
Axilla level, mm	5.44		0.64
Waist level, mm	-22.27		-21.78
Distance at axilla level, mm	12.12		9.28
Distance at waist level, mm	14.81		0.69
POTSI for trunk balance, percentage	3.97		0.44
POTSI for the left and right distance at axilla, percentage	4.35		3.42
POTSI for the left and right distance at waist, percentage	6.42		0.30
POTSI for shoulder level, percentage	0.79		0.31
POTSI for axilla level, percentage	1.52		0.17
POTSI for waist level, percentage	6.23		5.88
POTSI in total, percentage	23.28		10.52

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.

Figure 9.28 illustrates a graph representing the trunk's horizontal rotation. Table 9.19 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T12 level or around the level of the apical vertebrae of the thoracic curve.

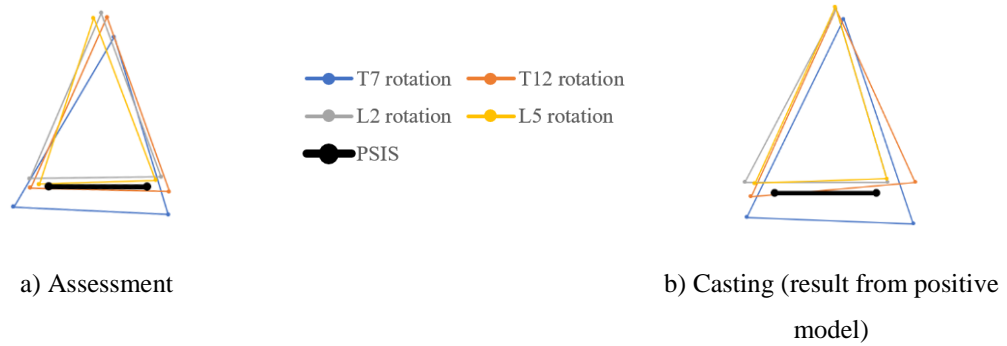


Figure 9.28 Spinal parameters in the transverse plane of participant 6, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.29 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.19 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

The result showed that the shoulder and waist levels remained the same, but the axilla level was improved. The left and right distances at the waist level was improved, but the axilla remained the same. The total POTSI index between assessment and casting was considerably decreased (from 23.28% to 10.52%). As a result, the shape of the trunk in the coronal plane obviously changed or improved after applying the forces to correct the spine.

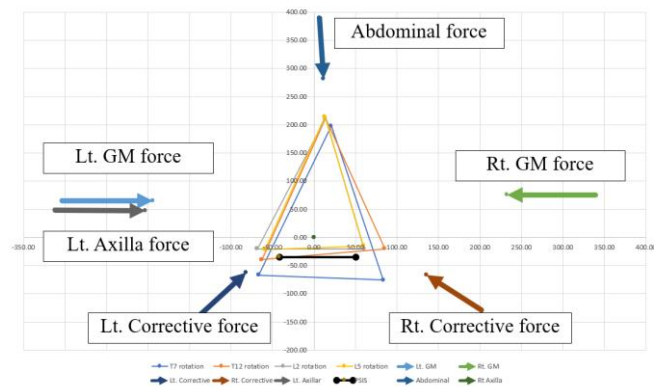


Figure 9.29 Spinal parameters in the trunk asymmetry of participant 6, where a) and b) the results from a postural measurement system during the assessment and casting.

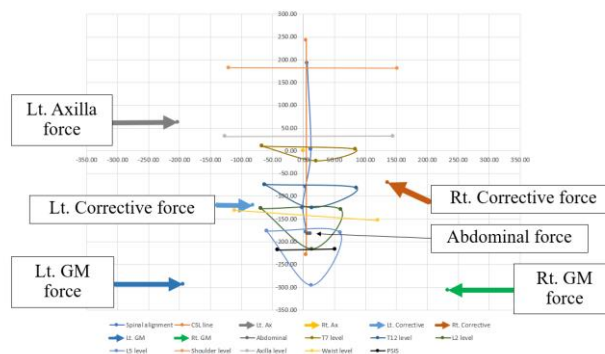


Figure 9.30 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 6 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.20 reports the magnitude of forces applied to Participant 6's trunk to correct the deformity.

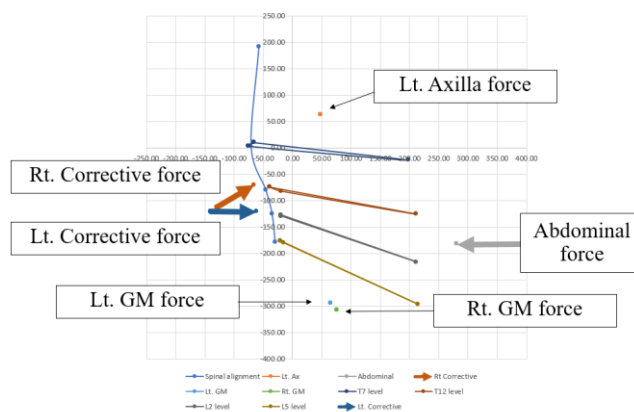
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 17.20 N to 47.86 N, and the total force was 169.79 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the lumbar curve towards the centre in the coronal plane and derotate the spine in the transverse plane. The thoracic corrective force, providing anterior, medial, and upward-directed force, could also shift the thoracic curve towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.30 Forces applied to participant 6's trunk in a) the transverse, b) coronal, and c) sagittal planes.

Table 9.20 Result of the forces that were applied to participant 6's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	47.86
Right corrective force	32.70
Left axilla force	29.53
Left gluteus medius force	18.09
Right gluteus medius force	17.20
Abdominal force	24.41
Total	169.79

### Participant 7

Participant 7 was a 14-year-seven-month-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a single curve to the left side with an apex at L1 and the upper- and lower-end vertebra at T11 and L4, respectively. The coronal Cobb angle at the thoracic curve was 40°. The Risser sign was grade 4. She had her period for two years, or since she was twelve. The coronal decompensation was slight to the left side by 30 mm. The physical examination revealed that the right shoulder, waist, and axilla were higher than the left side. She had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 18.75 kg/m<sup>2</sup>. She had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 7's curve pattern was a single left lumbar curve. After clinical consideration, six forces were applied to correct the deformity. Firstly, the lumbar corrective force was at the curve's apex from L1 to L3 on the left side. This force was applied in the anterior and medial directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T9 to T11 on the right side and the inferior counterforce at the gluteus medius on the right side. The left gluteus medius force was then applied to stabilise the pelvis. To reduce the coronal decompensation, the stabilising force was applied in the medial direction at the left side of the axilla area. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.31 and 9.32 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.21 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.22 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column was shifted from the left to the centre and looked was straighter. The coronal decompensation was improved and closed to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

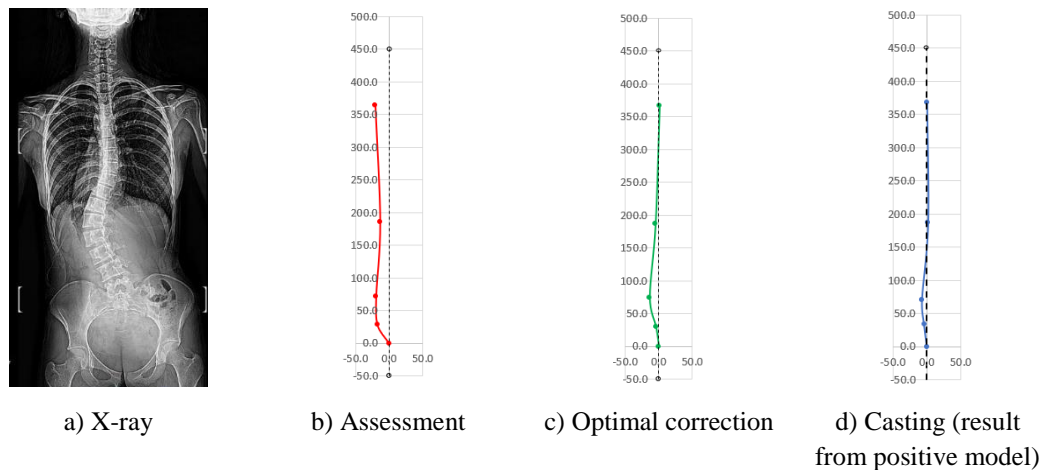


Figure 9.31 Spinal column in the coronal plane of participant 7, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

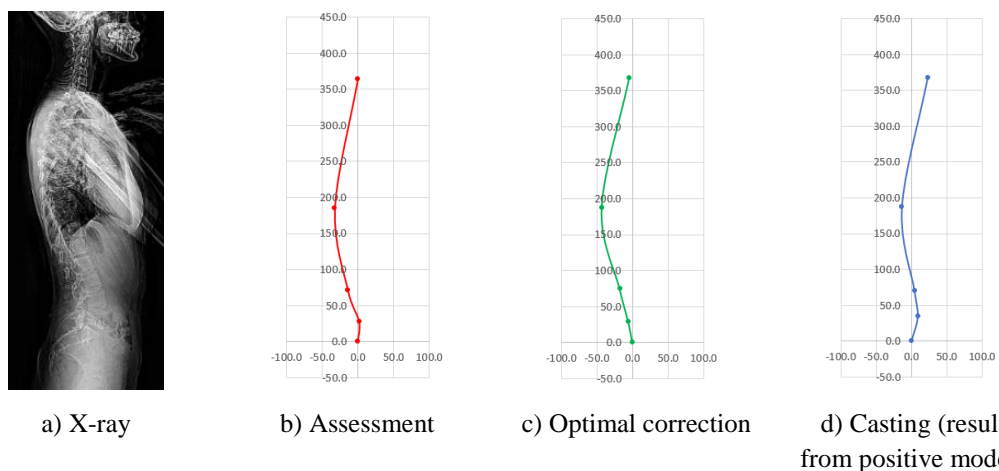


Figure 9.32 Spinal column in the sagittal plane of participant 7, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.21 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 7.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	94.00	92.36	88.10	90.47
CLTA, degree	80.00	86.90	85.72	85.71
CULA, degree	86.00	92.42	101.47	94.33
CLLA, degree	118.00	121.76	98.21	96.71
SUTA, degree	79.00	79.82	77.94	78.57
SLTA, degree	99.00	99.04	102.62	98.72
SULA, degree	105.00	109.80	104.66	97.33
SLLA, degree	86.00	86.79	101.30	75.03
Trunk balance, mm	-30.00	-20.47	1.08	0.56

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA

= Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity was between potential abnormal and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same zone. Some improved to a better zone, and some did not improve but increased the deformity. Overall, the deformity improved, and the deformity was between normal and mal-alignment after applying the force to correct the deformity.

SSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal spinal deformity was mostly between normal and potential abnormal alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same colour zone, and the deformity was mostly normal alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone. Some improved the deformity and changed to a better zone. The deformity was mostly normal alignment after applying the force to correct the deformity.

Table 9.22 Results of a low-cost postural measurement system for participant 7 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	79.56	77.80	78.56
3D-LTA, degree	99.54	103.28	99.69
3D-ULA, degree	109.93	108.31	98.49
3D-LLA, degree	58.13	103.85	73.72
Trunk rotation at T7, degree	-5.96		2.76
Trunk rotation at T12, degree	-13.07		-2.21
Trunk rotation at L2, degree	-6.90		-0.83
Trunk rotation at L5, degree	-9.07		1.49

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
Shoulder level, mm	20.21		-5.64
Axilla level, mm	21.88		-12.84
Waist level, mm	61.09		50.79
Distance at axilla level, mm	-13.94		5.48
Distance at waist level, mm	-4.84		0.20
POTSI for trunk balance, percentage	7.51		0.21
POTSI for the left and right distance at axilla, percentage	5.11		2.06
POTSI for the left and right distance at waist, percentage	2.23		0.10
POTSI for shoulder level, percentage	5.55		1.53
POTSI for axilla level, percentage	6.01		3.49
POTSI for waist level, percentage	16.77		13.82
POTSI in total, percentage	43.18		21.21

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre. n/a = not applicable or cannot measure.

Figure 9.33 illustrates a graph representing the trunk's horizontal rotation. Table 9.22 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process. The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the T12 and L2 levels or around the level of the apical vertebrae of the lumbar curve.

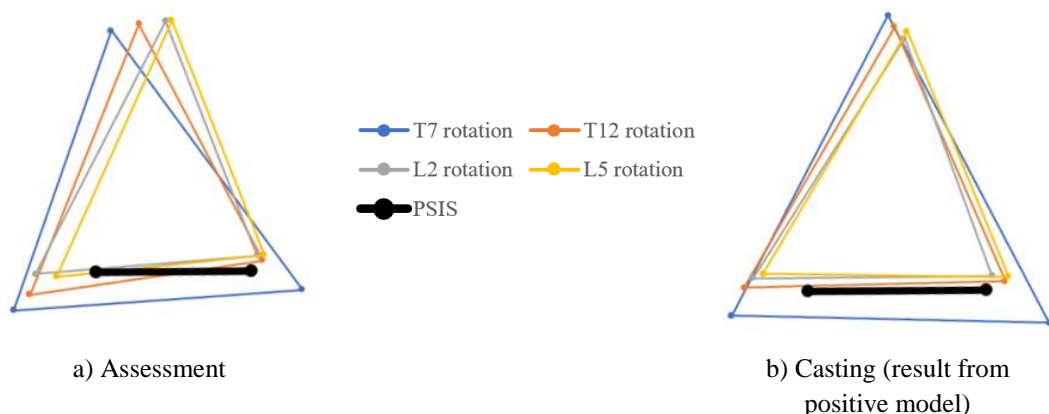


Figure 9.33 Spinal parameters in the transverse plane of participant 7, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.34 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.22 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces. The result showed that the shoulder level was improved, the waist level remained the same, but the axilla level went to the opposite side (from right to left sides). The left and right distances at the axilla and waist levels were improved. The total POTSI index between assessment and casting was considerably decreased (from 43.18% to 21.21%). As a result, the shape of the trunk in the coronal plane obviously changed or improved after applying the forces to correct the spine.

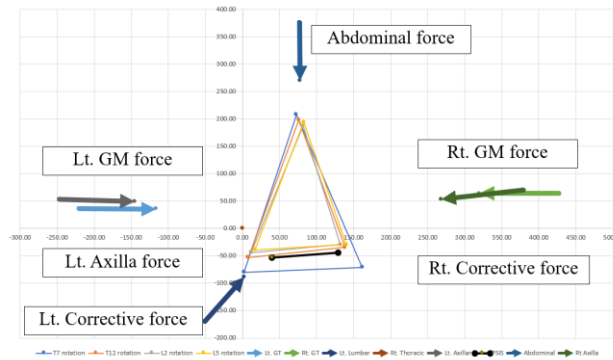


Figure 9.34 Spinal parameters in the trunk asymmetry of participant 7, where a) and b) the results from a postural measurement system during the assessment and casting.

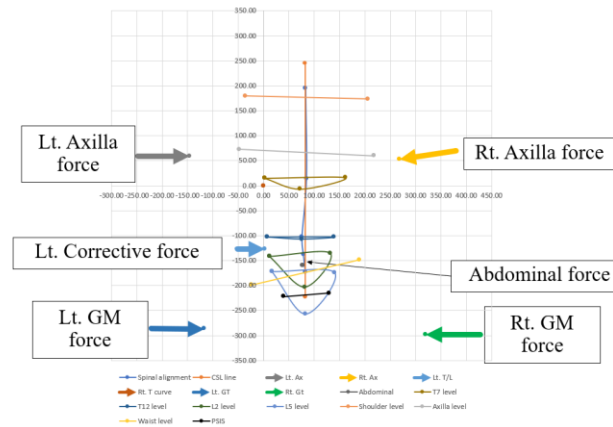
Figure 9.35 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 7 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.23 reports the magnitude of forces applied to Participant 7's trunk to correct the deformity.

The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 20.84 N to 46.47 N, and the total force was 181.07 N. Furthermore, the postural measurement system could illustrate the force direction

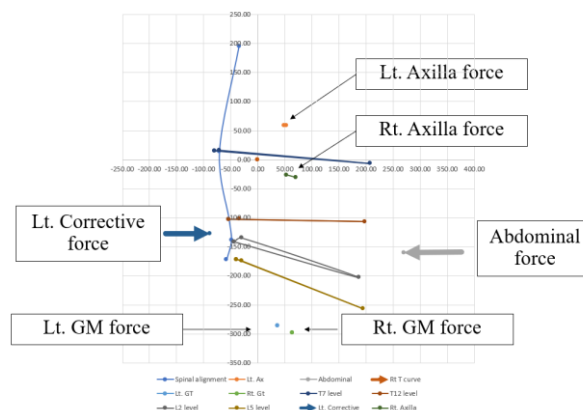
in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the lumbar curve towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.35 Forces applied to participant 7's trunk in a) the transverse, b) coronal, and c) sagittal planes.



Table 9.23 Result of the forces that were applied to participant 7's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	45.24
Left axilla force	22.63
Right axilla force	46.47
Left gluteus medius force	10.44
Right gluteus medius force	20.84
Abdominal force	35.45
Total	181.07

## Participant 8

Participant 8 was a 17-year-four-month-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a single curve to the left side with an apex at L2 and the upper- and lower-end vertebra at T11 and L4, respectively. The coronal Cobb angle at the thoracic curve was 29°. The Risser sign was grade 5. She had her period for three years, or since she was fourteen. The coronal decompensation was slight to the left side by 18 mm. The physical examination revealed that the right shoulder, waist, and axilla were higher than the left side. She had no LLD. A spinal curve's flexibility was semi-rigid. The BMI was 19.33 kg/m<sup>2</sup>. She had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 8's curve pattern was a single left lumbar curve. After clinical consideration, five forces were applied to correct the deformity. Firstly, the lumbar corrective force was at the curve's apex from L2 to L4 on the left side. This force was applied in the anterior and medial directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T7 to T9 on the right side and the inferior counterforce at the gluteus medius on the right side. The left gluteus medius force was then applied to stabilise the pelvis. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.36 and 9.37 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between

radiographs in assessment and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.24 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.25 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column was shifted from the left to the centre and looked was straighter. The coronal decompensation was improved and closed to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

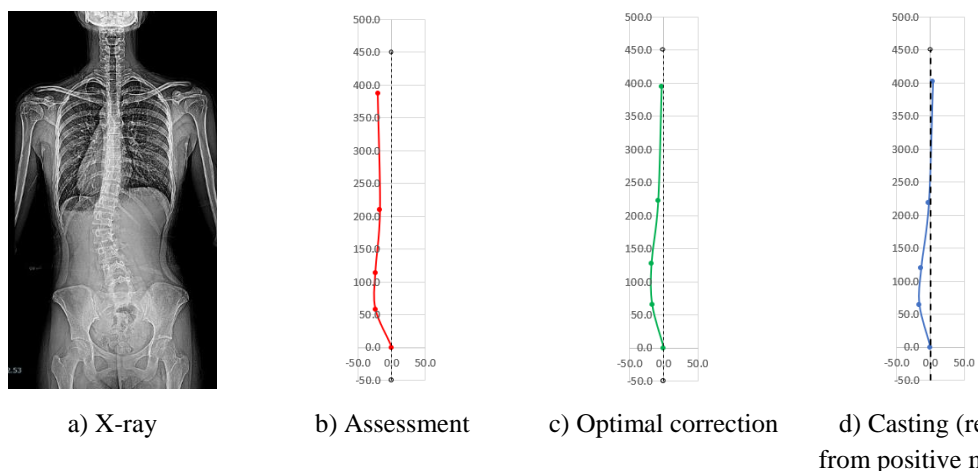


Figure 9.36 Spinal column in the coronal plane of participant 8, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

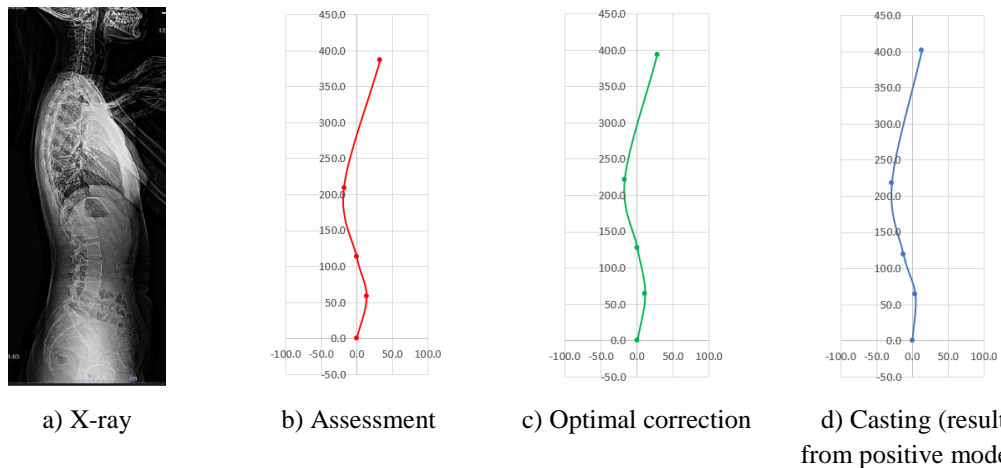


Figure 9.37 Spinal column in the sagittal plane of participant 8, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.24 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 8.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	91.00	91.04	91.64	91.90
CLTA, degree	85.00	85.94	84.24	83.55
CULA, degree	90.00	90.35	91.00	87.36
CLLA, degree	106.00	112.20	104.22	104.10
SUTA, degree	80.00	74.04	75.31	77.24
SLTA, degree	100.00	100.61	100.58	99.24
SULA, degree	100.00	104.14	99.83	106.45
SLLA, degree	82.00	76.59	80.50	86.21
Trunk balance, mm	-18.00	-20.55	-3.08	3.77

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that all spinal parameters were in a similar colour zone, and the coronal spinal deformity was between normal and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, all spinal segments remained in the same colour zone. Overall, the deformity improved, but the change was still in the same deformity zone or between normal and mal-alignment after applying the force to correct the deformity.

SSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal spinal deformity was between normal and potential abnormal alignment (mostly in normal). For SSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same colour zone, and the deformity was mostly normal alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone. Some improved the deformity and changed to a better zone. The deformity was mostly normal alignment after applying the force to correct the deformity.

Table 9.25 Results of a low-cost postural measurement system for participant 8 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	74.01	75.22	77.11
3D-LTA, degree	101.33	101.99	101.21
3D-ULA, degree	104.15	99.88	106.64
3D-LLA, degree	64.70	73.11	75.44
Trunk rotation at T7, degree	-1.14		2.67
Trunk rotation at T12, degree	-0.71		5.72
Trunk rotation at L2, degree	-14.28		10.41
Trunk rotation at L5, degree	1.99		3.50
Shoulder level, mm	16.14		-8.21
Axilla level, mm	19.99		-5.35
Waist level, mm	6.87		-0.73
Distance at axilla level, mm	-19.35		29.75
Distance at waist level, mm	-12.74		-28.23
POTSI for trunk balance, percentage	6.77		1.26
POTSI for the left and right distance at axilla, percentage	6.37		9.91
POTSI for the left and right distance at waist, percentage	6.16		12.13
POTSI for shoulder level, percentage	4.17		2.04
POTSI for axilla level, percentage	5.16		1.33
POTSI for waist level, percentage	1.77		0.18
POTSI in total, percentage	30.39		26.84

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.



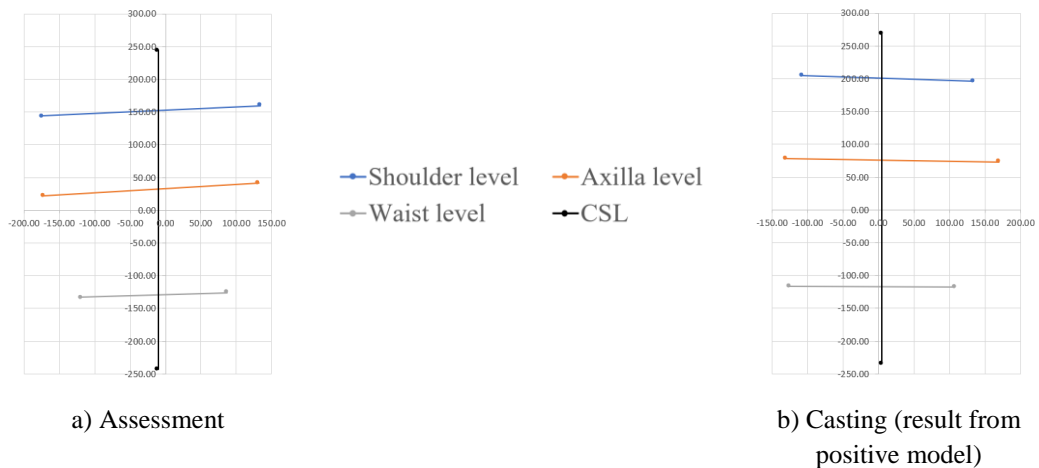
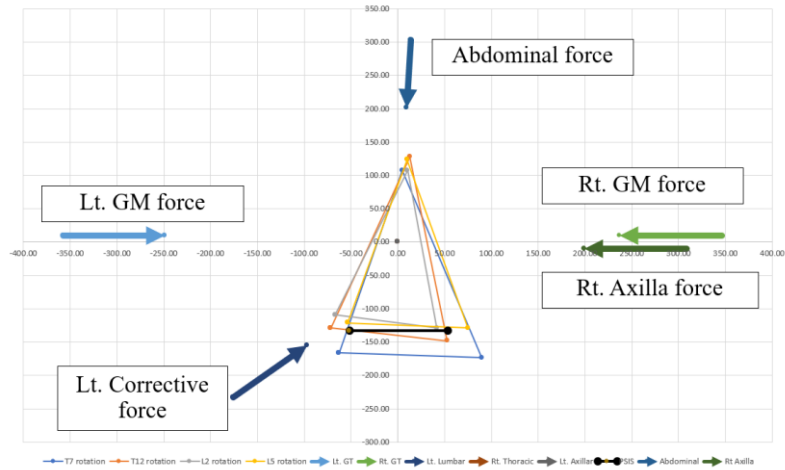


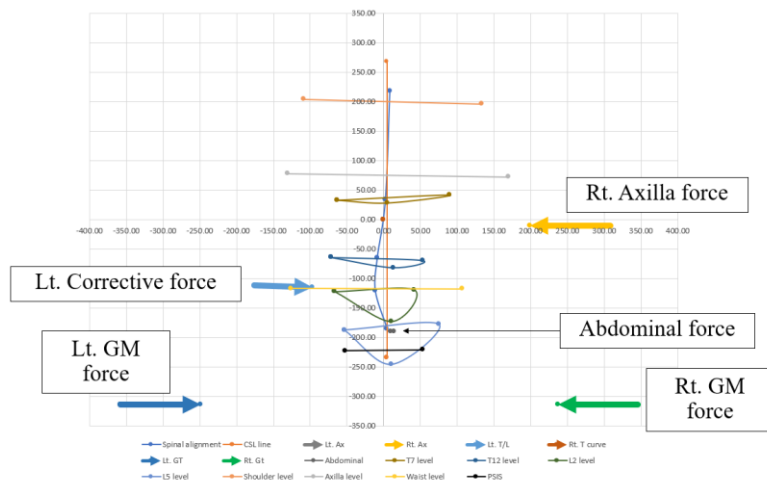
Figure 9.39 Spinal parameters in the trunk asymmetry of participant 8, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.40 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 8 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.26 reports the magnitude of forces applied to Participant 8's trunk to correct the deformity.

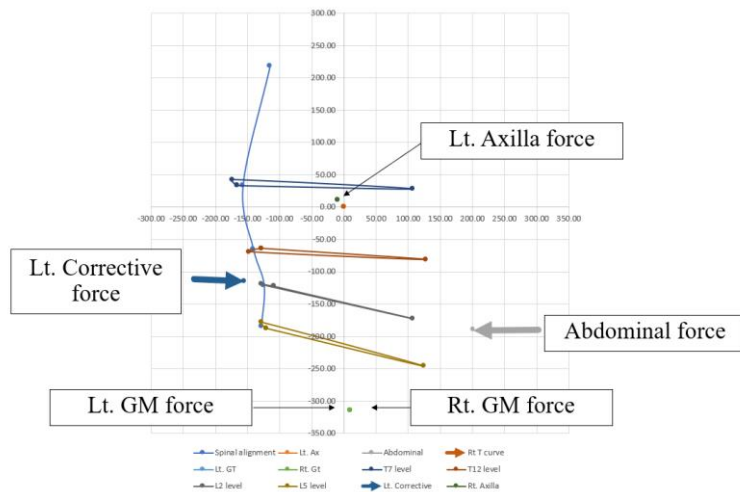
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 17.48 N to 24.93 N, and the total force was 101.90 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the lumbar curve towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.40 Forces applied to participant 8's trunk in the transverse plane.

Table 9.26 Result of the forces that were applied to participant 8's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	22.11
Right axilla force	17.48
Left gluteus medius force	15.89
Right gluteus medius force	21.48
Abdominal force	24.93
Total	101.90

### Participant 9

Participant 9 was a 15-year-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a single curve to the right side with an apex at T10 and the upper- and lower-end vertebra at T6 and L1, respectively. The coronal Cobb angle at the thoracic curve was 40°. The Risser sign was grade 3. She had her period for two months. The coronal decompensation was not present. The physical examination revealed that the right shoulder seemed level. The right axilla was higher than the left side, but the left waist was higher than the right waist. She had no LLD. A spinal curve's flexibility was flexible. The BMI was 18.67 kg/m<sup>2</sup>. She had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 9's curve pattern was a single right thoracic curve. After clinical consideration, five forces were applied to correct the deformity. Firstly, the thoracic corrective force was at the curve's apex from T10 to T12 on the right side. This force was applied in the anterior, medial, and upward directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. Another two main counterforces were required to counteract this corrective force in the medial direction: the superior counterforce at T6 to T8 on the left side and the inferior counterforce at the gluteus medius on the left side. The right gluteus medius force was then applied to stabilise the pelvis. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.41 and 9.42 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between



radiographs in assessment and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.27 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.28 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column was shifted from the right to the centre and looked was straighter. The coronal decompensation was improved and closed to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

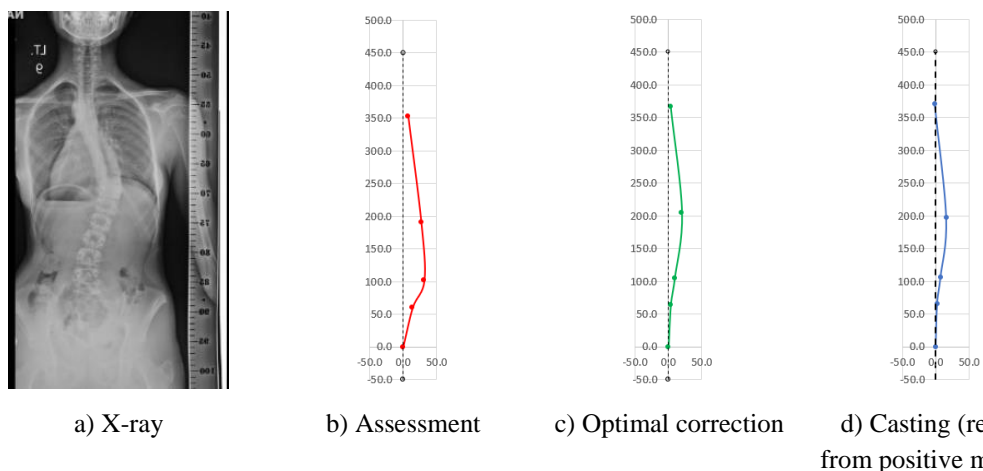


Figure 9.41 Spinal column in the coronal plane of participant 9, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

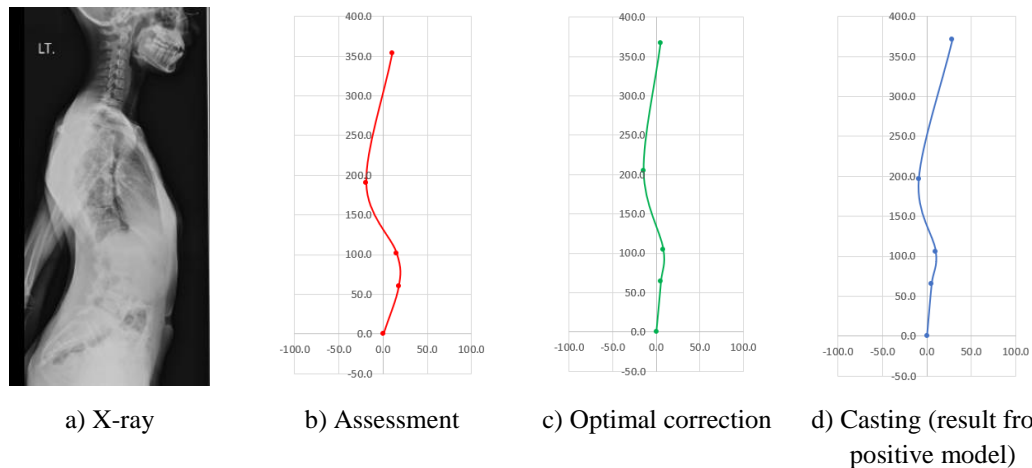


Figure 9.42 Spinal column in the sagittal plane of participant 9, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.27 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 9.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	98.00	96.77	95.96	95.64
CLTA, degree	98.00	87.39	82.64	83.21
CULA, degree	68.00	67.62	84.80	86.00
CLLA, degree	80.00	76.90	86.98	88.10
SUTA, degree	75.00	79.74	83.17	77.86
SLTA, degree	105.00	108.92	102.80	101.90
SULA, degree	107.00	93.76	86.06	83.50
SLLA, degree	84.00	72.05	85.25	85.39
Trunk balance, mm	0.00	7.20	2.99	-1.45

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity of this participant was between potential abnormal and mal-alignment (mostly in mal-alignment). For CSA from the postural measurement system in assessment VS optimal correction and casting, some spinal segments changed to a better zone. However, some remained in the same colour zone (red zone). Overall, the deformity was improved and ranged between normal and mal-alignment after applying the force to correct the deformity.

SSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal spinal deformity of this participant was between normal and potential abnormal alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone. However, some increased the deformity. The deformity varied between normal and mal-alignment after applying the force to correct the deformity.

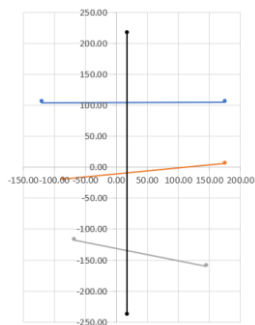
For 3DSA from the postural measurement system in assessment VS optimal correction and casting, some spinal parameters remained in the same colour zone. Some improved the deformity, and some did not improve but increased the deformity. The deformity varied between normal and mal-alignment after applying the force to correct the deformity.

Table 9.28 Results of a low-cost postural measurement system for participant 9 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	77.78	80.97	76.68
3D-LTA, degree	109.08	104.65	103.02
3D-ULA, degree	112.57	83.49	82.30
3D-LLA, degree	69.60	84.38	85.01
Trunk rotation at T7, degree	-2.85		-2.44
Trunk rotation at T12, degree	4.38		1.28
Trunk rotation at L2, degree	-2.60		2.32
Trunk rotation at L5, degree	-3.39		0.07
Shoulder level, mm	0.12		-0.27
Axilla level, mm	25.15		6.69
Waist level, mm	-42.11		-26.44
Distance at axilla level, mm	56.02		4.94
Distance at waist level, mm	44.22		18.01
POTSI for trunk balance, percentage	2.75		0.56
POTSI for the left and right distance at axilla, percentage	21.43		1.90
POTSI for the left and right distance at waist, percentage	20.93		8.44
POTSI for shoulder level, percentage	0.03		0.07
POTSI for axilla level, percentage	7.11		1.80
POTSI for waist level, percentage	11.91		7.12
POTSI in total, percentage	64.18		19.89

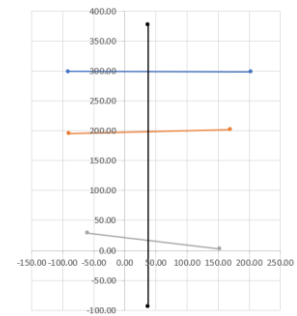
Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.





a) Assessment

— Shoulder level — Axilla level  
— Waist level — CSL

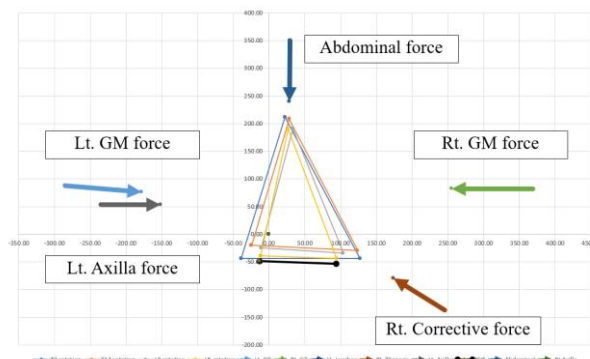


b) Casting (result from positive model)

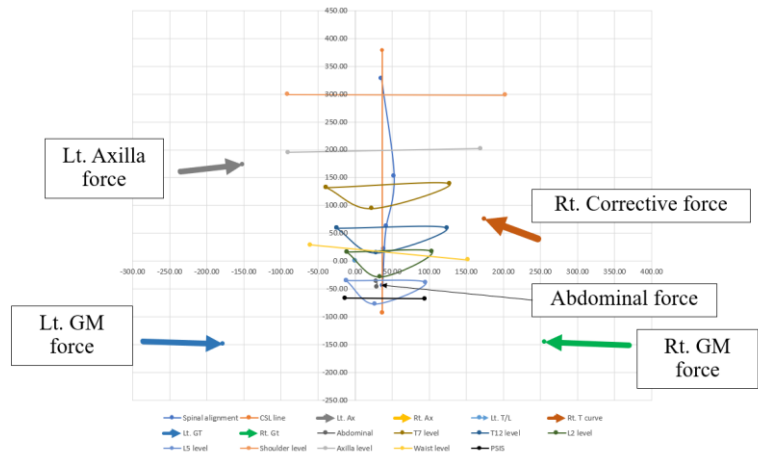
Figure 9.44 Spinal parameters in the trunk asymmetry of participant 9, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.45 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 9 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.29 reports the magnitude of forces applied to Participant 9's trunk to correct the deformity.

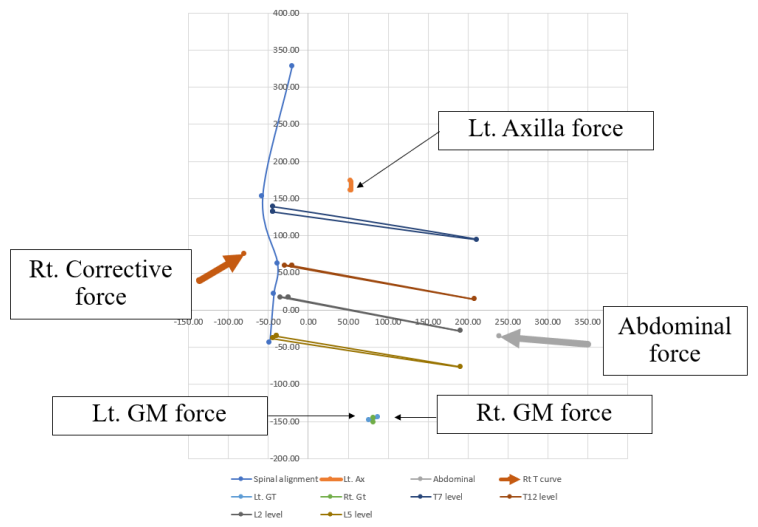
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 22.44 N to 53.26 N, and the total force was 178.21 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The thoracic corrective force, providing medial, anterior, and upward-directed force, could shift the thoracic curve towards the centre in the coronal plane and derotate the spine in the transverse plane. Other forces were well counteracted by the corrective force.



a)



b)



c)

Figure 9.45 Forces applied to participant 9's trunk in a) the transverse, b) coronal, and c) sagittal planes.

Table 9.29 Result of the forces that were applied to participant 9's trunk.

Area of Forces	Amount of Force (N)
Right corrective force	49.24
Left axilla force	53.26
Left gluteus medius force	29.79
Right gluteus medius force	22.44
Abdominal force	23.47
Total	178.21

## Participant 10

Participant 10 was a 13-year-five-month-old girl who was diagnosed with AIS by a rehabilitation doctor. The radiographic evaluation revealed that her spinal curve pattern was a double-curve, with right thoracic and left lumbar curves. At the thoracic curve, the apical vertebra was at T8, and the upper- and lower-end vertebra were at T6 and T11, respectively. The coronal Cobb angle at the thoracic curve was 20°. At the lumbar curve, the apical vertebra was at L1, and the upper- and lower-end vertebra were at T11 and L4, respectively. The coronal Cobb angle at the lumbar curve was 27°. The Risser sign was grade 3. The coronal decompensation was slight to the left side by 10 mm. The physical examination revealed slight coronal decompensation to the left side, similar to the X-ray result. The physical examination revealed that the right shoulder and waist were higher than the left side, and the axilla seemed level. She had no LLD. A spinal curve's flexibility was flexible. The BMI was 18.83 kg/m<sup>2</sup>. She had no history of spinal surgery or underlying disease, psychological or communication problems, and could stand and walk normally.

Participant 10's curve pattern was a double-curve, with right thoracic and left lumbar curves. After clinical consideration, six forces were applied to correct the deformity. Firstly, the lumbar corrective force was at the curve's apex from L1 to L3 on the left side. This force was applied in the anterior and medial directions to centralise the spine in the coronal plane and derotate the curve in the transverse plane. The corrective force for the thoracic curve was located on the right side at the curve's apex from T8 to T10. This force was applied mainly in the medial direction and had to be applied cautiously because it would shift the spine to the left as the apical vertebra was now located at the CSL line. This corrective force also acted as the counter force for the lumbar corrective force. The inferior counterforce at the right gluteus medius was another counterforce required to counteract the lumbar corrective force. The left gluteus medius force was then applied to stabilise the pelvis. The superior counterforce at the axilla on the left side was another counterforce required to counteract the thoracic corrective force. Finally, the abdominal force was applied to counteract with corrective force around the abdomen area, and the force was in the posterior direction.

Figures 9.46 and 9.47 illustrate the spinal column in the coronal and sagittal planes, respectively. All figures show the comparison of the spinal column between radiographs in assessment and a postural measurement system during the assessment, optimal correction, and casting processes. Table 9.30 reports the results of spinal parameters in the sagittal and coronal planes from radiographs and a postural measurement system. Table 9.31 reports the results of a postural measurement system for other spinal parameters, including the angle of the spinal column in 3D, trunk horizontal rotation, and trunk asymmetry.

The result showed that the graphs from the postural measurement system during assessment could replicate and illustrate the spinal alignment similar to the radiographs. During the optimal correction and casting, the graphs from the postural measurement system could illustrate the change in the spinal alignment in the coronal plane. The spinal column was shifted from the left to the centre and looked was straighter. The coronal decompensation was improved and closed to the centre. In the sagittal plane, the graphs from the postural measurement system could illustrate a slight change in the spinal column, but the overall curve was still maintained. The postural measurement system gave the results of the spinal angle in 3D. The following section analysed all spinal angles in detail using descriptive and inferential statistics.

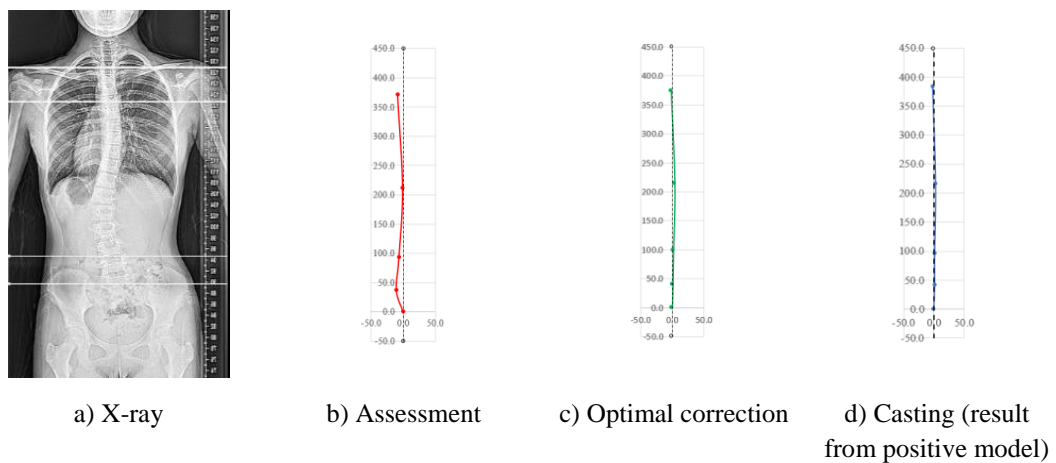


Figure 9.46 Spinal column in the coronal plane of participant 10, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.



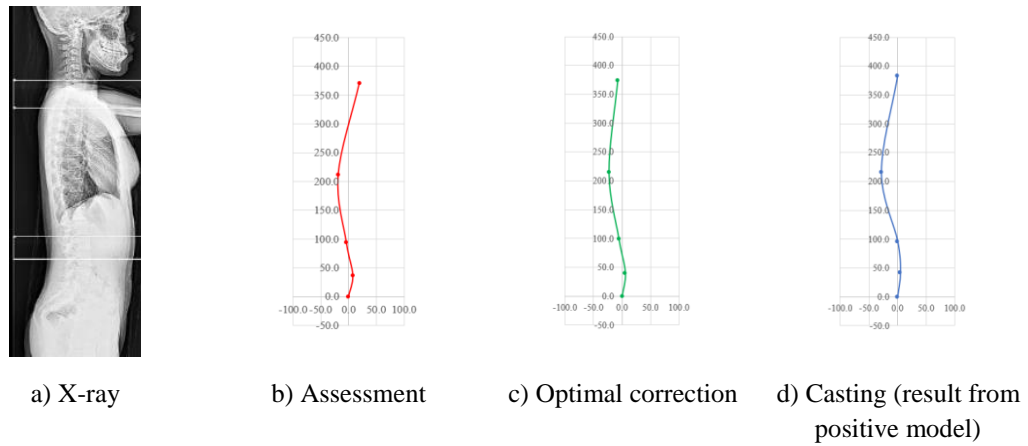


Figure 9.47 Spinal column in the sagittal plane of participant 10, where a) X-ray image and b) to d) the results from a postural measurement system during assessment, optimal correction, and casting.

Table 9.30 Results of spinal parameters in the sagittal and coronal planes and coronal decompensation from an X-ray and a low-cost postural measurement system for participant 10.

Spinal Parameters	Result from X-ray in Assessment	Result from postural measurement system		
		Assessment	Optimal	Casting
CUTA, degree	93.00	92.70	92.03	91.80
CLTA, degree	85.00	87.33	89.05	89.16
CULA, degree	82.00	86.12	89.08	89.93
CLLA, degree	105.00	105.53	88.63	88.10
SUTA, degree	80.00	76.60	84.52	80.53
SLTA, degree	95.00	97.19	98.53	103.12
SULA, degree	107.00	101.41	100.80	95.07
SLLA, degree	82.00	78.24	82.03	83.50
Trunk balance, mm	-10.00	-8.43	-1.79	-2.06

Note: CUTA = Coronal upper thoracic angle. CLTA = Coronal lower thoracic angle. CULA = Coronal upper lumbar angle. CLLA = Coronal lower lumbar angle. SUTA = Sagittal upper thoracic angle. SLTA = Sagittal lower thoracic angle. SULA = Sagittal upper lumbar angle. SLLA = Sagittal lower lumbar angle. mm = millimetre.

Considering the alignment zone based on Jang's study (Jang, 2018), CSA between radiographs in assessment and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the coronal spinal deformity of this participant was between potential abnormal and mal-alignment. For CSA from the postural measurement system in assessment VS optimal correction and casting, all spinal segments changed to the green zone, or the deformity changed to normal alignment after applying the force to correct the deformity.

SSA between radiographs and the postural measurement system in assessment showed that most spinal parameters were in a similar colour zone, and the sagittal

spinal deformity of this participant was between normal and potential abnormal alignment. For SSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters remained in the same colour zone, or the deformity remained in normal alignment after applying the force to correct the deformity.

For 3DSA from the postural measurement system in assessment VS optimal correction and casting, most spinal parameters had changed but remained in the same colour zone, or the deformity remained in normal alignment after applying the force to correct the deformity.

Table 9.31 Results of a low-cost postural measurement system for participant 10 for spinal parameters in 3D, trunk rotation, and trunk asymmetry.

Spinal Parameters	Result from postural measurement system		
	Assessment	Optimal	Casting
3D-UTA, degree	76.35	84.16	80.37
3D-LTA, degree	97.66	98.58	103.15
3D-ULA, degree	102.02	100.84	95.07
3D-LLA, degree	76.98	81.91	83.23
Trunk rotation at T7, degree	1.47		3.54
Trunk rotation at T12, degree	-4.24		-4.93
Trunk rotation at L2, degree	-1.86		-1.65
Trunk rotation at L5, degree	-6.12		-0.96
Shoulder level, mm	8.34		-3.68
Axilla level, mm	1.36		3.65
Waist level, mm	6.84		16.71
Distance at axilla level, mm	-32.18		-0.39
Distance at waist level, mm	1.08		-2.48
POTSI for trunk balance, percentage	2.63		0.70
POTSI for the left and right distance at axilla, percentage	10.04		0.13
POTSI for the left and right distance at waist, percentage	0.47		1.06
POTSI for shoulder level, percentage	2.25		0.96
POTSI for axilla level, percentage	0.37		0.95
POTSI for waist level, percentage	1.84		4.36
POTSI in total, percentage	17.59		8.16

Note: 3D-UTA = 3-dimensional upper thoracic angle. 3D-LTA = 3-dimensional lower thoracic angle. 3D-ULA = 3-dimensional upper lumbar angle. 3D-LLA = 3-dimensional lower lumbar angle. T = Thoracic. L = Lumbar. mm = millimetre.

Figure 9.48 illustrates a graph representing the trunk's horizontal rotation. Table 9.31 also reports the angles of horizontal trunk rotation at each level. The graphs show the change in trunk rotation compared between the assessment and the casting process.

The postural measurement system could illustrate the horizontal trunk rotation change before and when applying the forces to correct the deformity. As can be seen from the graph and table, the rotation decreased from the assessment to the casting process, especially at the L5 level or around the level of the lumbar curve.

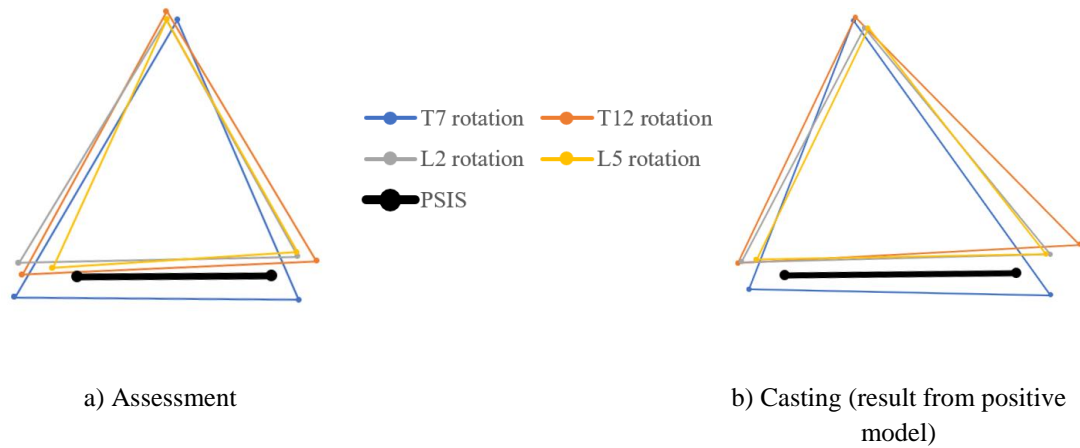


Figure 9.48 Spinal parameters in the transverse plane of participant 10, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.49 illustrates a graph representing the spinal parameters in the trunk asymmetry in the coronal plane. Table 9.31 also reports the results of the trunk asymmetry in the coronal plane and POTSI index. The graphs and results in the table from the postural measurement system could show these parameters in a way that was similar to the physical examination (before applying the forces). The postural measurement system could also report the actual value and illustrate the change in these parameters when applying forces.

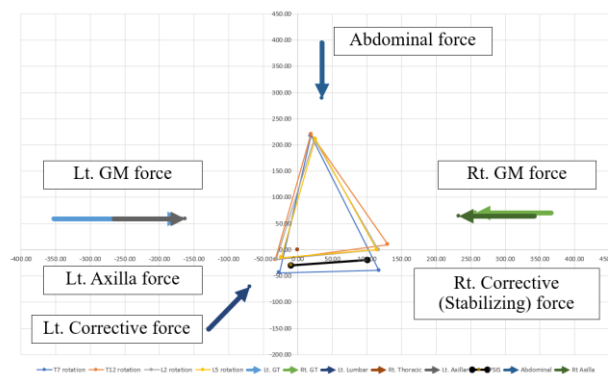
The result showed that the axilla level remained at the level, the shoulder level went to the opposite side (from right to left sides), and the waist level increased on the right side. The left and right distances at the waist remained the same, and the axilla was improved. The total POTSI index between assessment and casting was considerably decreased (from 17.59% to 8.16%). As a result, the shape of the trunk in the coronal plane obviously changed or improved after applying the forces to correct the spine.



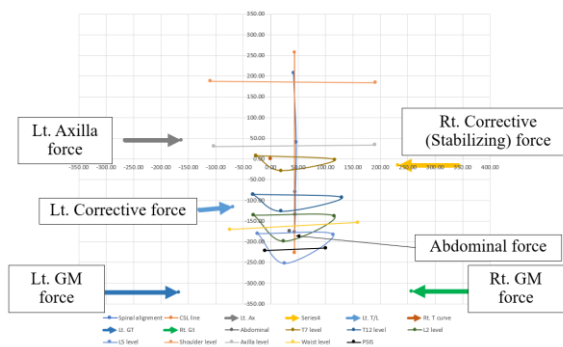
Figure 9.49 Spinal parameters in the trunk asymmetry of participant 10, where a) and b) the results from a postural measurement system during the assessment and casting.

Figure 9.50 illustrates a graph representing the corrective force and counterforces used to correct the spinal deformity in 3 dimensions for Participant 10 during the casting process. This figure is illustrated in the transverse, coronal, and sagittal planes, respectively. Finally, Table 9.32 reports the magnitude of forces applied to Participant 10's trunk to correct the deformity.

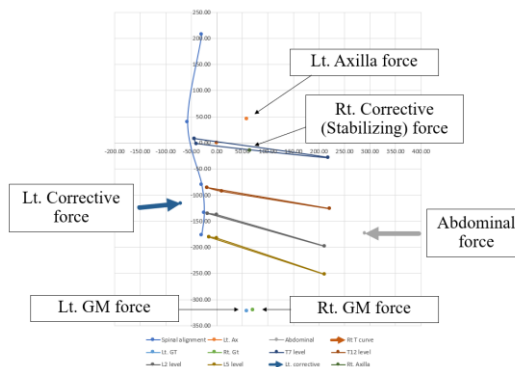
The system could report the results of the force magnitudes to correct the deformity. The forces ranged from 11.01 N to 35.31 N, and the total force was 153.09 N. Furthermore, the postural measurement system could illustrate the force direction in three dimensions. The system illustrated the force locations and directions the clinician aims to achieve during casting. The lumbar corrective force, providing medial and anterior-directed force, could shift the lumbar curve towards the centre in the coronal plane and derotate the spine in the transverse plane. The thoracic corrective force, providing mainly medial-directed force, could reduce the curve magnitude and stabilise the spine. Other forces were well counteracted by the corrective forces.



a)



b)



c)

Figure 9.50 Forces applied to participant 10's trunk in the transverse plane.

Table 9.32 Result of the forces that were applied to participant 10's trunk.

Area of Forces	Amount of Force (N)
Left corrective force	30.85
Right corrective (Stabilizing) force	35.31
Left axilla force	21.75
Left gluteus medius force	23.27
Right gluteus medius force	11.01
Abdominal force	30.90
Total	153.09

### 9.3. Conclusion

Ten AIS patients interested in joining the research project and following the inclusion criteria were recruited for the study. Most of them were females (70%) more than males (30%). Most of them had a single-curve type (70%), followed by a double-

curve type (30%). Most of them had the Risser sign with grade 4, and the curve flexibility was semi-rigid. All female participants already had menstruation, and the median menarche status was two years.

The data collection was performed in the SSPO research room. One researcher was the main person to collect the data to prevent bias. Participants had to come to the SSPO research room 2 times. The first time was for recruitment and obtaining the radiographs (if needed), and the second was for data collection using the developed system.

Four data groups were collected and compared, including the data from radiographs and the developed system for assessment, optimal correction, and casting processes. Chapter 9 focused on the experimental results for individual participants to demonstrate the feasibility of using the developed system to treat AIS patients and describe how the spinal changes before and when applying the force correction to treat scoliosis deformity. The next chapter, Chapter 10, further describes the summary of ten participants using descriptive and inferential statistics to validate the developed system and highlight the clinical findings.

In a low-cost postural measurement system, we used the system to collect the results from the marker locations attached to the participant's trunk and a positive model to quantify the spinal parameters and force directions in 3-dimensions.

Regarding the results between radiographs and a postural measurement system, the developed system could provide the results and illustrate the graphs for three spinal parameters, including the CSA and SSA in 4 segments and the trunk balance. The graphs from the postural measurement system could replicate and illustrate the spinal column alignment and trunk balance similar to the radiographs.

Regarding the CSA, SSA, and 3DSA in 4 segments and trunk balance, the developed system could also provide the results and illustrate the graphs during the assessment, optimal correction, and casting processes. For the CSA and trunk balance, the developed system could report the change between assessment and optimal correction. There was a slight change in these parameters from the optimal correction to casting process. For the SSA, the developed system could report the change between assessment, optimal correction, and casting processes. However, the overall curve in the SSA was still maintained. For the 3DSA, the developed system could also report

the change between assessment, optimal correction, and casting processes. This parameter could describe how the scoliosis deformity is present in three dimensions. Jang previously studied the angle of spinal parameters in 4 spinal segments in 3 dimensions with 20 non-scoliosis persons (Jang, 2018). The study divided the values of this angle into 3 groups, including the neutral alignment, the potentially abnormal alignment, and the mal-alignment. We then compared the results of individual participants to the previous research. Overall, the alignment zone between radiographs in assessment and the postural measurement system in assessment had a similar zone. The postural measurement system in assessment could provide spinal parameter values similar to those in radiographs. When applying the forces to correct the deformity, there was a change in the spinal parameters in 3 dimensions. Overall, the CSA was improved and changed to a better zone. The SSA slightly changed and remained in the same zone. The 3DSA also changed but remained in the same zone. The inferential statistics are needed to confirm the statistically significant change or similarity, which will be described in the following chapter.

Regarding the asymmetry of the trunk in the coronal plane, the developed system could provide the results and illustrate the graphs comparing the assessment and casting processes. Furthermore, the POsterior Trunk Symmetry Index (POTSI), a summary of the trunk's asymmetry percentage, could indicate the percentage change, and we could compare it between the assessment and casting processes. Even though this parameter was not the key parameter to evaluate the improvement or failure of AIS treatment, we could see how the shape of the trunk changes after applying the 3-dimensional force correction.

Regarding the angles of trunk rotation in the horizontal plane, the developed system could provide the results and illustrate the graphs comparing the assessment and casting processes. This parameter was a new parameter to see the trunk change on the horizontal plane before and when applying the 3-dimensional force correction. This parameter could describe how the scoliosis deformity is present in the transverse plane.

A scoliosis casting apparatus could be used to assist the researcher during the optimal correction and casting processes to correct the spinal deformity using 3-dimensional biomechanical force correction and to obtain the good quality of the negative cast, which can be used to produce the device for the treatment in the

following process. The postural measurement system could illustrate the force locations and directions the clinician aims to achieve during casting. The 3-dimensional biomechanical force correction from the casting apparatus could shift the spinal curve towards the centre in the coronal plane, de-rotate the curve in the transverse plane, and maintain the curve in the sagittal plane. The force measuring system could be used to measure the magnitude of forces in various areas. This parameter was new to describe how much force was used to correct the scoliosis deformity during the casting process. Combining the three developed systems could describe how the scoliosis deformity changes in 3 dimensions during the orthotic treatment of AIS patients.

The results from individual participants demonstrated the feasibility of using the developed system to treat AIS patients and describe how the spinal changes before and when applying the 3-dimensional biomechanical force correction to treat 3-dimensional deformity. To validate the system and make the conclusion, the data from ten participants will be analysed using descriptive and inferential statistics. Therefore, the next chapter, Chapter 10, describes the summary of descriptive and inferential statistics to evaluate the developed system and highlight the clinical findings.



**10. Chapter 10 Statistical Analysis and Result  
Interpretation from Clinical Experient with Adolescent  
Idiopathic Scoliosis**

## 10.1. Introduction

The previous section detailed the results of each participant, including the spinal angle in sagittal, coronal, and 3-dimensional planes, trunk balance, trunk asymmetry in the coronal plane, horizontal trunk rotation, and force magnitudes. This section further describes the data analysis from ten participants using descriptive and inferential statistics to analyse the results change or similarity compared between radiographs and the developed system in assessment, optimal correction, and casting. Some statistical analyses were performed using the SPSS program (Version 18.0.), and some used the NCSS program (Version 2023).

Before statistical analysis, all data were tested for the normal distribution based on Shapiro-Wilk's test (Shapiro and Wilk, 1965). The Shapiro-Wilk test is a standard statistical method to test the normal distribution of the data. If the data is not a normal distribution, the p-value will be smaller than 0.05. In contrast, if the data is normally distributed, the p-value is greater than or equal to 0.05. The results showed that most of the data variables collected had a p-value greater than 0.05, and the standard deviations (SD) were considered small compared to their means (see the statistical analysis table in the Appendix 13). Therefore, all data were considered normally distributed, and a parametric test was selected for the inferential statistics.

Six objectives were focused on for this clinical experiment and these are listed below.

1. To verify if a low-cost postural measurement system could quantify the spinal parameters in coronal and sagittal planes and to study the validity of the spinal parameters using a low-cost postural measurement system compared to the radiographic evaluation.
2. To examine if a low-cost postural measurement system could quantify the angles of the spinal column in 3 dimensions during the assessment, optimal correction, and casting process.
3. To examine if a low-cost postural measurement system could quantify the angle of trunk rotation in the horizontal plane during the assessment and casting process.

4. To examine if a low-cost postural measurement system could quantify the trunk asymmetry in the coronal plane during the assessment and casting process.
5. To examine if a scoliosis casting apparatus could be adjusted to correct the spinal deformity in 3 dimensions during optimal correction and casting process.
6. To examine if a force measuring system could measure the force magnitude and if a postural measurement system could illustrate the force directions when the forces were applied to correct the spinal deformity in 3 dimensions.

Regarding the statistics, Mean and standard deviation (SD) were selected to analyse all parameters for descriptive results. For the inferential statistics, the Pearson correlation coefficient was selected to analyse the relationship between the results from radiographs and the low-cost postural measurement system in assessment to evaluate the correlation's magnitude and the relationship's direction. The correlation was deemed significant if the p-value was below 0.05 level in a 2-tailed test. The relationship between the 3 spinal parameters were analysed using these statistics, including the CSA and SSA in 4 segments and the trunk balance. Regarding the magnitude of the correlation result (r-value), if the r-value was from 0.00 to 0.30 (or -0.30 to -0.00), the level of relationship was little. If the r-value was from 0.30 to 0.50 (or -0.50 to -0.30), the level of relationship was a low positive (or negative) correlation. If the r-value was from 0.50 to 0.70 (or -0.70 to -0.50), the level of relationship was a moderate positive (or negative) correlation. If the r-value was from 0.70 to 0.90 (or -0.90 to -0.70), the level of relationship was a high positive (or negative) correlation. If the r-value was from 0.90 to 1.00 (or -1.00 to -0.90), the level of relationship was a very high positive (or negative) correlation.

A Paired sample t-test was selected to determine the significance of the mean difference between the two groups. A statistically significant difference was determined at p-value below 0.05 for a 2-tailed test. In this thesis, the data were collected at 3 points. The assessment process was to evaluate the spinal parameters before applying the forces to correct the deformity. The optimal correction and casting processes were to evaluate the spinal parameters when applying the forces to correct

the deformity. Therefore, the Paired sample t-test was selected to compare the difference between before and when applying the forces to correct the deformity. Six spinal parameters were analysed using the paired sample t-test, including the CSA in assessment VS optimal correction, the CSA in assessment VS casting, the 3DSA in assessment VS optimal correction, the 3DSA in assessment VS casting, the trunk balance in assessment VS optimal correction, and the trunk balance in assessment VS casting. The null hypothesis ( $H_0$ ) assumes that the mean difference is equal to zero or no mean difference. In contrast, the alternative hypothesis ( $H_1$ ) is that the mean difference is not equal to zero or has a statistical significance difference. If the p-value was lower than 0.05, we rejected the null hypothesis, and the result was a statistically significant difference between both groups. In contrast, if the p-value was greater than 0.05, we failed to reject the null hypothesis, and the result was no statistically significant difference or not enough evidence (data) to find a difference. Lastly, a Paired t-test for equivalence using the Two One-Sided Test (TOST) was selected to determine whether both groups were equivalent. The statistically significant level was again set at  $p = 0.05$ . A margin of equivalence (M) had to be specified to analyse the equivalence of the data. The margin of (clinical) equivalence is chosen by defining the largest difference that is clinically acceptable so that a difference bigger than this would matter in practice. The margin of equivalence of this analysis was  $5^\circ$  or 5 mm because we allowed the spine to change within  $5^\circ$  and 5 mm. Six spinal parameters were analysed using the Paired t-test for equivalence, including the CSA in optimal correction VS casting, the SSA in assessment VS optimal correction, the SSA in assessment VS casting, and the SSA in optimal correction VS casting, the 3DSA in optimal correction VS casting, and the trunk balance in optimal correction VS casting. The null hypothesis ( $H_0$ ) assumes that both groups were not equivalent. In contrast, the alternative hypothesis ( $H_1$ ) is that both groups were equivalent. If the p-value was lower than 0.05, we rejected the null hypothesis, and the result was a statistical equivalence between both groups. In contrast, if the p-value was greater than 0.05, we failed to reject the null hypothesis, and the result was no statistical equivalence or not enough evidence (data) to find an equivalence.

## 10.2. Coronal spinal angles

The statistical analysis was begun with the spinal parameters in the coronal plane. Table 10.1 reports the mean and SD between coronal spinal angle (CSA) from radiographs and the low-cost postural measurement system in assessment, optimal correction, and casting. The mean CUTA results were 94.10° (SD = 3.21) in radiographs during the assessment process, 94.05° (SD = 2.90) in developed systems during the assessment process, 92.49° (SD = 2.35) in developed systems during the optimal correction process, and 92.91° (SD = 1.88) in developed systems during the casting process, respectively. The mean CLTA results were 85.90° (SD = 7.16) in radiographs during the assessment process, 86.09° (SD = 3.12) in developed systems during the assessment process, 86.70° (SD = 5.30) in developed systems during the optimal correction process, and 86.15° (SD = 3.28) in developed systems during the casting process, respectively. The mean CULA results were 82.90° (SD = 7.99) in radiographs during the assessment process, 86.24° (SD = 8.49) in developed systems during the assessment process, 87.69° (SD = 6.79) in developed systems during the optimal correction process, and 88.48° (SD = 4.19) in developed systems during the casting process, respectively. The mean CLLA results were 100.30° (SD = 12.82) in radiographs during the assessment process, 99.12° (SD = 14.07) in developed systems during the assessment process, 93.66° (SD = 6.24) in developed systems during the optimal correction process, and 92.33° (SD = 5.39) in developed systems during the casting process, respectively. Finally, the mean of all coronal spinal angles were 90.80° (SD = 10.72) in radiographs during the assessment process, 91.37° (SD = 9.88) in developed systems during the assessment process, 90.14° (SD = 6.04) in developed systems during the optimal correction process, and 89.96° (SD = 4.69) in developed systems during the casting process, respectively.

Table 10.1 Mean and SD of CSA from radiographs in assessment and the low-cost postural measurement system in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
CUTA, degree	94.10 (3.21)	94.05 (2.90)	92.49 (2.35)	92.91 (1.88)

Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
CLTA, degree	85.90 (7.16)	86.09 (3.12)	86.70 (5.30)	86.15 (3.28)
CULA, degree	82.90 (7.99)	86.24 (8.49)	87.69 (6.79)	88.48 (4.19)
CLLA, degree	100.30 (12.82)	99.12 (14.07)	93.66 (6.24)	92.33 (5.39)
All coronal spinal angles, degree	90.80 (10.72)	91.37 (9.88)	90.14 (6.04)	89.96 (4.69)

Note. CUTA = coronal upper thoracic angle, CLTA = coronal lower thoracic angle, CULA = coronal upper lumbar angle, CLLA = coronal lower lumbar angle. SD = standard deviation.

In the radiographs, the reduction percentage of the CSA was calculated to see the change in the spinal curve during the outcome measurement using the coronal Cobb angle magnitude of the with-out brace compared with the coronal Cobb angle magnitude of the with-in brace. However, we could not use this method to calculate the result of this experiment because the method to calculate the spinal parameter was different. Therefore, we proposed another method to calculate “**the reduction percentage of the apical translation**” by looking at the difference in the distance in the X-direction between the L5 marker and C7, T7, T12, and L2 markers.

The mean and SD of the apical translation of assessment, optimal correction, and casting in 10 participants were 12.82 mm (SD = 8.11), 6.26 mm (SD = 5.62), and 4.94 mm (SD = 4.68), respectively. Table 10.2 reports the reduction percentage of the apical translation comparing the assessment VS optimal correction and the casting process for individual and total participants. The results reported that the mean reduction percentage between the assessment VS optimal correction was 51.21% (SD = 13.42), and assessment VS casting was 59.53% (SD = 13.04). The mean difference between both comparisons was 9.73% (SD = 3.81).

Table 10.2 Reduction percentage of the apical translation between assessment, optimal correction, and casting process.

	Reduction % from Assessment to Optimal correction (A)	Reduction % from Assessment to Casting (B)	Difference of reduction percentage (A) – (B)
Participant 1	30.45	41.45	11.00
Participant 2	26.74	39.78	13.04
Participant 3	55.55	62.83	7.28
Participant 4	55.55	48.49	7.06

	Reduction % from Assessment to Optimal correction (A)	Reduction % from Assessment to Casting (B)	Difference of reduction percentage (A) – (B)
Participant 5	56.81	68.93	12.12
Participant 6	50.88	59.41	8.53
Participant 7	66.49	81.02	14.53
Participant 8	47.40	58.41	11.01
Participant 9	54.89	66.27	11.38
Participant 10	67.32	68.67	1.35
Total, mean (SD)	51.21 (13.42)	59.53 (13.04)	9.73 (3.81)

Note. (A) = Reduction percentage from assessment to optimal correction, (B) = Reduction percentage from assessment to casting, SD = standard deviation. % = percentage.

A paired sample t-test was then conducted to determine the mean difference in the reduction of the apical translation of assessment VS optimal correction and assessment VS casting (significant level at 0.05, 2-tailed). The results in Table 10.3 reported that the p-values were lower than 0.05, rejecting the null hypothesis and accepting the alternative hypothesis. It indicated a statistically significant difference in the reduction of apical translation of assessment VS optimal correction and assessment VS casting.

Table 10.3 Comparison the reduction of the apical translation between assessment VS optimal correction, and assessment VS casting, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
Assessment VS Optimal correction	4.50	8.62	6.45	<0.001*
Assessment VS Casting	5.73	10.03	7.42	<0.001*

Note: CI = Confidence interval. \* Statistically significant at the 0.05 level.

As a result, the mean reduction percentage after applying the forces to correct the deformity was greater than 50% in both comparisons, which followed the optimal correction percentage from the Boston brace principle (BostonOrthotics & Prosthetics, 2023). However, it could not directly compare with Boston concept because it used different methods to quantify the spinal parameters. Interestingly, the reduction percentage of assessment VS casting was slightly higher than the assessment VS optimal correction. There were two possible reasons to explain the difference. The first was from the spinal structure and self-correction by patients. As the spine was not a rigid structure, the patients could actively change their trunk position away from the pressure when applying the forces. It could help straighten the spine by self-correction.

The second was the error in quantifying the spinal parameters from the positive model. As the result from casting was measured by the positive model, not directly from the patients, it might cause some errors from this technique. When comparing the reduction of the apical translation between assessment VS optimal correction and assessment VS casting using a Paired sample t-test, the p-values were lower than 0.05, indicating a statistically significant difference in the reduction of apical translation of assessment VS optimal correction and assessment VS casting.

### 10.2.1. CSA between radiographs and the postural measurement in assessment

Figure 10.1 illustrates a line graph of the CSA between radiographs (Black line) and the postural measurement in assessment (Red line) in 4 spinal segments, including CUTA, CLTA, CULA, and CLLA. As can be seen from the graphs, the postural measurement system results went in the same trend as the radiographs in all spinal parameters, especially the CUTA and CLLA. There was a slight deviation of the angle values in CLTA and CULA. The statistical analysis was used to describe the relationship between both groups in the following step.

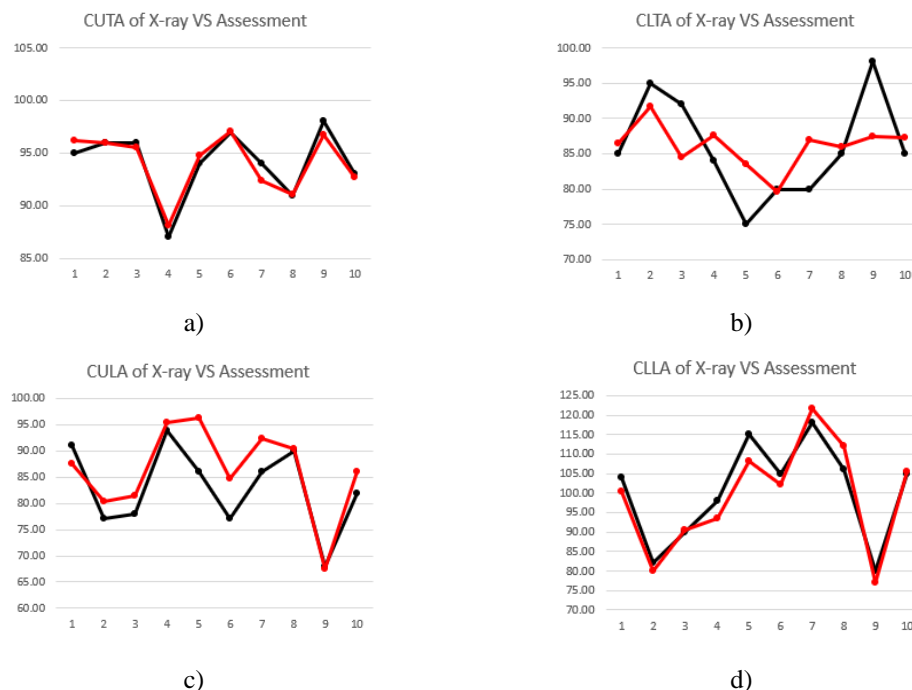
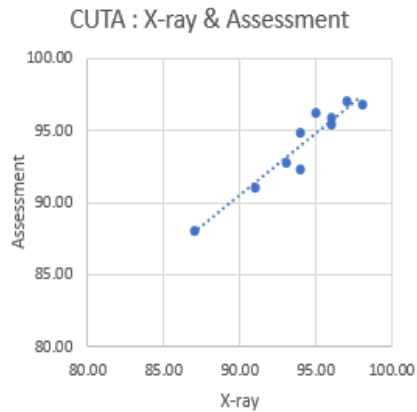


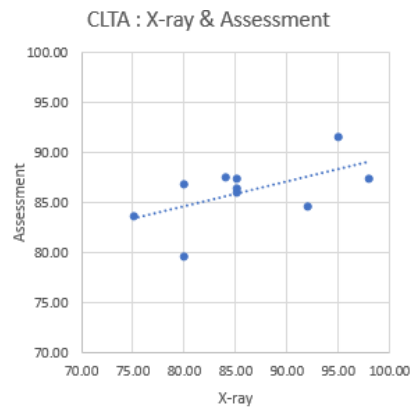
Figure 10.1 The line graph of the coronal spinal angles between radiographs (Black line) and the postural measurement system in assessment (Red line), where a) CUTA, b) CLTA, c) CULA, and d) CLLA.



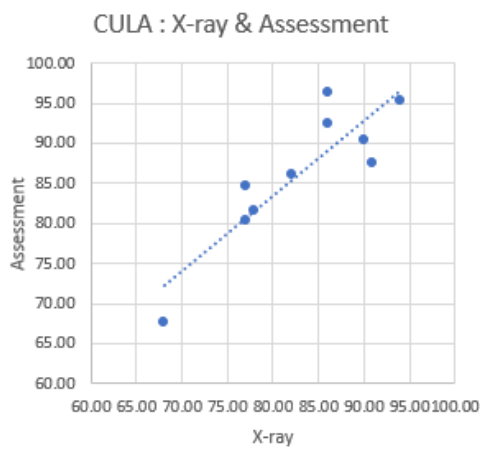
The Pearson correlation coefficient was then performed to analyse the relationship between radiographs and the postural measurement in assessment. Figure 10.2 illustrates the scatter plot of the coronal spinal angles between radiographs and the postural measurement in assessment in 4 spinal segments, including CUTA, CLTA, CULA, and CLLA. Furthermore, Table 10.4 reports the Pearson correlation results (r-value) for these spinal parameters.



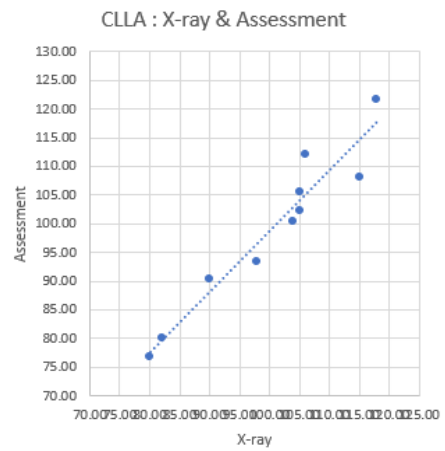
a)



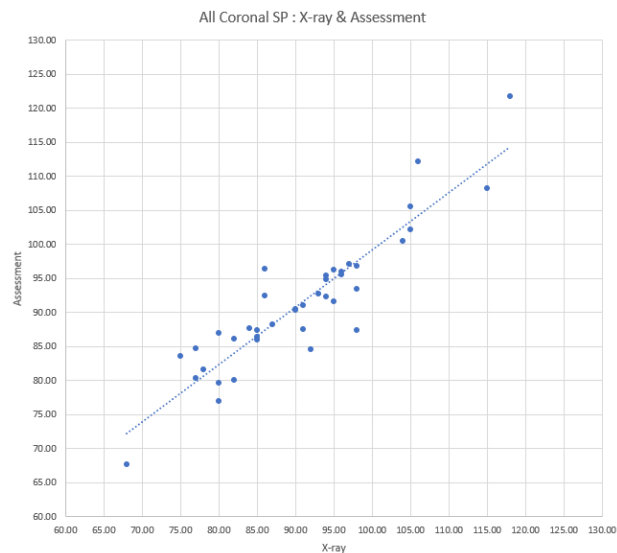
b)



c)



d)



e)

Figure 10.2 The scatter plot of the CSA between radiographs and the postural measurement system in assessment, a) CUTA, b) CLTA, c) CULA, d) CLLA, and e) all CSA. The X-axis is the data from radiographs, and the Y-axis is the data from the postural measurement system in assessment.

The results reported that  $p$ -values were lower than 0.05, and the correlation was significant at the 0.01 level (2-tailed) at CUTA, CULA, CLLA, and all CSA. Therefore, it rejected the null hypothesis ( $H_0$ ) and accepted the alternative hypothesis ( $H_1$ ). The relationship level was very high positive at CUTA ( $r = 0.958$ ), CLLA ( $r = 0.961$ ), and all CSA ( $r = 0.915$ ), high positive at CULA ( $r = 0.878$ ), and moderate positive at CLTA ( $r = 0.568$ ), respectively.

Table 10.4 Pearson correlation coefficient between CSA from radiographs and the postural measurement in assessment

Spinal Parameters	Pearson's correlation ( $r$ )	$p$ value	Relationship Level
CUTA	0.958	< 0.001**	Very high positive
CLTA	0.568	0.087	Moderate positive
CULA	0.878	< 0.001**	High positive
CLLA	0.961	< 0.001**	Very high positive
All coronal spinal angles	0.915	< 0.001**	Very high positive

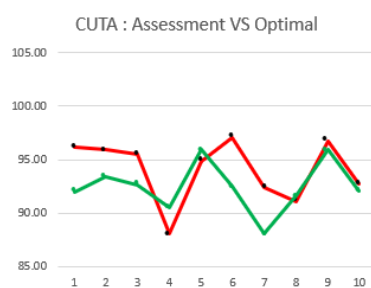
Note. CUTA = coronal upper thoracic angle, CLTA = coronal lower thoracic angle, CULA = coronal upper lumbar angle, CLLA = coronal lower lumbar angle. \* Correlation is significant at the 0.05 level (2-tailed), \*\* Correlation is significant at the 0.01 level (2-tailed).

Most CSA from the postural measurement in assessment had a high to very high positive correlation with CSA from radiographs, statistically significant at the 0.01

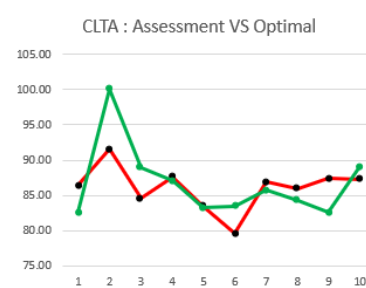
level (2-tails). Except for the CLTA, the p-value was greater than 0.05, and the r-value was 0.568. The relationship level was moderately positive, with no statistical significance. It could be concluded that the low-cost postural measurement system could be used to quantify the CSA that highly correlated to the radiographic results.

### 10.2.2. CSA between the postural measurement in assessment and optimal correction

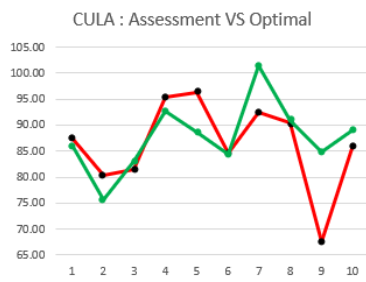
The scoliosis casting apparatus was designed in this thesis to assist the clinicians in achieving the best spinal correct during the orthotic treatment. The apparatus could be adjusted to apply the forces to correct the spinal deformity in 3 dimensions. A low-cost postural measurement system could be used to evaluate the change in the spinal parameters. Therefore, the first spinal parameter we analysed was the CSA in 4 segments, comparing assessment (before applying the force) and optimal correction (when applying the force). Figure 10.3 illustrates a line graph of the CSA between the postural measurement system in assessment (Red line) and optimal correction (Green line) in 4 spinal segments, including CUTA, CLTA, CULA, and CLLA. As can be seen from the graphs, there was a large deviation of the angle values in all spinal parameters or a spinal angle change when applying the 3-dimensional force to correct the spinal deformity. The statistical analysis was used to describe the difference between both groups in the following step.



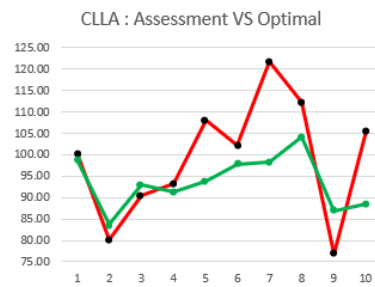
a)



b)



c)



d)

Figure 10.3 The line graph of the CSA between the postural measurement system in assessment (Red line) and optimal correction (Green line), a) CUTA, b) CLTA, c) CULA, and d) CLLA.

Then, a paired sample t-test was conducted to determine the mean difference in the CSA between the postural measurement in assessment and optimal correction (significant level at 0.05, 2-tailed). The results in Table 10.5 reported that all p-values were greater than 0.05, indicating a failure to reject the null hypothesis. It indicated a non-statistically significant difference in the CSA from the low-cost postural measurement system in the assessment and optimal correction at CUTA, CLTA, CULA, CLLA, and all CSA.

Table 10.5 Comparison between CSA from the postural measurement in assessment and optimal correction, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
CUTA	-0.19	3.30	2.014	0.750
CLTA	-3.49	2.27	-0.478	0.644
CULA	-6.56	3.65	-0.644	0.536
CLLA	-1.94	12.85	1.667	0.130
All coronal spinal angles	-0.99	3.47	1.119	0.270

Note: CUTA = coronal upper thoracic angle, CLTA = coronal lower thoracic angle, CULA = coronal upper lumbar angle, CLLA = coronal lower lumbar angle. CI = confidence interval, \* Significant at the 0.05 level (2-tailed).

### 10.2.3. CSA between the postural measurement in assessment and casting

The following spinal parameter we analysed was the CSA, comparing assessment and casting process to evaluate the difference between both groups. Figure 10.4 illustrates a line graph of the CSA between the postural measurement system in assessment (Red line) and casting (Blue line) in 4 spinal segments, including CUTA,

CLTA, CULA, and CLLA. As can be seen from the graphs, there was a large deviation of the angle values in all spinal parameters, similar to when comparing the assessment with the optimal correction. The statistical analysis was used to describe the difference between both groups in the following step.

A paired sample t-test was then conducted to determine the mean difference in the CSA between the postural measurement in assessment and casting (significant level at 0.05, 2-tailed). The results in Table 10.6 reported that all p-values were greater than 0.05, indicating a failure to reject the null hypothesis. It indicated a non-statistically significant difference in the CSA from the postural measurement in assessment and casting at CUTA, CLTA, CULA, CLLA, and all CSA.

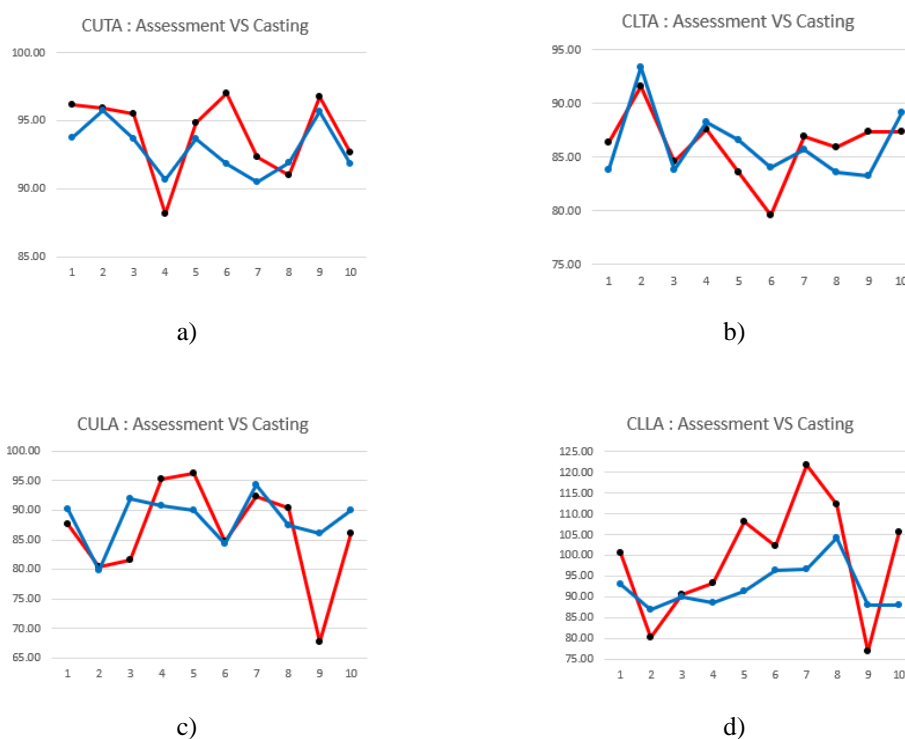


Figure 10.4 The line graph of the CSA between the postural measurement system in assessment (Red line) and casting (Blue line), a) CUTA, b) CLTA, c) CULA, and d) CLLA.

Table 10.6 Comparison between CSA from the postural measurement in assessment and casting, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Lower		
CUTA	-0.33	2.60	1.758	0.113
CLTA	-2.00	1.90	-0.064	0.951
CULA	-7.54	3.06	-0.956	0.364

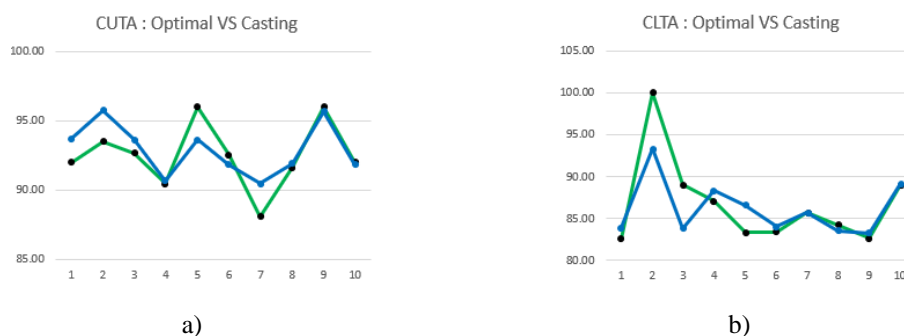
Spinal Parameters	95% CI		t	p-value
	Lower	Lower		
CLLA	-1.14	14.72	1.938	0.085
All coronal spinal angles	-0.96	3.78	1.200	0.237

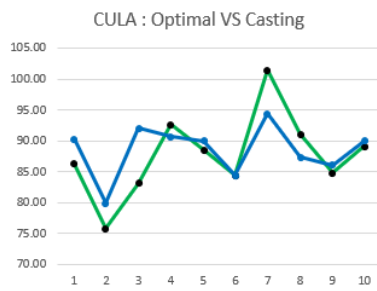
Note: CUTA = coronal upper thoracic angle, CLTA = coronal lower thoracic angle, CULA = coronal upper lumbar angle, CLLA = coronal lower lumbar angle. CI = confidence interval, \* Significant at the 0.05 level (2-tailed).

#### 10.2.4. CSA between the postural measurement in optimal correction and casting

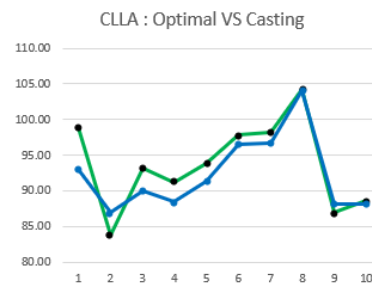
The following spinal parameter we analysed was the CSA, comparing the optimal correction and casting to evaluate the equivalence between both groups. The optimal correction was a process of spinal correction using scoliosis casting apparatus applying the forces to correct the deformity. The casting process was a continued process to capture the trunk's shape using plaster bandages, and the scoliosis casting apparatus still applied the forces to correct the deformity. Ideally, the spinal angles should remain at the same value, and we allowed the deviation not to be greater than 5°.

Figure 10.5 illustrates a line graph of the CSA between the postural measurement system in optimal correction (Green line) and casting (Blue line) in 4 spinal segments, including CUTA, CLTA, CULA, and CLLA. As can be seen from the graphs, the results from optimal correction went in the same trend as the casting process, and there was a slight deviation of the angle values in all spinal parameters. The statistical analysis was used to describe the equivalence between both groups in the following step.





c)



d)

Figure 10.5 The line graph of the CSA between the postural measurement system in optimal correction (Green line) and casting (Blue line), a) CUTA, b) CLTA, c) CULA, and d) CLLA.

A paired t-test for equivalence using the Two One-Sided Test (TOST) was performed to test the statistical hypothesis and determine whether both groups were equivalent, the postural measurement in optimal correction and casting. The margin of equivalence of this analysis was 5° because we allowed the spine to change within 5°, and the significant level was at 0.05. The results in Table 10.7 reported that all p-values were lower than 0.05, rejected by the null hypothesis. As a result, it indicated a statistically significant equivalence of the CSA from the postural measurement in optimal correction and casting process at CUTA, CLTA, CULA, CLLA, and all CSA.

Table 10.7 Equivalence testing between CSA from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.

Spinal Parameters	T-Statistic		p-value	
	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary
CUTA	9.87	-11.70	< 0.001*	< 0.001*
CLTA	5.76	-4.61	< 0.001*	< 0.001*
CULA	2.97	-4.08	0.008*	0.001*
CLLA	8.07	-4.66	< 0.001*	< 0.001*
All coronal spinal angles	10.66	-9.95	< 0.001*	< 0.001*

Note. CUTA = coronal upper thoracic angle, CLTA = coronal lower thoracic angle, CULA = coronal upper lumbar angle, CLLA = coronal lower lumbar angle. \* Statistically significant at the 0.05 level.

### 10.3. Sagittal spinal angles

Deformity in the sagittal plane can be seen in some patients. The patients may present hypokyphosis of the thoracic curve (flatback) and hyperlordosis of the lumbar

curve. This thesis used the new method to quantify the sagittal spinal angle (SSA) by dividing the spinal column into four segments, similar to the CSA, to evaluate the angle change in the sagittal plane. Table 10.8 reports the mean and SD between SSA from radiographs and the low-cost postural measurement system in assessment, optimal correction, and casting. The mean SUTA results were 80.90° (SD = 4.72) in radiographs during the assessment process, 76.49° (SD = 7.99) in developed systems during the assessment process, 78.53° (SD = 7.72) in developed systems during the optimal correction process, and 77.03° (SD = 6.92) in developed systems during the casting process, respectively. The mean SLTA results were 100.20° (SD = 3.33) in radiographs during the assessment process, 98.34° (SD = 6.12) in developed systems during the assessment process, 100.58° (SD = 7.27) in developed systems during the optimal correction process, and 98.37° (SD = 6.16) in developed systems during the casting process, respectively. The mean SULA results were 103.90° (SD = 3.45) in radiographs during the assessment process, 100.76° (SD = 5.41) in developed systems during the assessment process, 97.75° (SD = 7.38) in developed systems during the optimal correction process, and 96.70° (SD = 7.29) in developed systems during the casting process, respectively. The mean SLLA results were 90.00° (SD = 8.39) in radiographs during the assessment process, 84.77° (SD = 11.95) in developed systems during the assessment process, 89.48° (SD = 9.38) in developed systems during the optimal correction process, and 87.74° (SD = 10.11) in developed systems during the casting process, respectively. Finally, the mean of all sagittal spinal angles were 93.75° (SD = 10.48) in radiographs during the assessment process, 90.09° (SD = 12.82) in developed systems during the assessment process, 91.58° (SD = 11.58) in developed systems during the optimal correction process, and 89.96° (SD = 11.38) in developed systems during the casting process, respectively.

Table 10.8 Mean and SD of SSA from radiographs in assessment and the postural measurement in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
SUTA, degree	80.90 (4.72)	76.49 (7.99)	78.53 (7.72)	77.03 (6.92)
SLTA, degree	100.20 (3.33)	98.34 (6.12)	100.58 (7.27)	98.37 (6.16)
SULA, degree	103.90 (3.45)	100.76 (5.41)	97.75 (7.38)	96.70 (7.29)



Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
SLLA, degree	90.00 (8.39)	84.77 (11.95)	89.48 (9.38)	87.74 (10.11)
All sagittal spinal angles, degree	93.75 (10.48)	90.09 (12.82)	91.58 (11.58)	89.96 (11.38)

Note. SUTA = sagittal upper thoracic angle, SLTA = sagittal lower thoracic angle, SULA = sagittal upper lumbar angle, SLLA = sagittal lower lumbar angle, SD = standard deviation.

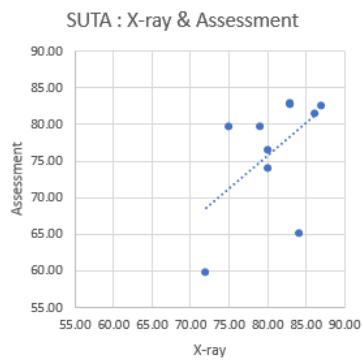
### 10.3.1. SSA between radiographs and the postural measurement in assessment

Figure 10.6 illustrates a line graph of the SSA between radiographs (Black line) and the postural measurement in assessment (Red line) in 4 spinal segments, including SUTA, SLTA, SULA, and SLLA. As can be seen from the graphs, the postural measurement system results followed the same trend as the radiographs in all spinal parameters. However, the parameters had a slight deviation between both groups, and the statistical analysis was used to describe the relationship between both groups in the following step.

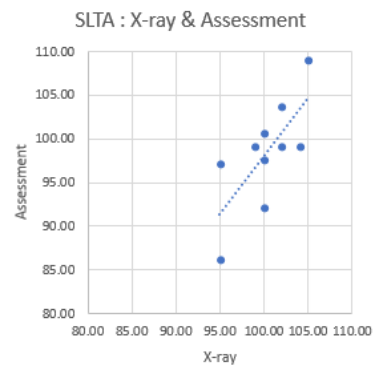


Figure 10.6 The line graph of the SSA between radiographs (Black line) and the postural measurement system for assessment (Red line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.

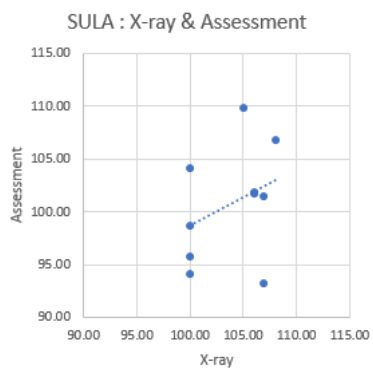
The Pearson correlation coefficient was then performed to analyse the relationship between radiographs and the postural measurement in assessment. Figure 10.7 illustrates the scatter plot of the SSA between radiographs and the postural measurement system in assessment in 4 spinal segments, including SUTA, SLTA, SULA, and SLLA. Furthermore, Table 10.9 reports the Pearson correlation results (r-value) for these spinal parameters.



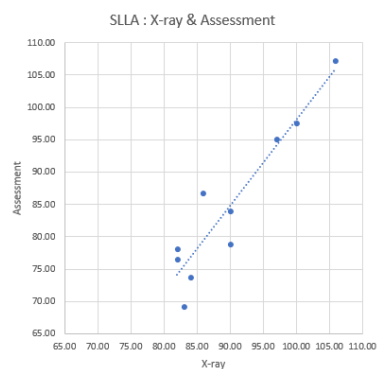
a)



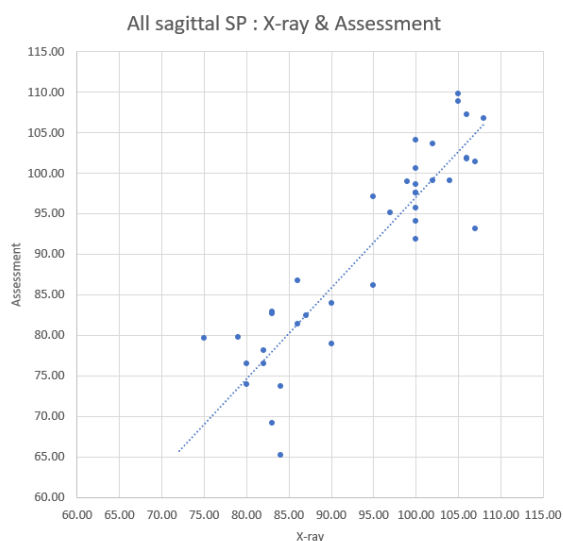
b)



c)



d)



e)

Figure 10.7 The scatter plot of the SSA between radiographs and the postural measurement system in assessment, a) SUTA, b) SLTA, c) SULA, d) SLLA, and e) all SSA. The X-axis is the data from radiographs, and the Y-axis is the data from the postural measurement system for assessment.

At SLTA, SLLA, and all SSA, the results reported that p-values were lower than 0.05, and the correlation was significant at the 0.05 or 0.01 level (2-tailed). It rejected the null hypothesis and accepted the alternative hypothesis. The relationship level was between a high and very high positive correlation. To clarify, the relationship level of SLTA ( $r = 0.731$ ) was highly positive, SLLA was very highly positive ( $r = 0.932$ ), and all sagittal spinal angles were also very highly positive ( $r = 0.912$ ). However, the relationship level of SUTA was moderately positive ( $r = 0.526$ ), and SULA was lowly positive ( $r = 0.348$ ). The p-values were greater than 0.05, and the correlation was not significant at the 0.05 level.

Table 10.9 Pearson correlation coefficient between SSA from radiographs and the postural measurement in assessment.

Spinal Parameters	Pearson's correlation (r)	p value	Relationship Level
SUTA	0.526	0.118	Moderate positive
SLTA	0.731	0.016*	High positive
SULA	0.348	0.324	Low positive
SLLA	0.935	< 0.001**	Very high positive
All sagittal spinal angles	0.912	< 0.001**	Very high positive

Note. SUTA = sagittal upper thoracic angle, SLTA = sagittal lower thoracic angle, SULA = sagittal upper lumbar angle, and SLLA = sagittal lower lumbar angle. \* Correlation is significant at the 0.05 level (2-tailed), \*\* Correlation is significant at the 0.01 level (2-tailed).

Half of the parameters from the postural measurement in assessment had a high to very high positive correlation with the radiographs, statistically significant at the 0.05 or 0.01 level (2-tails). In contrast, another half of the parameters from the postural measurement system in assessment had a moderate to low positive correlation with radiographs. The p-value was greater than 0.05 or not statistically significant. However, the r-value of all SSA had a very high positive correlation, statistically significant at the 0.01 level (2-tails). It can be concluded that the low-cost postural measurement system could be used to quantify the SSA, and the correlation ranged from low to very high positive correlation to the radiographic results.

### **10.3.2. SSA between the postural measurement in assessment and optimal correction**

The scoliosis casting apparatus was designed to assist the clinician in achieving the best spinal correction during the orthotic treatment. Ideally, the coronal curve and spinal rotation deformity would be reduced to the optimal positions. Moreover, the sagittal spinal curve should be maintained. Therefore, the following spinal parameter we analysed was the SSA, comparing assessment and optimal correction to evaluate the equivalence between both groups.

Figure 10.8 illustrates a line graph of the SSA between the postural measurement system in assessment (Red line) and optimal correction (Green line) in 4 spinal segments, including SUTA, SLTA, SULA, and SLLA. As can be seen from the graphs, the results from the assessment went in the same direction as the optimal correction, and there was a slight deviation of the angle values in all parameters. The statistical analysis was used to describe the equivalence between both groups in the following step.

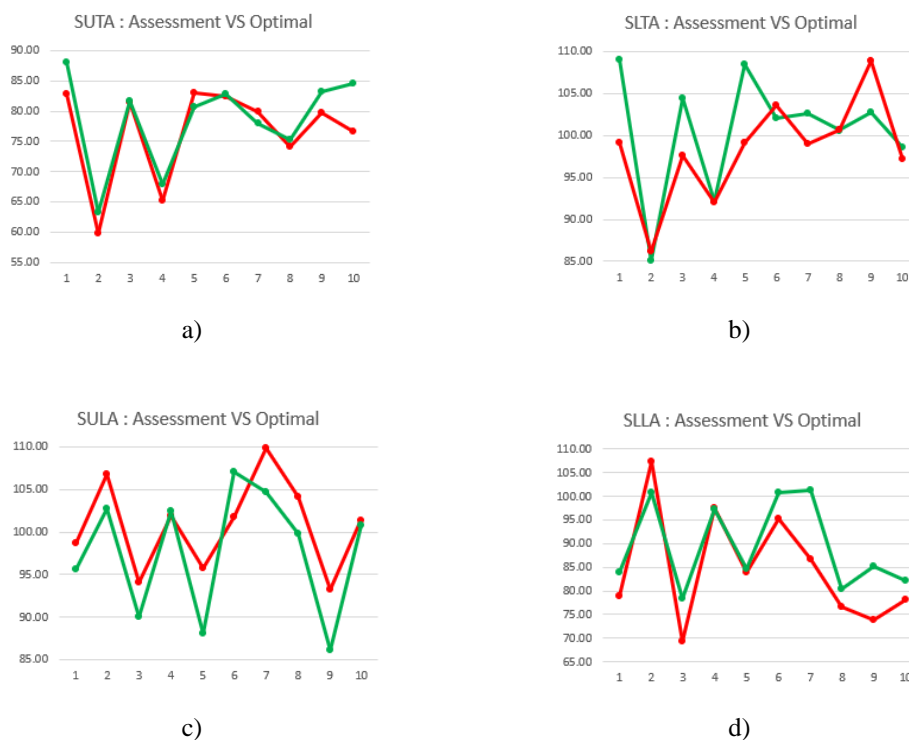


Figure 10.8 The line graph of the SSA between the postural measurement system in assessment (Red line) and optimal correction (Green line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.

A paired t-test for equivalence using the Two One-Sided Test (TOST) was performed to determine whether both groups were equivalent. The margin of equivalence of this analysis was  $5^\circ$  because we allowed the spine to change within  $5^\circ$ . The significance was at the 0.05 level. When considering individual segments (Table 10.10), only SUTA had a p-value of both boundaries lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence. In contrast, the p-values of SLTA, SULA, and SLLA had either lower or upper boundaries lower than 0.05, which could not reject the null hypothesis or indicate a statistically significant equivalence. Then, we varied the margin of equivalence to see which value produced p-values for all spinal parameters less than 0.05. However, the margin of equivalence should not be greater than  $10^\circ$  because it was the largest difference we allowed the spine to change. Finally, the margin of equivalence of this analysis was  $9^\circ$  (Table 10.10). The result showed that all p-values were lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence.

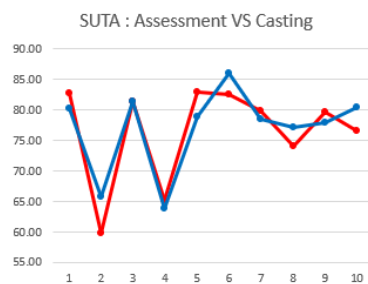
Table 10.10 Equivalence testing between SSA from the postural measurement in assessment and optimal correction, using Paired t-test for equivalence.

Spinal Parameters	Margin = 5°				Margin = 9°			
	T-Statistic		p-value		T-Statistic		p-value	
	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary
SUTA	2.95	-6.99	0.008*	< 0.001*	6.93	-10.98	< 0.001*	< 0.001*
SLTA	1.69	-4.44	0.063	< 0.001*	4.14	-6.89	< 0.001*	< 0.001*
SULA	6.56	-1.62	< 0.001*	0.070	9.83	-4.89	< 0.001*	< 0.001*
SLLA	0.15	-5.09	0.442	< 0.001*	2.24	-7.18	0.026*	< 0.001*
All sagittal spinal angles	4.15	-7.70	< 0.001*	< 0.001*	8.90	-12.45	< 0.001*	< 0.001*

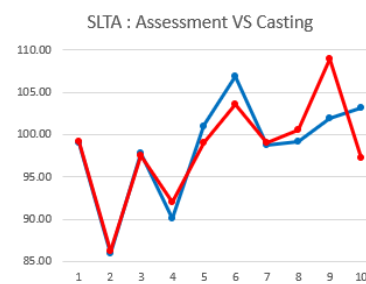
Note. SUTA = sagittal upper thoracic angle, SLTA = sagittal lower thoracic angle, SULA = sagittal upper lumbar angle, SLLA = sagittal lower lumbar angle. \* Statistically significant at the 0.05 level.

### 10.3.3. SSA between the postural measurement in assessment and casting

The following spinal parameter we analysed was the SSA, comparing the assessment and casting process to evaluate the equivalence between both groups. Figure 10.9 illustrates a line graph of the SSA between the postural measurement system in assessment (Red line) and casting (Blue line) in 4 spinal segments, including SUTA, SLTA, SULA, and SLLA. As can be seen from the graphs, the results from the assessment went in the same direction as the casting, but there was a slight deviation of the angle values in all parameters. The statistical analysis was used to describe the equivalence between both groups in the following step.



a)



b)

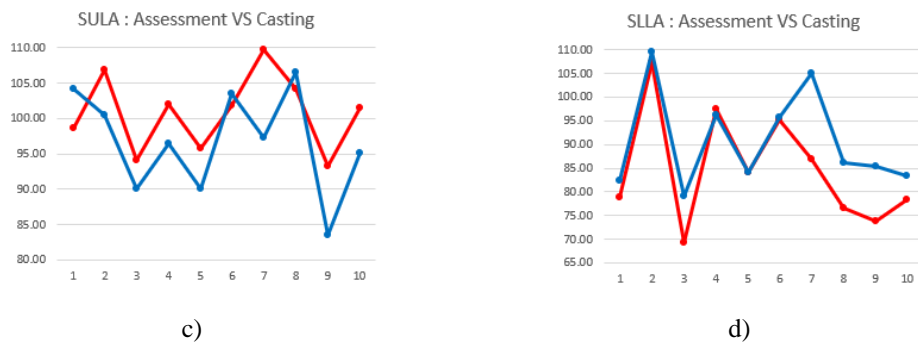


Figure 10.9 The line graph of the SSA between the postural measurement system in assessment (Red line) and casting (Blue line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.

A paired t-test for equivalence using the Two One-Sided Test (TOST) was performed to determine whether both groups were equivalent. The margin of equivalence of this analysis was  $5^\circ$ , and the significant level was at 0.05. When considering individual segments (Table 10.11), SUTA and SLTA had a p-value of both boundaries lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence. In contrast, the p-values of SULA and SLLA had either lower or upper boundaries lower than 0.05, which could not reject the null hypothesis or indicate a statistically significant equivalence. Then, we varied the margin of equivalence to see which value produced p-values for all spinal parameters less than 0.05. However, the margin of equivalence should not be greater than  $10^\circ$  because it was the largest difference we allowed the spine to change. Finally, the margin of equivalence of this analysis was  $8^\circ$  (Table 10.11). The result showed that all p-values were lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence.

Table 10.11 Equivalence testing between SSA from the postural measurement in assessment and casting, using Paired t-test for equivalence.

Spinal Parameters	Margin = $5^\circ$				Margin = $8^\circ$			
	T-Statistic		p-value		T-Statistic		p-value	
	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary
SUTA	4.25	-5.28	0.001*	< 0.001*	7.11	-8.14	< 0.001*	< 0.001*
SLTA	4.60	-4.67	< 0.001*	< 0.001*	7.38	-7.45	< 0.001*	< 0.001*
SULA	5.10	-0.53	< 0.001*	0.306	6.79	-2.21	< 0.001*	0.027*
SLLA	0.94	-3.69	0.186	0.003*	2.33	-5.08	0.022*	< 0.001*
All sagittal spinal angles	5.94	-5.64	< 0.001*	< 0.001*	9.41	-9.11	< 0.001*	< 0.001*

Note. SUTA = sagittal upper thoracic angle, SLTA = sagittal lower thoracic angle, SULA = sagittal upper lumbar angle, SLLA = sagittal lower lumbar angle. \* Statistically significant at the 0.05 level.

### 10.3.4. SSA between the postural measurement in optimal correction and casting

The following spinal parameter we analysed was the SSA, comparing the optimal correction and casting process to evaluate the equivalence between both groups. Figure 10.10 illustrates a line graph of the SSA between the postural measurement system in optimal correction (Green line) and casting (Blue line) in 4 spinal segments, including SUTA, SLTA, SULA, and SLLA. As can be seen from the graphs, the results from the optimal correction went in the same direction as the casting, but there was a slight deviation of the angle values in all parameters. The statistical analysis was used to describe the equivalence between both groups in the following step.

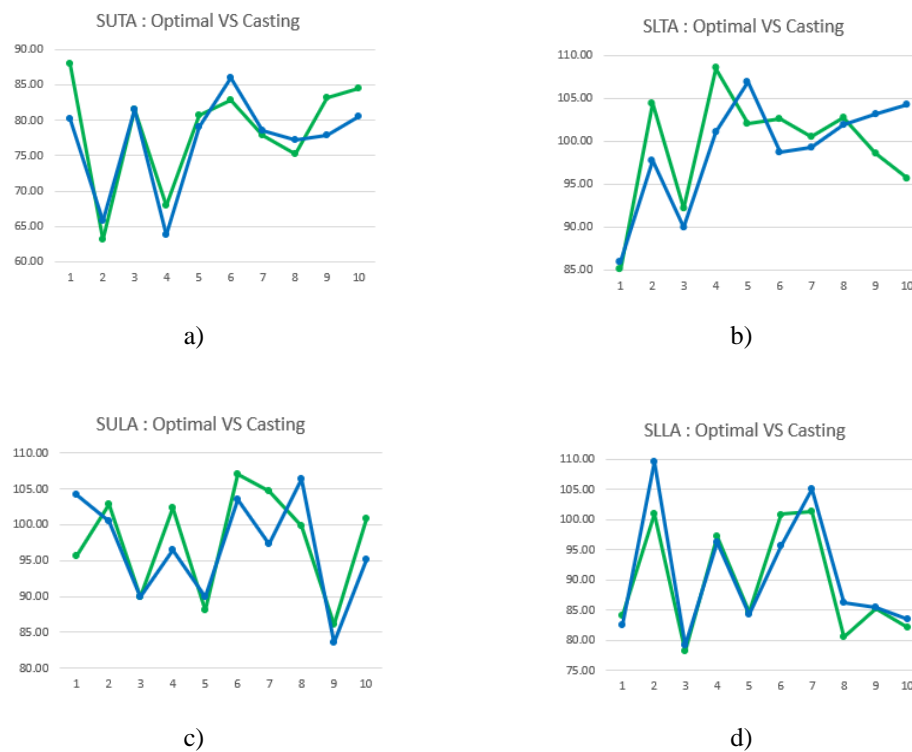


Figure 10.10 The line graph of the SSA between the postural measurement system in optimal correction (Green line) and casting (Blue line), a) SUTA, b) SLTA, c) SULA, and d) SLLA.

A paired t-test for equivalence using the Two One-Sided Test (TOST) was performed to determine whether both groups were equivalent. The margin of equivalence of this analysis was  $5^\circ$ , and the significant level was at 0.05. When



considering individual segments (Table 10.12), SUTA and SULA had a p-value of both boundaries lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence. In contrast, the p-values of SLTA and SLLA had either lower or upper boundaries lower than 0.05, which could not reject the null hypothesis or indicate a statistically significant equivalence. Then, we varied the margin of equivalence to see which value produced p-values for all spinal parameters less than 0.05. However, the margin of equivalence should not be greater than 10° because it was the largest difference we allowed the spine to change. Finally, the margin of equivalence of this analysis was 8° (Table 10.12). The result showed that all p-values were lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence.

Table 10.12 Equivalence testing between SSA from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.

Spinal Parameters	Margin = 5°				Margin = 8°			
	T-Statistic		p-value		T-Statistic		p-value	
	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary
SUTA	5.55	-2.99	< 0.001*	0.008*	8.11	-5.55	< 0.001*	< 0.001*
SLTA	4.61	-1.78	< 0.001*	0.054	6.53	-3.70	< 0.001*	< 0.002*
SULA	3.58	-2.34	0.003*	0.022*	5.35	-4.11	< 0.001*	0.001*
SLLA	2.26	-1.09	0.025*	0.152	3.26	-2.10	0.005*	0.033*
All sagittal spinal angles	6.97	-3.55	< 0.001*	< 0.001*	10.12	-6.71	< 0.001*	< 0.001*

Note. SUTA = sagittal upper thoracic angle, SLTA = sagittal lower thoracic angle, SULA = sagittal upper lumbar angle, SLLA = sagittal lower lumbar angle. \* Significant at the 0.05 level (2-tailed).

#### 10.4. 3-dimensional spinal angles

The 3-dimensional spinal angle (3DSA) was a new parameter to describe the scoliosis deformity in 3 dimensions. The thesis used this method to quantify this parameter by dividing the spinal column into 4 segments, similar to the CSA and SSA, to evaluate the change of the 3DSA respective to the horizontal plane.

Table 10.13 reports the mean and SD between 3DSA from the low-cost postural measurement system in assessment, optimal correction, and casting. The results showed that the mean 3D-UTA in the assessment, optimal correction, and casting were 75.37° (SD = 7.43), 77.49° (SD = 6.92), and 76.60° (SD = 6.85), respectively. The mean 3D-LTA in the assessment, optimal correction, and casting were 99.37° (SD =

6.52), 101.09° (SD = 9.35), and 99.37° (SD = 6.53), respectively. The mean 3D-ULA in the assessment, optimal correction, and casting were 102.06° (SD = 5.13), 99.34° (SD = 8.55), and 97.42° (SD = 7.83), respectively. The mean 3D-LLA in the assessment, optimal correction, and casting were 79.57° (SD = 17.63), 88.69° (SD = 11.76), and 86.73° (SD = 11.41), respectively. Finally, the mean of all 3D spinal angles was 89.09° (SD = 15.59) in assessment, 91.65° (SD = 13.10) in optimal correction, and 90.03° (SD = 12.26) in casting.

Table 10.13 Mean and standard deviation of 3DSA from the postural measurement in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)		
	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
3D-UTA, degree	75.37 (7.43)	77.49 (6.92)	76.60 (6.85)
3D-LTA, degree	99.37 (6.52)	101.09 (9.35)	99.37 (6.53)
3D-ULA, degree	102.06 (5.13)	99.34 (8.55)	97.42 (7.83)
3D-LLA, degree	79.57 (17.63)	88.69 (11.76)	86.73 (11.41)
All 3D spinal angles, degree	89.09 (15.59)	91.65 (13.10)	90.03 (12.26)

Note: 3D-UTA = 3-dimensional upper thoracic angle, 3D-LTA = 3-dimensional lower thoracic angle, 3D-ULA = 3-dimensional upper lumbar angle, and 3D-LLA = 3-dimensional lower lumbar angle.

#### 10.4.1. 3DSA between the postural measurement in assessment and optimal correction

The following spinal parameter we analysed was the 3DSA, comparing assessment and optimal correction to evaluate the difference between both groups. Figure 10.11 illustrates a line graph of the 3DSA between the postural measurement system in assessment (Red line) and optimal correction (Green line) in 4 spinal segments, including 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA. As can be seen from the graphs, the results from the assessment went in the same direction as the optimal correction, and there was a slight deviation of the angle values in all parameters. The statistical analysis was used to describe the difference between both groups in the following step.

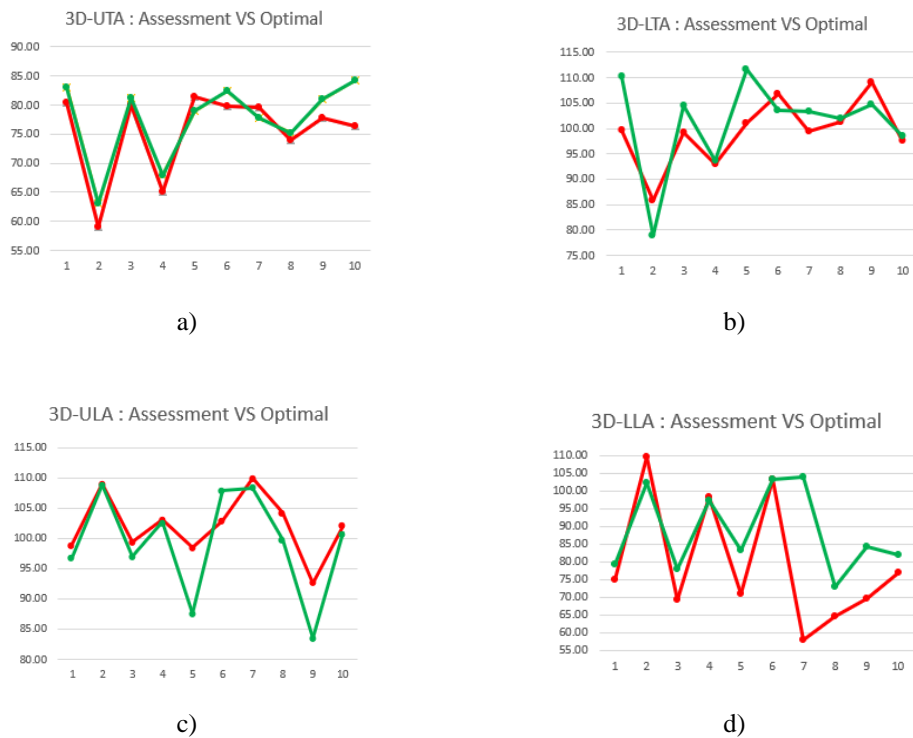


Figure 10.11 The line graph of the 3DSA between the postural measurement system in assessment (Red line) and optimal correction (Green line), a) 3D-UTA, b) 3D-LTA, c) 3D-ULA, and d) 3D-LLA.

Then, a paired sample t-test was conducted to determine the mean difference in the 3DSA between the postural measurement in assessment and optimal correction (significant level at 0.05, 2-tailed). The results in Table 10.14 reported that only one spinal segment, 3D-UTA, had a p-value lower than 0.05. However, other spinal parameters at 3D-LTA, 3D-ULA, and 3D-LLA had a p-value greater than 0.05, failed to reject the null hypothesis, and indicated a non-significant difference between the 3DSA from the postural measurement system in assessment and optimal correction. Moreover, the p-value of all 3DSA was greater than 0.05, indicating a non-significant difference.

Table 10.14 Comparison between 3DSA from the postural measurement in assessment and optimal correction, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
3D-UTA, degree	-4.19	-0.38	-2.30	0.047*
3D-LTA, degree	-5.95	2.49	-0.93	0.377
3D-ULA, degree	-0.53	5.97	1.89	0.091
3D-LLA, degree	-19.44	1.20	-1.99	0.077

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
All 3D spinal angles, degree	-5.44	0.32	-1.80	0.080

Note. 3D-UTA = 3-dimensional upper thoracic angle, 3D-LTA = 3-dimensional lower thoracic angle, 3D-ULA = 3-dimensional upper lumbar angle, and 3D-LLA = 3-dimensional lower lumbar angle.

\* Significant at the 0.05 level (2-tailed).

### 10.4.2. 3DSA between the postural measurement in assessment and casting

The following spinal parameter we analysed was the 3DSA, comparing assessment and casting to evaluate the difference between both groups. Figure 10.12 illustrates a line graph of the 3DSA between the postural measurement system in assessment (Red line) and casting (Blue line) in 4 spinal segments, including 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA. As can be seen from the graphs, the results from the assessment went in the same direction as the casting, and there was a slight deviation of the angle values in all parameters. The statistical analysis was used to describe the difference between both groups in the following step.

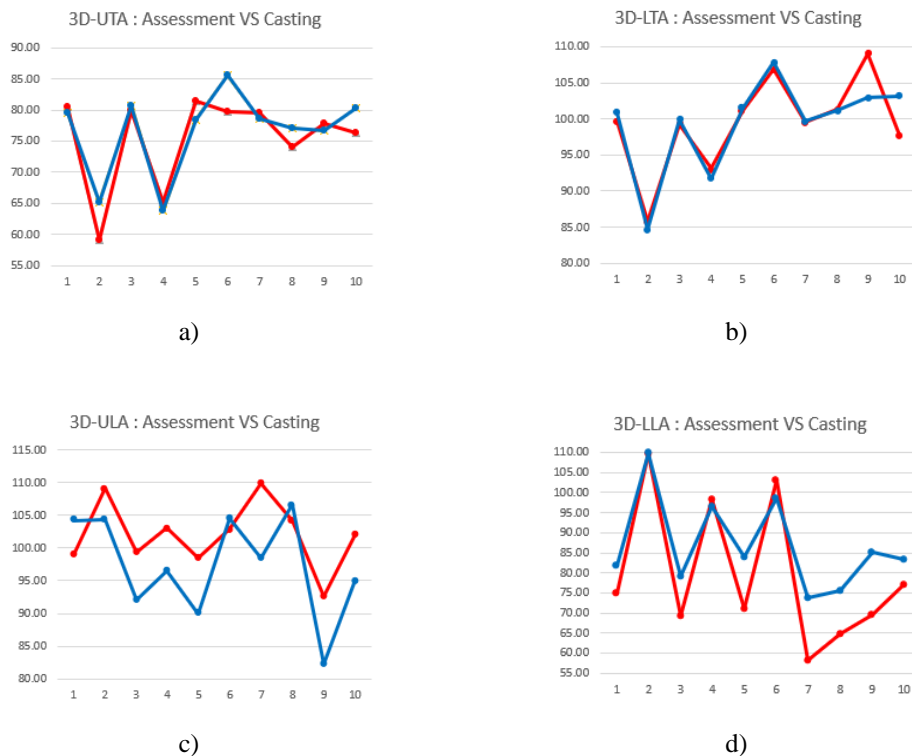


Figure 10.12 The line graph of the 3DSA between the postural measurement system in assessment (Red line) and casting (Blue line), a) 3D-UTA, b) 3D-LTA, c) 3D-ULA, and d) 3D-LLA.

A paired sample t-test was conducted to determine the mean difference in the 3DSA between the postural measurement in assessment and casting (significant level at 0.05, 2-tailed). The results in Table 10.15 reported that the spinal segment at 3D-ULA and 3D-LLA had a p-value lower than 0.05, rejected the null hypothesis, and indicated a significant difference. In contrast, the spinal parameters at 3D-UTA and 3D-LTA had a p-value greater than 0.05, failed to reject the null hypothesis, and indicated a non-significant difference. Moreover, the p-value of all 3DSA was greater than 0.05, indicating a non-significant difference. Three out of five spinal parameters in this analysis had a p-value greater than 0.05, indicating a non-statistically significant difference. There was a change in the 3DSA between assessment and casting, but there was no statistically significant difference.

Table 10.15 Comparison between 3DSA from the postural measurement in assessment and casting, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
3D-UTA, degree	-3.57	1.11	-1.187	0.266
3D-LTA, degree	-2.04	2.03	-0.001	0.999
3D-ULA, degree	0.53	8.76	2.56	0.031*
3D-LLA, degree	-12.27	-2.05	-3.168	0.011*
All 3D spinal angles, degree	-3.00	1.13	-0.914	0.366

Note. 3D-UTA = 3-dimensional upper thoracic angle, 3D-LTA = 3-dimensional lower thoracic angle, 3D-ULA = 3-dimensional upper lumbar angle, 3D-LLA = 3-dimensional lower lumbar angle, CI = Confidence interval, \* Significant at the 0.05 level (2-tailed).

#### 10.4.3. 3DSA between the postural measurement in optimal correction and casting

The following spinal parameter we analysed was the 3DSA, comparing optimal correction and casting to evaluate the equivalence between both groups. Figure 10.13 illustrates a line graph of the 3DSA between the postural measurement system in optimal correction (Green line) and casting (Blue line) in 4 spinal segments, including 3D-UTA, 3D-LTA, 3D-ULA, and 3D-LLA. As can be seen from the graphs, the results from the optimal correction went in the same direction as the casting, and there was a slight deviation of the angle values in all parameters. The statistical analysis was used to describe the equivalence between both groups in the following step.

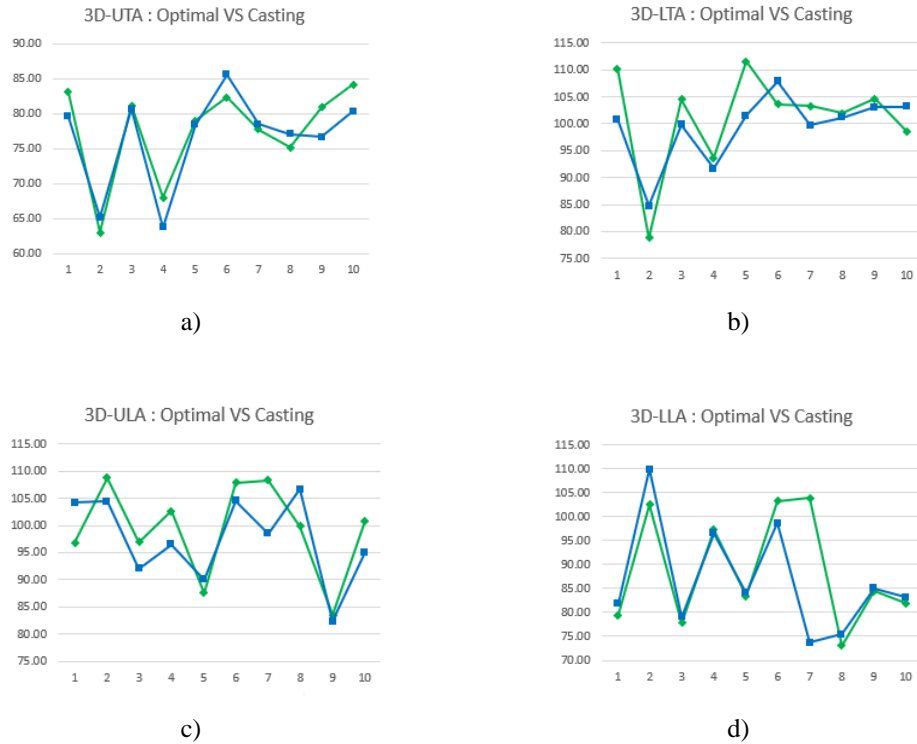


Figure 10.13 The line graph of the 3DSA between the postural measurement system in optimal correction (Green line) and casting (Blue line), a) 3D-UTA, b) 3D-LTA, c) 3D-ULA, and d) 3D-LLA.

A paired t-test for equivalence using the Two One-Sided Test (TOST) was performed to determine whether both groups were equivalent. The margin of equivalence of this analysis was  $5^\circ$  because we allowed the spine to change within  $5^\circ$ . The significance was at the 0.05 level. When considering individual segments (Table 10.16), the 3D-UTA and 3D-LTA had a p-value of both boundaries lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence. In contrast, the p-values of 3D-ULA and 3D-LLA had either lower or upper boundaries lower than 0.05, which could not reject the null hypothesis or indicate a statistically significant equivalence. Then, we varied the margin of equivalence to see which value produced p-values for all spinal parameters less than 0.05. However, the margin of equivalence should not be greater than  $10^\circ$  because it was the largest difference we allowed the spine to change. Finally, the margin of equivalence of this analysis was  $8^\circ$  (Table 10.16). The result showed that all p-values were lower than 0.05, rejected the null hypothesis, and indicated a statistically significant equivalence.

Table 10.16 Equivalence testing between 3DSA from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.

Spinal Parameters	Margin = 5°				Margin = 8°			
	T-Statistic		p-value		T-Statistic		p-value	
	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary
3D-UTA	6.46	-4.52	< 0.001*	0.012*	9.76	-7.81	< 0.001*	< 0.001*
3D-LTA	3.86	-1.88	0.002*	0.047*	5.58	-3.60	< 0.001*	0.003*
3D-ULA	3.83	-1.70	0.002*	0.062	5.49	-3.36	< 0.001*	0.004*
3D-LLA	2.13	-0.93	0.02*	0.188	3.05	-1.85	0.007*	0.049*
All 3D spinal angles	6.52	-3.32	< 0.001*	< 0.001*	9.47	-6.27	< 0.001*	< 0.001*

#### 10.4. Trunk balance

The trunk balance in the coronal plane or coronal decompensation was a primary parameter clinicians used to observe the trunk as a whole spine. If the result was zero millimetres, the trunk was balanced or had no coronal decompensation or trunk leaning. In contrast, if the result was greater than zero, the trunk was not balanced, or the coronal decompensation presented to either the left or right side. While the patients are wearing the spinal orthosis, clinicians expect that the patients should not present the trunk leaning, or this distance should be close to zero as much as possible to keep the whole spine in balance. In this clinical experiment, the trunk balance was collected 4 times, including the radiographs and the low-cost postural measurement system in assessment, optimal correction, and casting.

Table 10.17 reports the mean and SD between trunk balance from radiographs and the low-cost postural measurement system in assessment, optimal correction, and casting. The mean trunk balance results were -11.40 mm (SD = 10.38) in radiographs during the assessment process, -8.19 mm (SD = 11.58) in developed systems during the assessment process, -1.25 mm (SD = 4.56) in developed systems during the optimal correction process, and -0.71 mm (SD = 3.32) in developed systems during the casting process, respectively. It notes that the minus (-) value meant the trunk leaned to the left side. In contrast, the plus (+) value meant the trunk leaned to the right side.

Table 10.17 Mean and SD of trunk balance from radiographs and the postural measurement in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
Trunk balance	-11.40 (10.38)	-8.19 (11.58)	-1.25 (4.56)	-0.71 (3.32)

Note. SD = standard deviation.

#### 10.4.1. Trunk balance between radiographs and the postural measurement in assessment

The following spinal parameter we analysed was trunk balance, comparing radiographs and the postural measurement in assessment to evaluate the relationship between both groups. Figure 10.14 illustrates a line graph of the trunk balance between radiographs (Black line) and the postural measurement in assessment (Red line). As can be seen from the graph, the postural measurement system result followed the same trend as the radiographs for all participants. There was a slight deviation in this parameter. The statistical analysis was used to describe the relationship between both groups in the following step.

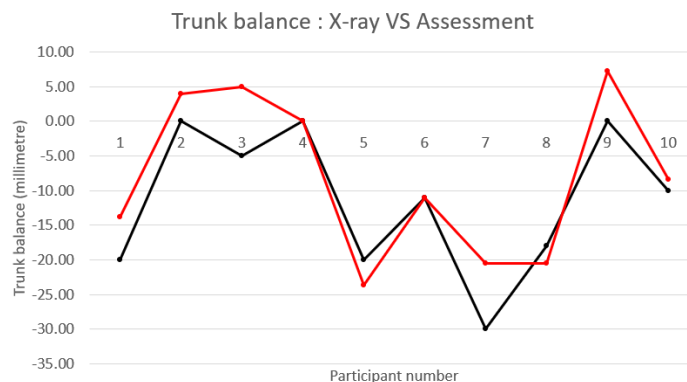


Figure 10.14 The line graph of the trunk balance between radiographs (Black line) and the postural measurement system in assessment (Red line).

The Pearson correlation coefficient was then performed to analyse the relationship between radiographs and the postural measurement in assessment. Figure 10.15 illustrates the scatter plot of the trunk balance between radiographs and the postural



measurement system in assessment. Furthermore, Table 10.18 reports the Pearson correlation results (r-value) for this spinal parameter. The results showed that the r-value was 0.907 with a p-value lower than 0.01. Therefore, it rejected the null hypothesis and accepted the alternative hypothesis with statistically significant at the 0.01 level (2-tails). The relationship level was very high and in a positive direction. It could be concluded that the low-cost postural measurement system could be used to quantify the trunk balance, which was very highly correlated to the radiographic results.

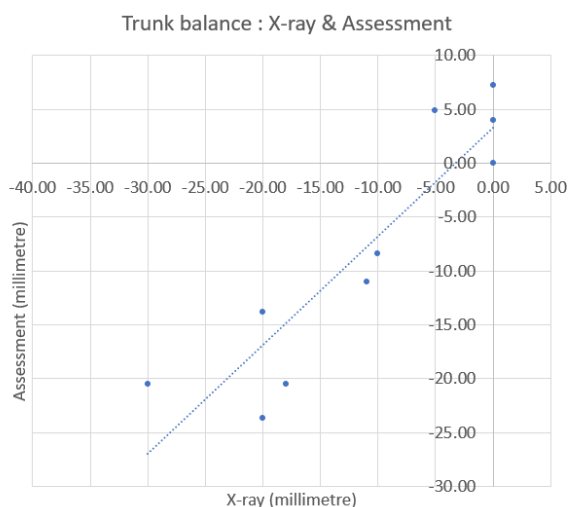


Figure 10.15 The scatter plot of the trunk balance between radiographs and the postural measurement system in assessment. The X-axis is the data from radiographs in assessment, and the Y-axis is the data from the postural measurement system in assessment.

Table 10.18 Pearson correlation coefficient between trunk balance from radiographs and the postural measurement in assessment.

Spinal Parameters	Pearson's correlation (r)	p value	Relationship Level
Trunk balance	0.907	<0.001**	Very high positive

Note: mm = millimetre, SD = standard deviation. \* Correlation is significant at the 0.05 level (2-tailed), \*\* Correlation is significant at the 0.01 level (2-tailed).

#### 10.4.2 Trunk balance between the postural measurement in assessment and optimal correction

The following spinal parameter we analysed was the trunk balance, comparing assessment and optimal correction to evaluate the difference between both groups.

Figure 10.16 illustrates a line graph of the trunk balance between the postural measurement system in assessment (Red line) and optimal correction (Green line). As can be seen from the graph, the result from the assessment highly deviated from the optimal correction. The statistical analysis was used to describe the difference between both groups in the following step.

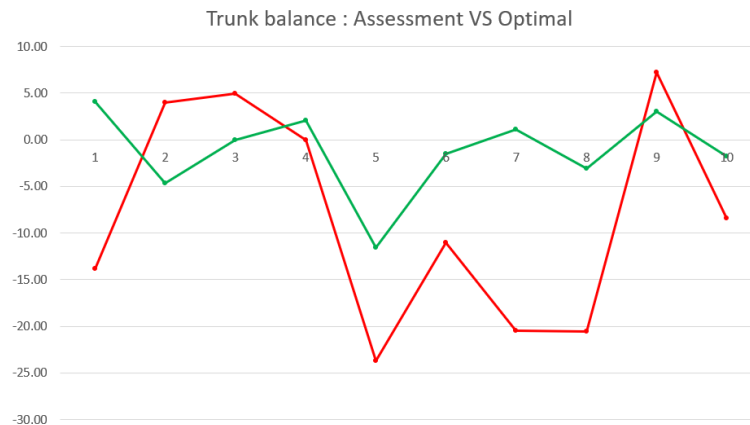


Figure 10.16 The line graph of the trunk balance between the postural measurement system in assessment (Red line) and optimal correction (Green line).

A paired sample t-test was conducted to determine the mean difference in trunk balance between the low-cost postural measurement system in assessment and optimal correction (significant level at 0.05, 2-tailed). The results in Table 10.19 reported that the p-value was higher than 0.05, failed to reject the null hypothesis, and indicated a non-significant difference. It could be concluded that there was a change in the trunk balance or coronal decompensation between the low-cost postural measurement system in assessment (mean = -8.19 mm, SD = 11.58) and optimal correction (mean = -1.25 mm, SD = 4.56), but no statistically significant difference. However, the mean of the optimal correction was close to zero, indicating a high improvement in the trunk balance.

Table 10.19 Comparison between trunk balance from the postural measurement in assessment and optimal correction, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
Trunk balance	-14.51	0.63	-2.073	0.068

Note: CI = Confidence interval. \* Statistically significant at the 0.05 level.

### 10.4.3. Trunk balance between the postural measurement in assessment and casting

The following spinal parameter we analysed was the trunk balance, comparing assessment and casting to evaluate the difference between both groups. Figure 10.17 illustrates a line graph of the trunk balance between the postural measurement system in assessment (Red line) and casting (Blue line). As can be seen from the graph, the result from the assessment highly deviated from the casting. The statistical analysis was used to describe the difference between both groups in the following step.

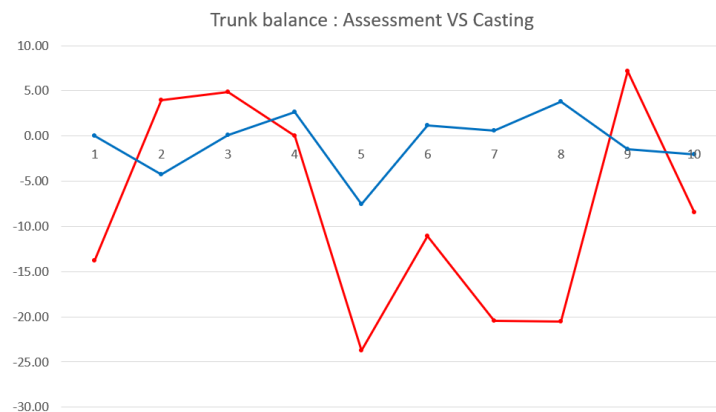


Figure 10.17 The line graph of the trunk balance between the postural measurement system in assessment (Red line) and casting (Blue line).

A paired sample t-test was conducted to determine the mean difference in trunk balance between the postural measurement in assessment and casting (significant level at 0.05, 2-tailed). The results in Table 10.20 reported that the p-value was higher than 0.05, failed to reject the null hypothesis, and indicated a non-significant difference. It could be concluded that there was a change in the trunk balance or coronal decompensation between the low-cost postural measurement system in assessment (mean = -8.19 mm, SD = 11.58) and casting (mean = -0.71 mm, SD = 3.32), but no statistically significant difference. However, the mean of the casting was close to zero, similar to the optimal correction, indicating a high improvement in the trunk balance.

Table 10.20 Comparison between trunk balance from the postural measurement in assessment and casting, using Paired sample t-test.

Spinal Parameters	95% CI		t	p-value
	Lower	Upper		
Trunk balance	-16.04	1.09	-1.975	0.080

Note: CI = Confidence interval. \* Statistically significant at the 0.05 level.

#### 10.4.4. Trunk balance between the postural measurement in optimal correction and casting

The following spinal parameter were analysed was the trunk balance, comparing optimal correction and casting to evaluate the equivalence between both groups. Figure 10.18 illustrates a line graph of the trunk balance between the postural measurement system in optimal correction (Green line) and casting (Blue line). As can be seen from the graph, the result from the optimal correction went in the same direction as the casting, and there was a slight deviation of the values in all participants. The statistical analysis was used to describe the equivalence between both groups in the following step.

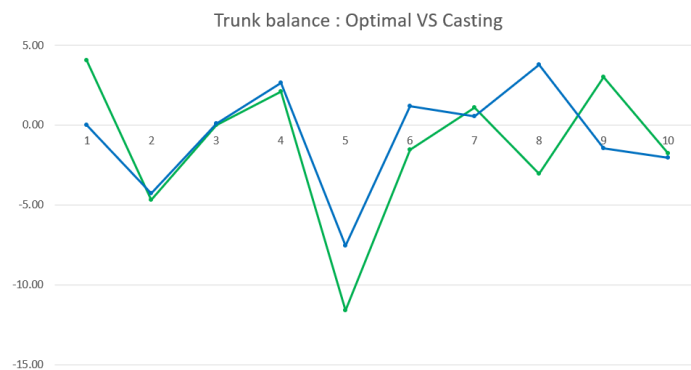


Figure 10.18 The line graph of the trunk balance between the postural measurement system in optimal correction (Green line) and casting (Blue line).

A paired t-test for equivalence using the Two One-Sided Test (TOST) was performed to determine whether both groups were equivalent. The margin of equivalence of this analysis was 5 mm because we allowed the trunk balance to change within 5 mm, and the significant level was at 0.05. The results in Table 10.21 reported that the p-value was lower than 0.05, rejecting the null hypothesis. As a result, it

indicated a statistically significant equivalence of trunk balance from the low-cost postural measurement system in the optimal correction and casting process. Then, we varied the margin of equivalence to see which lowest value gave a p-value less than 0.05. Finally, the lowest value of the margin of equivalence was 3 mm. It could be concluded that the trunk balance from the optimal correction and casting processes were equivalent, and their average difference was confined within a small margin of 3 mm.

Table 10.21 Equivalence testing between trunk balance from the postural measurement in optimal correction and casting, using Paired t-test for equivalence.

Spinal Parameters	Margin = 5 mm				Margin = 3 mm			
	T-Statistic		p-value		T-Statistic		p-value	
	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary	Lower Boundary	Upper Boundary
Trunk balance	4.13	-5.13	0.0013*	<0.001*	2.28	-3.28	0.024*	0.005*

Note: \* Statistically significant at the 0.05 level.

### 10.5. Trunk asymmetry in coronal plane between the low-cost postural measurement system in assessment and casting

Trunk asymmetry is commonly observed during physical examination to evaluate the body asymmetry in the coronal plane. Six parameters were collected for analysing these parameters, including the trunk balance, the left and right distance at the axilla level, the left and right distance at the waist level, the shoulder level, the axilla level, and the waist level. Table 10.22 reports the summary results of the POsterior Trunk Symmetry Index (POTSI) at each index and compares the mean and SD between the low-cost postural measurement system in the assessment and casting.

The mean difference of both comparisons showed that all indices were improved, or the asymmetry percentages were decreased from assessment to casting processes. The 3-dimensional forces applied during casting could improve the trunk asymmetry in the coronal plane in AIS patients. The maximum asymmetry percentage difference was at the left and right distance at the waist (3.54%), followed by trunk balance as a second order (3.13%). The third order was the waist level (1.28%). Finally, the left

and right distance at the axilla, axilla level, and shoulder level were the fourth order and presented with a slight improvement (ranging between 0.91% – 0.94%).

Table 10.22 The results of the POTSI in 6 parameters compared between the assessment and casting.

POTSI	Assessment Mean (SD)	Casting Mean (SD)	Mean Diff between Assessment & Casting
Trunk balance, percentage	3.95 (2.70)	0.82 (0.80)	3.13
Left and right distance at axilla, percentage	7.92 (6.03)	6.98 (5.55)	0.94
Left and right distance at waist, percentage	8.44 (6.43)	4.90 (4.76)	3.54
Shoulder level, percentage	2.55 (2.01)	1.64 (1.85)	0.91
Axilla level, percentage	3.10 (2.41)	2.16 (1.67)	0.94
Waist level, percentage	7.59 (5.03)	6.31 (4.81)	1.28

Note. SD = standard deviation, POTSI = POsterior Trunk Symmetry Index.

Table 10.23 reports the total POTSI percentage in the assessment and casting processes. The table shows that the mean POTSI percentage in the assessment was 33.54% (SD = 16.23), and the mean POTSI percentage in the casting process was 22.80% (SD = 12.41). It could be concluded that the POTSI or asymmetry percentage was reduced or improved from the assessment to the casting with a mean difference of 10.74% (SD = 14.14). This result could be converted to the percentage of reduction, and it accounted for a 28.27% reduction. The 3-dimensional biomechanical force correction from the scoliosis casting apparatus could assist the clinician in correcting the deformity and reducing or improving the asymmetry of the trunk in the coronal plane.

Table 10.23 The summary results of the POTSI compared between the assessment and casting.

POTSI	Mean	SD
Assessment, percentage	33.54	16.23
Casting, percentage	22.80	12.41
Difference between assessment and casting, percentage	10.74	14.15
Percent reduction between assessment and casting, percentage	28.27	

Note. SD = standard deviation, POTSI = POsterior Trunk Symmetry Index.

## **10.6. Trunk horizontal rotation between the low-cost postural measurement system in assessment and casting**

The horizontal trunk rotation is one of the characteristics of scoliotic deformity that can be seen in the top view. The horizontal trunk rotation commonly happens on the same side of the apical vertebrae. For example, the thoracic rotation is in the clockwise direction if the patient has a right thoracic curve, or the lumbar rotation is in the anticlockwise direction if the patient has a left lumbar curve. The left and right markers were attached to the participant's trunk at the same level as T7, T12, L2, and L5. These marker levels were compared with the horizontal angle from the PSIS.

Table 10.24 reports individual participants' total angle of horizontal trunk rotation during the assessment and casting processes using the low-cost postural measurement system. The minute sign (-) value indicated that the total angle rotated in an anticlockwise direction. The plus value (+) meant that the total angle rotated in a clockwise direction. The same table also reports the reduction angle from the assessment to the casting processes. The reduction angle of participant 1 was 10.22°, participant 2 was 13.34°, participant 3 was 21.68°, participant 4 was 5.56°, participant 5 was 3.39°, participant 6 was 9.07°, participant 7 was 36.20°, participant 8 was 36.43°, participant 9 was 5.68°, and participant 10 was 6.75°, respectively. The mean of the total reduction of the horizontal trunk rotation angle was 14.83° (SD = 12.44).

It could be concluded that the scoliosis casting apparatus could assist the clinician in correcting the deformity by reducing the curve magnitude in the coronal plane, maintaining the curve in the sagittal plane, and de-rotating the curve in the transverse plane. The low-cost postural measurement system could also provide the results of horizontal trunk rotation to evaluate the change of scoliosis deformity in the transverse. The horizontal trunk rotation was improved when the 3-dimensional biomechanical force correction was applied to correct the deformity. The reduction angle would depend on the patient's characteristics, such as curve flexibility. Overall, the total horizontal trunk rotation reduction angle was approximately 15°.

Table 10.24 Total angles of horizontal trunk rotation of individual participants during the assessment and casting process

	Assessment (Degree)	Casting (Degree)	Reduction Angle (Degree)
Participant 1	-6.42	3.80	10.22
Participant 2	20.51	7.17	13.34
Participant 3	29.76	8.08	21.68
Participant 4	-17.35	-11.79	5.56
Participant 5	3.49	6.89	3.39
Participant 6	2.13	-6.93	9.07
Participant 7	-34.99	1.21	36.20
Participant 8	-14.13	22.30	36.43
Participant 9	-4.45	1.23	5.68
Participant 10	-10.75	-4.00	6.75
All participants, mean (SD)			14.83 (12.44)

Note. SD = standard deviation. The minute sign (-) value indicates that the total angle rotates in an anticlockwise direction. The plus value (+) means that the total angle rotates in a clockwise direction.

### 10.7. Force magnitude during optimal correction process

The corrective and counter forces were applied to the participant's trunks to correct or reduce the spinal deformity. This process was performed using the scoliosis casting apparatus to apply the forces during the optimal correction and casting. The load cells or force sensors were attached to the manipulator's end or the padding area where it directly contacted the participant's skin. All forces were measured after reaching the maximum force the participant could tolerate. Seven areas of applied force could be measured, including the left gluteus medius, right gluteus medius, left corrective, right corrective, left axilla, right axilla, and abdominal area. However, the forces' locations varied depending on the curve pattern, curve type, and other clinical considerations.

Table 10.25 reports the mean and SD of applied forces in seven areas during the optimal correction. The mean force of the left gluteus medius area was 26.31 N (SD = 10.46), the right gluteus medius area was 20.82 N (SD = 5.68), the left corrective area was 33.36 N (SD = 11.74), the right corrective area was 33.97 N (SD = 9.99), the left axilla area was 28.34 N (SD = 11.34), the right axilla was 27.15 N (SD = 11.50) and the abdominal area was 30.11 N (SD = 6.82). The mean of total forces each participant had to tolerate while correcting the deformity was 153.84 N (SD = 26.35).



It could be concluded that the scoliosis casting apparatus could assist the clinician in correcting the deformity, and the force-measuring system could measure the force magnitude during optimal correction and casting to get the exact force value. Overall, the force to correct the deformity at each area was approximately 30 N, and the total force each patient had to tolerate during optimal correction and casting was approximately 150 N.

Table 10.25 Mean and standard deviation of forces in various areas to correct scoliosis during the optimal correction (n=10).

Area of force	Mean (N)	SD (N)
Left gluteus medius	26.31	10.46
Right gluteus medius	20.82	5.68
Left corrective	33.36	11.74
Right corrective	33.97	9.99
Left axilla	28.34	11.34
Right axilla	27.15	11.50
Abdominal	30.11	6.82
Total forces per participant	153.84	26.35

Note. N = newton, SD = standard deviation.

## 10.8. Conclusion

This chapter reported the results from data analysis in ten AIS participants using descriptive and inferential statistics. The results analysed the relationship between radiographs and the developed system in assessment and the change or similarity between the developed system in assessment, optimal correction, and casting. All data were normally distributed, and a parametric test was selected for the inferential statistics. Table 10.26 summarises the main results from the clinical experiment.

Regarding the relationship between the radiographs in assessment and the postural measurement system in assessment, there was a moderate to very high positive correlation in the CSA, a very high positive correlation in the trunk balance, and a low to very high positive correlation in the SSA. When comparing the assessment VS optimal correction and the assessment VS casting in the CSA, there was a high reduction percentage of apical translation with a statistically significant difference.

However, there was no statistically significant difference in the angle change (Márkus *et al.*). For the equivalence of the optimal correction VS casting in the CSA, there was a statistically significant equivalence when the margin of equivalence was 5°.

In sagittal spinal angles, we aimed not to significantly change the spinal angle at this plane when applying the forces to correct the deformity. The result showed no statistically significant equivalence when the margin of equivalence was 5°. However, there was a statistically significant equivalence when the margin of equivalence changed to 9° for assessment VS optimal correction and 8° for assessment VS casting and optimal correction VS casting, respectively.

For the 3-dimensional spinal angle, there was no statistically significant difference for assessment VS optimal correction and assessment VS casting. There was no statistically significant equivalence when the margin of equivalence was 5° for optimal correction VS casting. However, there was a statistically significant equivalence when the margin of equivalence changed to 8°.

For the trunk balance, there was a high reduction from assessment to optimal correction and casting. However, there was no statistically significant difference in both comparisons. For optimal correction VS casting, there was a statistically significant equivalence when the margin of equivalence was 3 mm.

For the POTSI, the trunk asymmetry improved from 33.54% (SD = 16.23) in assessment to 22.80% (SD = 12.41) in casting. The maximum asymmetry percentage difference (improvement) was at the left and right distance at the waist and trunk balance. Waist level, the left and right distance at the axilla, axilla level, and shoulder level were slightly improved.

For the horizontal trunk rotation, the mean of the total reduction of the horizontal trunk rotation angle was 14.83° (SD = 12.44).

For the force magnitude, the force to correct the deformity at each area was approximately 30 N, and the total force each patient had to tolerate during optimal correction and casting was approximately 150 N. The highest force was present at the left and right corrective forces, followed by the abdominal force. A slightly lower force was present at the left and right axilla forces. The smallest forces were present at the left and right gluteus medius forces.

Table 10.26 Summary of results from the clinical experiment.

Parameters	Type of descriptive statistics	Type of inferential statistics	Result
1. CSA: Radiographs in assessment VS Postural measurement system in assessment	Mean (SD)	Pearson correlation coefficient	<ul style="list-style-type: none"> <li>• CUTA (<math>r = 0.958</math>), CLTA (<math>r = 0.568</math>), CULA (<math>r = 0.878</math>), CLLA (<math>r = 0.961</math>), and all CSA (<math>r = 0.915</math>)</li> <li>• Moderate to very high positive correlation</li> </ul>
2. CSA: Postural measurement system in assessment VS Postural measurement system in optimal correction	Mean (SD)	Paired t-test	<ul style="list-style-type: none"> <li>• Reduction percentage: 51.21% (SD = 13.42)</li> <li>• No statistically significant difference in CSA</li> <li>• Statistically significant difference in reduction of apical translation</li> </ul>
3. CSA: Postural measurement system in assessment VS Postural measurement system in casting	Mean (SD)	Paired t-test	<ul style="list-style-type: none"> <li>• Reduction percentage: 59.53% (SD = 13.04)</li> <li>• No statistically significant difference in CSA</li> <li>• Statistically significant difference in reduction of apical translation</li> </ul>
4. CSA: Postural measurement system in optimal correction VS Postural measurement system in casting	Mean (SD)	Paired t-test for equivalence using TOST	<ul style="list-style-type: none"> <li>• Statistically significant equivalence when the margin of equivalence was 5°.</li> </ul>
5. SSA: Radiographs in assessment VS Postural measurement system in assessment	Mean (SD)	Pearson correlation coefficient	<ul style="list-style-type: none"> <li>• SUTA (<math>r = 0.526</math>), SLTA (<math>r = 0.731</math>), SULA (<math>r = 0.348</math>), SLLA (<math>r = 0.932</math>), and all SSA (<math>r = 0.912</math>)</li> <li>• Low to very high positive correlation</li> </ul>
6. SSA: Postural measurement system in assessment VS Postural measurement system in optimal correction	Mean (SD)	Paired t-test for equivalence using TOST	<ul style="list-style-type: none"> <li>• Statistically significant equivalence when the margin of equivalence was 9°.</li> </ul>

Parameters	Type of descriptive statistics	Type of inferential statistics	Result
7. SSA: Postural measurement system in assessment VS Postural measurement system in casting	Mean (SD)	Paired t-test for equivalence using TOST	<ul style="list-style-type: none"> <li>Statistically significant equivalence when the margin of equivalence was 8°.</li> </ul>
8. SSA: Postural measurement system in optimal correction VS Postural measurement system in casting	Mean (SD)	Paired t-test for equivalence using TOST	<ul style="list-style-type: none"> <li>Statistically significant equivalence when the margin of equivalence was 8°.</li> </ul>
9. 3DSA: Postural measurement system in assessment VS Postural measurement system in optimal correction	Mean (SD)	Paired t-test	<ul style="list-style-type: none"> <li>No statistically significant difference</li> </ul>
10. 3DSA: Postural measurement system in assessment VS Postural measurement system in casting	Mean (SD)	Paired t-test	<ul style="list-style-type: none"> <li>No statistically significant difference</li> </ul>
11. 3DSA: Postural measurement system in optimal correction VS Postural measurement system in casting	Mean (SD)	Paired t-test for equivalence using TOST	<ul style="list-style-type: none"> <li>Statistically significant equivalence when the margin of equivalence was 8°.</li> </ul>
12. Trunk balance: Radiographs in assessment VS Postural measurement system in assessment	Mean (SD)	Pearson correlation coefficient	<ul style="list-style-type: none"> <li>r-value = 0.91</li> <li>Very high positive correlation</li> </ul>
13. Trunk balance: Postural measurement system in assessment VS Postural measurement system in optimal correction	Mean (SD)	Paired t-test	<ul style="list-style-type: none"> <li>Reduction from -8.19 mm (SD = 11.58) in assessment to -1.25 mm (SD = 4.56) in optimal correction</li> <li>No statistically significant difference</li> </ul>
14. Trunk balance: Postural measurement system in assessment VS Postural measurement system in casting	Mean (SD)	Paired t-test	<ul style="list-style-type: none"> <li>Reduction from -8.19 mm (SD = 11.58) in assessment to -0.71 mm (SD = 3.32) in casting</li> <li>No statistically significant difference</li> </ul>

Parameters	Type of descriptive statistics	Type of inferential statistics	Result
15. Trunk balance: Postural measurement system in optimal correction VS Postural measurement system in casting	Mean (SD)	Paired t-test for equivalence using TOST	<ul style="list-style-type: none"> <li>Statistically significant equivalence when the margin of equivalence was 3 mm.</li> </ul>
16. Trunk asymmetry in coronal plane	Mean (SD)		<ul style="list-style-type: none"> <li>POTSI from 33.54% (SD = 16.23) in assessment to 22.80% (SD = 12.41) in casting.</li> <li>Maximum asymmetry percentage difference (improvement) was at the left and right distance at the waist (3.54%) and trunk balance (3.13%). Waist level (1.28%), the left and right distance at the axilla (0.94%), axilla level (0.94%), and shoulder level (0.91%) were slightly improved.</li> </ul>
17. Horizontal trunk rotation	Mean (SD)		<ul style="list-style-type: none"> <li>The mean of the total reduction of the horizontal trunk rotation angle was 14.83° (SD = 12.44).</li> </ul>
18. Force magnitude	Mean (SD)		<ul style="list-style-type: none"> <li>The force to correct the deformity at each area was approximately 30 N, and the total force each patient had to tolerate during optimal correction and casting was approximately 150 N.</li> </ul>

Note: SA = Spinal angle, SD = standard deviation, VS = versus, TOST = Two One-Sided Test, 3D = 3-dimensional. CSA = coronal spinal angle, SSA = sagittal spinal angle, 3DSA = 3-dimensional spinal angle. r = Pearson correlation coefficient result.

## **11. Chapter 11 Discussion, Limitation, Recommendation, and Conclusion**

## 11.1. Introduction

Scoliosis is a 3-dimensional deformity of the spine. The spinal column laterally deviates from the centre in the coronal plane, reduces or increases the spinal curve in the sagittal plane, and the spine rotates in the transverse plane. Radiographs have become a standard tool for quantifying spinal deformity in scoliosis, and most research has focused on spinal deformity change in the coronal and sagittal planes, but there is still a lack of understanding of spinal deformity in three dimensions. The casting is an essential process to capture the shape of the scoliosis trunk under the forces applied by the clinician's hands. It requires skill and teamwork from clinicians to achieve a good negative cast. The use of a casting frame is rare in clinical practice because it is relatively expensive and complicated to use. There is still a lack of information about the effectiveness and benefits of using a casting frame to assist clinicians in obtaining a good negative cast. Furthermore, there is still a lack of information about the magnitude of forces applied to correct the scoliosis deformity and the locations and directions of forces in three dimensions during the casting process.

Therefore, this thesis developed three systems to solve these issues, a low-cost postural measurement system, a scoliosis casting apparatus, and a force-measuring system. The main purposes of this thesis were to develop a system that could quantify the spinal deformity of AIS patients in three dimensions, a system that could apply forces to correct the spinal deformity in three dimensions, and a system that could measure the magnitude of forces and illustrate those force directions in three dimensions. The second purpose of this thesis was to demonstrate the feasibility of using the developed system to treat AIS patients in clinical practice. As a whole system, this thesis aimed to demonstrate, evaluate, and improve the understanding of the spinal deformity change in three dimensions during the orthotic treatment of AIS patients.

After successful development, the next step was to demonstrate and evaluate the system using the developed systems to treat AIS patients. Ten AIS patients were recruited for the study. The study design was a pilot study, and ten cases could provide enough information to determine the success or otherwise of the system and provide sufficient feedback to evaluate its usefulness and any future modifications. Participants

had to come to the SSPO research room for two appointments for recruitment and data collection processes. Four data groups were collected and compared, namely the results from radiographs and the developed system in assessment, optimal correction, and casting processes. Descriptive and inferential statistics were used to analyse the data, validate the developed system, and highlight the clinical findings in AIS treatment. Six objectives were focused on this clinical experiment and will be discussed in the next section. The chapter then discusses the study's limitations, the recommendations for future research, and the conclusion at the end.

## **11.2. Discussion**

Six objectives are discussed in this section.

- 1. “To verify if the developed postural measurement system could quantify the spinal parameters in coronal and sagittal planes and to study the validity of the spinal parameters using the low-cost postural measurement system compared to the radiographic evaluation.”.**

The spinal parameters for this objective consisted of the coronal spinal angle (CSA), the sagittal spinal angles (SSA), and the trunk balance in the coronal plane. Jang previously introduced the new spinal parameters by dividing the spinal column into four segments (Jang, 2018), and we adopted this method for this thesis for the CSA, SSA, and 3DSA.

To validate the developed system, we studied the relationship between the standard outcome measurement from radiographs and the low-cost postural measurement system in assessment. Three spinal parameters were included in this analysis, namely CSA, SSA, and trunk balance.

- Coronal spinal angles (CSA)**

Regarding the CSA between the radiographs in assessment and the postural measurement system in assessment, the correlation results ranged from 0.57 to 0.96



with the overall correlation of 0.92. To clarify, the correlation results of CUTA, CLTA, CULA, and CLLA were 0.96, 0.57, 0.88, and 0.96, respectively. The relationship level was moderate to very high positive correlation. A similar study was conducted earlier by Jang with 13 AIS patients (Jang, 2018). The results of the previous study showed that the CUTA, CLTA, CULA, and CLLA were 0.87, 0.79, 0.81, and 0.70, respectively. Comparing the studies, correlation results were slightly different, but the relationship level was quite similar ranging from a high to very high positive correlation.

For other motion analysis studies, Solomito, Lee and Peterson used a Vicon system to study the relationship with radiographs in 5 AIS patients and 5 normal subjects (Solomito, Lee and Peterson, 2011). The correlation result was  $>0.75$  (they did not mention the specific number), and the relationship level was a high positive correlation, similar to our study. However, this research studied the correlation of the whole spine, not dividing the spine into segments. Pesenti and colleagues used high-resolution infrared cameras to study the relationship with radiographs in 62 AIS patients (Pesenti *et al.*, 2020). The correlation result was 0.58 for the thoracic and -0.21 for the lumbar, and the relationship level was a moderate positive correlation for the thoracic and a little negative correlation for the lumbar. The correlation in the thoracic was slightly lower than ours, but the correlation in the lumbar was highly different from ours. However, their research studied the correlation by dividing the spine into two segments and their method to quantify the spinal parameters was also different from our study.

Li and colleagues used a 3-dimensional clinical ultrasound system to study the relationship with radiographs in 43 AIS patients (Li *et al.*, 2010, Li *et al.*, 2012). The correlation result was 0.98, and the relationship level was a very high positive correlation. The correlation result was slightly higher than our study, but the relationship level was quite similar. This research studied the correlation of the whole spine, not dividing the spine into segments. A similar study was conducted by Zheng and colleagues in 2016 (Zheng *et al.*, 2016) and Brink and colleagues in 2018 (Brink *et al.*, 2018). Both studies used the Scolioscan to study the relationship to radiographs in 49 AIS patients by Zheng and 33 AIS patients by Brink. The correlation result was 0.72, and the relationship level was a high positive correlation reported in the study from Zheng. The correlation result in the study from Brink was 0.97, and the

relationship level was a very high positive correlation. The studies reported a similar relationship level to our study, ranging from a high to a very positive correlation. This research studied the correlation of the whole spine, not dividing the spine into segments. Lastly, Wong and colleagues used the Scolioscan to study the relationship to radiographs in a big group of scoliosis patients, 952 IS patients (Wong *et al.*, 2019). The correlation result was 0.629 above T6.5, 0.873 at T7 to T12/L1 and 0.740 below L1. The relationship level was a moderate to high positive correlation. The correlation result was slightly lower than our study.

Thometz used the Quantec Spinal Imaging System (QGIS) to study the relationship of surface topography to radiographs in 149 IS patients (Thometz *et al.*, 2000). The spinal column was divided into two segments, thoracic and lumbar. The correlation result was 0.65 in the thoracic and 0.63 in the lumbar, and the relationship level was a moderate positive correlation in both levels, slightly lower than our study. Knott used the Orthoscan to study the relationship to radiographs in 42 scoliosis patients (Knott *et al.*, 2006). The spinal column was also divided into two segments, thoracic and lumbar. The correlation result was 0.65 in the thoracic and 0.13 in the lumbar, and the correlation level was moderate and little positive correlation. The correlation at the thoracic was slightly lower than ours, but the correlation at the lumbar looked highly different from our study and the previous study from Thometz. Frerich used the Formetric 4D surface topography to study the relationship to radiographs in 14 AIS patients (Frerich *et al.*, 2012). The spinal column was also divided into two segments, thoracic and lumbar. The correlation result was 0.87 in the thoracic and 0.76 in the lumbar, and the correlation level was a high positive correlation in both levels, similar to our study. Knott used the DIERS Formetric to study the relationship to radiographs in 193 paediatric scoliosis patients (Knott *et al.*, 2016). The spinal column was also divided into two segments, thoracic and lumbar. The correlation result was 0.73 in the thoracic and 0.49 in the lumbar, and the correlation level was high and low positive correlation. The correlation result at thoracic was slightly lower than our study, but the relationship level was similar. However, the correlation result at the lumbar was highly different to our study. Lastly, the meta-analysis from Navarro and colleagues (Navarro, Candotti and Rosa, 2019) reported that the correlation results at the thoracic and

lumbar were 0.74 and 0.55, respectively. The relationship level was between moderate and high positive correlation, slightly lower than our study.

We conclude that the developed postural measurement system could quantify the coronal spinal angles to evaluate the coronal spinal deformity, and the results were well correlated with radiographs. The correlation between the developed postural measurement system in assessment and the radiographs was between moderate and very high positive correlation, similar to the study from Jang. Compared to other motion analysis studies, the correlation of other studies was between moderate and high positive correlation, which was slightly lower than ours. However, we could not directly compare their results to ours because they used different methods to calculate the spinal parameters. Some studies considered the spine as one segment, and some divided the spine into two segments (thoracic and lumbar regions). Compared to ultrasound analysis studies, the correlation of most studies was between high and very high positive correlation, similar to ours. Again, we could not directly compare their results to ours because they used different methods to calculate the spinal parameters. Some studies considered the spine as one segment, and some divided the spine into two segments (thoracic and lumbar regions). Interestingly, the latest article conducted using ultrasound with a big group of IS patients (952 participants) reported a correlation between moderate and high positive correlation, a slightly lower correlation than other ultrasound studies and our study. Compared to surface topography analysis studies, the correlation was between low and high positive correlations, slightly lower than the postural measurement system, ultrasound system, and our study. The correlation results varied and depending on the type of scanning system.

Regarding the CSA between assessment and optimal correction, the study found that the reduction percentage of the apical translation from assessment to optimal correction was 51.21% (SD = 13.42). The reduction percentage was higher than 50%, following the suggestion from the Boston principle to achieve the in-brace reduction percentage (Boston Orthotics & Prosthetics, 2023). However, it could not directly compare the result with the Boston concept because it used a different method to calculate the reduction percentage. Two inferential statistics were used to analyse the change. For the reduction of the apical translation, the p-value from the paired sample

t-test was lower than 0.05. There was a statistically significant difference between the apical translation in assessment and optimal correction. For the CSA, the p-value from the paired sample t-test was higher than 0.05. The CSA or spinal deformity in the coronal plane was corrected from assessment to optimal correction. However, there was no statistically significant difference in the angle change or insufficient data to indicate the change.

Regarding the CSA between the assessment and casting process, the reduction percentage of the apical translation was 59.53% (SD = 13.04). The reduction percentage was higher than 50%, following the suggestion from the Boston principle to achieve the in-brace reduction percentage (Boston Orthotics & Prosthetics, 2023), similar to assessment VS optimal correction. The reduction percentage of assessment VS casting was slightly higher than the assessment VS optimal correction. There were two possible reasons to explain the difference. The first was from the spinal structure and self-correction by patients. As the spine was not a rigid structure, the patients could actively change their trunk position away from the forces applied. It could help straighten the spine by self-correction. The second was the error in quantifying the spinal parameters from the positive model. As the result, the spinal parameters in casting process was measured by the positive model, not directly from the patients, it might cause some errors from this technique. Two inferential statistics were used to analyse the change. For the reduction of the apical translation, the p-value from the paired sample t-test was lower than 0.05. There was a statistically significant difference between the apical translation in assessment and casting. For the CSA, the p-value from the paired sample t-test was higher than 0.05. The CSA or spinal deformity in the coronal plane was corrected from assessment to casting. However, there was no statistically significant difference in the angle change or insufficient data to indicate the change.

Regarding the CSA between the optimal correction and casting process, the p-value from the paired t-test for equivalence was lower than 0.05, indicating a statistically significant equivalence of these spinal parameters. The spinal angles between both groups were equivalence and confined within a small margin of 5°.

Jang previously studied CSA in four spinal segments with 20 non-scoliosis persons (Jang, 2018). The study divided the values of this angle into three groups, including

the neutral alignment, the potentially abnormal alignment, and the mal-alignment, as shown in Table 11.1. Table 10.1 reports the coronal spinal angles from our study (refer to Chapter 10) with the colour ranges. To clarify, we added the colour in the table to indicate the zone of the alignment. The green box in the table indicates the neutral alignment, the yellow box indicates the potentially abnormal alignment, and the red box indicates the mal-alignment.  $\pm 1$ SD was approximately 5°.

Table 11.1 CSA of non-scoliosis persons proposed by Jang (n = 20)

SP	Mal-alignment	Potentially abnormal alignment	Neutral alignment			Potentially abnormal alignment	Mal-alignment
	< -2 SD	-2 SD	-1 SD	Mean	+ 1 SD	+ 2 SD	> +2 SD
CUTA	<86.15	86.15	88.12	90.08	92.05	94.02	>94.02
CLTA	<85.21	85.21	87.36	89.51	91.66	93.81	>93.81
CULA	<85.07	85.07	87.23	89.39	91.54	93.70	>93.70
CLLA	<85.35	85.35	87.25	89.16	91.06	92.96	>92.96

Note. SP = spinal parameter, SD = standard deviation.

Table 10.1. (Refer to Chapter 10) Mean and SD of CSA from radiographs and the postural measurement in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
CUTA, degree	94.10 (3.21)	94.05 (2.90)	92.49 (2.35)	92.91 (1.88)
CLTA, degree	85.90 (7.16)	86.09 (3.12)	86.70 (5.30)	86.15 (3.28)
CULA, degree	82.90 (7.99)	86.24 (8.49)	87.69 (6.79)	88.48 (4.19)
CLLA, degree	100.30 (12.82)	99.12 (14.07)	93.66 (6.24)	92.33 (5.39)

Note. CUTA = coronal upper thoracic angle, CLTA = coronal lower thoracic angle, CULA = coronal upper lumbar angle, CLLA = coronal lower lumbar angle. SD = standard deviation.

When comparing this result to our result, all spinal parameters in the radiographs in assessment and most spinal parameters in the postural measurement system in the assessment are shown with the red box, indicating the mal-alignment of the CSA. When comparing the assessment and optimal correction results, most spinal parameters were improved, and the colour changed from red to yellow or yellow to green, indicating the improvement of the CSA. Although the CLLA remained in the red zone, the mean was decreased and closer to the neutral alignment zone. When comparing the assessment and casting results, most spinal parameters were improved,

similar to assessment VS optimal correction. The colour changed from red to yellow or yellow to green, indicating the improvement of the coronal spinal angle. Although the CLLA remained in the yellow zone, the mean was decreased and closer to the neutral alignment zone. When comparing the optimal correction and casting results, most spinal parameters remained in the same zone, indicating the similarity of the coronal spinal angle in both processes. The CLLA changed the colour zone from red to yellow. However, the mean of both groups was quite similar.

The developed postural measurement system could quantify the CSA during assessment, optimal correction, and casting processes. The reduction percentage showed that the coronal deformity improved with a reduction percentage of more than 50%, following the suggestion from the Boston principle to achieve the in-brace reduction percentage. However, it could not directly compare the result with the Boston concept because it used a different method to calculate the reduction percentage. For inferential statistics, the p-value of CSA from the paired sample t-test in assessment VS optimal correction and assessment VS casting were higher than 0.05. The spinal deformity was corrected during optimal correction and casting processes using 3-dimensional force correction from the scoliosis casting apparatus. However, there was no statistically significant difference in the angle change or insufficient data to indicate the change. For the reduction of apical translation, the p-value from the paired sample t-test in assessment VS optimal correction and assessment VS casting were lower than 0.05, indicating a statistically significant difference in the apical translation. The p-value from the paired t-test for equivalence was lower than 0.05 when comparing the optimal correction VS casting. The spinal angles between both groups were equivalence and confined within a small margin of 5°.

- **Sagittal spinal angles (SSA)**

Regarding the SSA between the radiographs in assessment and the postural measurement system in assessment, the correlation results ranged from 0.35 to 0.94 with the overall correlation of 0.91. To clarify, the correlation results of SUTA, SLTA, SULA, and SLLA were 0.53, 0.73, 0.35, and 0.94, respectively. The relationship level was low to very high positive correlation. A similar study was conducted by Jang with

13 AIS patients (Jang, 2018). The results of the previous study showed that the SUTA, SLTA, SULA, and SLLA were 0.92, 0.96, 0.89, and 0.83, respectively, and the relationship level was between high and very high positive correlation.

For other motion analysis studies, Solomito, Lee and Peterson used a Vicon system to study the relationship with radiographs in 5 AIS patients and 5 normal subjects (Solomito, Lee and Peterson, 2011). The correlation result was  $>0.75$  (they did not mention the specific number), and the relationship level was a high positive correlation. However, this research studied the correlation of the whole spine, not dividing the spine into segments. Pesenti and colleagues used high-resolution infrared cameras to study the relationship with radiographs in 62 AIS patients (Pesenti *et al.*, 2020). The correlation result was 0.57 for the thoracic kyphosis and 0.44 for the lumbar lordosis, and the relationship level was a moderate positive correlation for the thoracic and a low positive correlation for the lumbar. The correlation in the thoracic and lumbar areas was slightly lower than ours. However, this research studied the correlation by dividing the spine into two segments, and the method used to quantify the spinal parameters differed from our study. For the ultrasound system, interestingly, no research has studied the correlation between ultrasound and radiographs in the sagittal plane.

For the surface topography system, Frerich used the Formetric 4D surface topography to study the relationship to radiographs in 14 AIS patients (Frerich *et al.*, 2012). The spinal column was also divided into two segments, thoracic kyphosis and lumbar lordosis. The correlation result was 0.80 in the thoracic kyphosis and 0.81 in the lumbar lordosis, and the relationship level was a high positive correlation in both levels. Knott used the DIERS Formetric to study the relationship to radiographs in 193 paediatric scoliosis patients (Knott *et al.*, 2016). The spinal column was also divided into two segments, thoracic kyphosis and lumbar lordosis. The correlation result was 0.83 in the thoracic kyphosis and 0.82 in the lumbar lordosis, and the correlation level was a high positive correlation. The correlation result at thoracic and lumbar was slightly higher than our study. Lastly, the meta-analysis from Navarro and colleagues (Navarro, Candotti and Rosa, 2019) reported that the correlation results at the thoracic kyphosis and lumbar lordosis were 0.84 and 0.82, respectively. The relationship level was a high positive correlation, slightly higher than our study.

We concluded the developed postural measurement system could quantify the sagittal spinal angles to evaluate the sagittal spinal deformity. The correlation between the developed postural measurement system in assessment and the radiographs in assessment was between low and very high positive correlation, slightly lower than the study from Jang. There were two possible reasons to explain the difference. The first was from the trunk position while obtaining the radiographs in assessment and the postural measurement system in our study. Many participants already had the radiographs before attending the study. As a result, we could not control the participant's position while taking the radiographs. Furthermore, the participants had to flex the shoulder and elbow 90° to prevent the unclear spine image from the upper limb while taking the radiographs. However, the participants had to stand relaxed while recording the data from the postural measurement system. As a result, the spinal column might change the range of motion slightly while flexing the upper limb. Moreover, the results from the radiographs in assessment and postural measurement system were collected at different times and on different days, and three months was the maximum difference. Therefore, the participant's position during the radiographs and the postural measurement system in assessment were not exactly the same and would make the error happen during the clinical experiment. The second was from the patient's characteristics. Our study's and Jang's participants had the same pathology, AIS patients. However, the patient's characteristics might not be the same, such as age, the magnitude of the Cobb angle, BMI, and the type of curve. Moreover, both studies were conducted in a small sample size. As a result, we could not directly compare our study to hers, and further study in a big group of AIS patients should be done to clarify this issue.

Compared to other motion analysis studies, the correlation of other studies was between low and high positive correlation. Overall, it was pretty similar to our study's relationship level. However, we could not directly compare their results to ours because they used different methods to calculate the spinal parameters. Some studies considered the spine as one segment, and some divided the spine into two segments (thoracic kyphosis and lumbar lordosis). Compared to surface topography analysis studies, all studies reported a similar result. The relationship level was a high positive correlation for thoracic kyphosis and lumbar lordosis, slightly lower than Jang's study



but slightly higher than other motion systems and our study. Again, we could not directly compare their results to ours because they used different methods to calculate the spinal parameters.

Regarding the SSA between the assessment and optimal correction, the p-value from the paired t-test for equivalence when the margin of equivalence at  $5^\circ$  was higher than 0.05, indicating a non-statistically significant equivalence of these spinal parameters. However, the p-value was lower than 0.05 when the margin of equivalence was  $9^\circ$ , indicating a statistically significant equivalence of these spinal parameters. As a result, the spinal angles between both groups were changed and confined within a small margin of  $9^\circ$ . Table 11.2 reports the SSA from Jang's study (Jang, 2018). The result showed that  $\pm 1SD$  of the sagittal curve change was approximately  $10^\circ$ . No research studied the margin of equivalence of the sagittal angle change in scoliosis. Therefore, we used  $\pm 1SD$  of this study for the maximum sagittal curve change.

Regarding the SSA between the assessment and casting, the p-value from the paired t-test for equivalence when the margin of equivalence at  $5^\circ$  was higher than 0.05, indicating a non-statistically significant equivalence of these spinal parameters. However, the p-value was lower than 0.05 when the margin of equivalence was  $8^\circ$ , indicating a statistically significant equivalence of these spinal parameters. As a result, the spinal angles between both groups were changed and confined within a small margin of  $8^\circ$ , which was lower than the maximum sagittal curve change.

Regarding the SSA between the optimal correction and casting, the p-value from the paired t-test for equivalence when the margin of equivalence at  $5^\circ$  was higher than 0.05, indicating a non-statistically significant equivalence of these spinal parameters. However, the p-value was lower than 0.05 when the margin of equivalence was  $8^\circ$ , indicating a statistically significant equivalence of these spinal parameters. As a result, the spinal angles between both groups were changed and confined within a small margin of  $8^\circ$ , which was lower than the maximum sagittal curve change.

In clinical practice, we try not to decrease or increase the angle of thoracic kyphosis and lumbar lordosis during orthotic treatment. However, no study mentioned how many degrees of the maximum angle could allow the spine to change in this plane. As a result, we set the margin of equivalence or allowed the spinal angle to change at  $5^\circ$  and should not be greater than  $10^\circ$  in the sagittal plane. The inferential statistics result

showed that the margin of equivalence was between 8° and 9° during the assessment, optimal correction, and casting processes, lower than 10°, indicating the equivalence of the sagittal spinal angle of this clinical experiment.

Jang previously studied the SSA in four spinal segments with 20 non-scoliosis persons (Jang, 2018). The study divided the values of this angle into three groups, including the neutral alignment, the potentially abnormal alignment, and the mal-alignment, as shown in Table 11.2. Table 10.8 reports the SSA from our study (refer to Chapter 10) with the colour ranges. To clarify, we added the colour in the table to indicate the zone of the alignment, similar to CSA.  $\pm 1SD$  was approximately 10°, and we used this value as a maximum angle change in SSA.

Table 11.2 SSA of non-scoliosis persons proposed by Jang (n = 20)

SP	Mal-alignment	Potentially abnormal alignment	Neutral alignment			Potentially abnormal alignment	Mal-alignment
	< -2 SD	-2 SD	-1 SD	Mean	+ 1 SD	+ 2 SD	> +2 SD
SUTA	<66.96	66.96	72.37	77.79	83.20	88.61	>88.61
SLTA	<86.48	86.48	93.09	99.70	106.31	112.92	>112.92
SULA	<88.37	88.37	93.05	97.74	102.42	107.10	>107.10
SLLA	<59.90	59.90	68.11	76.32	84.53	92.73	>92.73

Note. SP = spinal parameter, SD = standard deviation.

Table 10.8. (Refer to Chapter 10) Mean and SD of sagittal spinal angles from radiographs in assessment and the low-cost postural measurement system in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)			
	Radiographs in Assessment	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
SUTA, degree	80.90 (4.72)	76.49 (7.99)	78.53 (7.72)	77.03 (6.92)
SLTA, degree	100.20 (3.33)	98.34 (6.12)	100.58 (7.27)	98.37 (6.16)
SULA, degree	103.90 (3.45)	100.76 (5.41)	97.75 (7.38)	96.70 (7.29)
SLLA, degree	90.00 (8.39)	84.77 (11.95)	89.48 (9.38)	87.74 (10.11)

Note. SUTA = sagittal upper thoracic angle, SLTA = sagittal lower thoracic angle, SULA = sagittal upper lumbar angle, SLLA = sagittal lower lumbar angle, SD = standard deviation.

When comparing this result to our result, most spinal parameters in the radiographs in assessment and the postural measurement system in assessment had a similar colour zone, and the spinal parameters ranged from green to red zones, indicating neutral alignment in some segments and potentially abnormal to mal-alignment in others.

When comparing the assessment and optimal correction results, all spinal parameters presented in a similar colour zone ranging from green to red, indicating the change of the sagittal spinal angle, but the change remained in the same alignment zone. When comparing the assessment and casting results, all spinal parameters presented in a similar colour zone ranging from green to red, indicating the change of the sagittal spinal angle, but the change remained in the same alignment zone. When comparing the optimal correction and casting results, all spinal parameters remained in the same zone, indicating the similarity of the sagittal spinal angle in both processes. The spinal parameters ranged from green to red, indicating the change in the sagittal spinal angle, but the change remained in the same alignment zone. Overall, most spinal parameters remained in the same colour zone during the assessment, optimal correction, and casting processes.

- **Trunk balance**

Regarding the trunk balance between the radiographs in assessment and the postural measurement system in assessment, the correlation result was 0.91, and the relationship level was a very high positive correlation. Interestingly, no previous research studied the correlation of the trunk balance between the radiographs and other systems (ultrasound, surface topography, and high-cost motion capture systems).

Regarding the trunk balance between assessment and optimal correction, the mean trunk balance changed from -8.19 mm (SD = 11.58) in the assessment to -1.25 mm (SD = 4.56) in the optimal correction. For inferential statistics, the p-value from the paired sample t-test was higher than 0.05. The result from optimal correction was close to zero, indicating an improvement in the trunk balance from assessment to optimal correction. However, this parameter had no statistically significant difference or insufficient data to indicate the change.

Regarding the trunk balance between assessment and casting, the mean trunk balance changed from -8.19 mm (SD = 11.58) in the assessment to -0.71 mm (SD = 3.32) in the casting process. For inferential statistics, the p-value from the paired sample t-test was higher than 0.05, similar to assessment VS optimal correction. The result from casting was close to zero, indicating an improvement in the trunk balance

from the assessment to the casting. However, this parameter had no statistically significant difference or insufficient data to indicate the change.

Regarding the trunk balance between the optimal correction and casting, the p-value from the paired t-test for equivalence when the margin of equivalence at 5 mm was lower than 0.05, indicating a statistically significant equivalence of these spinal parameters. Moreover, the minimum margin of equivalence that still got a p-value lower than 0.05 was 3 mm. As a result, the trunk balance between both groups was confined within a small margin of 3 mm. It could be implied that the trunk balance between the optimal correction and casting processes had a similar value.

The developed postural measurement system could quantify the trunk balance to evaluate the coronal spinal deformity during assessment, optimal correction, and casting. The scoliosis casting apparatus could be adjusted to apply the 3-dimensional force correction. The apparatus could be used to correct the spinal deformity in three dimensions and reduce or improve the trunk balance in the coronal plane. The Scoliosis Research Society (SRS) defined the “compensation” as the distance between C7 and the sacrum, which was less than 1.5 cm. Furthermore, some studies suggested a 2.0 cm or more to define the coronal decompensation with a poor balance (Karami, Maleki and Mazda, 2016, Lenke, 2000). There is no standard value for optimal coronal decompensation. However, the clinicians aim to minimise this distance to nearly zero as much as possible to improve the trunk balance for orthotic treatment and spinal surgery, and our result was nearly zero following the clinical practice.

In conclusion, we determined that the developed postural measurement system could quantify the spinal parameters in coronal and sagittal planes with sufficient accuracy across the 4 spinal zones and that the spinal parameters so produced using the low-cost postural measurement system showed high concurrent validity when compared to planar radiographic evaluation.

## **2. “To examine if the developed postural measurement system could quantify the 3-dimensional spinal angles”.**

The spinal parameter for this objective was the 3-dimensional spinal angle (3DSA). Jang previously introduced this spinal parameter by dividing the spinal

column into four segments (Jang, 2018), and we adopted this method for this thesis. This angle was a 3-dimensional angle respective to the horizontal plane. This spinal parameter was the new parameter we proposed to clarify the unclear 3-dimensional deformity of AIS patients. To examine the developed system, we compared the result between the developed system before (assessment) and when applying the 3-dimensional force correction from the scoliosis casting apparatus (optimal correction and casting). We also studied the equivalence of the results between the optimal correction and casting to examine the equivalence between before and after applying the plaster bandages.

Regarding the 3DSA between assessment and optimal correction, the p-value from the paired sample t-test was higher than 0.05. The 3DSA changed from assessment to optimal correction. However, there was no statistically significant difference in the angle change or insufficient data to indicate the change.

Regarding the 3DSA between assessment and casting, the p-value from the paired sample t-test was higher than 0.05. The 3DSA changed from assessment to casting. However, there was no statistically significant difference in the angle change or insufficient data to indicate the change.

Table 11.3 reports the 3DSA from Jang's study (Jang, 2018). The result showed that  $\pm 1SD$  of the 3DSA change was approximately  $10^\circ$ . No research studied the margin of equivalence of the 3DSA change in scoliosis. Therefore, we used  $\pm 1SD$  of this study for the maximum 3DSA change. Regarding the 3DSA between the optimal correction and casting, the p-value from the paired t-test for equivalence when the margin of equivalence at  $5^\circ$  was higher than 0.05, indicating a non-statistically significant equivalence of these spinal parameters. However, the p-value was lower than 0.05 when the margin of equivalence was  $8^\circ$ , indicating a statistically significant equivalence of these spinal parameters. As a result, the spinal angles between both groups were confined within a small margin of  $8^\circ$ , which was lower than the maximum 3D spinal angle change.

Jang previously studied the 3DSA in four spinal segments with 20 non-scoliosis persons (Jang, 2018). The study divided the values of this angle into three groups, including the neutral alignment, the potentially abnormal alignment, and the mal-alignment, as shown in Table 11.3. Table 10.13 reports the 3DSA from our study (refer

to Chapter 10) with the colour ranges. To clarify, we added the colour in the table to indicate the zone of the alignment, similar to CSA and SSA.  $\pm 1SD$  was approximately  $10^\circ$ , and we used this value as a maximum angle change in 3DSA.

Table 11.3 Three-dimensional spinal angles of non-scoliosis persons proposed by Jang (n = 20)

SP	Mal-alignment	Potentially abnormal alignment	Neutral alignment			Potentially abnormal alignment	Mal-alignment
	< -2 SD	-2 SD	-1 SD	Mean	+ 1 SD	+ 2 SD	> +2 SD
3D-UTA	<66.96	66.96	72.29	77.61	82.93	88.26	>88.26
3D-LTA	<89.01	89.01	93.5	100	106.49	112.98	>112.98
3D-ULA	<89.01	89.01	93.56	98.1	102.65	107.2	>107.20
3D-LLA	<59.97	59.97	68.03	76.1	84.17	92.24	>92.24

Note. SP = spinal parameter, n = number of participants, SD = standard deviation.

Table 10.13. (refer to Chapter 10) Mean and standard deviation of 3-dimensional spinal angles from the low-cost postural measurement system in assessment, optimal correction, and casting.

Spinal Parameters	Mean (SD)		
	Postural measurement in Assessment	Postural measurement in Optimal correction	Postural measurement in Casting
3D-UTA, degree	75.37 (7.43)	77.49 (6.92)	76.60 (6.85)
3D-LTA, degree	99.37 (6.52)	101.09 (9.35)	99.37 (6.53)
3D-ULA, degree	102.06 (5.13)	99.34 (8.55)	97.42 (7.83)
3D-LLA, degree	79.57 (17.63)	88.69 (11.76)	86.73 (11.41)

Note: 3D-UTA = 3-dimensional upper thoracic angle, 3D-LTA = 3-dimensional lower thoracic angle, 3D-ULA = 3-dimensional upper lumbar angle, and 3D-LLA = 3-dimensional lower lumbar angle.

When comparing the assessment and optimal correction results, most spinal parameters were presented in the green zone, indicating the change in the 3-dimensional spinal angle. However, the change remained in the same alignment zone. Except for 3D-LLA in optimal correction, it was presented in the yellow zone, indicating the potentially abnormal alignment in this segment. When comparing the assessment and casting results, the colour was presented similarly to assessment VS optimal correction. Most spinal parameters were presented in the green zone, indicating the change in the 3-dimensional spinal angle. However, the change remained in the same alignment zone. Except for 3D-LLA in the casting process, it was presented in the yellow zone, indicating the potentially abnormal alignment in this segment. When comparing the optimal correction and casting results, all spinal

parameters remained in the same zone, indicating the similarity of the 3-dimensional spinal angle in both processes. The spinal parameters ranged from green to yellow (but mostly were green), indicating the change in the 3-dimensional spinal angle, but the change remained in the same alignment zone.

The developed postural measurement system could quantify the 3-dimensional spinal angle to evaluate the 3-dimensional spinal deformity during assessment, optimal correction, and casting. The scoliosis casting apparatus could be adjusted to apply the 3-dimensional force correction to correct the spinal deformity, and the developed postural measurement system could provide the results of spinal angles in three dimensions.

We concluded that the developed postural measurement system could quantify the 3-dimensional spinal angles. This parameter can help clarify how the spine changes in three dimensions.

### **3. “To examine if the developed postural measurement system could quantify the angle of trunk rotation in the horizontal plane”.**

The spinal parameter for this objective was the angle of horizontal trunk rotation at the T7, T12, L2, and L5 levels in relation to the PSIS level. Previous research from Jang studied the horizontal rotation at interior angle of the scapula (or T7 level) and paraspinal muscles around the L3/L4 area (Jang, 2018). This thesis continued to use this parameter at T7 and added more levels to cover the trunk shape. This spinal parameter was the new parameter we proposed to clarify the unclear deformity of AIS patients in the transverse plane. To examine the developed system, we compared the result from the developed system before (assessment) and when applying the 3-dimensional force correction from the scoliosis casting apparatus in the casting.

Regarding the trunk rotation in the transverse plane, there was an improvement for all participants from the assessment to casting, and the mean reduction angle was  $14.83^\circ$  (SD = 12.44). As this was the new spinal parameter to describe the deformity of scoliosis in the transverse plane, we could not compare and discuss the result with other researchers.

The developed postural measurement system could quantify the horizontal trunk rotation to evaluate the spinal deformity in the transverse plane at four levels (T7, T12, L2, and L5) compared to the PSIS level. The system could report the rotation angles and illustrate the graph to compare the change of the trunk rotation in the transverse plane. The spinal deformity was corrected from the assessment to casting using 3-dimensional force correction from the scoliosis casting apparatus. As a result, it could help clarify the unclear 3-dimensional deformity of AIS patients, especially on the transverse plane (assessment), and how the trunk changes when applying the forces to correct the deformity (casting).

We concluded that the developed postural measurement system could quantify the angle of trunk rotation in the horizontal plane. This result gave a valuable insight into the rotation of the spine in the transverse plane that was not previously reported.

#### **4. “To examine if the developed postural measurement system could quantify the trunk asymmetry in the coronal plane”.**

The spinal parameter for this objective was the trunk asymmetry in the coronal plane. Suzuki and colleagues introduced a method to calculate the asymmetry of the trunk in the coronal plane, called the “POsterior Trunk Symmetry Index (POTSI)” (Suzuki *et al.*, 1999, Inami *et al.*, 1999). To clarify, the POTSI consisted of six indices. If the summation of six indices was equal to or less than 10%, it indicated normal or trunk symmetry. In contrast, if the POTSI index was greater than 10%, it indicated the spinal pathology or asymmetry of the trunk in the coronal plane.

Regarding the result of this study, the POTSI was improved from assessment (mean = 33.54%, SD = 16.23) to casting (mean = 22.80%, SD = 12.41), with the highest improvement percentage at the left and right distance at the waist and the trunk balance. There were a slight improvement percentage at waist level, axilla level, shoulder level, and the left and right distance at the axilla.

Kotwicki and colleagues compared the POTSI with 50 IS patients who had a coronal Cobb angle between 25° to 40° (24 patients in wearing the Cheneau brace and 26 in the non-brace) (Kotwicki *et al.*, 2007). The result showed that the POTSI of the bracing group was 26.1% (SD = 18.3), while the non-bracing group was 24.9% (SD =



11.8). The results in both groups were slightly lower than ours in the assessment. Chonggov and colleagues studied the improvement of POTSI before and after performing the Schroth physiotherapeutic exercises with 128 AIS patients (Chonggov *et al.*, 2017). The result showed that the POTSI before-exercise group was 31.27% (SD = 17.1), while the after-exercise group was 23.08% (SD = 14.38). There was a statistically significant difference between both groups before and after the Schroth physiotherapeutic exercises. The results in both groups were quite similar to ours in the assessment and casting. Lastly, Yagci and Yakut studied the improvement of POTSI before and after performing two types of exercises with 30 AIS patients (Core stabilization exercise and Scientific exercises approach to scoliosis) (Yagci and Yakut, 2019). The result showed a statistically significant difference between before and after exercises in both groups and an equivalence of the treatment after exercises in both types. In the core stabilization exercise group, the POTSI before treatment was 32.9% (SD = 11.8), while the POTSI after treatment was 23.2% (SD = 7.7). In the scientific exercises approach to the scoliosis group, the POTSI before treatment was 33.7% (SD = 13.5), while the POTSI after treatment was 20.7% (SD = 10.1). The results in both groups were quite similar to our study in the assessment (before treatment group) and casting (after treatment group).

The developed postural measurement system could quantify the trunk asymmetry in the coronal plane to evaluate the trunk change. The POTSI was calculated at six indices to see the percentage change at each location and the summary of all percentages. We could compare the percentage change before and when applying the forces to correct the deformity. The system could report the trunk asymmetry values and illustrate the graph to compare the change in the coronal plane. The spinal deformity was corrected from the assessment to casting using 3-dimensional force correction from the scoliosis casting apparatus. As a result, it could help clarify how the trunk of scoliosis patients changes in the coronal plane when applying the forces to correct the deformity.

We concluded that the developed postural measurement system could quantify the trunk asymmetry in the coronal plane.

**5. “To examine if the scoliosis casting apparatus could be adjusted to correct the spinal deformity in 3 dimensions”.**

The parameters used to examine this objective were the spinal parameters from objectives 1 to 4, including the CSA, SSA, 3DSA, trunk rotation in the horizontal plane, and trunk asymmetry in the coronal plane, to evaluate the change of the spinal deformity in three dimensions during assessment, optimal correction, and casting processes. As a result of the previous objectives, the clinicians could adjust the scoliosis casting apparatus to apply the 3-dimensional force correction to correct the 3-dimensional spinal deformity of scoliosis patients during the optimal correction and casting. The magnitude of the coronal curve was reduced, and the spinal column shifted to the centre and was straighter. The sagittal curve could be maintained within 8° to 9° deviation. The 3-dimensional spinal angle was improved and mostly in the neutral alignment zone. The trunk asymmetry in the coronal plane and horizontal trunk rotation were improved.

We concluded that the scoliosis casting apparatus could be adjusted to correct the spinal deformity in 3 dimensions and can be used to treat scoliosis patients in clinical practice.

**6. “To examine if the force measuring system could measure the force magnitude and the postural measurement system could illustrate the force directions when the forces were applied to correct the spinal deformity in 3 dimensions”.**

The parameters used to examine this objective were the force magnitude to correct the deformity applied by the scoliosis casting apparatus and the direction of forces in three dimensions obtained by the low-cost postural measurement system. The developed program could measure the force magnitude in seven areas. However, the number of required forces to correct the deformity depended on the individual patient's condition. The force magnitudes were varied and depended on many factors, such as the curve type, trunk balance, curve flexibility, age, gender, BMI, and other clinical considerations.

Regarding the force magnitude from this study, the force at each area was approximately 30 N (ranging from 20.82 N at the gluteus medius area to 33.97 N at the corrective curve area). The total force each patient had to tolerate during the optimal correction and casting was approximately 150 N, ranging from 101.90 N to 181.07 N. Regarding the force direction, the system could illustrate the force direction in three dimensions following the forces that the researcher aimed to apply during optimal correction and casting to correct the deformity in three dimensions.

No previous research studied the force magnitudes and force directions applied while casting scoliosis patients. The previous research mostly used sensors to measure the pressure or force inside the spinal orthoses. Pham and colleagues used the TekScan1 system to measure the pressure inside the Cheneau brace in 32 AIS patients (Pham *et al.*, 2008). The study measured the pressure in three areas, including the left sub-axillary support, the right thoracic support, and the left lumbar support. The pressure ranged from 7.06 kPa to 9.5 kPa before and after strap adjustment and in various spinal activities. van Den Hout and van Den Munckhof used the PEDAR measuring device to measure the forces inside the Boston brace in 16 AIS patients (Hout and Munckhof, 2002). The study measured the force and the area of force in three areas, including the lumbar pad, thoracic pad, and axillary extension. The force ranged from 43 N with 29 cm<sup>2</sup> to 329 N with 60 cm<sup>2</sup> in various spinal positions. The study mentioned that the changes in body posture resulted in statistically significant alterations in the exerted forces. Loukos and colleagues used the F-Socket 9801 pressure sensor and the MatScan Research BETA STAM 6.30 software to measure the force inside the dynamic derotation brace at the corrective force area in 44 AIS patients (Loukos *et al.*, 2011). The force ranged from 70 N to 176 N during minimum and maximum strap tension adjustment and in various spinal activities.

Comparing the results of our study and others, the force from our study were lower than that of other studies. However, the forces from other studies were the force measured inside the orthoses. The orthoses were made of plastic shells and did not allow the patients to adjust their posture freely. As a result, the force and pressure were high inside the orthosis. In contrast, the force from our study was the force measured on the manipulators used to correct the deformity in five to six areas. The patients could adjust their posture away from the forces freely. As a result, the patients

performed active or self-correction, and the force from casting could be lower than the force inside the orthosis.

In summary, the spinal deformity was corrected from the assessment to the optimal correction and casting using 3-dimensional force correction from the scoliosis casting apparatus. The forces were then measured by the developed force-measuring program, and the developed postural measurement system illustrated the direction of forces in three dimensions. As a result, it could help clarify how the 3-dimensional forces correct the 3-dimensional spinal deformity in AIS patients. We concluded that the force-measuring system could measure the force magnitude and the postural measurement system could illustrate the force directions when the forces were applied to correct the spinal deformity in 3 dimensions.

## **7. Message to the scoliosis treatment**

Scoliosis is a pathology that is commonly seen in hospital, and adolescent idiopathic scoliosis (AIS) is the largest group of scoliotic patients seeking treatment. The patients must obtain an X-ray to clarify the structure of the spinal deformity and plan for the treatment. The X-ray is mainly obtained in the coronal plane to evaluate the lateral deviation of the spine. Some cases will be obtained in the sagittal plane to evaluate the angle of thoracic kyphosis and lumbar lordosis. After radiographic evaluation, a doctor considers the best treatment option and discuss it with patients and their relatives. Spinal orthoses made of plastic are commonly prescribed to the patients and the objective of wearing the orthosis is to stop or delay the curve progression. The orthosis can apply 3-dimensional biomechanical force correction and a 3-point-force system to correct the 3-dimensional spinal deformity. The X-ray is obtained again to evaluate the deformity change while wearing the orthosis. The deformity must be reduced while wearing the orthosis. The patients regularly visit the hospitals for follow-ups and obtain in-brace X-rays every six months to one year to evaluate the treatment outcome. The new treatment may be necessary if the curve rapidly progresses or the current treatment is not effective.

Radiographs are a standard tool for illustrating and measuring spinal deformity and are commonly taken in coronal and sagittal planes. However, this method is insufficient to describe scoliosis deformity because scoliosis is a 3-dimensional spinal deformity. There is still a lack of outcome measurement in transverse and 3-dimensional deformity. The radiographs have been used for a long time and are widely used in hospitals. The clinicians must use this method to evaluate the treatment outcome because they do not have a better option. Other technologies have been studied for the possibility of quantifying scoliosis deformity in 3 dimensions. However, no one currently uses it in clinical practice, and the developing systems are still in the research phase. The 3-dimensional deformity of scoliosis treated by 3-dimensional biomechanical force correction has been introduced for a long time. However, displaying and measuring the deformity in 3 dimensions is still an issue, especially in clinical practice.

One of the most important factors in providing an excellent orthotic treatment is how much the orthosis can correct the spinal deformity. The orthosis has to reduce the magnitude of deformity and return the spinal column to the neutral alignment as much as possible. The coronal curve should be straighter, the sagittal curve should be similar to its original curve or should not increase the kyphosis and lordosis curves, and the transverse plane should reverse the horizontal rotation of the spine and trunk. To achieve this alignment, casting is an important process to capture the shape with the force correction applied. As the scoliosis deformity has a complex deformity, two to three clinicians are required to perform a cast by using their hands to apply the 3-dimensional force correction to correct the deformity. It is not common to use casting frames in scoliosis casting because it is expensive, complicated to use, and does not have strong evidence to support its effectiveness. The magnitude of force to correct the deformity is also important to know. If insufficient forces are applied to the spine, the deformity may not highly improve. In contrast, if the exaggerated forces are applied, the patient can not tolerate the force, pain or abrasion may occur, patients may discontinue the use of the orthosis, and the curve will progress. Therefore, measuring the spinal deformity in 3 dimensions and applying the 3-dimensional biomechanical force correction is important for treating scoliosis patients, and it is still challenging for clinicians.

This thesis developed three systems to solve these issues to improve the unclear of 3-dimensional scoliosis deformity and improve the quality of orthotic treatment. We developed a low-cost postural measurement system to quantify the scoliosis deformity in 3 dimensions. The experiments in this thesis proved that the developed postural measurement system had a high concurrent validity compared to planar radiographic evaluation. The system had a suitable accuracy in quantifying spinal parameters of scoliosis patients in clinical practice. The system could measure the spinal angles and other parameters and illustrated the spine change in 3 dimensions. The system could also illustrate the direction of the forces when applied to correct 3-dimensional deformity. We developed a scoliosis casting apparatus that could be manually adjusted to apply the 3-dimensional biomechanical force to correct the deformity. The system could assist the clinicians during casting and produce a good negative cast for making the spinal orthosis in the following process. To evaluate how much force we used to correct the deformity, we developed a force-measuring system to measure the forces during casting. Combining all developed systems, we could quantify the spinal deformity in 3 dimensions. We could measure the force magnitude used to correct deformity, illustrate some spinal parameters commonly evaluated in clinics, report force magnitude, and illustrate force direction to evaluate the spine change in 3 dimensions during the treatment.

The current orthotic outcome measurement in clinical practice is mainly based on the change in the coronal spinal parameters. However, many spinal evaluations are missing due to the limitations of technology. Therefore, the new system that can quantify spinal parameters in 3 dimensions should be used in clinical practice to evaluate the real change or improvement of the deformity in 3 dimensions. The clinical outcome measurement in scoliosis should cooperate with the current concept of 3-dimensional force correction and the new scientific method and evidence-based practice to improve the quality of scoliosis treatment and quality of life of scoliosis patients.

It is time to evaluate the scoliosis deformity change in 3 dimensions. The radiographs are still needed at the beginning of treatment to evaluate the change inside the body, and using the new system, such as our developed system, is necessary during the whole treatment process to quantify the deformity in 3 dimensions. Applying 3-

dimensional force correction during casting is not a new concept for clinicians in orthotic treatment. However, the scoliosis casting apparatus we developed can assist in this process to achieve a good negative cast. Compared with the traditional casting method, clinicians with scoliosis casting apparatus will have more time to evaluate the correction force applied during casting. The developed postural measurement system can help to evaluate the deformity change in 3 dimensions. The forces from manipulators can be adjusted by slightly increasing or decreasing until reaching the maximum correction. Furthermore, using the apparatus can reduce the number of clinicians during casting, and clinicians will have more time to provide treatment for other patients. The force-measuring system can help to measure the force magnitude during casting. The manipulators attached to the apparatus are released for some reason. However, the force magnitude can help return the manipulators to the same correction after re-adjustment.

## **8. Reflection on the developed system**

Evaluating the 3-dimensional spinal deformity in scoliosis is challenging because the deformity occurs inside the body, and the standard method can measure the change inside the body but only describes it in 2 dimensions. It is different from outcome measurement in other pathologies, such as cerebral vascular accident (CVA or stroke) and cerebral palsy (CP), that we can measure the outcome while they are walking or performing activities and use a motion capture system to measure and analyse the movement change. The motion capture system is commonly used to analyse the movement in 3 dimensions, but there needs to be more evidence in scoliosis outcome measurement. The outcome measurement in scoliosis is the spinal and postural change, which can be measured in static movement. Therefore, a motion capture system or a postural measurement system with enough accuracy to quantify the scoliosis parameters should be developed.

No research clearly explained a method to calculate a marker position in motion capture system in 3 dimensions, and the researches mostly focused on the application of the system. During the developing phase, we developed several programs and performed several experiments to evaluate the system and find the possibility of a

developing system for scoliosis applications. It spent quite a long time for the development before we got the accurate system. The factor that was highly affected during development was the COVID-19 pandemic because there was no appropriate space and tools for installing and testing the system. We finally got a system that accurately calculates marker positions for scoliosis applications. The system could calculate marker positions one by one and could not give the output in real time. The system is suitable for testing the accuracy. However, the results should be reported in real-time in the following development to assist the clinicians during clinical decision-making to evaluate the spine change. There was no significant issue during the development of the scoliosis casting apparatus and the force-measuring system. The scoliosis casting apparatus could be adjusted to apply forces in 3 dimensions, and the force-measuring system could measure the forces to correct the deformity.

During clinical experiments, the high challenge was the recruitment of participants in the study. Because participants were children, they could not make any decisions, and we had to deal with their relatives. Many relatives did not allow their children to participate in the project because they were afraid of the side effects of radiation from X-rays and COVID-19 infection, and they did not want their children to be absent from school. To quantify spinal parameters during casting was a second challenge. We tried several techniques to quantify spinal parameters when plaster bandages were wrapped on the trunk. It was hard to measure the parameters with participants during casting, and we finally used a positive cast to measure parameters instead. The structure to lock and unlock the manipulators attached to the casting apparatus was not easy to adjust. It was challenging during the casting process because I had to apply plaster bandages on the whole trunk, re-adjust the manipulators, and re-measure the force magnitude within 10 to 15 minutes.

There are several things I have learned and achieved from this project. Time management is the most important because the project's progression does not rely on anyone except me. Technology is rapidly progressing, cheap and allows everyone to develop their own application. Raspberry Pi is one of the best examples. Raspberry Pi can do a similar performance to the computer. We can connect it with other hardware, write the computer code, create the program or application, etc. The high-cost system available for use is good, but it is not available for everyone because of pricing, and



the system may be overkill for your application. I have learned how to develop the available low-cost technology to implement in clinical practice. Even though the accuracy of the developing system is lower than that of the high-cost one, it is better than nothing in clinical practice. Using subjective opinion to evaluate the change, especially in medical fields, should be minimised. Scientific outcome measurement and evidence-based practice should be more implemented in clinical practice. However, the developing system should have enough accuracy to work in your applications.

The thesis proved that the developed postural measurement system had a high concurrent validity compared to planar radiographic evaluation. The system had a suitable accuracy in quantifying spinal parameters of scoliosis patients in clinical practice. The scoliosis casting apparatus could manually adjust to apply the 3-dimensional biomechanical force to correct the deformity. The force measuring system could measure the forces during casting. The second version of the system should be done before applying it in clinical practice. The motion system should report the result in real time, the force-measuring program should be combined with the same program as the postural measurement system, and the manipulators of the casting apparatus should be easily locked and unlocked during adjustment.

### **11.3. Limitations and recommendations**

The limitations and recommendations of this study will be discussed in four aspects, including a low-cost postural measurement system, a scoliosis casting apparatus, a force-measuring system, and clinical experiment.

- **A low-cost postural measurement system**

Regarding the low-cost postural measurement system, this system had high accuracy and was suitable for quantifying spinal parameters in scoliosis patients in 3 dimensions. The current system is based on the evaluation of scoliosis deformities in the static posture in a standing position during assessment, optimal correction, and

casting. However, scoliosis deformities can also change the pattern while walking or performing trunk movements, and the current system could not assess that change. Therefore, the following research should develop a system that can evaluate the spinal parameters in a static position to evaluate the posture of scoliosis patients and dynamic movement to evaluate how the scoliosis curve changes in 3 dimensions during gait or performing the trunk motions, which can help clarify how scoliosis deformities change during dynamic movement.

The current program could calculate a marker position from a white marker, and the background or unwanted area had to be in black. As a result, the black curtain was hung around the camera frame to turn the background black. The current program could also not label the marker's name and required the user to calculate the position marker-by-marker. The program also allowed the user to crop or select the marker area on each image and exclude the object outside the selected area before calculating the marker's position. Furthermore, the spinal parameters were calculated manually in the pre-excel spreadsheet, and the formulas were embedded in the file. As a result, it required a lengthy time during the post-data processing. Therefore, the next development should find a method to label the marker's name or modify the program to calculate a marker position from various marker colours and specify one colour for one anatomical landmark. As a result, we may no longer need to turn the background black and shorten the time spent calculating marker positions and spinal parameters.

The current program could not report the marker positions and spinal parameters in real time because it was mainly used to calculate the marker positions, validate the system's accuracy, and demonstrate its feasibility for clinical practice. As a result, the data collection during optimal correction and casting still came from the clinician's opinion, and the system could not help the clinician make the final decision. Therefore, the next development should find a method to calculate marker positions and spinal parameters and then report and illustrate the results in real time. This method may help the clinicians to evaluate the spinal parameters in real time and allow the clinicians to adjust the forces from the apparatus by decreasing or increasing the forces until getting the actual optimal position of each patient in the orthotic treatment.

The results of spinal parameters in this experiment came from one pair of Raspberry Pi cameras (cameras 1&2), and other pairs (cameras 1&2, 3&4, 5&6,7&8)

were used to calculate and illustrate force directions. The RMSE of one pair was lower than 5 mm, and the RMSE of multiple pairs was higher than 5 mm due to the issue of the rotation matrix and translation vector of all camera pairs. As a result, we decided to use one pair to calculate the spinal parameters and other pairs for illustration. Therefore, the next development should find a method to calculate the rotation matrix and translation vector of all camera pairs and use them to calculate a marker position by averaging the value from multiple pairs of cameras.

There were six programs used to calculate marker positions in the developed postural measurement system, including the “getpicameraimages” and “calibpicamera” for camera calibration, the “getpistereoisimages” and “calibpistereocamera” for stereo camera calibration, and “getimagepairs” and “reconstructfromimages” for marker calculation. As a result, it was not easy for others to use the programs. Therefore, the next development should combine all programs in one program and create a graphical user interface (GUI) to run the program and display the results in real time, assisting the user during the system's operation and allowing the user to select and run the program by clicking the buttons on the GUI.

Regarding the camera frame, the current frame was 2,000 mm in all dimensions. It allowed the clinician to perform all data corrections with the patients. However, the clinician had to be careful and pay more attention during data collection not to touch the calibrated cameras. Otherwise, the new stereo camera calibration was required. Therefore, the next development should be to increase the size of the camera frame or attach the camera to the permanent wall, making it difficult to touch the camera so the clinicians can work more freely during the orthotic treatment.

Lastly, the current system used eight Raspberry Pi and eight associated cameras to capture the images for marker calculation. Each camera was positioned on the camera frame and could be adjusted freely using the connector, bolt and nut. Each camera sent its picture to the central computer, where it was analysed by the operator using the software. The software was developed to allow camera locations at any point in 3D space. However, the experiments conducted found that the two cameras (stereo cameras) should be parallel to each other with a fixed separation to achieve the low errors required.

Therefore, the next development should use a specific mount to accurately position the cameras in parallel to one another with a suitable separation. It could be achieved by 3D printing. Moreover, the new version of the Raspberry Pi, the Raspberry Pi 5 released in September 2023 in the UK allows users to plug in two cameras into one Raspberry Pi. Therefore, the next development should use one Raspberry Pi 5 with two attached cameras, fixed rigidly and permanently parallel to one another with suitable separation between cameras for the application (250 mm) to calculate the marker positions automatically.

The Raspberry Pi 5 now has sufficient computing power to run the OpenCV routines used in this developed prototype system directly on the pi, and OpenCV is now available for the Pi. Hence, each pi and pair of cameras could be individually calibrated as a standalone system, and rather than send the captured pictures to a central computer for processing as was done in the current prototype. The Pi could analyse the pictures on board and only send the locations of the markers to the central computer, a process which can be done in real time.

The other reason for the time-consuming, non-real-time nature of the developed prototype was this issue of conflicting markers where a marker appeared at the same vertical level on the image as another marker, and then marker pairs between images could be confused. Hence, the operator was required to solve the conflict. However, the use of fixed parallel cameras would mean that marker order in both images would be consistent, i.e. if a marker is to the left of another marker in one image, it is also on the left of the other marker in the other image and hence automatic marker recognition and processing on board the Raspberry Pi 5 with two cameras should be possible. These developments imply that real-time processing of the pairs of images on board the Pi in an automatic way should be possible and would simplify the system to four Raspberry Pi 5 processors each with two parallel cameras and each sending 3D marker data in real time to the visualisation computer. Such a system would mean real-time feedback to the orthotist during orthotic treatment was possible. It would include feedback on the 3D biomechanical force system applied and its effect on spinal posture.

- **A scoliosis casting apparatus**

Regarding the scoliosis casting apparatus, the manipulator was fixed using the connector, bolt and nut, and the user had to use the Allen key to lock and unlock the connector during manipulator adjustment. Five to six manipulators needed to be adjusted during the optimal correction and casting process, and the current connector was not easy to adjust and needed more time to do it. The most challenging task during the adjustment was the casting process because the plaster bandages were commonly set within 15 minutes. The user needed to complete all processes within that period, including wrapping the plaster bandage, adjusting the manipulators, and re-measure the force magnitude if needed. Therefore, the next development should re-design the connector or add any structure to lock and unlock the connector of manipulators in a short time.

After the optimal correction process, the user unlocked three manipulators to allow the patients to move outside the apparatus and prepare for the casting process. The user had to create a line mark on the manipulators before unlocking, and the user could refer back to this position by checking the line mark during the casting process. Therefore, the next development should add the scale along the manipulators, and the user can use this scale when they want to adjust the manipulators to the same position during the casting process.

Lastly, the user had to move the scoliosis casting apparatus into the camera frame during the optimal correction and casting, and the current apparatus was quite heavy and difficult to move from one place to another. Therefore, the next development should add wheels to assist the clinicians when moving the scoliosis casting apparatus from one place to another, and the wheels should be able to lock and unlock.

- **A force-measuring system**

Regarding the force-measuring program, the current program was written in Python computer language and run on Visual Studio 2017 in Windows 10. It was separated from the low-cost postural measurement program. Both programs could not open at the same time, and the user had to close one program before running another

program. Therefore, the next development should combine all programs in one program and re-write the code from Python to C++ computer language, similar to the low-cost postural measurement program. All programs should also run and show on the same graphical user interface (GUI) to assist the clinicians during the system's operation.

Lastly, we found a small issue during the optimal correction and casting processes. The electrical wire of the load cells sometimes obstructed the user during these processes. The user needed to pay more attention not to pull or damage the wire. Therefore, the next development tries using the wireless load cell or force sensors to measure the force during the optimal correction and casting. It may be beneficial if the sensor sends the result wirelessly to the Raspberry Pi or computer, and the system illustrates the result on the GUI in real-time.

- **Clinical experiment**

Regarding the clinical experiment, ten cases were recruited. They could provide enough information to determine the success or otherwise of the system and provide sufficient feedback to evaluate its usefulness and any future modifications. However, some inferential statistics in this clinical experiment showed no statistical significance, which might come from insufficient data to make the conclusion. Therefore, the next research should calculate a sample size that can be enough to draw conclusions from the research objectives. Moreover, the next research should study subgroup analysis, such as the different results between the single and double curves in AIS patients or between a smaller and a bigger coronal Cobb angle.

This clinical experiment quantified the spinal parameters of the casting process using the positive model, not directly from the participants. As a result, an error could occur from this method. This experiment could not directly calculate marker positions during casting with the participants because of the colour of the plaster bandages, how to keep the markers on the plaster bandages, and time consumption. The plaster bandages were white, the same colour as the markers. If we want to calculate the marker positions, the negative cast must be turned to the black colour using a painting technique or something similar, and it will take a longer time with the participants

during this process. Therefore, the next development should find a method to quantify the spinal parameters during the casting process with direct measurements from the patients. One possible solution is to modify the low-cost postural measurement program to detect and calculate marker positions in other colours of markers, and colour from plaster bandages does not affect the marker calculation.

The participants recruited in this thesis were patients who came to get the treatment at SSPO. Many participants regularly followed up with a doctor every six months or one year and obtained radiographs before attending the study. As a result, we could not control the participant's position while taking the radiographs. Moreover, results from radiographs and the postural measurement system were collected at different times and on different days, and three months was the maximum difference. Therefore, the participant's position during the radiographs in assessment and the postural measurement system in assessment were not exactly the same and would make the error happen during the clinical experiment. Therefore, the next research should control the participant's position during the assessment process. The radiographs should be obtained on the same day as the postural measurement system. Moreover, other systems, such as a high-cost motion capture system and ultrasound, should be used to collect the data simultaneously with the low-cost postural measurement system, and the results should then be compared and analysed.

Lastly, the current thesis mainly focused on how the scoliosis casting apparatus applied the forces to correct the deformity in 3 dimensions. However, the apparatus's effectiveness and satisfaction still need to be studied. Therefore, the next research should focus on the study of the effectiveness of scoliosis casting apparatus in terms of function, appearance, satisfaction and so on. The second version of the scoliosis casting apparatus may be needed after getting the results in all aspects from the users.

#### **11.4. Conclusion**

Idiopathic scoliosis is a common pathology of the spine that causes deformities in three dimensions, including lateral deviation in the coronal plane, increasing or decreasing kyphotic or lordotic curvature in the sagittal plane, and horizontal rotation

of vertebrae in the transverse plane. Scoliosis can be classified according to age range, and this study focused on adolescent idiopathic scoliosis (AIS) occurring between the ages of 10 and 18 years because it was a big group of patients to get the orthotic treatments. In the general treatment of AIS patients, there are two main types of treatment, which are conservative treatment and surgical treatment, and the choice between them is based on the severity of the deformity.

Radiographs are the standard method to measure the outcome of scoliosis because they are relatively low-cost and available in hospitals. Although radiographs have been a standard method, most research has concentrated on spinal deformity changes in the coronal and sagittal planes. There still needs to be more understanding of spinal deformity in three dimensions. Casting is an important step in capturing the contour of the scoliosis trunk under the forces produced by the clinician's hands. A good negative cast involves clinical skill and collaboration. The casting frame is uncommon in clinical practice due to its high cost and difficulty in usage. There is still a lack of data on the efficacy and advantages of employing a casting frame to help practitioners get an effective negative cast and orthotic treatment. In addition, there is currently a lack of information about the magnitude of forces used to correct the scoliosis deformity, as well as the locations and directions of forces in three dimensions throughout the casting process.

Firstly, the literature review was conducted and discussed in **Chapter 3** to clarify the current evidence related to scoliosis. This chapter aimed to describe the definition and characteristics of scoliosis, scoliosis treatment, standard and alternative scoliosis outcome measurement, high-cost and low-cost postural measurement systems in outcome measurement and its accuracy, and casting frames to assist clinicians during casting. As described above, radiographs have become a standard method to quantify the spinal parameters in scoliosis, mainly in coronal and sagittal deformity. Some studies use MRI and CT scans to quantify spinal parameters in scoliosis. However, these technologies are not commonly used in AIS patients because they take a long time to complete scanning, only scan in the lying position, and are relatively expensive. As a result, doctors primarily use these systems to examine the spine in severe cases, not typical AIS patients. Ultrasound has become a popular alternative method for measuring spinal parameters in scoliosis. However, most current studies use



ultrasound to quantify spinal deformity in the coronal plane, lacking research in the sagittal and transverse planes and spinal column in 3 dimensions. Surface topography and motion analysis are two new technologies to quantify the spinal parameters in scoliosis. However, stronger evidence is still needed to support the effectiveness of these systems. Low-cost motion capture systems become increasingly popular for use in motion analysis. The Raspberry Pi and camera is a low-cost and small computer widely used in many applications, such as robotics, security, and smart agriculture. Surprisingly, no one has used the Raspberry Pi and camera to analyse human motion. Some scoliosis casting frames have been introduced but have yet to become widely used in clinics because they are relatively expensive, lack research evidence to support their efficacy, are relatively difficult to use and only apply corrective force in one or two dimensions. Furthermore, there is a lack of evidence regarding the magnitude of force to correct the deformity during casting, and no research illustrates the direction of forces applied to correct the spine in 3 dimensions.

This thesis developed three systems to solve research gaps and improve orthotic treatment. The developed systems included a low-cost postural measurement system, a scoliosis casting apparatus, and a force measuring system. The primary goals of this thesis were to create a system that could quantify the spinal deformity of AIS patients in three dimensions, a system that could apply forces to correct the spinal deformity in three dimensions, and a system that could measure the magnitude of forces and illustrate those force directions in three dimensions. The second goal of this thesis was to demonstrate the feasibility of using the developed system to treat AIS patients in clinical practice.

The development was begun with the scoliosis casting apparatus. The development of the scoliosis casting apparatus was discussed in **Chapter 4**. This development aimed to assist clinicians during the optimal correction and casting to obtain a good negative cast for orthotic treatment in the following step. The apparatus could apply the 3-dimensional force correction to correct the 3-dimensional spinal deformity in AIS patients. The apparatus was composed of 8 parts, including the main structural element, sitting bar, arm-rest, abdominal manipulator, gluteus medius manipulators (left and right sides), axillary manipulators (left and right sides), left thoracic or lumbar corrective manipulator, and right thoracic or lumbar corrective manipulator. However,

the number of forces required during casting varied and depended on the clinical consideration for individual cases.

The second development was a force-measuring system, which was discussed in **Chapter 5**. This development aimed to measure the force magnitudes from the scoliosis casting apparatus that applies the 3-dimensional force correction to correct the 3-dimensional spinal deformity of scoliosis patients during the optimal correction and casting. The system was composed of 2 parts, including the hardware (6 load cells, 2 PhidgetBridges, computer, mouse, keyboard, and monitor) and software (load cell calibration program and force-measuring program). All load cells were calibrated by measuring the voltage change and known mass. The calibration was performed during increasing and decreasing mass of 40 Kg, and a simple linear regression was calculated to get the simple linear regression formula for each load cell. These formulas were ready to measure the force magnitude in the following program. The load cells were attached to the end of the manipulators and removable pads to measure the forces during optimal correction and casting. All programs were written in Python computer language and run on Visual Studio 2017 in Windows 10. The user adjusted the scoliosis casting apparatus to correct deformity and then ran the program to collect the force magnitudes. GUI showed on the monitor while running the program, the user clicked the buttons on the GUI to run the command and display the force's result.

The last development was a low-cost postural measurement system discussed in **Chapters 6 and 7**. This development aimed to calculate marker positions and the spinal parameters of AIS patients in 3 dimensions. The system was composed of 2 parts, including the hardware (Raspberry Pis and cameras, computer, mouse, keyboard, monitor, camera frame, whiteboard, chessboard pattern with 9 x 6, and Phidget switch) and software (camera calibration, stereo camera calibration, and marker calculation programs). The development started using 2 and 3 Raspberry Pis with the associated cameras and a simple calculation technique (the right triangle similarity theorem and trigonometry) to calculate a marker position in 3-dimensional space, as described in Chapter 6. Several experiments were done to evaluate the system's accuracy. However, the RMSE of the experiments was still unsatisfied, and the system had to be improved to reduce the error. We tried another concept to calculate marker positions using the stereo camera concept and OpenCV library. This

technique was previously used in image processing applications and robotics but not in human movement analysis. We used the OpenCV library and further developed the program to match our application. All programs were written in C++ computer language and run on Visual Studio 2019 in Windows 10. There were 4 processes involved in calculating marker positions for this development. Camera calibration is a process of determining an individual camera's intrinsic and extrinsic parameters. Stereo camera calibration and image rectification are the processes to rectify and row-align the image plane between the left and right cameras. Correspondence is the process of finding the same features or points between the left and right images. Lastly, Reprojection is a process for calculating the depth of an object or markers in 3-dimensional space using the disparity map from the previous process and a triangulation calculation between marker and cameras. Several experiments were done to evaluate the system's accuracy. The RMSE was lower than 5 mm when using one stereo camera pair, following our goal. However, the RMSE was higher than 5 mm when using multiple stereo camera pairs to calculate marker positions. The primary problem of the current development was the relationship between multiple stereo camera pairs. Future research should find a method to calculate the rotation matrix and translation vector of each stereo camera pair and link them to respect the new global reference. As a result, we decided to use one stereo camera pair to calculate marker positions and spinal parameters, and other camera pairs were used only to illustrate the results.

After the successful development of three systems, the next step was to demonstrate and evaluate the system to treat AIS patients. **Chapter 8** discussed the clinical research methodology, including participant inclusion and exclusion criteria, sample size selection, study design, research protocol and procedure, data collection, and data analysis. This chapter also detailed a method to calculate spinal parameters using the results from each marker position. The project had already been approved by the ethical committees from the University of Strathclyde and Mahidol University. The developed system was installed in the SSPO research room, and the data collection was conducted there. Participants were the outpatients who usually came to get treatment at SSPO. Participants had to come to SSPO for two appointments for recruitment and data collection from radiographs and the developed system in

assessment, optimal correction, and casting. Several parameters were collected to evaluate the developed system and highlight the clinical findings. The parameters consisted of coronal spinal angle, sagittal spinal angle, 3-dimensional spinal angle, trunk balance, trunk asymmetry in the coronal plane, horizontal trunk rotation, force magnitudes, and force locations and directions. Descriptive statistics was used to describe the participant characteristics and spinal parameter results (mean and standard deviation, minimum and maximum, and percentage). Inferential statistics was then used to analyse the relationship between radiographs and the developed system in assessment (Pearson correlation coefficient) and to analyse the statistically significant difference (Paired sample t-test) or equivalence (Paired t-test for equivalence using the Two One-Sided Test) of the spinal parameter results.

Ten AIS patients were recruited for the study. The study design was a pilot study, and ten cases could provide enough information to determine the success or otherwise of the system and provide sufficient feedback to evaluate its usefulness and any future modifications. **Chapter 9** described the results of individual cases in detail. This chapter aimed to demonstrate the feasibility of using the developed system to treat AIS patients and describes how the spinal changes before and when applying the 3-dimensional biomechanical force correction to treat 3-dimensional deformity.

**Chapter 10** described the data analysis from ten participants using descriptive and inferential statistics to analyse the relationship or the difference or similarity compared between radiographs and the developed system in assessment, optimal correction, and casting. Some statistical analyses were performed using the SPSS program, and some used the NCSS program.

**Chapter 11** then discussed the clinical experiment results to validate the developed system and address the clinical findings in scoliosis treatment. Regarding the relationship between the results from radiographs in assessment and the postural measurement system in assessment, there was a moderate to very high positive correlation in the coronal spinal angles, a very high positive correlation in the trunk balance, and a low to very high positive correlation in the sagittal spinal angles. Regarding the comparison of the postural measurement in assessment VS optimal correction and the assessment VS casting in the coronal spinal angles, there was a high reduction percentage from assessment to optimal correction and casting, 51.21% (SD

= 13.42) in optimal correction and 59.53% (SD = 13.04) in casting. There was a statistically significant difference in the reduction of apical translation. However, there was no statistically significant difference in coronal spinal angle, or there was not enough data to indicate the difference in angle change. For the equivalence of the postural measurement in optimal correction VS casting in the coronal spinal angles, there was a statistically significant equivalence when the margin of equivalence was 5°. In sagittal spinal angles, we aimed not to significantly change the spinal angle at this plane when applying the forces to correct the deformity. The result showed no statistically significant equivalence when the margin of equivalence was 5°. However, there was a statistically significant equivalence when the margin of equivalence changed to 9° for assessment VS optimal correction and 8° for assessment VS casting and optimal correction VS casting, respectively, which was lower than the maximum SSA change of 10°. For the 3-dimensional spinal angle, there was no statistically significant difference for postural measurement in assessment VS optimal correction and assessment VS casting. There was no statistically significant equivalence when the margin of equivalence was 5° for optimal correction VS casting. However, there was a statistically significant equivalence when the margin of equivalence changed to 8°, which was lower than the maximum 3DSA change of 10°. For the trunk balance, there was a high reduction from assessment (-8.19 mm, SD = 11.58) to optimal correction (-1.25 mm, SD = 4.56) and casting (-0.71 mm, SD = 3.32). However, there was no statistically significant difference in both comparisons, or it did not have enough data to indicate the difference. For optimal correction VS casting, there was a statistically significant equivalence when the margin of equivalence was 3 mm. For the POTSI, the trunk asymmetry improved from 33.54% (SD = 16.23) in postural measurement in assessment to 22.80% (SD = 12.41) in casting. The maximum asymmetry percentage difference (improvement) was at the left and right distance at the waist (3.54%) and trunk balance (3.13%). Waist level (1.28%), the left and right distance at the axilla (0.94%), axilla level (0.94%), and shoulder level (0.91%) were slightly improved. For the horizontal trunk rotation, the mean of the total reduction of the horizontal trunk rotation angle was 14.83° (SD = 12.44). For the force magnitude, the force to correct the deformity at each area was approximately 30 N, and the total force each patient had to tolerate during optimal correction and casting was

approximately 150 N. The highest force was present at the left and right corrective forces, followed by the abdominal force. A slightly lower force was present at the left and right axilla forces. The smallest forces were present at the left and right gluteus medius forces.

All experiments showed that the low-cost postural measurement system could calculate marker positions and spinal parameters. The system could also illustrate the spinal alignments, trunk asymmetry in the coronal plane, horizontal trunk rotation, and force locations and directions to describe how the scoliosis deformity changed in three dimensions. The researcher could adjust the scoliosis casting apparatus to apply the 3-dimensional force to correct the 3-dimensional spinal deformity of scoliosis patients during the optimal correction and casting. All spinal parameters were improved from assessment to optimal correction and casting. As a result, the scoliosis casting apparatus could be adjusted to correct the spinal deformity in 3 dimensions and can already be used to treat scoliosis patients in clinical practice. The forces could be measured during optimal correction and casting or when the scoliosis casting apparatus was adjusted to apply the forces to correct the deformity.

When combining three developed systems, the low-cost postural measurement system could quantify the spinal deformity of AIS patients and illustrate some spinal parameters and the force directions when correcting the deformity in three dimensions. The scoliosis casting apparatus could be adjusted to apply the forces to correct the spinal deformity in three dimensions. The force-measuring system could measure the magnitude of forces used to correct the spinal deformity during optimal correction and casting. Therefore, it is time to apply the systems in clinical practice to assist clinicians during orthotic treatment for AIS patients and improve the outcome measurement in scoliosis by considering the deformity change in 3 dimensions.

## **References**

## References

- Aaro, S. and Dahlborn, M. (1981) 'Estimation of vertebral rotation and the spinal and rib cage deformity in scoliosis by computer tomography'. *Spine*, 6 (5), pp. 460-467.
- Adam, C.J., Askin, G.N. and Pearcy, M.J. (2008) 'Gravity-induced torque and intravertebral rotation in idiopathic scoliosis'. *Spine (Phila Pa 1976)*, 33 (2), pp. E30-37.
- Amer, G., Mudassir, S. and Malik, M. (2015) Published. 'Design and operation of Wi-Fi agribot integrated system'. *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, 2015. IEEE, pp.207-212.
- Ameri, E. *et al.* (2015) 'The role of routine preoperative magnetic resonance imaging in idiopathic scoliosis: a ten years review'. *Asian spine journal*, 9 (4), pp. 511.
- Andriacchi, T.P. and Alexander, E.J. (2000) 'Studies of human locomotion: past, present and future'. *J Biomech*, 33 (10), pp. 1217-1224.
- Ate, A. and Abdelrahim, M. (2018) Published. 'Controlling the temperature reactor based on Raspberry Pi system control'. *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, 2018. IEEE, pp.215-218.
- Aulisa, A.G. *et al.* (2015) 'Lyon bracing in adolescent females with thoracic idiopathic scoliosis: a prospective study based on SRS and SOSORT criteria'. *BMC Musculoskelet Disord*, 16 316.
- Balasubramaniyan, C. and Manivannan, D. (2016) 'Iot enabled air quality monitoring system (AQMS) using raspberry Pi'. *Indian Journal of Science and Technology*, 9 (39), pp. 1-6.
- Ballhause, T.M. *et al.* (2019) 'Serial casting in early onset scoliosis: syndromic scoliosis is no contraindication'. *BMC Musculoskelet Disord*, 20 (1), pp. 554.
- Banerjee, S. *et al.* (2013) Published. 'Secure sensor node with Raspberry Pi'. *IMPACT-2013*, 2013. IEEE, pp.26-30.
- Bayer, L.M.L.M. (1959) '*RADIOGRAPHIC ATLAS OF SKELETAL DEVELOPMENT OF THE HAND AND WRIST: Second Edition*'.
- Belyea, B.C. *et al.* (2015) 'Validity and Intrarater Reliability of a 2-Dimensional Motion Analysis Using a Handheld Tablet Compared With Traditional 3-Dimensional Motion Analysis'. *Journal of sport rehabilitation*, 24 (4), pp.
- Birchall, D. *et al.* (1997) 'Measurement of vertebral rotation in adolescent idiopathic scoliosis using three-dimensional magnetic resonance imaging'. *Spine*, 22 (20), pp. 2403-2407.



- BostonOrthotics&Prosthetics (2023) *Scoliosis Bracing Results & Clinical Outcomes*. Available at: <https://www.bostonoandp.com/for-patients/scoliosis-and-spine/scoliosis-outcomes/> (Accessed: 25 December 2023).
- Brink, R.C. *et al.* (2018) 'A reliability and validity study for different coronal angles using ultrasound imaging in adolescent idiopathic scoliosis'. *The Spine Journal*, 18 (6), pp. 979-985.
- Brooks, H.L. *et al.* (1975) 'Scoliosis: A prospective epidemiological study'. *J Bone Joint Surg Am*, 57 (7), pp. 968-972.
- Brown, D. (1971) Published. 'Close-Range Camera Calibration'. 1971.
- Burwell, R.G. *et al.* (1983) 'Standardised trunk asymmetry scores. A study of back contour in healthy school children'. *J Bone Joint Surg Br*, 65 (4), pp. 452-463.
- Cabaccan, C.N., Cruz, F.R.G. and Agulto, I.C. (2017) Published. 'Wireless sensor network for agricultural environment using raspberry pi based sensor nodes'. *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2017. IEEE, pp.1-5.
- Canavese, F. *et al.* (2016) 'Serial elongation, derotation and flexion (EDF) casting under general anesthesia and neuromuscular blocking drugs improve outcome in patients with juvenile scoliosis: preliminary results'. *European Spine Journal*, 25 (2), pp. 487-494.
- Carman, D., Browne, R. and Birch, J. (1990) 'Measurement of scoliosis and kyphosis radiographs. Intraobserver and interobserver variation'. *JBSJ*, 72 (3), pp. 328-333.
- Carr, W.A. *et al.* (1980) 'Treatment of idiopathic scoliosis in the Milwaukee brace'. *J Bone Joint Surg Am*, 62 (4), pp. 599-612.
- Chaudhari, K.G. (2019) 'Windmill Monitoring System Using Internet of Things with Raspberry Pi'. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 8 (2), pp. 482-485.
- Chêneau, J. (1994) *Corset-Chêneau : manuel d'orthopédie des scolioses suivant la technique originale*. Paris: Editions Frison-Roche Paris.
- Chongov, B. *et al.* (2017) 'Correlation between scoliosis deformity type and trunk symmetry before and after schroth physiotherapeutic exercises'. *Comptes rendus de l'Académie bulgare des Sciences*, 70 (10), pp. 1455-1463.
- Clark, R.A. *et al.* (2013) 'Validity of the Microsoft Kinect for providing lateral trunk lean feedback during gait retraining'. *Gait Posture*, 38 (4), pp. 1064-1066.
- Cobb, J. (1948a) 'Outline for the study of scoliosis'. *Instructional course lecture*.

- Cobb, J. (1948b) 'Outline for the study of scoliosis'. *AAOS Instr Course Lect* (5), pp. 261-275.
- Coillard, C., Circo, A.B. and Rivard, C.H. (2014) 'A prospective randomized controlled trial of the natural history of idiopathic scoliosis versus treatment with the SpineCor brace. Sosort Award 2011 winner'. *Eur J Phys Rehabil Med*, 50 (5), pp. 479-487.
- Coillard, C. *et al.* (2007) 'Effectiveness of the SpineCor Brace Based on the New Standardized Criteria Proposed by the Scoliosis Research Society for Adolescent Idiopathic Scoliosis'. *Journal of Pediatric Orthopaedics*, 27 (4), pp. 375-379.
- Cotrel, Y. and Morel, G. (1964) '[THE ELONGATION-DEROTATION-FLEXION TECHNIC IN THE CORRECTION OF SCOLIOSIS]'. *Rev Chir Orthop Reparatrice Appar Mot*, 50 59-75.
- Danielsson, A.J. *et al.* (2007) 'A prospective study of brace treatment versus observation alone in adolescent idiopathic scoliosis: a follow-up mean of 16 years after maturity'. *Spine (Phila Pa 1976)*, 32 (20), pp. 2198-2207.
- Danita, M. *et al.* (2018) Published. 'IoT based automated greenhouse monitoring system'. *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018. IEEE, pp.1933-1937.
- de Mauroy, J.C. *et al.* (2014) 'Prospective study and new concepts based on scoliosis detorsion of the first 225 early in-brace radiological results with the new Lyon brace: ARTbrace'. *Scoliosis*, 9.
- De Smet, A.A. (1985) 'Radiology of spinal curvature'.
- Dede, O. and Sturm, P.F. (2016) 'A brief history and review of modern casting techniques in early onset scoliosis'. *Journal of children's orthopaedics*, 10 (5), pp. 405-411.
- DeSmet, A.A. *et al.* (1982) 'A clinical study of the differences between the scoliotic angles measured on posteroanterior and anteroposterior radiographs'. *J Bone Joint Surg Am*, 64 (4), pp. 489-493.
- Dickson, R.A. (1983) 'Scoliosis in the community'. *Br Med J (Clin Res Ed)*, 286 (6365), pp. 615-618.
- Dickson, R.A. *et al.* (1984) 'The pathogenesis of idiopathic scoliosis. Biplanar spinal asymmetry'. *The Journal of bone and joint surgery. British volume*, 66 (1), pp. 8-15.
- Dickson, R.A. and Weinstein, S.L. (1999) 'Bracing (and screening)--yes or no?'. *J Bone Joint Surg Br*, 81 (2), pp. 193-198.

- Dimeglio, A. (2001) 'Growth in Pediatric Orthopaedics'. *Journal of Pediatric Orthopaedics*, 21 (4), pp. 549-555.
- Eltoukhy, M. *et al.* (2012) 'Evaluation of the Performance of Digital Video Analysis of Human Motion: Dartfish Tracking System'. *International Journal of Scientific and Engineering Research*, 3.
- Emans, J.B. (2003) *Reference manual for the Boston brace*. Boston: Boston : Boston Brace International.
- Emans, J.B. *et al.* (1986) 'The Boston bracing system for idiopathic scoliosis. Follow-up results in 295 patients'. *Spine (Phila Pa 1976)*, 11 (8), pp. 792-801.
- Fabricant, P.D. *et al.* (2013) 'A Radiographic Study of the Ossification of the Posterior Wall of the Acetabulum: Implications for the Diagnosis of Pediatric and Adolescent Hip Disorders'. *JBJS*, 95 (3), pp. 230-236.
- Fairbank, J. (2004) 'Historical perspective: William Adams, the forward bending test, and the spine of Gideon Algernon Mantell'. *Spine*, 29 (17), pp. 1953-1955.
- Fang, M.-Q. *et al.* (2015a) 'Long-term effects of the Chêneau brace on coronal and sagittal alignment in adolescent idiopathic scoliosis'. *Journal of Neurosurgery: Spine SPI*, 23 (4), pp. 505-509.
- Fang, M.Q. *et al.* (2015b) 'Long-term effects of the Chêneau brace on coronal and sagittal alignment in adolescent idiopathic scoliosis'. *J Neurosurg Spine*, 23 (4), pp. 505-509.
- Fayssoux, R.S., Cho, R.H. and Herman, M.J. (2010) 'A History of Bracing for Idiopathic Scoliosis in North America'. *Clin Orthop Relat Res*, 468 (3), pp. 654-664.
- Fernández-González, P. *et al.* (2020) 'Reliability of Kinovea® Software and Agreement with a Three-Dimensional Motion System for Gait Analysis in Healthy Subjects'. *Sensors*, 20 (11), pp. 3154.
- Fetters, L.T.J. (2012) 'Evidence based physical therapy'.
- FileZilla (2023) *FileZilla The free FTP solution*. Available at: <https://filezilla-project.org/> (Accessed: 1 December 2023).
- Findikcioglu, K. *et al.* (2013) 'The impact of breast reduction surgery on the vertebral column'. *Annals of plastic surgery*, 70 (6), pp. 639-642.
- Fisher, D.A., Rapp, G.F. and Emkes, M. (1987) 'Idiopathic scoliosis: transcutaneous muscle stimulation versus the Milwaukee brace'. *Spine (Phila Pa 1976)*, 12 (10), pp. 987-991.

- Frerich, J.M. *et al.* (2012) 'Comparison of radiographic and surface topography measurements in adolescents with idiopathic scoliosis'. *The open orthopaedics journal*, 6 261-265.
- Fryer, J.G. and Brown, D.C. (1986) 'Lens distortion for close-range photogrammetry'. *Photogrammetric Engineering and Remote Sensing*, 52 51.
- Girdler, S. *et al.* (2020) 'Emerging techniques in diagnostic imaging for idiopathic scoliosis in children and adolescents: a review of the literature'. *World neurosurgery*, 136 128-135.
- Gore, D.R. *et al.* (1981) 'Scoliosis screening: results of a community project'. *Pediatrics*, 67 (2), pp. 196-200.
- Gray, A.D. *et al.* (2017) 'Development and Validation of a Portable and Inexpensive Tool to Measure the Drop Vertical Jump Using the Microsoft Kinect V2'. *Sports Health*, 9 (6), pp. 537-544.
- Greenwood, D. and Bogar, W. (2014) 'Congenital scoliosis in non-identical twins case reports and literature review'. *Journal of the Canadian Chiropractic Association*, 58 291+.
- Grivas, T.B. *et al.* (2003) 'The effect of a modified Boston brace with anti-rotatory blades on the progression of curves in idiopathic scoliosis: aetiologic implications'. *Pediatr Rehabil*, 6 (3-4), pp. 237-242.
- Grivas, T.B. *et al.* (2006a) 'Association between adolescent idiopathic scoliosis prevalence and age at menarche in different geographic latitudes'. *Scoliosis*, 1 9.
- Grivas, T.B. *et al.* (2006b) 'Geographic latitude and prevalence of adolescent idiopathic scoliosis'. *Stud Health Technol Inform*, 123 84-89.
- Grivas, T.B. *et al.* (2007) 'SOSORT consensus paper: school screening for scoliosis. Where are we today?'. *Scoliosis*, 2 (1), pp. 17.
- Gum, J.L. *et al.* (2007) 'Transverse plane pelvic rotation in adolescent idiopathic scoliosis: primary or compensatory?'. *European spine journal : official publication of the European Spine Society, the European Spinal Deformity Society, and the European Section of the Cervical Spine Research Society*, 16 (10), pp. 1579-1586.
- Hacquebord, J.H. and Leopold, S.S. (2012) 'In brief: The Risser classification: a classic tool for the clinician treating adolescent idiopathic scoliosis'. *Clin Orthop Relat Res*, 470 (8), pp. 2335-2338.
- Harris, G.F. *et al.* (1996) *Human motion analysis : current applications and future directions*. New York: New York : Institute of Electrical and Electronics Engineers.

- Hattori, T. *et al.* (2011) 'In vivo three-dimensional segmental analysis of adolescent idiopathic scoliosis'. *European Spine Journal*, 20 (10), pp. 1745-1750.
- Ho, E.K. *et al.* (1993) 'New methods of measuring vertebral rotation from computed tomographic scans: an intraobserver and interobserver study on girls with scoliosis'. *Spine*, 18 (9), pp. 1173-1177.
- Huang, S.C. (1997) 'Cut-off point of the Scoliometer in school scoliosis screening'. *Spine (Phila Pa 1976)*, 22 (17), pp. 1985-1989.
- Idris, R., Alkooheji, A. and Jameel, A. (2022) 'Image Processing Based Smart Home Automation using Raspberry Pi and Python'.
- Ilharreborde, B. *et al.* (2016) 'EOS microdose protocol for the radiological follow-up of adolescent idiopathic scoliosis'. *European Spine Journal*, 25 526-531.
- Illés, T. and Somoskeöy, S. (2012) 'The EOS™ imaging system and its uses in daily orthopaedic practice'. *International orthopaedics*, 36 (7), pp. 1325-1331.
- Inami, K. *et al.* (1999) 'Analysis of posterior trunk symmetry index (POTSI) in scoliosis. Part 2'. *Research into spinal deformities 2*. IOS Press, pp. 85-88.
- Jang, S.H. (2018) *Development of a novel non-invasive digital clinical assessment tool and its 3D measurable parameters for the orthotic treatment of adolescent idiopathic scoliosis*. Thesis [Ph. D.] -- University of Strathclyde, 2018.
- Janicki, J.A. and Alman, B. (2007a) 'Scoliosis: Review of diagnosis and treatment'. *Paediatr Child Health*, 12 (9), pp. 771-776.
- Janicki, J.A. and Alman, B. (2007b) 'Scoliosis: Review of diagnosis and treatment'. *Paediatr Child Health*, 12 (9), pp. 771-776.
- Jaremko, J.L. *et al.* (2002) 'Indices of torso asymmetry related to spinal deformity in scoliosis'. *Clin Biomech (Bristol, Avon)*, 17 (8), pp. 559-568.
- Kaehler, A. and Bradski, G. (2016) *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc.
- Kamal, N. and Ghosal, P. (2018) Published. 'Three tier architecture for iot driven health monitoring system using raspberry pi'. *2018 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*, 2018. IEEE, pp.167-170.
- Karaaslan, O. *et al.* (2013) 'The effect of reduction mammoplasty on the vertebral column: a radiologic study'. *The Scientific World Journal*, 2013.
- Karami, M., Maleki, A. and Mazda, K. (2016) 'Assessment of coronal radiographic parameters of the spine in the treatment of adolescent idiopathic scoliosis'. *Archives of Bone and Joint Surgery*, 4 (4), pp. 376.

- Karpiel, I. *et al.* (2021) 'A Survey of Methods and Technologies Used for Diagnosis of Scoliosis'. *Sensors*, 21 (24), pp. 8410.
- Karthikeyan, S. *et al.* (2023) 'A Systematic Analysis on Raspberry Pi Prototyping: Uses, Challenges, Benefits, and Drawbacks'. *IEEE Internet of Things Journal*.
- Kato, S., Debaud, C. and Zeller, R.D. (2017) 'Three-dimensional EOS analysis of apical vertebral rotation in adolescent idiopathic scoliosis'. *Journal of Pediatric Orthopaedics*, 37 (8), pp. e543-e547.
- Katz, D.E. *et al.* (2010) 'Brace wear control of curve progression in adolescent idiopathic scoliosis'. *J Bone Joint Surg Am*, 92 (6), pp. 1343-1352.
- Katz, D.E. *et al.* (1997) 'A comparison between the Boston brace and the Charleston bending brace in adolescent idiopathic scoliosis'. *Spine (Phila Pa 1976)*, 22 (12), pp. 1302-1312.
- Keiser, R.P. and Shufflebarger, H.L. (1976) 'The Milwaukee brace in idiopathic scoliosis: evaluation of 123 completed cases'. *Clin Orthop Relat Res*, (118), pp. 19-24.
- Khanali, M. *et al.* (2015) 'The influence of thoracolumbosacral orthoses on standing balance in patients with adolescent idiopathic scoliosis: a pilot study'. *Disability and Rehabilitation: Assistive Technology*, 10 (6), pp. 452-457.
- Kharazi, M. *et al.* (2015) Published. 'Validity of microsoft kinect™ for measuring gait parameters'. *2015 22nd Iranian Conference on Biomedical Engineering (ICBME)*, 2015. IEEE, pp.375-379.
- Kim, A. *et al.* (2015) 'A wearable smartphone-enabled camera-based system for gait assessment'. *Gait & posture*, 42 (2), pp. 138-144.
- Kim, S.-H. *et al.* (2019) 'Ossification of the triradiate cartilage and posterior acetabulum'. *Journal of the Korean Society of Radiology*, 80 (3), pp. 503-512.
- King, H.A. *et al.* (1983) 'The selection of fusion levels in thoracic idiopathic scoliosis'. *J Bone Joint Surg Am*, 65 (9), pp. 1302-1313.
- Knott, P. *et al.* (2006) 'Electromagnetic topographical technique of curve evaluation for adolescent idiopathic scoliosis'. *Spine*, 31 (24), pp. E911-E915.
- Knott, P. *et al.* (2016) 'Multicenter comparison of 3D spinal measurements using surface topography with those from conventional radiography'. *Spine deformity*, 4 (2), pp. 98-103.
- Komeili, A. *et al.* (2015a) 'Correlation Between a Novel Surface Topography Asymmetry Analysis and Radiographic Data in Scoliosis'. *Spine Deform*, 3 (4), pp. 303-311.

- Komeili, A. *et al.* (2015b) 'Monitoring for idiopathic scoliosis curve progression using surface topography asymmetry analysis of the torso in adolescents'. *The Spine Journal*, 15 (4), pp. 743-751.
- Komeili, A. *et al.* (2014) 'Surface topography asymmetry maps categorizing external deformity in scoliosis'. *The Spine Journal*, 14 (6), pp. 973-983.e972.
- Kotwicki, T. (2008) 'Evaluation of scoliosis today: examination, X-rays and beyond'. *Disabil Rehabil*, 30 (10), pp. 742-751.
- Kotwicki, T. and Cheneau, J. (2008) 'Biomechanical action of a corrective brace on thoracic idiopathic scoliosis: Cheneau 2000 orthosis'. *Disabil Rehabil Assist Technol*, 3 (3), pp. 146-153.
- Kotwicki, T. *et al.* (2007) 'Discrepancy in clinical versus radiological parameters describing deformity due to brace treatment for moderate idiopathic scoliosis'. *Scoliosis*, 2 (1), pp. 18.
- Koukourakis, I. *et al.* (1997) 'Screening school children for scoliosis on the island of Crete'. *J Spinal Disord*, 10 (6), pp. 527-531.
- Krismer, M. *et al.* (1996) 'Axial rotation measurement of scoliotic vertebrae by means of computed tomography scans'. *Spine*, 21 (5), pp. 576-581.
- Kumar, S. and Jasuja, A. (2017) Published. 'Air quality monitoring system based on IoT using Raspberry Pi'. *2017 International conference on computing, communication and automation (ICCCA)*, 2017. IEEE, pp.1341-1346.
- Lam, G.C. *et al.* (2008) 'Vertebral rotation measurement: a summary and comparison of common radiographic and CT methods'. *Scoliosis*, 3 1-10.
- Lange, J.E. *et al.* (2011) 'Long-term results after Boston brace treatment in late-onset juvenile and adolescent idiopathic scoliosis'. *Scoliosis*, 6 18.
- Lau, K.K.-L. *et al.* (2022) 'Reliability of a three-dimensional spinal proprioception assessment for patients with adolescent idiopathic scoliosis'. *European Spine Journal*, 31 (11), pp. 3013-3019.
- Laulund, T., Søjbjerg, J.O. and Hørlyck, E. (1982) 'Moiré topography in school screening for structural scoliosis'. *Acta Orthop Scand*, 53 (5), pp. 765-768.
- Lavanya, S., Lavanya, G. and Divyabharathi, J. (2017) Published. 'Remote prescription and I-Home healthcare based on IoT'. *2017 international conference on innovations in green energy and healthcare technologies (IGEHT)*, 2017. IEEE, pp.1-3.
- Lee, C.S. *et al.* (2012) 'Effectiveness of the Charleston Night-time Bending Brace in the Treatment of Adolescent Idiopathic Scoliosis'. *Journal of Pediatric Orthopaedics*, 32 (4), pp. 368-372.

- Lenke, C. (2000) 'SRS Terminology committee and working group on spinal classification'. *Revised Glossary of Terms. Scoliosis Research Society, Milwaukee, US.*
- Lenke, L.G. *et al.* (2001a) 'Adolescent Idiopathic Scoliosis : A New Classification to Determine Extent of Spinal Arthrodesis'. *J Bone Joint Surg Am*, 83 (8), pp. 1169-1181.
- Lenke, L.G. *et al.* (2001b) 'Adolescent idiopathic scoliosis: a new classification to determine extent of spinal arthrodesis'. *JBJS*, 83 (8), pp. 1169-1181.
- Li, M. *et al.* (2010) 'Application of 3-D ultrasound in assisting the fitting procedure of spinal orthosis to patients with adolescent idiopathic scoliosis'. *Stud Health Technol Inform*, 158 34-37.
- Li, M. *et al.* (2012) 'Could clinical ultrasound improve the fitting of spinal orthosis for the patients with AIS?'. *European Spine Journal*, 21 (10), pp. 1926-1935.
- Liu, X.C. *et al.* (2001) 'Functional classification of patients with idiopathic scoliosis assessed by the Quantec system: a discriminant functional analysis to determine patient curve magnitude'. *Spine (Phila Pa 1976)*, 26 (11), pp. 1274-1278; discussion 1279.
- Lonstein, J.E. (2006) 'Scoliosis: surgical versus nonsurgical treatment'. *Clin Orthop Relat Res*, 443 248-259.
- Lonstein, J.E. and Winter, R.B. (1994) 'The Milwaukee brace for the treatment of adolescent idiopathic scoliosis. A review of one thousand and twenty patients'. *J Bone Joint Surg Am*, 76 (8), pp. 1207-1221.
- Lou, E., Hill, D. and Raso, J. (2008) 'Brace treatment for adolescent idiopathic scoliosis'. *Studies in health technology and informatics*, 135 265.
- Lou, E.H. *et al.* (2015) 'Ultrasound-assisted brace casting for adolescent idiopathic scoliosis, IRSSD Best research paper 2014'. *Scoliosis*, 10 13-13.
- Lou, E.H. *et al.* (2017) 'Results of ultrasound-assisted brace casting for adolescent idiopathic scoliosis'. *Scoliosis and spinal disorders*, 12 23-23.
- Loukos, I. *et al.* (2011) 'Analysis of the corrective forces exerted by a dynamic derotation brace (DDB)'. *Prosthetics and Orthotics International*, 35 (4), pp. 365-372.
- Lyon, R. *et al.* (2004) 'Reproducibility of spinal back-contour measurements taken with raster stereography in adolescent idiopathic scoliosis'. *American journal of orthopedics (Belle Mead, N.J.)*, 33 (2), pp. 67-70.
- Maclean, M. *et al.* (2020) 'Non-ionizing 405 nm Light as a Potential Bactericidal Technology for Platelet Safety: Evaluation of in vitro Bacterial Inactivation



- and in vivo Platelet Recovery in Severe Combined Immunodeficient Mice'. *Frontiers in Medicine*, 6 (331), pp.
- Mahajan, R. *et al.* (2020) 'Evolution of casting techniques in early-onset and congenital scoliosis'. *Journal of clinical orthopaedics and trauma*, 11 (5), pp. 810-815.
- Márkus, I. *et al.* (2018) 'The effect of coronal decompensation on the biomechanical parameters in lower limbs in adolescent idiopathic scoliosis'. *Orthopaedics & Traumatology: Surgery & Research*, 104 (5), pp. 609-616.
- McCoy and Barry, D. (1996) '*Apparatus for forming a scoliosis brace*'.
- Medium (2023) *Camera Calibration with OpenCV*. Available at: <https://medium.com/analytics-vidhya/camera-calibration-with-opencv-f324679c6eb7> (Accessed: 1 December 2023).
- Melhem, E. *et al.* (2016) 'EOS® biplanar X-ray imaging: concept, developments, benefits, and limitations'. *Journal of children's orthopaedics*, 10 (1), pp. 1-14.
- Monticone, M. *et al.* (2014) 'Active self-correction and task-oriented exercises reduce spinal deformity and improve quality of life in subjects with mild adolescent idiopathic scoliosis. Results of a randomised controlled trial'. *Eur Spine J*, 23 (6), pp. 1204-1214.
- Morais, T., Bernier, M. and Turcotte, F. (1985) 'Age- and sex-specific prevalence of scoliosis and the value of school screening programs'. *Am J Public Health*, 75 (12), pp. 1377-1380.
- Morel, B. *et al.* (2018) 'Dose, image quality and spine modeling assessment of biplanar EOS micro-dose radiographs for the follow-up of in-brace adolescent idiopathic scoliosis patients'. *European Spine Journal*, 27 (5), pp. 1082-1088.
- Morrissy, R.T. *et al.* (1990) 'Measurement of the Cobb angle on radiographs of patients who have scoliosis. Evaluation of intrinsic error'. *JBJS*, 72 (3), pp. 320-327.
- Nagymáté, G. and Kiss, R.M. (2018) 'Motion capture system validation with surveying techniques'. *Materials Today: Proceedings*, 5 (13, Part 2), pp. 26501-26506.
- NASH, C.L.J. and MOE, J.H. (1969) 'A Study of Vertebral Rotation'. *JBJS*, 51 (2), pp. 223-229.
- Navarro, I., Candotti, C. and Rosa, B. (2019) 'Anatomic markers, evaluation parameters and validity of surface topography for idiopathic adolescent scoliosis: a systematic review with meta-analysis'. *Gait & Posture*, 69.
- Negrini, S. *et al.* (2014) 'Letter to the editor concerning: "active self-correction and task-oriented exercises reduce spinal deformity and improve quality of life in subjects with mild adolescent idiopathic scoliosis. Results of a randomised controlled trial" by Monticone M, Ambrosini E, Cazzaniga D, Rocca B,

- Ferrante S (2014). *Eur Spine J*; DOI:10.1007/s00586-014-3241-y'. *Eur Spine J*, 23 (10), pp. 2218-2220.
- Negrini, S. *et al.* (2018) '2016 SOSORT guidelines: orthopaedic and rehabilitation treatment of idiopathic scoliosis during growth'. *Scoliosis Spinal Disord*, 13 3.
- Negrini, S. *et al.* (2006) 'Why do we treat adolescent idiopathic scoliosis? What we want to obtain and to avoid for our patients. SOSORT 2005 Consensus paper'. *Scoliosis*, 1 4.
- Negrini, S. *et al.* (2015) 'Recommendations for research studies on treatment of idiopathic scoliosis: Consensus 2014 between SOSORT and SRS non-operative management committee'. *Scoliosis*, 10.
- Ng, S.-Y. and Bettany-Saltikov, J. (2017) 'Suppl-9, M5: Imaging in the Diagnosis and Monitoring of Children with Idiopathic Scoliosis'. *The open orthopaedics journal*, 11 1500.
- Nissinen, M. *et al.* (1993) 'Trunk asymmetry and screening for scoliosis: a longitudinal cohort study of pubertal schoolchildren'. *Acta Paediatr*, 82 (1), pp. 77-82.
- Ozturk, C. *et al.* (2010) 'The role of routine magnetic resonance imaging in the preoperative evaluation of adolescent idiopathic scoliosis'. *International orthopaedics*, 34 543-546.
- Parent, S., Newton, P.O. and Wenger, D.R. (2005) 'Adolescent idiopathic scoliosis: etiology, anatomy, natural history, and bracing'. *Instr Course Lect*, 54 529-536.
- Park, M.S. *et al.* (2013) 'The effect of age on cervical sagittal alignment: normative data on 100 asymptomatic subjects'. *Spine*, 38 (8), pp. E458-E463.
- Parks, M.T., Wang, Z. and Siu, K.-C. (2019) 'Current Low-Cost Video-Based Motion Analysis Options for Clinical Rehabilitation: A Systematic Review'. *Phys Ther*, 99 (10), pp. 1405-1425.
- Parvaresh, K.C. *et al.* (2018) 'Secondary Ossification Center Appearance and Closure in the Pelvis and Proximal Femur'. *Journal of Pediatric Orthopaedics*, 38 (8), pp. 418-423.
- Pasquini, G. *et al.* (2016) 'The outcome of a modified version of the Cheneau brace in adolescent idiopathic scoliosis (AIS) based on SRS and SOSORT criteria: a retrospective study'. *Eur J Phys Rehabil Med*, 52 (5), pp. 618-629.
- Payne, W.K., 3rd *et al.* (1997) 'Does scoliosis have a psychological impact and does gender make a difference?'. *Spine (Phila Pa 1976)*, 22 (12), pp. 1380-1384.
- Pennella, D. *et al.* (2013) 'Effectiveness of pressure biofeedback / pbu (pressure biofeedback unit) in the process of learning of self-correction in patients with scoliosis: a pilot study'. *Scoliosis*, 8 (1), pp. P10.

- Pepke, W. *et al.* (2023) 'Outcome of Conservative Therapy of Adolescent Idiopathic Scoliosis (AIS) with Chêneau-Brace'. *Journal of Clinical Medicine*, 12 NA.
- Perdriolle, R. (1979) 'La scoliose: Son e'tude Tridimensionnelle'. *Ed. Maloine, Paris*.
- Pesenti, S. *et al.* (2020) 'Does static trunk motion analysis reflect its true position during daily activities in adolescent with idiopathic scoliosis?'. *Orthopaedics & Traumatology: Surgery & Research*, 106 (7), pp. 1251-1256.
- Pfister, A. *et al.* (2014) 'Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis'. *Journal of medical engineering & technology*, 38 (5), pp. 274-280.
- Pham, V.M. *et al.* (2007) '[Evaluation of the Chêneau brace in adolescent idiopathic scoliosis]'. *Ann Readapt Med Phys*, 50 (3), pp. 125-133.
- Pham, V.M. *et al.* (2008) 'Study of the pressures applied by a Chêneau brace for correction of adolescent idiopathic scoliosis'. *Prosthetics and Orthotics International*, 32 (3), pp. 345-355.
- Phidgets (2023a) *Button Load Cell - 50kg*. Available at: <https://www.phidgets.com/?prodid=227>.
- Phidgets (2023b) *PhidgetBridge 4-Input*. Available at: <https://www.phidgets.com/?prodid=1270> (Accessed: 1 December 2023).
- Pierre-Aurelien, B. *et al.* (2018) 'Management of split cord malformation in children: the Lyon experience'. *Child's Nervous System*, 34 883-891.
- Pin, L.H. *et al.* (1985) 'Early diagnosis of scoliosis based on school-screening'. *J Bone Joint Surg Am*, 67 (8), pp. 1202-1205.
- Ploumis, A. *et al.* (2009) 'A correlation of radiographic and functional measurements in adult degenerative scoliosis'. *Spine*, 34 (15), pp. 1581-1584.
- Price, C.T. *et al.* (1990) 'Nighttime Bracing for Adolescent Idiopathic Scoliosis with the Charleston Bending Brace: Preliminary Report'. *Spine*, 15 (12), pp.
- RaspberryPi (2023) *Raspberry Pi*. Available at: <https://www.raspberrypi.com/> (Accessed: 1 December 2023).
- RealVNC (2023) *RealVNC*. Available at: <https://www.realvnc.com/en/> (Accessed: 1 December 2023).
- Reither, L.R. *et al.* (2018) 'Upper extremity movement reliability and validity of the Kinect version 2'. *DISABIL REHABIL-ASSI*, 13 (1), pp. 54-59.
- Richards, B.S. (1992) 'Measurement error in assessment of vertebral rotation using the Perdriolle torsionmeter'. *Spine (Phila Pa 1976)*, 17 (5), pp. 513-517.

- Richards, B.S. *et al.* (2005) 'Standardization of Criteria for Adolescent Idiopathic Scoliosis Brace Studies: SRS Committee on Bracing and Nonoperative Management'. *Spine*, 30 (18), pp. 2068-2075.
- Richards, J.G. (1999) 'The measurement of human motion: A comparison of commercially available systems'. *Human Movement Science*, 18 (5), pp. 589-602.
- Rigo, M. and Weiss, H. (2008) 'The Chêneau concept of bracing-Biomechanical aspects'. *Studies in Health Technology and Informatics*, 135 303.
- Riseborough, E.J. and Wynne-Davies, R. (1973) 'A genetic survey of idiopathic scoliosis in Boston, Massachusetts'. *J Bone Joint Surg Am*, 55 (5), pp. 974-982.
- Risser, J.C. (1958) 'The Iliac apophysis; an invaluable sign in the management of scoliosis'. *Clin Orthop*, 11 111-119.
- Robinson, C. and Fatone, S. (2012) 'You've heard about outcome measures, so how do you use them? Integrating clinically relevant outcome measures in orthotic management of stroke'. *Prosthet Orthot Int*, 37 (1), pp. 30-42.
- Rogala, E.J., Drummond, D.S. and Gurr, J. (1978) 'Scoliosis: incidence and natural history. A prospective epidemiological study'. *J Bone Joint Surg Am*, 60 (2), pp. 173-176.
- Romano, M. *et al.* (2012) 'Exercises for adolescent idiopathic scoliosis'. *Cochrane Database Syst Rev*, 2012 (8), pp. Cd007837.
- Romano, M. *et al.* (2013) 'Exercises for adolescent idiopathic scoliosis: a Cochrane systematic review'. *Spine (Phila Pa 1976)*, 38 (14), pp. E883-893.
- Sachian, M.-A. *et al.* (2020) Published. 'Cyber-physical healthcare security system based on a Raspberry Pi'. *Advanced Topics in Optoelectronics, Microelectronics and Nanotechnologies X*, 2020. SPIE, pp.517-525.
- Sanders, J.O. (2016) 'Casting for Infantile Idiopathic Scoliosis'. *Operative Techniques in Orthopaedics*, 26 (4), pp. 218-221.
- Sanders, J.O. and D'Astous, J. (2009) 'Cast Treatment of Infantile or Early Onset Scoliosis'. *Operative Techniques in Orthopaedics*, 19 (1), pp. 2-5.
- Sanders, J.O., Johnston, C.E. and D'Astous, J. (2012) 'Casting in Early-Onset Scoliosis'. *Seminars in Spine Surgery*, 24 (3), pp. 144-148.
- Schiller, J.R., Thakur, N.A. and Ebersson, C.P. (2010) 'Brace management in adolescent idiopathic scoliosis'. *Clin Orthop Relat Res*, 468 (3), pp. 670-678.
- Schmid, S. *et al.* (2015) 'Using Skin Markers for Spinal Curvature Quantification in Main Thoracic Adolescent Idiopathic Scoliosis: An Explorative Radiographic Study'. *PloS one*, 10 (8), pp. e0135689-e0135689.

- Schmid, S. *et al.* (2016) 'Quantifying spinal gait kinematics using an enhanced optical motion capture approach in adolescent idiopathic scoliosis'. *Gait & Posture*, 44 231-237.
- Schmitz, A. *et al.* (2001) 'A new MRI technique for imaging scoliosis in the sagittal plane'. *European Spine Journal*, 10 114-117.
- Schülein, S. *et al.* (2013) 'Rasterstereographic evaluation of interobserver and intraobserver reliability in postsurgical adolescent idiopathic scoliosis patients'. *J Spinal Disord Tech*, 26 (4), pp. E143-149.
- Schurr, S.A. *et al.* (2017) 'Two-dimensional video analysis is comparable to 3D motion capture in lower extremity movement assessment'. *International journal of sports physical therapy*, 12 (2), pp. 163.
- Shands, A.R., Jr. and Eisberg, H.B. (1955) 'The incidence of scoliosis in the state of Delaware; a study of 50,000 minifilms of the chest made during a survey for tuberculosis'. *J Bone Joint Surg Am*, 37-a (6), pp. 1243-1249.
- Shapiro, S.S. and Wilk, M.B. (1965) 'An analysis of variance test for normality (complete samples)'. *Biometrika*, 52 (3-4), pp. 591-611.
- Shaughnessy, W.J. (2007) 'Advances in scoliosis brace treatment for adolescent idiopathic scoliosis'. *Orthopedic Clinics of North America*, 38 (4), pp. 469-475.
- Solomito, M.J., Lee, M.C. and Peterson, D.R. (2011) Published. 'Using motion analysis technology to reduce radiographs in patients with scoliosis'. *2011 IEEE 37th Annual Northeast Bioengineering Conference (NEBEC)*, 1-3 April 2011 2011. pp.1-2.
- Sooryavanshi, P., Urganlawar, S. and Bhosle, A. (2017) Published. 'Implementation of node.js server on Raspberry pi to control a remote vehicle for defense use'. *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, 2017. IEEE, pp.816-819.
- Sornalatha, K. and Kavitha, V. (2017) Published. 'IoT based smart museum using Bluetooth Low Energy'. *2017 third international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB)*, 2017. IEEE, pp.520-523.
- Soucacos, P.N. *et al.* (1997) 'School-screening for scoliosis. A prospective epidemiological study in northwestern and central Greece'. *J Bone Joint Surg Am*, 79 (10), pp. 1498-1503.
- Sruthy, S. and George, S.N. (2017) Published. 'WiFi enabled home security surveillance system using Raspberry Pi and IoT module'. *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 2017. IEEE, pp.1-6.

- Stokes, I.A. (1989) 'Axial rotation component of thoracic scoliosis'. *J Orthop Res*, 7 (5), pp. 702-708.
- Stokes, I.A., Bigalow, L.C. and Moreland, M.S. (1987) 'Three-dimensional spinal curvature in idiopathic scoliosis'. *J Orthop Res*, 5 (1), pp. 102-113.
- Sturm, P.F. and Maybank, S.J. (1999) Published. 'On plane-based camera calibration: A general algorithm, singularities, applications'. *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 23-25 June 1999. pp.432-437 Vol. 431.
- Sung, P.S. and Park, M.S. (2020) 'Lumbar spine coordination during axial trunk rotation in adolescents with and without right thoracic idiopathic scoliosis'. *Human Movement Science*, 73 102680.
- Suzuki, N. *et al.* (1999) 'Analysis of posterior trunk symmetry index (POTSI) in scoliosis. Part 1'. *Research into spinal deformities 2*. IOS Press, pp. 81-84.
- Tay, V.S., Kornberg, A. and Cook, M. (2007) 'CHAPTER 38 - SPINE AND SPINAL CORD: DEVELOPMENTAL DISORDERS'. In: Schapira, A.H.V. *et al.* (eds.) *Neurology and Clinical Neuroscience*. Philadelphia: Mosby, pp. 488-506.
- Theologis, T.N. *et al.* (1997) 'Early detection of progression in adolescent idiopathic scoliosis by measurement of changes in back shape with the Integrated Shape Imaging System scanner'. *Spine (Phila Pa 1976)*, 22 (11), pp. 1223-1227; discussion 1228.
- Thometz, J. *et al.* (2000) 'Relationship between Quantec measurement and Cobb angle in patients with idiopathic scoliosis'. *Journal of Pediatric Orthopaedics*, 20 (4), pp. 512-516.
- Topley, M. and Richards, J.G. (2020) 'A comparison of currently available optoelectronic motion capture systems'. *J Biomech*, 106 109820-109820.
- van Den Hout, J. and van Den Munckhof, R. (2002) 'Interface corrective force measurements in Boston brace treatment'. *European Spine Journal*, 11 (4), pp. 332-335.
- Vijvermans, V., Fabry, G. and Nijs, J. (2004) 'Factors determining the final outcome of treatment of idiopathic scoliosis with the Boston brace: a longitudinal study'. *J Pediatr Orthop B*, 13 (3), pp. 143-149.
- Wang, J. *et al.* (2018) 'Measurement of scoliosis Cobb angle by end vertebra tilt angle method'. *Journal of Orthopaedic Surgery and Research*, 13 (1), pp. 223.
- Wang, L. *et al.* (2022) 'Optimized scheme for paired transverse corrective forces in S-shaped scoliosis via ultrasound and application in Chêneau brace: a pilot study'. *Prosthet Orthot Int*, 46 (1), pp. 42-49.

- Wang, W. *et al.* (2014) 'The Use of Finite Element Models to Assist Understanding and Treatment For Scoliosis: A Review Paper'. *Spine Deformity*, 2 (1), pp. 10-27.
- Weinstein, S.L. *et al.* (2003) 'Health and function of patients with untreated idiopathic scoliosis: a 50-year natural history study'. *Jama*, 289 (5), pp. 559-567.
- Weinstein, S.L. and Ponseti, I.V. (1983) 'Curve progression in idiopathic scoliosis'. *J Bone Joint Surg Am*, 65 (4), pp. 447-455.
- Weiß, H.-R., Rigo, M. and Chêneau, J. (2000) *Praxis der Cheneau-Korsettversorgung in der skoliosetherapie*. Thieme.
- Weiss, H. and Rigo, M. (2008) 'The cheneau concept of bracing-actual standards'. *Studies in health technology and informatics*, 135 291.
- Wiley, J.W. *et al.* (2000) 'Effectiveness of the boston brace in treatment of large curves in adolescent idiopathic scoliosis'. *Spine (Phila Pa 1976)*, 25 (18), pp. 2326-2332.
- Willers, U. *et al.* (1993) 'Long-term results of Boston brace treatment on vertebral rotation in idiopathic scoliosis'. *Spine (Phila Pa 1976)*, 18 (4), pp. 432-435.
- Willner, S. and Udén, A. (1982) 'A prospective prevalence study of scoliosis in Southern Sweden'. *Acta Orthop Scand*, 53 (2), pp. 233-237.
- Wong, H.K. *et al.* (2005) 'Idiopathic scoliosis in Singapore schoolchildren: a prevalence study 15 years into the screening program'. *Spine (Phila Pa 1976)*, 30 (10), pp. 1188-1196.
- Wong, Y.-s. *et al.* (2019) 'Is Radiation-Free Ultrasound Accurate for Quantitative Assessment of Spinal Deformity in Idiopathic Scoliosis (IS): A Detailed Analysis With EOS Radiography on 952 Patients'. *Ultrasound in Medicine & Biology*, 45 (11), pp. 2866-2877.
- Wright, N. (2000) 'Imaging in scoliosis'. *Archives of disease in childhood*, 82 (1), pp. 38-40.
- Xu, L. *et al.* (2019) 'Brace Treatment in Adolescent Idiopathic Scoliosis Patients with Curve Between 40° and 45°: Effectiveness and Related Factors'. *World Neurosurgery*, 126 e901-e906.
- Xu, T.x. and Feng–ying, H. (2021) Published. 'Stored Grain Pests Monitoring System Based on Raspberry Pi'. *2021 International Symposium on Computer Technology and Information Science (ISCTIS)*, 2021. IEEE, pp.59-62.
- Yagci, G. and Yakut, Y. (2019) 'Core stabilization exercises versus scoliosis-specific exercises in moderate idiopathic scoliosis treatment'. *Prosthetics and orthotics international*, 43 (3), pp. 301-308.

- Yawn, B. and Yawn, R.A. (2001) 'Efficacy of school scoliosis screening'. *Orthopedics*, 24 (4), pp. 317.
- Yeung, L. *et al.* (2014) 'Evaluation of the Microsoft Kinect as a clinical assessment tool of body sway'. *Gait & posture*, 40 (4), pp. 532-538.
- Yrjönen, T. *et al.* (2007) 'Results of brace treatment of adolescent idiopathic scoliosis in boys compared with girls: a retrospective study of 102 patients treated with the Boston brace'. *European spine journal : official publication of the European Spine Society, the European Spinal Deformity Society, and the European Section of the Cervical Spine Research Society*, 16 (3), pp. 393-397.
- Zabjek, K. *et al.* (2008) 'Postural Characteristics of Adolescents With Idiopathic Scoliosis'. *Journal of pediatric orthopedics*, 28 218-224.
- Zaborowska-Sapeta, K. *et al.* (2011a) 'Effectiveness of Cheneau brace treatment for idiopathic scoliosis: prospective study in 79 patients followed to skeletal maturity.(Research)(Clinical report)'. *Scoliosis*, 6 2.
- Zaborowska-Sapeta, K. *et al.* (2011b) 'Effectiveness of Chêneau brace treatment for idiopathic scoliosis: prospective study in 79 patients followed to skeletal maturity'. *Scoliosis*, 6 (1), pp. 2.
- Zabrowska-Sapeta, K. *et al.* (2010) 'Evaluation of the effectiveness of Chêneau brace treatment for idiopathic scoliosis – own observations'. *Polish Annals of Medicine*, 17 (1), pp. 44-53.
- Zhang, R. *et al.* (1999) 'Shape from Shading: A Survey'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21 690-706.
- Zhang, Z. (2000) 'A flexible new technique for camera calibration'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (11), pp. 1330-1334.
- Zheng, Y.-P. *et al.* (2016) 'A reliability and validity study for Scolioscan: a radiation-free scoliosis assessment system using 3D ultrasound imaging'. *Scoliosis and Spinal Disorders*, 11 (1), pp. 13.
- Zhou, G. *et al.* (2017) 'Automatic Measurement of Spine Curvature on 3-D Ultrasound Volume Projection Image With Phase Features'. *IEEE Transactions on Medical Imaging*, 36 (6), pp. 1250-1262.



## **Appendix**

## Appendix 1 Coding for load cell calibration for experiment analysis (Chapter 5)

```
from Phidget22.Phidget import *
from Phidget22.Devices.VoltageRatioInput import *
import time
import csv

def VoltageRatioChangeHandler_0(self, voltageRatio):
    return
def onVoltageRatioChange_0(self, voltageRatio):
    return

def main():
    voltageRatioInput0 = VoltageRatioInput()
    voltageRatioInput0.setDeviceSerialNumber(468981)
    voltageRatioInput0.setChannel(1)
    voltageRatioInput0.setOnVoltageRatioChangeHandler(onVoltageRatioChange_0)
    voltageRatioInput0.openWaitForAttachment(5000)
    voltageRatioInput0.setDataInterval(1000)
    print("81/0 Start getting VoltageRatio value: ")

    with open('test.csv', 'w', newline='') as f:
        writer = csv.writer(f, delimiter=' ')
        for row in range(0, 1000000000000000000):
            var1 = time.strftime('%S')
            var2 = str(round(time.time()*1000))
            var3 = voltageRatioInput0.getVoltageRatio()*1000000
            print(var1 + ':' + var2 + ':' + str('{:0.4f}'.format(var3)))
            fieldnames = ['var1', 'var2', 'var3']
            writer = csv.DictWriter(f, fieldnames=fieldnames)
            writer.writerow({'var1': var1, 'var2': var2, 'var3': var3})
        voltageRatioInput0.close()
    print("81/0 Stop getting VoltageRatio value")

main()
print("Recording CVS file successfully :)")
```

## Appendix 2 Coding for load cell calibration for GUI (Chapter 5)

```
import tkinter as tk
from PIL import Image, ImageTk
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,NavigationToolbar2Tk)
from matplotlib.backend_bases import key_press_handler
from matplotlib.figure import Figure
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import style
import csv
from tkinter import *
from tkinter import ttk
import increase_ch0
import increase_ch1
import increase_ch2
import increase_ch3
import increase_ch4
import increase_ch5
import decrease_ch0
import decrease_ch1
import decrease_ch2
import decrease_ch3
import decrease_ch4
import decrease_ch5

class Page(tk.Frame):
    def __init__(self, *args, **kwargs):
        tk.Frame.__init__(self, *args, **kwargs)
    def show(self):
        self.lift()
```

```

class Page1(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Load Cell Channel: 81/0", font = "Helvetica 16 bold")
        lb1.grid(row=0, column=0, padx = 5, pady = 5)
        lb2 = tk.Label(self, text="Increasing Mass (from 0 Kg to 40 Kg)", font = "Helvetica 10 bold")
        lb2.place(x=100,y=35)
        lb3 = tk.Label(self, text="Decreasing Mass (from 40 Kg to 0 Kg)", font = "Helvetica 10 bold")
        lb3.place(x=500,y=35)
#-----ch0 0 Kg increasing mass-----
    #Click to Record load cells
    def click_increase_ch0_0kg_record():
        increase_ch0.increase_ch0_0kg_record()
    #Click to calculate mean
    def click_increase_ch0_0kg_calmean():
        increase_ch0.increase_ch0_0kg_calmean()
    #Click to display mean
    def click_increase_ch0_0kg_showmean():
        with open("increase_ch0_0kg_mean.csv", newline="") as file:
            reader = csv.reader(FileZilla)
            r = 0
            for col in reader:
                c = 0
                for row in col:
                    label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                    label.place(x=290, y=62)
lb4 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb4.place(x=30,y=60)
btn5 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_0kg_record)
btn5.place(x=80,y=60)
btn6 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_0kg_calmean)
btn6.place(x=150,y=60)
btn7 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_0kg_showmean)
btn7.place(x=220,y=60)
lb8 = tk.Label(self, text="v/v")
lb8.place(x=360,y=60)
#-----ch0 40kg decreasing mass-----
    def click_decrease_ch0_40kg_record():
        decrease_ch0.decrease_ch0_40kg_record()
    def click_decrease_ch0_40kg_calmean():
        decrease_ch0.decrease_ch0_40kg_calmean()
    def click_decrease_ch0_40kg_showmean():
        with open("decrease_ch0_40kg_mean.csv", newline="") as file:
            reader = csv.reader(FileZilla)
            r = 0
            for col in reader:
                c = 0
                for row in col:
                    label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                    label.place(x=700, y=62)
lb9 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb9.place(x=440,y=60)
btn10 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_40kg_record)
btn10.place(x=490,y=60)
btn11 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_40kg_calmean)
btn11.place(x=560,y=60)
btn12 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_40kg_showmean)
btn12.place(x=630,y=60)
lb13 = tk.Label(self, text="v/v.")
lb13.place(x=770,y=60)
#-----ch0 5 Kg increasing mass-----
    #Click to Record load cells
    def click_increase_ch0_5kg_record():
        increase_ch0.increase_ch0_5kg_record()
    #Click to calculate mean
    def click_increase_ch0_5kg_calmean():
        increase_ch0.increase_ch0_5kg_calmean()
    #Click to display mean
    def click_increase_ch0_5kg_showmean():
        with open("increase_ch0_5kg_mean.csv", newline="") as file:
            reader = csv.reader(FileZilla)
            r = 0

```

```

    for col in reader:
        c = 0
        for row in col:
            label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
            label.place(x=290, y=102)
lb14 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb14.place(x=30, y=100)
btn15 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_5kg_record)
btn15.place(x=80, y=100)
btn16 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_5kg_calmean)
btn16.place(x=150, y=100)
btn17 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_5kg_showmean)
btn17.place(x=220, y=100)
lb18 = tk.Label(self, text="v/v")
lb18.place(x=360, y=100)
#-----ch0 35kg decreasing mass-----
def click_decrease_ch0_35kg_record():
    decrease_ch0.decrease_ch0_35kg_record()
def click_decrease_ch0_35kg_calmean():
    decrease_ch0.decrease_ch0_35kg_calmean()
def click_decrease_ch0_35kg_showmean():
    with open("decrease_ch0_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=102)
lb19 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb19.place(x=440, y=100)
btn20 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_35kg_record)
btn20.place(x=490, y=100)
btn21 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_35kg_calmean)
btn21.place(x=560, y=100)
btn22 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_35kg_showmean)
btn22.place(x=630, y=100)
lb23 = tk.Label(self, text="v/v.")
lb23.place(x=770, y=100)
#-----ch0 10 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch0_10kg_record():
    increase_ch0.increase_ch0_10kg_record()
#Click to calculate mean
def click_increase_ch0_10kg_calmean():
    increase_ch0.increase_ch0_10kg_calmean()
#Click to display mean
def click_increase_ch0_10kg_showmean():
    with open("increase_ch0_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=142)
lb24 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb24.place(x=30, y=140)
btn25 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_10kg_record)
btn25.place(x=80, y=140)
btn26 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_10kg_calmean)
btn26.place(x=150, y=140)
btn27 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_10kg_showmean)
btn27.place(x=220, y=140)
lb28 = tk.Label(self, text="v/v")
lb28.place(x=360, y=140)
#-----ch0 30kg decreasing mass-----
def click_decrease_ch0_30kg_record():
    decrease_ch0.decrease_ch0_30kg_record()
def click_decrease_ch0_30kg_calmean():
    decrease_ch0.decrease_ch0_30kg_calmean()
def click_decrease_ch0_30kg_showmean():

```

```

with open("decrease_ch0_30kg_mean.csv", newline="") as file:
    reader = csv.reader(FileZilla)
    r = 0
    for col in reader:
        c = 0
        for row in col:
            label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
            label.place(x=700, y=142)
lb29 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb29.place(x=440, y=140)
btn30 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_30kg_record)
btn30.place(x=490, y=140)
btn31 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_30kg_calmean)
btn31.place(x=560, y=140)
btn32 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_30kg_showmean)
btn32.place(x=630, y=140)
lb33 = tk.Label(self, text="v/v.")
lb33.place(x=770, y=140)
#-----ch0 15 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch0_15kg_record():
    increase_ch0.increase_ch0_15kg_record()
#Click to calculate mean
def click_increase_ch0_15kg_calmean():
    increase_ch0.increase_ch0_15kg_calmean()
#Click to display mean
def click_increase_ch0_15kg_showmean():
    with open("increase_ch0_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=182)
lb34 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb34.place(x=30, y=180)
btn35 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_15kg_record)
btn35.place(x=80, y=180)
btn36 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_15kg_calmean)
btn36.place(x=150, y=180)
btn37 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_15kg_showmean)
btn37.place(x=220, y=180)
lb38 = tk.Label(self, text="v/v")
lb38.place(x=360, y=180)
#-----ch0 25kg decreasing mass-----
def click_decrease_ch0_25kg_record():
    decrease_ch0.decrease_ch0_25kg_record()
def click_decrease_ch0_25kg_calmean():
    decrease_ch0.decrease_ch0_25kg_calmean()
def click_decrease_ch0_25kg_showmean():
    with open("decrease_ch0_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=182)
lb39 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb39.place(x=440, y=180)
btn40 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_25kg_record)
btn40.place(x=490, y=180)
btn41 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_25kg_calmean)
btn41.place(x=560, y=180)
btn42 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_25kg_showmean)
btn42.place(x=630, y=180)
lb43 = tk.Label(self, text="v/v.")
lb43.place(x=770, y=180)
#-----ch0 20 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch0_20kg_record():

```

```

        increase_ch0.increase_ch0_20kg_record()
    #Click to calculate mean
    def click_increase_ch0_20kg_calmean():
        increase_ch0.increase_ch0_20kg_calmean()
    #Click to display mean
    def click_increase_ch0_20kg_showmean():
        with open("increase_ch0_20kg_mean.csv", newline="") as file:
            reader = csv.reader(FileZilla)
            r = 0
            for col in reader:
                c = 0
                for row in col:
                    label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                    label.place(x=290, y=222)
lb44 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb44.place(x=30, y=220)
btn45 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_20kg_record)
btn45.place(x=80, y=220)
btn46 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_20kg_calmean)
btn46.place(x=150, y=220)
btn47 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_20kg_showmean)
btn47.place(x=220, y=220)
lb48 = tk.Label(self, text="v/v")
lb48.place(x=360, y=220)
#-----ch0 20kg decreasing mass-----
    def click_decrease_ch0_20kg_record():
        decrease_ch0.decrease_ch0_20kg_record()
    def click_decrease_ch0_20kg_calmean():
        decrease_ch0.decrease_ch0_20kg_calmean()
    def click_decrease_ch0_20kg_showmean():
        with open("decrease_ch0_20kg_mean.csv", newline="") as file:
            reader = csv.reader(FileZilla)
            r = 0
            for col in reader:
                c = 0
                for row in col:
                    label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                    label.place(x=700, y=222)
lb49 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb49.place(x=440, y=220)
btn50 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_20kg_record)
btn50.place(x=490, y=220)
btn51 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_20kg_calmean)
btn51.place(x=560, y=220)
btn52 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_20kg_showmean)
btn52.place(x=630, y=220)
lb53 = tk.Label(self, text="v/v.")
lb53.place(x=770, y=220)
#-----ch0 25 Kg increasing mass-----
    #Click to Record load cells
    def click_increase_ch0_25kg_record():
        increase_ch0.increase_ch0_25kg_record()
    #Click to calculate mean
    def click_increase_ch0_25kg_calmean():
        increase_ch0.increase_ch0_25kg_calmean()
    #Click to display mean
    def click_increase_ch0_25kg_showmean():
        with open("increase_ch0_25kg_mean.csv", newline="") as file:
            reader = csv.reader(FileZilla)
            r = 0
            for col in reader:
                c = 0
                for row in col:
                    label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                    label.place(x=290, y=262)
lb54 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb54.place(x=30, y=260)
btn55 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_25kg_record)
btn55.place(x=80, y=260)
btn56 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_25kg_calmean)
btn56.place(x=150, y=260)
btn57 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_25kg_showmean)

```

```

btn57.place(x=220,y=260)
lb58 = tk.Label(self, text="v/v")
lb58.place(x=360,y=260)
#-----ch0 15kg decreasing mass-----
def click_decrease_ch0_15kg_record():
    decrease_ch0.decrease_ch0_15kg_record()
def click_decrease_ch0_15kg_calmean():
    decrease_ch0.decrease_ch0_15kg_calmean()
def click_decrease_ch0_15kg_showmean():
    with open("decrease_ch0_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=262)
lb59 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb59.place(x=440,y=260)
btn60 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_15kg_record)
btn60.place(x=490,y=260)
btn61 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_15kg_calmean)
btn61.place(x=560,y=260)
btn62 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_15kg_showmean)
btn62.place(x=630,y=260)
lb63 = tk.Label(self, text="v/v.")
lb63.place(x=770,y=260)
#-----ch0 30 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch0_30kg_record():
    increase_ch0.increase_ch0_30kg_record()
#Click to calculate mean
def click_increase_ch0_30kg_calmean():
    increase_ch0.increase_ch0_30kg_calmean()
#Click to display mean
def click_increase_ch0_30kg_showmean():
    with open("increase_ch0_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=302)
lb64 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb64.place(x=30,y=300)
btn65 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_30kg_record)
btn65.place(x=80,y=300)
btn66 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_30kg_calmean)
btn66.place(x=150,y=300)
btn67 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_30kg_showmean)
btn67.place(x=220,y=300)
lb68 = tk.Label(self, text="v/v")
lb68.place(x=360,y=300)
#-----ch0 10kg decreasing mass-----
def click_decrease_ch0_10kg_record():
    decrease_ch0.decrease_ch0_10kg_record()
def click_decrease_ch0_10kg_calmean():
    decrease_ch0.decrease_ch0_10kg_calmean()
def click_decrease_ch0_10kg_showmean():
    with open("decrease_ch0_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=302)
lb69 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb69.place(x=440,y=300)
btn70 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_10kg_record)
btn70.place(x=490,y=300)

```

```

btn71 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_10kg_calmean)
btn71.place(x=560,y=300)
btn72 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_10kg_showmean)
btn72.place(x=630,y=300)
lb73 = tk.Label(self, text="v/v.")
lb73.place(x=770,y=300)
#-----ch0 35 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch0_35kg_record():
    increase_ch0.increase_ch0_35kg_record()
#Click to calculate mean
def click_increase_ch0_35kg_calmean():
    increase_ch0.increase_ch0_35kg_calmean()
#Click to display mean
def click_increase_ch0_35kg_showmean():
    with open("increase_ch0_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=342)
lb74 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb74.place(x=30,y=340)
btn75 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_35kg_record)
btn75.place(x=80,y=340)
btn76 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_35kg_calmean)
btn76.place(x=150,y=340)
btn77 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_35kg_showmean)
btn77.place(x=220,y=340)
lb78 = tk.Label(self, text="v/v")
lb78.place(x=360,y=340)
#-----ch0 5kg decreasing mass-----
def click_decrease_ch0_5kg_record():
    decrease_ch0.decrease_ch0_5kg_record()
def click_decrease_ch0_5kg_calmean():
    decrease_ch0.decrease_ch0_5kg_calmean()
def click_decrease_ch0_5kg_showmean():
    with open("decrease_ch0_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=342)
lb79 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb79.place(x=440,y=340)
btn80 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_5kg_record)
btn80.place(x=490,y=340)
btn81 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_5kg_calmean)
btn81.place(x=560,y=340)
btn82 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_5kg_showmean)
btn82.place(x=630,y=340)
lb83 = tk.Label(self, text="v/v.")
lb83.place(x=770,y=340)
#-----ch0 40 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch0_40kg_record():
    increase_ch0.increase_ch0_40kg_record()
#Click to calculate mean
def click_increase_ch0_40kg_calmean():
    increase_ch0.increase_ch0_40kg_calmean()
#Click to display mean
def click_increase_ch0_40kg_showmean():
    with open("increase_ch0_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:

```



```

        label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
        label.place(x=290, y=382)
lb84 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb84.place(x=30, y=380)
btn85 = tk.Button(self, text="Record", width = 8, command = click_increase_ch0_40kg_record)
btn85.place(x=80, y=380)
btn86 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch0_40kg_calmean)
btn86.place(x=150, y=380)
btn87 = tk.Button(self, text="Display", width = 8, command= click_increase_ch0_40kg_showmean)
btn87.place(x=220, y=380)
lb88 = tk.Label(self, text="v/v")
lb88.place(x=360, y=380)
#-----ch0 0kg decreasing mass-----
def click_decrease_ch0_0kg_record():
    decrease_ch0.decrease_ch0_0kg_record()
def click_decrease_ch0_0kg_calmean():
    decrease_ch0.decrease_ch0_0kg_calmean()
def click_decrease_ch0_0kg_showmean():
    with open("decrease_ch0_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=382)
lb89 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb89.place(x=440, y=380)
btn90 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch0_0kg_record)
btn90.place(x=490, y=380)
btn91 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch0_0kg_calmean)
btn91.place(x=560, y=380)
btn92 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch0_0kg_showmean)
btn92.place(x=630, y=380)
lb93 = tk.Label(self, text="v/v.")
lb93.place(x=770, y=380)
class Page2(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Load Cell Channel: 81/1", font = "Helvetica 16 bold")
        lb1.grid(row=0, column=0, padx = 5, pady = 5)
        lb2 = tk.Label(self, text="Increasing Mass (from 0 Kg to 40 Kg)", font = "Helvetica 10 bold")
        lb2.place(x=100, y=35)
        lb3 = tk.Label(self, text="Decreasing Mass (from 40 Kg to 0 Kg)", font = "Helvetica 10 bold")
        lb3.place(x=500, y=35)
#-----ch1 0 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_0kg_record():
    increase_ch1.increase_ch1_0kg_record()
#Click to calculate mean
def click_increase_ch1_0kg_calmean():
    increase_ch1.increase_ch1_0kg_calmean()
#Click to display mean
def click_increase_ch1_0kg_showmean():
    with open("increase_ch1_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=62)
lb4 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb4.place(x=30, y=60)
btn5 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_0kg_record)
btn5.place(x=80, y=60)
btn6 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_0kg_calmean)
btn6.place(x=150, y=60)
btn7 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_0kg_showmean)
btn7.place(x=220, y=60)
lb8 = tk.Label(self, text="v/v")
lb8.place(x=360, y=60)

```

```

#-----ch1 40kg decreasing mass-----
def click_decrease_ch1_40kg_record():
    decrease_ch1.decrease_ch1_40kg_record()
def click_decrease_ch1_40kg_calmean():
    decrease_ch1.decrease_ch1_40kg_calmean()
def click_decrease_ch1_40kg_showmean():
    with open("decrease_ch1_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=62)
lb9 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb9.place(x=440, y=60)
btn10 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_40kg_record)
btn10.place(x=490, y=60)
btn11 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_40kg_calmean)
btn11.place(x=560, y=60)
btn12 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_40kg_showmean)
btn12.place(x=630, y=60)
lb13 = tk.Label(self, text="v/v.")
lb13.place(x=770, y=60)
#-----ch1 5 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_5kg_record():
    increase_ch1.increase_ch1_5kg_record()
#Click to calculate mean
def click_increase_ch1_5kg_calmean():
    increase_ch1.increase_ch1_5kg_calmean()
#Click to display mean
def click_increase_ch1_5kg_showmean():
    with open("increase_ch1_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=102)
lb14 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb14.place(x=30, y=100)
btn15 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_5kg_record)
btn15.place(x=80, y=100)
btn16 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_5kg_calmean)
btn16.place(x=150, y=100)
btn17 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_5kg_showmean)
btn17.place(x=220, y=100)
lb18 = tk.Label(self, text="v/v")
lb18.place(x=360, y=100)
#-----ch1 35kg decreasing mass-----
def click_decrease_ch1_35kg_record():
    decrease_ch1.decrease_ch1_35kg_record()
def click_decrease_ch1_35kg_calmean():
    decrease_ch1.decrease_ch1_35kg_calmean()
def click_decrease_ch1_35kg_showmean():
    with open("decrease_ch1_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=102)
lb19 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb19.place(x=440, y=100)
btn20 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_35kg_record)
btn20.place(x=490, y=100)
btn21 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_35kg_calmean)
btn21.place(x=560, y=100)
btn22 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_35kg_showmean)

```

```

btn22.place(x=630,y=100)
lb23 = tk.Label(self, text="v/v.")
lb23.place(x=770,y=100)
#-----ch1 10 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_10kg_record():
    increase_ch1.increase_ch1_10kg_record()
#Click to calculate mean
def click_increase_ch1_10kg_calmean():
    increase_ch1.increase_ch1_10kg_calmean()
#Click to display mean
def click_increase_ch1_10kg_showmean():
    with open("increase_ch1_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=142)
lb24 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb24.place(x=30,y=140)
btn25 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_10kg_record)
btn25.place(x=80,y=140)
btn26 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_10kg_calmean)
btn26.place(x=150,y=140)
btn27 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_10kg_showmean)
btn27.place(x=220,y=140)
lb28 = tk.Label(self, text="v/v")
lb28.place(x=360,y=140)
#-----ch1 30kg decreasing mass-----
def click_decrease_ch1_30kg_record():
    decrease_ch1.decrease_ch1_30kg_record()
def click_decrease_ch1_30kg_calmean():
    decrease_ch1.decrease_ch1_30kg_calmean()
def click_decrease_ch1_30kg_showmean():
    with open("decrease_ch1_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=142)
lb29 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb29.place(x=440,y=140)
btn30 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_30kg_record)
btn30.place(x=490,y=140)
btn31 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_30kg_calmean)
btn31.place(x=560,y=140)
btn32 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_30kg_showmean)
btn32.place(x=630,y=140)
lb33 = tk.Label(self, text="v/v.")
lb33.place(x=770,y=140)
#-----ch1 15 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_15kg_record():
    increase_ch1.increase_ch1_15kg_record()
#Click to calculate mean
def click_increase_ch1_15kg_calmean():
    increase_ch1.increase_ch1_15kg_calmean()
#Click to display mean
def click_increase_ch1_15kg_showmean():
    with open("increase_ch1_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=182)
lb34 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")

```

```

lb34.place(x=30,y=180)
btn35 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_15kg_record)
btn35.place(x=80,y=180)
btn36 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_15kg_calmean)
btn36.place(x=150,y=180)
btn37 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_15kg_showmean)
btn37.place(x=220,y=180)
lb38 = tk.Label(self, text="v/v")
lb38.place(x=360,y=180)
#-----ch1 25kg decreasing mass-----
def click_decrease_ch1_25kg_record():
    decrease_ch1.decrease_ch1_25kg_record()
def click_decrease_ch1_25kg_calmean():
    decrease_ch1.decrease_ch1_25kg_calmean()
def click_decrease_ch1_25kg_showmean():
    with open("decrease_ch1_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=182)
lb39 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb39.place(x=440,y=180)
btn40 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_25kg_record)
btn40.place(x=490,y=180)
btn41 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_25kg_calmean)
btn41.place(x=560,y=180)
btn42 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_25kg_showmean)
btn42.place(x=630,y=180)
lb43 = tk.Label(self, text="v/v.")
lb43.place(x=770,y=180)
#-----ch1 20 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_20kg_record():
    increase_ch1.increase_ch1_20kg_record()
#Click to calculate mean
def click_increase_ch1_20kg_calmean():
    increase_ch1.increase_ch1_20kg_calmean()
#Click to display mean
def click_increase_ch1_20kg_showmean():
    with open("increase_ch1_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=222)
lb44 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb44.place(x=30,y=220)
btn45 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_20kg_record)
btn45.place(x=80,y=220)
btn46 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_20kg_calmean)
btn46.place(x=150,y=220)
btn47 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_20kg_showmean)
btn47.place(x=220,y=220)
lb48 = tk.Label(self, text="v/v")
lb48.place(x=360,y=220)
#-----ch1 20kg decreasing mass-----
def click_decrease_ch1_20kg_record():
    decrease_ch1.decrease_ch1_20kg_record()
def click_decrease_ch1_20kg_calmean():
    decrease_ch1.decrease_ch1_20kg_calmean()
def click_decrease_ch1_20kg_showmean():
    with open("decrease_ch1_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:

```

```

        label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
        label.place(x=700, y=222)
lb49 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb49.place(x=440, y=220)
btn50 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_20kg_record)
btn50.place(x=490, y=220)
btn51 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_20kg_calmean)
btn51.place(x=560, y=220)
btn52 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_20kg_showmean)
btn52.place(x=630, y=220)
lb53 = tk.Label(self, text="v/v.")
lb53.place(x=770, y=220)
#-----ch1 25 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_25kg_record():
    increase_ch1.increase_ch1_25kg_record()
#Click to calculate mean
def click_increase_ch1_25kg_calmean():
    increase_ch1.increase_ch1_25kg_calmean()
#Click to display mean
def click_increase_ch1_25kg_showmean():
    with open("increase_ch1_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=262)
lb54 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb54.place(x=30, y=260)
btn55 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_25kg_record)
btn55.place(x=80, y=260)
btn56 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_25kg_calmean)
btn56.place(x=150, y=260)
btn57 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_25kg_showmean)
btn57.place(x=220, y=260)
lb58 = tk.Label(self, text="v/v")
lb58.place(x=360, y=260)
#-----ch1 15kg decreasing mass-----
def click_decrease_ch1_15kg_record():
    decrease_ch1.decrease_ch1_15kg_record()
def click_decrease_ch1_15kg_calmean():
    decrease_ch1.decrease_ch1_15kg_calmean()
def click_decrease_ch1_15kg_showmean():
    with open("decrease_ch1_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=262)
lb59 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb59.place(x=440, y=260)
btn60 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_15kg_record)
btn60.place(x=490, y=260)
btn61 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_15kg_calmean)
btn61.place(x=560, y=260)
btn62 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_15kg_showmean)
btn62.place(x=630, y=260)
lb63 = tk.Label(self, text="v/v.")
lb63.place(x=770, y=260)
#-----ch1 30 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_30kg_record():
    increase_ch1.increase_ch1_30kg_record()
#Click to calculate mean
def click_increase_ch1_30kg_calmean():
    increase_ch1.increase_ch1_30kg_calmean()
#Click to display mean
def click_increase_ch1_30kg_showmean():

```

```

with open("increase_ch1_30kg_mean.csv", newline="") as file:
    reader = csv.reader(FileZilla)
    r = 0
    for col in reader:
        c = 0
        for row in col:
            label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
            label.place(x=290, y=302)
lb64 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb64.place(x=30, y=300)
btn65 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_30kg_record)
btn65.place(x=80, y=300)
btn66 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_30kg_calmean)
btn66.place(x=150, y=300)
btn67 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_30kg_showmean)
btn67.place(x=220, y=300)
lb68 = tk.Label(self, text="v/v")
lb68.place(x=360, y=300)
#-----ch1 10kg decreasing mass-----
def click_decrease_ch1_10kg_record():
    decrease_ch1.decrease_ch1_10kg_record()
def click_decrease_ch1_10kg_calmean():
    decrease_ch1.decrease_ch1_10kg_calmean()
def click_decrease_ch1_10kg_showmean():
    with open("decrease_ch1_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=302)
lb69 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb69.place(x=440, y=300)
btn70 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_10kg_record)
btn70.place(x=490, y=300)
btn71 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_10kg_calmean)
btn71.place(x=560, y=300)
btn72 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_10kg_showmean)
btn72.place(x=630, y=300)
lb73 = tk.Label(self, text="v/v.")
lb73.place(x=770, y=300)
#-----ch1 35 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_35kg_record():
    increase_ch1.increase_ch1_35kg_record()
#Click to calculate mean
def click_increase_ch1_35kg_calmean():
    increase_ch1.increase_ch1_35kg_calmean()
#Click to display mean
def click_increase_ch1_35kg_showmean():
    with open("increase_ch1_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=342)
lb74 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb74.place(x=30, y=340)
btn75 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_35kg_record)
btn75.place(x=80, y=340)
btn76 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_35kg_calmean)
btn76.place(x=150, y=340)
btn77 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_35kg_showmean)
btn77.place(x=220, y=340)
lb78 = tk.Label(self, text="v/v")
lb78.place(x=360, y=340)
#-----ch1 5kg decreasing mass-----
def click_decrease_ch1_5kg_record():
    decrease_ch1.decrease_ch1_5kg_record()

```

```

def click_decrease_ch1_5kg_calmean():
    decrease_ch1.decrease_ch1_5kg_calmean()
def click_decrease_ch1_5kg_showmean():
    with open("decrease_ch1_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=342)
lb79 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb79.place(x=440, y=340)
btn80 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_5kg_record)
btn80.place(x=490, y=340)
btn81 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_5kg_calmean)
btn81.place(x=560, y=340)
btn82 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_5kg_showmean)
btn82.place(x=630, y=340)
lb83 = tk.Label(self, text="v/v.")
lb83.place(x=770, y=340)
#-----ch1 40 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch1_40kg_record():
    increase_ch1.increase_ch1_40kg_record()
#Click to calculate mean
def click_increase_ch1_40kg_calmean():
    increase_ch1.increase_ch1_40kg_calmean()
#Click to display mean
def click_increase_ch1_40kg_showmean():
    with open("increase_ch1_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=382)
lb84 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb84.place(x=30, y=380)
btn85 = tk.Button(self, text="Record", width = 8, command = click_increase_ch1_40kg_record)
btn85.place(x=80, y=380)
btn86 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch1_40kg_calmean)
btn86.place(x=150, y=380)
btn87 = tk.Button(self, text="Display", width = 8, command= click_increase_ch1_40kg_showmean)
btn87.place(x=220, y=380)
lb88 = tk.Label(self, text="v/v")
lb88.place(x=360, y=380)
#-----ch1 0kg decreasing mass-----
def click_decrease_ch1_0kg_record():
    decrease_ch1.decrease_ch1_0kg_record()
def click_decrease_ch1_0kg_calmean():
    decrease_ch1.decrease_ch1_0kg_calmean()
def click_decrease_ch1_0kg_showmean():
    with open("decrease_ch1_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=382)
lb89 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb89.place(x=440, y=380)
btn90 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch1_0kg_record)
btn90.place(x=490, y=380)
btn91 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch1_0kg_calmean)
btn91.place(x=560, y=380)
btn92 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch1_0kg_showmean)
btn92.place(x=630, y=380)
lb93 = tk.Label(self, text="v/v.")
lb93.place(x=770, y=380)

```

```

class Page3(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Load Cell Channel: 81/2", font = "Helvetica 16 bold")
        lb1.grid(row=0, column=0, padx = 5, pady = 5)
        lb2 = tk.Label(self, text="Increasing Mass (from 0 Kg to 40 Kg)", font = "Helvetica 10 bold")
        lb2.place(x=100,y=35)
        lb3 = tk.Label(self, text="Decreasing Mass (from 40 Kg to 0 Kg)", font = "Helvetica 10 bold")
        lb3.place(x=500,y=35)

#-----ch1 0 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_0kg_record():
    increase_ch2.increase_ch2_0kg_record()
#Click to calculate mean
def click_increase_ch2_0kg_calmean():
    increase_ch2.increase_ch2_0kg_calmean()
#Click to display mean
def click_increase_ch2_0kg_showmean():
    with open("increase_ch2_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=62)
lb4 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb4.place(x=30,y=60)
btn5 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_0kg_record)
btn5.place(x=80,y=60)
btn6 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_0kg_calmean)
btn6.place(x=150,y=60)
btn7 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_0kg_showmean)
btn7.place(x=220,y=60)
lb8 = tk.Label(self, text="v/v")
lb8.place(x=360,y=60)

#-----ch2 40kg decreasing mass-----
def click_decrease_ch2_40kg_record():
    decrease_ch2.decrease_ch2_40kg_record()
def click_decrease_ch2_40kg_calmean():
    decrease_ch2.decrease_ch2_40kg_calmean()
def click_decrease_ch2_40kg_showmean():
    with open("decrease_ch2_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=62)
lb9 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb9.place(x=440,y=60)
btn10 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_40kg_record)
btn10.place(x=490,y=60)
btn11 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_40kg_calmean)
btn11.place(x=560,y=60)
btn12 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_40kg_showmean)
btn12.place(x=630,y=60)
lb13 = tk.Label(self, text="v/v.")
lb13.place(x=770,y=60)

#-----ch2 5 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_5kg_record():
    increase_ch2.increase_ch2_5kg_record()
#Click to calculate mean
def click_increase_ch2_5kg_calmean():
    increase_ch2.increase_ch2_5kg_calmean()
#Click to display mean
def click_increase_ch2_5kg_showmean():
    with open("increase_ch2_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)

```



```

r = 0
for col in reader:
    c = 0
    for row in col:
        label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
        label.place(x=290, y=102)
lb14 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb14.place(x=30, y=100)
btn15 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_5kg_record)
btn15.place(x=80, y=100)
btn16 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_5kg_calmean)
btn16.place(x=150, y=100)
btn17 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_5kg_showmean)
btn17.place(x=220, y=100)
lb18 = tk.Label(self, text="v/v")
lb18.place(x=360, y=100)
#-----ch2 35kg decreasing mass-----
def click_decrease_ch2_35kg_record():
    decrease_ch2.decrease_ch2_35kg_record()
def click_decrease_ch2_35kg_calmean():
    decrease_ch2.decrease_ch2_35kg_calmean()
def click_decrease_ch2_35kg_showmean():
    with open("decrease_ch2_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=102)
lb19 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb19.place(x=440, y=100)
btn20 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_35kg_record)
btn20.place(x=490, y=100)
btn21 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_35kg_calmean)
btn21.place(x=560, y=100)
btn22 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_35kg_showmean)
btn22.place(x=630, y=100)
lb23 = tk.Label(self, text="v/v.")
lb23.place(x=770, y=100)
#-----ch2 10 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_10kg_record():
    increase_ch2.increase_ch2_10kg_record()
#Click to calculate mean
def click_increase_ch2_10kg_calmean():
    increase_ch2.increase_ch2_10kg_calmean()
#Click to display mean
def click_increase_ch2_10kg_showmean():
    with open("increase_ch2_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=142)
lb24 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb24.place(x=30, y=140)
btn25 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_10kg_record)
btn25.place(x=80, y=140)
btn26 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_10kg_calmean)
btn26.place(x=150, y=140)
btn27 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_10kg_showmean)
btn27.place(x=220, y=140)
lb28 = tk.Label(self, text="v/v")
lb28.place(x=360, y=140)
#-----ch2 30kg decreasing mass-----
def click_decrease_ch2_30kg_record():
    decrease_ch2.decrease_ch2_30kg_record()
def click_decrease_ch2_30kg_calmean():
    decrease_ch2.decrease_ch2_30kg_calmean()

```

```

def click_decrease_ch2_30kg_showmean():
    with open("decrease_ch2_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=142)
lb29 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb29.place(x=440, y=140)
btn30 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_30kg_record)
btn30.place(x=490, y=140)
btn31 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_30kg_calmean)
btn31.place(x=560, y=140)
btn32 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_30kg_showmean)
btn32.place(x=630, y=140)
lb33 = tk.Label(self, text="v/v.")
lb33.place(x=770, y=140)
#-----ch2 15 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_15kg_record():
    increase_ch2.increase_ch2_15kg_record()
#Click to calculate mean
def click_increase_ch2_15kg_calmean():
    increase_ch2.increase_ch2_15kg_calmean()
#Click to display mean
def click_increase_ch2_15kg_showmean():
    with open("increase_ch2_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=182)
lb34 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb34.place(x=30, y=180)
btn35 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_15kg_record)
btn35.place(x=80, y=180)
btn36 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_15kg_calmean)
btn36.place(x=150, y=180)
btn37 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_15kg_showmean)
btn37.place(x=220, y=180)
lb38 = tk.Label(self, text="v/v")
lb38.place(x=360, y=180)
#-----ch2 25kg decreasing mass-----
def click_decrease_ch2_25kg_record():
    decrease_ch2.decrease_ch2_25kg_record()
def click_decrease_ch2_25kg_calmean():
    decrease_ch2.decrease_ch2_25kg_calmean()
def click_decrease_ch2_25kg_showmean():
    with open("decrease_ch2_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=182)
lb39 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb39.place(x=440, y=180)
btn40 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_25kg_record)
btn40.place(x=490, y=180)
btn41 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_25kg_calmean)
btn41.place(x=560, y=180)
btn42 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_25kg_showmean)
btn42.place(x=630, y=180)
lb43 = tk.Label(self, text="v/v.")
lb43.place(x=770, y=180)
#-----ch2 20 Kg increasing mass-----
#Click to Record load cells

```

```

def click_increase_ch2_20kg_record():
    increase_ch2.increase_ch2_20kg_record()
#Click to calculate mean
def click_increase_ch2_20kg_calmean():
    increase_ch2.increase_ch2_20kg_calmean()
#Click to display mean
def click_increase_ch2_20kg_showmean():
    with open("increase_ch2_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=222)
lb44 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb44.place(x=30, y=220)
btn45 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_20kg_record)
btn45.place(x=80, y=220)
btn46 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_20kg_calmean)
btn46.place(x=150, y=220)
btn47 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_20kg_showmean)
btn47.place(x=220, y=220)
lb48 = tk.Label(self, text="v/v")
lb48.place(x=360, y=220)
#-----ch2 20kg decreasing mass-----
def click_decrease_ch2_20kg_record():
    decrease_ch2.decrease_ch2_20kg_record()
def click_decrease_ch2_20kg_calmean():
    decrease_ch2.decrease_ch2_20kg_calmean()
def click_decrease_ch2_20kg_showmean():
    with open("decrease_ch2_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=222)
lb49 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb49.place(x=440, y=220)
btn50 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_20kg_record)
btn50.place(x=490, y=220)
btn51 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_20kg_calmean)
btn51.place(x=560, y=220)
btn52 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_20kg_showmean)
btn52.place(x=630, y=220)
lb53 = tk.Label(self, text="v/v.")
lb53.place(x=770, y=220)
#-----ch2 25 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_25kg_record():
    increase_ch2.increase_ch2_25kg_record()
#Click to calculate mean
def click_increase_ch2_25kg_calmean():
    increase_ch2.increase_ch2_25kg_calmean()
#Click to display mean
def click_increase_ch2_25kg_showmean():
    with open("increase_ch2_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=262)
lb54 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb54.place(x=30, y=260)
btn55 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_25kg_record)
btn55.place(x=80, y=260)
btn56 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_25kg_calmean)
btn56.place(x=150, y=260)

```

```

btn57 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_25kg_showmean)
btn57.place(x=220,y=260)
lb58 = tk.Label(self, text="v/v")
lb58.place(x=360,y=260)
#-----ch2 15kg decreasing mass-----
def click_decrease_ch2_15kg_record():
    decrease_ch2.decrease_ch2_15kg_record()
def click_decrease_ch2_15kg_calmean():
    decrease_ch2.decrease_ch2_15kg_calmean()
def click_decrease_ch2_15kg_showmean():
    with open("decrease_ch2_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=262)
lb59 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb59.place(x=440,y=260)
btn60 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_15kg_record)
btn60.place(x=490,y=260)
btn61 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_15kg_calmean)
btn61.place(x=560,y=260)
btn62 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_15kg_showmean)
btn62.place(x=630,y=260)
lb63 = tk.Label(self, text="v/v.")
lb63.place(x=770,y=260)
#-----ch2 30 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_30kg_record():
    increase_ch2.increase_ch2_30kg_record()
#Click to calculate mean
def click_increase_ch2_30kg_calmean():
    increase_ch2.increase_ch2_30kg_calmean()
#Click to display mean
def click_increase_ch2_30kg_showmean():
    with open("increase_ch2_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=302)
lb64 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb64.place(x=30,y=300)
btn65 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_30kg_record)
btn65.place(x=80,y=300)
btn66 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_30kg_calmean)
btn66.place(x=150,y=300)
btn67 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_30kg_showmean)
btn67.place(x=220,y=300)
lb68 = tk.Label(self, text="v/v")
lb68.place(x=360,y=300)
#-----ch2 10kg decreasing mass-----
def click_decrease_ch2_10kg_record():
    decrease_ch2.decrease_ch2_10kg_record()
def click_decrease_ch2_10kg_calmean():
    decrease_ch2.decrease_ch2_10kg_calmean()
def click_decrease_ch2_10kg_showmean():
    with open("decrease_ch2_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=302)
lb69 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb69.place(x=440,y=300)
btn70 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_10kg_record)

```

```

btn70.place(x=490,y=300)
btn71 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_10kg_calmean)
btn71.place(x=560,y=300)
btn72 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_10kg_showmean)
btn72.place(x=630,y=300)
lb73 = tk.Label(self, text="v/v.")
lb73.place(x=770,y=300)
#-----ch2 35 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_35kg_record():
    increase_ch2.increase_ch2_35kg_record()
#Click to calculate mean
def click_increase_ch2_35kg_calmean():
    increase_ch2.increase_ch2_35kg_calmean()
#Click to display mean
def click_increase_ch2_35kg_showmean():
    with open("increase_ch2_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=342)
lb74 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb74.place(x=30,y=340)
btn75 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_35kg_record)
btn75.place(x=80,y=340)
btn76 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_35kg_calmean)
btn76.place(x=150,y=340)
btn77 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_35kg_showmean)
btn77.place(x=220,y=340)
lb78 = tk.Label(self, text="v/v")
lb78.place(x=360,y=340)
#-----ch2 5kg decreasing mass-----
def click_decrease_ch2_5kg_record():
    decrease_ch2.decrease_ch2_5kg_record()
def click_decrease_ch2_5kg_calmean():
    decrease_ch2.decrease_ch2_5kg_calmean()
def click_decrease_ch2_5kg_showmean():
    with open("decrease_ch2_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=342)
lb79 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb79.place(x=440,y=340)
btn80 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_5kg_record)
btn80.place(x=490,y=340)
btn81 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_5kg_calmean)
btn81.place(x=560,y=340)
btn82 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_5kg_showmean)
btn82.place(x=630,y=340)
lb83 = tk.Label(self, text="v/v.")
lb83.place(x=770,y=340)
#-----ch2 40 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch2_40kg_record():
    increase_ch2.increase_ch2_40kg_record()
#Click to calculate mean
def click_increase_ch2_40kg_calmean():
    increase_ch2.increase_ch2_40kg_calmean()
#Click to display mean
def click_increase_ch2_40kg_showmean():
    with open("increase_ch2_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0

```

```

        for row in col:
            label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
            label.place(x=290, y=382)
lb84 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb84.place(x=30, y=380)
btn85 = tk.Button(self, text="Record", width = 8, command = click_increase_ch2_40kg_record)
btn85.place(x=80, y=380)
btn86 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch2_40kg_calmean)
btn86.place(x=150, y=380)
btn87 = tk.Button(self, text="Display", width = 8, command= click_increase_ch2_40kg_showmean)
btn87.place(x=220, y=380)
lb88 = tk.Label(self, text="v/v")
lb88.place(x=360, y=380)
#-----ch2 0kg decreasing mass-----
def click_decrease_ch2_0kg_record():
    decrease_ch2.decrease_ch2_0kg_record()
def click_decrease_ch2_0kg_calmean():
    decrease_ch2.decrease_ch2_0kg_calmean()
def click_decrease_ch2_0kg_showmean():
    with open("decrease_ch2_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=382)
lb89 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb89.place(x=440, y=380)
btn90 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch2_0kg_record)
btn90.place(x=490, y=380)
btn91 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch2_0kg_calmean)
btn91.place(x=560, y=380)
btn92 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch2_0kg_showmean)
btn92.place(x=630, y=380)
lb93 = tk.Label(self, text="v/v.")
lb93.place(x=770, y=380)

class Page4(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Load Cell Channel: 81/3", font = "Helvetica 16 bold")
        lb1.grid(row=0, column=0, padx = 5, pady = 5)
        lb2 = tk.Label(self, text="Increasing Mass (from 0 Kg to 40 Kg)", font = "Helvetica 10 bold")
        lb2.place(x=100, y=35)
        lb3 = tk.Label(self, text="Decreasing Mass (from 40 Kg to 0 Kg)", font = "Helvetica 10 bold")
        lb3.place(x=500, y=35)
#-----ch1 0 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_0kg_record():
    increase_ch3.increase_ch3_0kg_record()
#Click to calculate mean
def click_increase_ch3_0kg_calmean():
    increase_ch3.increase_ch3_0kg_calmean()
#Click to display mean
def click_increase_ch3_0kg_showmean():
    with open("increase_ch3_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=62)
lb4 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb4.place(x=30, y=60)
btn5 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_0kg_record)
btn5.place(x=80, y=60)
btn6 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_0kg_calmean)
btn6.place(x=150, y=60)
btn7 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_0kg_showmean)
btn7.place(x=220, y=60)

```

```

lb8 = tk.Label(self, text="v/v")
lb8.place(x=360,y=60)
#-----ch3 40kg decreasing mass-----
def click_decrease_ch3_40kg_record():
    decrease_ch3.decrease_ch3_40kg_record()
def click_decrease_ch3_40kg_calmean():
    decrease_ch3.decrease_ch3_40kg_calmean()
def click_decrease_ch3_40kg_showmean():
    with open("decrease_ch3_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=62)
lb9 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb9.place(x=440,y=60)
btn10 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_40kg_record)
btn10.place(x=490,y=60)
btn11 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_40kg_calmean)
btn11.place(x=560,y=60)
btn12 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_40kg_showmean)
btn12.place(x=630,y=60)
lb13 = tk.Label(self, text="v/v.")
lb13.place(x=770,y=60)
#-----ch3 5 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_5kg_record():
    increase_ch3.increase_ch3_5kg_record()
#Click to calculate mean
def click_increase_ch3_5kg_calmean():
    increase_ch3.increase_ch3_5kg_calmean()
#Click to display mean
def click_increase_ch3_5kg_showmean():
    with open("increase_ch3_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=102)
lb14 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb14.place(x=30,y=100)
btn15 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_5kg_record)
btn15.place(x=80,y=100)
btn16 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_5kg_calmean)
btn16.place(x=150,y=100)
btn17 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_5kg_showmean)
btn17.place(x=220,y=100)
lb18 = tk.Label(self, text="v/v")
lb18.place(x=360,y=100)
#-----ch3 35kg decreasing mass-----
def click_decrease_ch3_35kg_record():
    decrease_ch3.decrease_ch3_35kg_record()
def click_decrease_ch3_35kg_calmean():
    decrease_ch3.decrease_ch3_35kg_calmean()
def click_decrease_ch3_35kg_showmean():
    with open("decrease_ch3_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=102)
lb19 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb19.place(x=440,y=100)
btn20 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_35kg_record)
btn20.place(x=490,y=100)
btn21 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_35kg_calmean)

```

```

btn21.place(x=560,y=100)
btn22 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_35kg_showmean)
btn22.place(x=630,y=100)
lb23 = tk.Label(self, text="v/v.")
lb23.place(x=770,y=100)
#-----ch3 10 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_10kg_record():
    increase_ch3.increase_ch3_10kg_record()
#Click to calculate mean
def click_increase_ch3_10kg_calmean():
    increase_ch3.increase_ch3_10kg_calmean()
#Click to display mean
def click_increase_ch3_10kg_showmean():
    with open("increase_ch3_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=142)
lb24 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb24.place(x=30,y=140)
btn25 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_10kg_record)
btn25.place(x=80,y=140)
btn26 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_10kg_calmean)
btn26.place(x=150,y=140)
btn27 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_10kg_showmean)
btn27.place(x=220,y=140)
lb28 = tk.Label(self, text="v/v")
lb28.place(x=360,y=140)
#-----ch3 30kg decreasing mass-----
def click_decrease_ch3_30kg_record():
    decrease_ch3.decrease_ch3_30kg_record()
def click_decrease_ch3_30kg_calmean():
    decrease_ch3.decrease_ch3_30kg_calmean()
def click_decrease_ch3_30kg_showmean():
    with open("decrease_ch3_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=142)
lb29 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb29.place(x=440,y=140)
btn30 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_30kg_record)
btn30.place(x=490,y=140)
btn31 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_30kg_calmean)
btn31.place(x=560,y=140)
btn32 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_30kg_showmean)
btn32.place(x=630,y=140)
lb33 = tk.Label(self, text="v/v.")
lb33.place(x=770,y=140)
#-----ch3 15 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_15kg_record():
    increase_ch3.increase_ch3_15kg_record()
#Click to calculate mean
def click_increase_ch3_15kg_calmean():
    increase_ch3.increase_ch3_15kg_calmean()
#Click to display mean
def click_increase_ch3_15kg_showmean():
    with open("increase_ch3_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)

```



```

        label.place(x=290, y=182)
lb34 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb34.place(x=30,y=180)
btn35 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_15kg_record)
btn35.place(x=80,y=180)
btn36 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_15kg_calmean)
btn36.place(x=150,y=180)
btn37 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_15kg_showmean)
btn37.place(x=220,y=180)
lb38 = tk.Label(self, text="v/v")
lb38.place(x=360,y=180)
#-----ch3 25kg decreasing mass-----
def click_decrease_ch3_25kg_record():
    decrease_ch3.decrease_ch3_25kg_record()
def click_decrease_ch3_25kg_calmean():
    decrease_ch3.decrease_ch3_25kg_calmean()
def click_decrease_ch3_25kg_showmean():
    with open("decrease_ch3_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=182)
lb39 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb39.place(x=440,y=180)
btn40 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_25kg_record)
btn40.place(x=490,y=180)
btn41 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_25kg_calmean)
btn41.place(x=560,y=180)
btn42 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_25kg_showmean)
btn42.place(x=630,y=180)
lb43 = tk.Label(self, text="v/v.")
lb43.place(x=770,y=180)
#-----ch3 20 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_20kg_record():
    increase_ch3.increase_ch3_20kg_record()
#Click to calculate mean
def click_increase_ch3_20kg_calmean():
    increase_ch3.increase_ch3_20kg_calmean()
#Click to display mean
def click_increase_ch3_20kg_showmean():
    with open("increase_ch3_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=222)
lb44 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb44.place(x=30,y=220)
btn45 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_20kg_record)
btn45.place(x=80,y=220)
btn46 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_20kg_calmean)
btn46.place(x=150,y=220)
btn47 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_20kg_showmean)
btn47.place(x=220,y=220)
lb48 = tk.Label(self, text="v/v")
lb48.place(x=360,y=220)
#-----ch3 20kg decreasing mass-----
def click_decrease_ch3_20kg_record():
    decrease_ch3.decrease_ch3_20kg_record()
def click_decrease_ch3_20kg_calmean():
    decrease_ch3.decrease_ch3_20kg_calmean()
def click_decrease_ch3_20kg_showmean():
    with open("decrease_ch3_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:

```

```

c = 0
for row in col:
    label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
    label.place(x=700, y=222)
lb49 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb49.place(x=440, y=220)
btn50 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_20kg_record)
btn50.place(x=490, y=220)
btn51 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_20kg_calmean)
btn51.place(x=560, y=220)
btn52 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_20kg_showmean)
btn52.place(x=630, y=220)
lb53 = tk.Label(self, text="v/v.")
lb53.place(x=770, y=220)
#-----ch3 25 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_25kg_record():
    increase_ch3.increase_ch3_25kg_record()
#Click to calculate mean
def click_increase_ch3_25kg_calmean():
    increase_ch3.increase_ch3_25kg_calmean()
#Click to display mean
def click_increase_ch3_25kg_showmean():
    with open("increase_ch3_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=262)
lb54 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb54.place(x=30, y=260)
btn55 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_25kg_record)
btn55.place(x=80, y=260)
btn56 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_25kg_calmean)
btn56.place(x=150, y=260)
btn57 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_25kg_showmean)
btn57.place(x=220, y=260)
lb58 = tk.Label(self, text="v/v")
lb58.place(x=360, y=260)
#-----ch3 15kg decreasing mass-----
def click_decrease_ch3_15kg_record():
    decrease_ch3.decrease_ch3_15kg_record()
def click_decrease_ch3_15kg_calmean():
    decrease_ch3.decrease_ch3_15kg_calmean()
def click_decrease_ch3_15kg_showmean():
    with open("decrease_ch3_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=262)
lb59 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb59.place(x=440, y=260)
btn60 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_15kg_record)
btn60.place(x=490, y=260)
btn61 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_15kg_calmean)
btn61.place(x=560, y=260)
btn62 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_15kg_showmean)
btn62.place(x=630, y=260)
lb63 = tk.Label(self, text="v/v.")
lb63.place(x=770, y=260)
#-----ch3 30 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_30kg_record():
    increase_ch3.increase_ch3_30kg_record()
#Click to calculate mean
def click_increase_ch3_30kg_calmean():
    increase_ch3.increase_ch3_30kg_calmean()

```

```

#Click to display mean
def click_increase_ch3_30kg_showmean():
    with open("increase_ch3_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=302)
lb64 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb64.place(x=30, y=300)
btn65 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_30kg_record)
btn65.place(x=80, y=300)
btn66 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_30kg_calmean)
btn66.place(x=150, y=300)
btn67 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_30kg_showmean)
btn67.place(x=220, y=300)
lb68 = tk.Label(self, text="v/v")
lb68.place(x=360, y=300)
#-----ch3 10kg decreasing mass-----
def click_decrease_ch3_10kg_record():
    decrease_ch3.decrease_ch3_10kg_record()
def click_decrease_ch3_10kg_calmean():
    decrease_ch3.decrease_ch3_10kg_calmean()
def click_decrease_ch3_10kg_showmean():
    with open("decrease_ch3_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=302)
lb69 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb69.place(x=440, y=300)
btn70 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_10kg_record)
btn70.place(x=490, y=300)
btn71 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_10kg_calmean)
btn71.place(x=560, y=300)
btn72 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_10kg_showmean)
btn72.place(x=630, y=300)
lb73 = tk.Label(self, text="v/v.")
lb73.place(x=770, y=300)
#-----ch3 35 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_35kg_record():
    increase_ch3.increase_ch3_35kg_record()
#Click to calculate mean
def click_increase_ch3_35kg_calmean():
    increase_ch3.increase_ch3_35kg_calmean()
#Click to display mean
def click_increase_ch3_35kg_showmean():
    with open("increase_ch3_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=342)
lb74 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb74.place(x=30, y=340)
btn75 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_35kg_record)
btn75.place(x=80, y=340)
btn76 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_35kg_calmean)
btn76.place(x=150, y=340)
btn77 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_35kg_showmean)
btn77.place(x=220, y=340)
lb78 = tk.Label(self, text="v/v")
lb78.place(x=360, y=340)
#-----ch3 5kg decreasing mass-----

```

```

def click_decrease_ch3_5kg_record():
    decrease_ch3.decrease_ch3_5kg_record()
def click_decrease_ch3_5kg_calmean():
    decrease_ch3.decrease_ch3_5kg_calmean()
def click_decrease_ch3_5kg_showmean():
    with open("decrease_ch3_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=342)
lb79 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb79.place(x=440, y=340)
btn80 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_5kg_record)
btn80.place(x=490, y=340)
btn81 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_5kg_calmean)
btn81.place(x=560, y=340)
btn82 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_5kg_showmean)
btn82.place(x=630, y=340)
lb83 = tk.Label(self, text="v/v.")
lb83.place(x=770, y=340)
#-----ch3 40 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch3_40kg_record():
    increase_ch3.increase_ch3_40kg_record()
#Click to calculate mean
def click_increase_ch3_40kg_calmean():
    increase_ch3.increase_ch3_40kg_calmean()
#Click to display mean
def click_increase_ch3_40kg_showmean():
    with open("increase_ch3_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=382)
lb84 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb84.place(x=30, y=380)
btn85 = tk.Button(self, text="Record", width = 8, command = click_increase_ch3_40kg_record)
btn85.place(x=80, y=380)
btn86 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch3_40kg_calmean)
btn86.place(x=150, y=380)
btn87 = tk.Button(self, text="Display", width = 8, command= click_increase_ch3_40kg_showmean)
btn87.place(x=220, y=380)
lb88 = tk.Label(self, text="v/v")
lb88.place(x=360, y=380)
#-----ch3 0kg decreasing mass-----
def click_decrease_ch3_0kg_record():
    decrease_ch3.decrease_ch3_0kg_record()
def click_decrease_ch3_0kg_calmean():
    decrease_ch3.decrease_ch3_0kg_calmean()
def click_decrease_ch3_0kg_showmean():
    with open("decrease_ch3_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=382)
lb89 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb89.place(x=440, y=380)
btn90 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch3_0kg_record)
btn90.place(x=490, y=380)
btn91 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch3_0kg_calmean)
btn91.place(x=560, y=380)
btn92 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch3_0kg_showmean)
btn92.place(x=630, y=380)

```

```

lb93 = tk.Label(self, text="v/v.")
lb93.place(x=770,y=380)

class Page5(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Load Cell Channel: 18/0", font = "Helvetica 16 bold")
        lb1.grid(row=0, column=0, padx = 5, pady = 5)
        lb2 = tk.Label(self, text="Increasing Mass (from 0 Kg to 40 Kg)", font = "Helvetica 10 bold")
        lb2.place(x=100,y=35)
        lb3 = tk.Label(self, text="Decreasing Mass (from 40 Kg to 0 Kg)", font = "Helvetica 10 bold")
        lb3.place(x=500,y=35)
#-----ch1 0 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_0kg_record():
    increase_ch4.increase_ch4_0kg_record()
#Click to calculate mean
def click_increase_ch4_0kg_calmean():
    increase_ch4.increase_ch4_0kg_calmean()
#Click to display mean
def click_increase_ch4_0kg_showmean():
    with open("increase_ch4_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=62)
lb4 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb4.place(x=30,y=60)
btn5 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_0kg_record)
btn5.place(x=80,y=60)
btn6 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_0kg_calmean)
btn6.place(x=150,y=60)
btn7 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_0kg_showmean)
btn7.place(x=220,y=60)
lb8 = tk.Label(self, text="v/v")
lb8.place(x=360,y=60)
#-----ch4 40kg decreasing mass-----
def click_decrease_ch4_40kg_record():
    decrease_ch4.decrease_ch4_40kg_record()
def click_decrease_ch4_40kg_calmean():
    decrease_ch4.decrease_ch4_40kg_calmean()
def click_decrease_ch4_40kg_showmean():
    with open("decrease_ch4_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=62)
lb9 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb9.place(x=440,y=60)
btn10 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_40kg_record)
btn10.place(x=490,y=60)
btn11 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_40kg_calmean)
btn11.place(x=560,y=60)
btn12 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_40kg_showmean)
btn12.place(x=630,y=60)
lb13 = tk.Label(self, text="v/v.")
lb13.place(x=770,y=60)
#-----ch4 5 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_5kg_record():
    increase_ch4.increase_ch4_5kg_record()
#Click to calculate mean
def click_increase_ch4_5kg_calmean():
    increase_ch4.increase_ch4_5kg_calmean()
#Click to display mean
def click_increase_ch4_5kg_showmean():

```

```

with open("increase_ch4_5kg_mean.csv", newline="") as file:
    reader = csv.reader(FileZilla)
    r = 0
    for col in reader:
        c = 0
        for row in col:
            label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
            label.place(x=290, y=102)
lb14 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb14.place(x=30, y=100)
btn15 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_5kg_record)
btn15.place(x=80, y=100)
btn16 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_5kg_calmean)
btn16.place(x=150, y=100)
btn17 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_5kg_showmean)
btn17.place(x=220, y=100)
lb18 = tk.Label(self, text="v/v")
lb18.place(x=360, y=100)
#-----ch4 35kg decreasing mass-----
def click_decrease_ch4_35kg_record():
    decrease_ch4.decrease_ch4_35kg_record()
def click_decrease_ch4_35kg_calmean():
    decrease_ch4.decrease_ch4_35kg_calmean()
def click_decrease_ch4_35kg_showmean():
    with open("decrease_ch4_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=102)
lb19 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb19.place(x=440, y=100)
btn20 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_35kg_record)
btn20.place(x=490, y=100)
btn21 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_35kg_calmean)
btn21.place(x=560, y=100)
btn22 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_35kg_showmean)
btn22.place(x=630, y=100)
lb23 = tk.Label(self, text="v/v.")
lb23.place(x=770, y=100)
#-----ch4 10 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_10kg_record():
    increase_ch4.increase_ch4_10kg_record()
#Click to calculate mean
def click_increase_ch4_10kg_calmean():
    increase_ch4.increase_ch4_10kg_calmean()
#Click to display mean
def click_increase_ch4_10kg_showmean():
    with open("increase_ch4_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=142)
lb24 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb24.place(x=30, y=140)
btn25 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_10kg_record)
btn25.place(x=80, y=140)
btn26 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_10kg_calmean)
btn26.place(x=150, y=140)
btn27 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_10kg_showmean)
btn27.place(x=220, y=140)
lb28 = tk.Label(self, text="v/v")
lb28.place(x=360, y=140)
#-----ch4 30kg decreasing mass-----
def click_decrease_ch4_30kg_record():
    decrease_ch4.decrease_ch4_30kg_record()

```

```

def click_decrease_ch4_30kg_calmean():
    decrease_ch4.decrease_ch4_30kg_calmean()
def click_decrease_ch4_30kg_showmean():
    with open("decrease_ch4_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=142)
lb29 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb29.place(x=440, y=140)
btn30 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_30kg_record)
btn30.place(x=490, y=140)
btn31 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_30kg_calmean)
btn31.place(x=560, y=140)
btn32 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_30kg_showmean)
btn32.place(x=630, y=140)
lb33 = tk.Label(self, text="v/v.")
lb33.place(x=770, y=140)
#-----ch4 15 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_15kg_record():
    increase_ch4.increase_ch4_15kg_record()
#Click to calculate mean
def click_increase_ch4_15kg_calmean():
    increase_ch4.increase_ch4_15kg_calmean()
#Click to display mean
def click_increase_ch4_15kg_showmean():
    with open("increase_ch4_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=182)
lb34 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb34.place(x=30, y=180)
btn35 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_15kg_record)
btn35.place(x=80, y=180)
btn36 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_15kg_calmean)
btn36.place(x=150, y=180)
btn37 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_15kg_showmean)
btn37.place(x=220, y=180)
lb38 = tk.Label(self, text="v/v")
lb38.place(x=360, y=180)
#-----ch4 25kg decreasing mass-----
def click_decrease_ch4_25kg_record():
    decrease_ch4.decrease_ch4_25kg_record()
def click_decrease_ch4_25kg_calmean():
    decrease_ch4.decrease_ch4_25kg_calmean()
def click_decrease_ch4_25kg_showmean():
    with open("decrease_ch4_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=182)
lb39 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb39.place(x=440, y=180)
btn40 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_25kg_record)
btn40.place(x=490, y=180)
btn41 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_25kg_calmean)
btn41.place(x=560, y=180)
btn42 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_25kg_showmean)
btn42.place(x=630, y=180)
lb43 = tk.Label(self, text="v/v.")
lb43.place(x=770, y=180)

```

```

#-----ch4 20 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_20kg_record():
    increase_ch4.increase_ch4_20kg_record()
#Click to calculate mean
def click_increase_ch4_20kg_calmean():
    increase_ch4.increase_ch4_20kg_calmean()
#Click to display mean
def click_increase_ch4_20kg_showmean():
    with open("increase_ch4_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=222)
lb44 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb44.place(x=30,y=220)
btn45 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_20kg_record)
btn45.place(x=80,y=220)
btn46 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_20kg_calmean)
btn46.place(x=150,y=220)
btn47 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_20kg_showmean)
btn47.place(x=220,y=220)
lb48 = tk.Label(self, text="v/v")
lb48.place(x=360,y=220)
#-----ch4 20kg decreasing mass-----
def click_decrease_ch4_20kg_record():
    decrease_ch4.decrease_ch4_20kg_record()
def click_decrease_ch4_20kg_calmean():
    decrease_ch4.decrease_ch4_20kg_calmean()
def click_decrease_ch4_20kg_showmean():
    with open("decrease_ch4_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=222)
lb49 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb49.place(x=440,y=220)
btn50 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_20kg_record)
btn50.place(x=490,y=220)
btn51 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_20kg_calmean)
btn51.place(x=560,y=220)
btn52 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_20kg_showmean)
btn52.place(x=630,y=220)
lb53 = tk.Label(self, text="v/v.")
lb53.place(x=770,y=220)
#-----ch4 25 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_25kg_record():
    increase_ch4.increase_ch4_25kg_record()
#Click to calculate mean
def click_increase_ch4_25kg_calmean():
    increase_ch4.increase_ch4_25kg_calmean()
#Click to display mean
def click_increase_ch4_25kg_showmean():
    with open("increase_ch4_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=262)
lb54 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb54.place(x=30,y=260)
btn55 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_25kg_record)
btn55.place(x=80,y=260)

```



```

btn56 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_25kg_calmean)
btn56.place(x=150,y=260)
btn57 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_25kg_showmean)
btn57.place(x=220,y=260)
lb58 = tk.Label(self, text="v/v")
lb58.place(x=360,y=260)
#-----ch4 15kg decreasing mass-----
def click_decrease_ch4_15kg_record():
    decrease_ch4.decrease_ch4_15kg_record()
def click_decrease_ch4_15kg_calmean():
    decrease_ch4.decrease_ch4_15kg_calmean()
def click_decrease_ch4_15kg_showmean():
    with open("decrease_ch4_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=262)
lb59 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb59.place(x=440,y=260)
btn60 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_15kg_record)
btn60.place(x=490,y=260)
btn61 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_15kg_calmean)
btn61.place(x=560,y=260)
btn62 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_15kg_showmean)
btn62.place(x=630,y=260)
lb63 = tk.Label(self, text="v/v.")
lb63.place(x=770,y=260)
#-----ch4 30 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_30kg_record():
    increase_ch4.increase_ch4_30kg_record()
#Click to calculate mean
def click_increase_ch4_30kg_calmean():
    increase_ch4.increase_ch4_30kg_calmean()
#Click to display mean
def click_increase_ch4_30kg_showmean():
    with open("increase_ch4_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=302)
lb64 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb64.place(x=30,y=300)
btn65 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_30kg_record)
btn65.place(x=80,y=300)
btn66 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_30kg_calmean)
btn66.place(x=150,y=300)
btn67 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_30kg_showmean)
btn67.place(x=220,y=300)
lb68 = tk.Label(self, text="v/v")
lb68.place(x=360,y=300)
#-----ch4 10kg decreasing mass-----
def click_decrease_ch4_10kg_record():
    decrease_ch4.decrease_ch4_10kg_record()
def click_decrease_ch4_10kg_calmean():
    decrease_ch4.decrease_ch4_10kg_calmean()
def click_decrease_ch4_10kg_showmean():
    with open("decrease_ch4_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=302)
lb69 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")

```

```

lb69.place(x=440,y=300)
btn70 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_10kg_record)
btn70.place(x=490,y=300)
btn71 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_10kg_calmean)
btn71.place(x=560,y=300)
btn72 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_10kg_showmean)
btn72.place(x=630,y=300)
lb73 = tk.Label(self, text="v/v.")
lb73.place(x=770,y=300)
#-----ch4 35 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_35kg_record():
    increase_ch4.increase_ch4_35kg_record()
#Click to calculate mean
def click_increase_ch4_35kg_calmean():
    increase_ch4.increase_ch4_35kg_calmean()
#Click to display mean
def click_increase_ch4_35kg_showmean():
    with open("increase_ch4_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=342)
lb74 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb74.place(x=30,y=340)
btn75 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_35kg_record)
btn75.place(x=80,y=340)
btn76 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_35kg_calmean)
btn76.place(x=150,y=340)
btn77 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_35kg_showmean)
btn77.place(x=220,y=340)
lb78 = tk.Label(self, text="v/v")
lb78.place(x=360,y=340)
#-----ch4 5kg decreasing mass-----
def click_decrease_ch4_5kg_record():
    decrease_ch4.decrease_ch4_5kg_record()
def click_decrease_ch4_5kg_calmean():
    decrease_ch4.decrease_ch4_5kg_calmean()
def click_decrease_ch4_5kg_showmean():
    with open("decrease_ch4_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=342)
lb79 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb79.place(x=440,y=340)
btn80 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_5kg_record)
btn80.place(x=490,y=340)
btn81 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_5kg_calmean)
btn81.place(x=560,y=340)
btn82 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_5kg_showmean)
btn82.place(x=630,y=340)
lb83 = tk.Label(self, text="v/v.")
lb83.place(x=770,y=340)
#-----ch4 40 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch4_40kg_record():
    increase_ch4.increase_ch4_40kg_record()
#Click to calculate mean
def click_increase_ch4_40kg_calmean():
    increase_ch4.increase_ch4_40kg_calmean()
#Click to display mean
def click_increase_ch4_40kg_showmean():
    with open("increase_ch4_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0

```

```

        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=382)
lb84 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb84.place(x=30, y=380)
btn85 = tk.Button(self, text="Record", width = 8, command = click_increase_ch4_40kg_record)
btn85.place(x=80, y=380)
btn86 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch4_40kg_calmean)
btn86.place(x=150, y=380)
btn87 = tk.Button(self, text="Display", width = 8, command= click_increase_ch4_40kg_showmean)
btn87.place(x=220, y=380)
lb88 = tk.Label(self, text="v/v")
lb88.place(x=360, y=380)
#-----ch4 0kg decreasing mass-----
def click_decrease_ch4_0kg_record():
    decrease_ch4.decrease_ch4_0kg_record()
def click_decrease_ch4_0kg_calmean():
    decrease_ch4.decrease_ch4_0kg_calmean()
def click_decrease_ch4_0kg_showmean():
    with open("decrease_ch4_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=382)
lb89 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb89.place(x=440, y=380)
btn90 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch4_0kg_record)
btn90.place(x=490, y=380)
btn91 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch4_0kg_calmean)
btn91.place(x=560, y=380)
btn92 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch4_0kg_showmean)
btn92.place(x=630, y=380)
lb93 = tk.Label(self, text="v/v.")
lb93.place(x=770, y=380)

class Page6(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Load Cell Channel: 18/1", font = "Helvetica 16 bold")
        lb1.grid(row=0, column=0, padx = 5, pady = 5)
        lb2 = tk.Label(self, text="Increasing Mass (from 0 Kg to 40 Kg)", font = "Helvetica 10 bold")
        lb2.place(x=100, y=35)
        lb3 = tk.Label(self, text="Decreasing Mass (from 40 Kg to 0 Kg)", font = "Helvetica 10 bold")
        lb3.place(x=500, y=35)
#-----ch1 0 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_0kg_record():
    increase_ch5.increase_ch5_0kg_record()
#Click to calculate mean
def click_increase_ch5_0kg_calmean():
    increase_ch5.increase_ch5_0kg_calmean()
#Click to display mean
def click_increase_ch5_0kg_showmean():
    with open("increase_ch5_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=62)
lb4 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb4.place(x=30, y=60)
btn5 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_0kg_record)
btn5.place(x=80, y=60)
btn6 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_0kg_calmean)
btn6.place(x=150, y=60)

```

```

btn7 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_0kg_showmean)
btn7.place(x=220,y=60)
lb8 = tk.Label(self, text="v/v")
lb8.place(x=360,y=60)
#-----ch5 40kg decreasing mass-----
def click_decrease_ch5_40kg_record():
    decrease_ch5.decrease_ch5_40kg_record()
def click_decrease_ch5_40kg_calmean():
    decrease_ch5.decrease_ch5_40kg_calmean()
def click_decrease_ch5_40kg_showmean():
    with open("decrease_ch5_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=62)
lb9 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb9.place(x=440,y=60)
btn10 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_40kg_record)
btn10.place(x=490,y=60)
btn11 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_40kg_calmean)
btn11.place(x=560,y=60)
btn12 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_40kg_showmean)
btn12.place(x=630,y=60)
lb13 = tk.Label(self, text="v/v.")
lb13.place(x=770,y=60)
#-----ch5 5 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_5kg_record():
    increase_ch5.increase_ch5_5kg_record()
#Click to calculate mean
def click_increase_ch5_5kg_calmean():
    increase_ch5.increase_ch5_5kg_calmean()
#Click to display mean
def click_increase_ch5_5kg_showmean():
    with open("increase_ch5_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=102)
lb14 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb14.place(x=30,y=100)
btn15 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_5kg_record)
btn15.place(x=80,y=100)
btn16 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_5kg_calmean)
btn16.place(x=150,y=100)
btn17 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_5kg_showmean)
btn17.place(x=220,y=100)
lb18 = tk.Label(self, text="v/v")
lb18.place(x=360,y=100)
#-----ch5 35kg decreasing mass-----
def click_decrease_ch5_35kg_record():
    decrease_ch5.decrease_ch5_35kg_record()
def click_decrease_ch5_35kg_calmean():
    decrease_ch5.decrease_ch5_35kg_calmean()
def click_decrease_ch5_35kg_showmean():
    with open("decrease_ch5_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=102)
lb19 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb19.place(x=440,y=100)
btn20 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_35kg_record)

```

```

btn20.place(x=490,y=100)
btn21 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_35kg_calmean)
btn21.place(x=560,y=100)
btn22 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_35kg_showmean)
btn22.place(x=630,y=100)
lb23 = tk.Label(self, text="v/v.")
lb23.place(x=770,y=100)
#-----ch5 10 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_10kg_record():
    increase_ch5.increase_ch5_10kg_record()
#Click to calculate mean
def click_increase_ch5_10kg_calmean():
    increase_ch5.increase_ch5_10kg_calmean()
#Click to display mean
def click_increase_ch5_10kg_showmean():
    with open("increase_ch5_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=290, y=142)
lb24 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb24.place(x=30,y=140)
btn25 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_10kg_record)
btn25.place(x=80,y=140)
btn26 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_10kg_calmean)
btn26.place(x=150,y=140)
btn27 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_10kg_showmean)
btn27.place(x=220,y=140)
lb28 = tk.Label(self, text="v/v")
lb28.place(x=360,y=140)
#-----ch5 30kg decreasing mass-----
def click_decrease_ch5_30kg_record():
    decrease_ch5.decrease_ch5_30kg_record()
def click_decrease_ch5_30kg_calmean():
    decrease_ch5.decrease_ch5_30kg_calmean()
def click_decrease_ch5_30kg_showmean():
    with open("decrease_ch5_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1,text=row, relief=tk.RIDGE)
                label.place(x=700, y=142)
lb29 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb29.place(x=440,y=140)
btn30 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_30kg_record)
btn30.place(x=490,y=140)
btn31 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_30kg_calmean)
btn31.place(x=560,y=140)
btn32 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_30kg_showmean)
btn32.place(x=630,y=140)
lb33 = tk.Label(self, text="v/v.")
lb33.place(x=770,y=140)
#-----ch5 15 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_15kg_record():
    increase_ch5.increase_ch5_15kg_record()
#Click to calculate mean
def click_increase_ch5_15kg_calmean():
    increase_ch5.increase_ch5_15kg_calmean()
#Click to display mean
def click_increase_ch5_15kg_showmean():
    with open("increase_ch5_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0

```

```

        for row in col:
            label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
            label.place(x=290, y=182)
lb34 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb34.place(x=30, y=180)
btn35 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_15kg_record)
btn35.place(x=80, y=180)
btn36 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_15kg_calmean)
btn36.place(x=150, y=180)
btn37 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_15kg_showmean)
btn37.place(x=220, y=180)
lb38 = tk.Label(self, text="v/v")
lb38.place(x=360, y=180)
#-----ch5 25kg decreasing mass-----
def click_decrease_ch5_25kg_record():
    decrease_ch5.decrease_ch5_25kg_record()
def click_decrease_ch5_25kg_calmean():
    decrease_ch5.decrease_ch5_25kg_calmean()
def click_decrease_ch5_25kg_showmean():
    with open("decrease_ch5_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=182)
lb39 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb39.place(x=440, y=180)
btn40 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_25kg_record)
btn40.place(x=490, y=180)
btn41 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_25kg_calmean)
btn41.place(x=560, y=180)
btn42 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_25kg_showmean)
btn42.place(x=630, y=180)
lb43 = tk.Label(self, text="v/v.")
lb43.place(x=770, y=180)
#-----ch5 20 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_20kg_record():
    increase_ch5.increase_ch5_20kg_record()
#Click to calculate mean
def click_increase_ch5_20kg_calmean():
    increase_ch5.increase_ch5_20kg_calmean()
#Click to display mean
def click_increase_ch5_20kg_showmean():
    with open("increase_ch5_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=222)
lb44 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb44.place(x=30, y=220)
btn45 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_20kg_record)
btn45.place(x=80, y=220)
btn46 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_20kg_calmean)
btn46.place(x=150, y=220)
btn47 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_20kg_showmean)
btn47.place(x=220, y=220)
lb48 = tk.Label(self, text="v/v")
lb48.place(x=360, y=220)
#-----ch5 20kg decreasing mass-----
def click_decrease_ch5_20kg_record():
    decrease_ch5.decrease_ch5_20kg_record()
def click_decrease_ch5_20kg_calmean():
    decrease_ch5.decrease_ch5_20kg_calmean()
def click_decrease_ch5_20kg_showmean():
    with open("decrease_ch5_20kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)

```

```

r = 0
for col in reader:
    c = 0
    for row in col:
        label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
        label.place(x=700, y=222)
lb49 = tk.Label(self, text="20 Kg", font = "Helvetica 10 bold")
lb49.place(x=440, y=220)
btn50 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_20kg_record)
btn50.place(x=490, y=220)
btn51 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_20kg_calmean)
btn51.place(x=560, y=220)
btn52 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_20kg_showmean)
btn52.place(x=630, y=220)
lb53 = tk.Label(self, text="v/v.")
lb53.place(x=770, y=220)
#-----ch5 25 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_25kg_record():
    increase_ch5.increase_ch5_25kg_record()
#Click to calculate mean
def click_increase_ch5_25kg_calmean():
    increase_ch5.increase_ch5_25kg_calmean()
#Click to display mean
def click_increase_ch5_25kg_showmean():
    with open("increase_ch5_25kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=262)
lb54 = tk.Label(self, text="25 Kg", font = "Helvetica 10 bold")
lb54.place(x=30, y=260)
btn55 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_25kg_record)
btn55.place(x=80, y=260)
btn56 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_25kg_calmean)
btn56.place(x=150, y=260)
btn57 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_25kg_showmean)
btn57.place(x=220, y=260)
lb58 = tk.Label(self, text="v/v")
lb58.place(x=360, y=260)
#-----ch5 15kg decreasing mass-----
def click_decrease_ch5_15kg_record():
    decrease_ch5.decrease_ch5_15kg_record()
def click_decrease_ch5_15kg_calmean():
    decrease_ch5.decrease_ch5_15kg_calmean()
def click_decrease_ch5_15kg_showmean():
    with open("decrease_ch5_15kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=262)
lb59 = tk.Label(self, text="15 Kg", font = "Helvetica 10 bold")
lb59.place(x=440, y=260)
btn60 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_15kg_record)
btn60.place(x=490, y=260)
btn61 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_15kg_calmean)
btn61.place(x=560, y=260)
btn62 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_15kg_showmean)
btn62.place(x=630, y=260)
lb63 = tk.Label(self, text="v/v.")
lb63.place(x=770, y=260)
#-----ch5 30 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_30kg_record():
    increase_ch5.increase_ch5_30kg_record()
#Click to calculate mean

```

```

def click_increase_ch5_30kg_calmean():
    increase_ch5.increase_ch5_30kg_calmean()
#Click to display mean
def click_increase_ch5_30kg_showmean():
    with open("increase_ch5_30kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=302)
lb64 = tk.Label(self, text="30 Kg", font = "Helvetica 10 bold")
lb64.place(x=30, y=300)
btn65 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_30kg_record)
btn65.place(x=80, y=300)
btn66 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_30kg_calmean)
btn66.place(x=150, y=300)
btn67 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_30kg_showmean)
btn67.place(x=220, y=300)
lb68 = tk.Label(self, text="v/v")
lb68.place(x=360, y=300)

#-----ch5 10kg decreasing mass-----
def click_decrease_ch5_10kg_record():
    decrease_ch5.decrease_ch5_10kg_record()
def click_decrease_ch5_10kg_calmean():
    decrease_ch5.decrease_ch5_10kg_calmean()
def click_decrease_ch5_10kg_showmean():
    with open("decrease_ch5_10kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=302)
lb69 = tk.Label(self, text="10 Kg", font = "Helvetica 10 bold")
lb69.place(x=440, y=300)
btn70 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_10kg_record)
btn70.place(x=490, y=300)
btn71 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_10kg_calmean)
btn71.place(x=560, y=300)
btn72 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_10kg_showmean)
btn72.place(x=630, y=300)
lb73 = tk.Label(self, text="v/v.")
lb73.place(x=770, y=300)

#-----ch5 35 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_35kg_record():
    increase_ch5.increase_ch5_35kg_record()
#Click to calculate mean
def click_increase_ch5_35kg_calmean():
    increase_ch5.increase_ch5_35kg_calmean()
#Click to display mean
def click_increase_ch5_35kg_showmean():
    with open("increase_ch5_35kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=342)
lb74 = tk.Label(self, text="35 Kg", font = "Helvetica 10 bold")
lb74.place(x=30, y=340)
btn75 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_35kg_record)
btn75.place(x=80, y=340)
btn76 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_35kg_calmean)
btn76.place(x=150, y=340)
btn77 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_35kg_showmean)
btn77.place(x=220, y=340)
lb78 = tk.Label(self, text="v/v")

```



```

lb78.place(x=360,y=340)
#-----ch5 5kg decreasing mass-----
def click_decrease_ch5_5kg_record():
    decrease_ch5.decrease_ch5_5kg_record()
def click_decrease_ch5_5kg_calmean():
    decrease_ch5.decrease_ch5_5kg_calmean()
def click_decrease_ch5_5kg_showmean():
    with open("decrease_ch5_5kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=342)
lb79 = tk.Label(self, text="5 Kg", font = "Helvetica 10 bold")
lb79.place(x=440,y=340)
btn80 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_5kg_record)
btn80.place(x=490,y=340)
btn81 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_5kg_calmean)
btn81.place(x=560,y=340)
btn82 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_5kg_showmean)
btn82.place(x=630,y=340)
lb83 = tk.Label(self, text="v/v.")
lb83.place(x=770,y=340)
#-----ch5 40 Kg increasing mass-----
#Click to Record load cells
def click_increase_ch5_40kg_record():
    increase_ch5.increase_ch5_40kg_record()
#Click to calculate mean
def click_increase_ch5_40kg_calmean():
    increase_ch5.increase_ch5_40kg_calmean()
#Click to display mean
def click_increase_ch5_40kg_showmean():
    with open("increase_ch5_40kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=290, y=382)
lb84 = tk.Label(self, text="40 Kg", font = "Helvetica 10 bold")
lb84.place(x=30,y=380)
btn85 = tk.Button(self, text="Record", width = 8, command = click_increase_ch5_40kg_record)
btn85.place(x=80,y=380)
btn86 = tk.Button(self, text="Calculate", width = 8, command= click_increase_ch5_40kg_calmean)
btn86.place(x=150,y=380)
btn87 = tk.Button(self, text="Display", width = 8, command= click_increase_ch5_40kg_showmean)
btn87.place(x=220,y=380)
lb88 = tk.Label(self, text="v/v")
lb88.place(x=360,y=380)
#-----ch5 0kg decreasing mass-----
def click_decrease_ch5_0kg_record():
    decrease_ch5.decrease_ch5_0kg_record()
def click_decrease_ch5_0kg_calmean():
    decrease_ch5.decrease_ch5_0kg_calmean()
def click_decrease_ch5_0kg_showmean():
    with open("decrease_ch5_0kg_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=8, height=1, text=row, relief=tk.RIDGE)
                label.place(x=700, y=382)
lb89 = tk.Label(self, text="0 Kg", font = "Helvetica 10 bold")
lb89.place(x=440,y=380)
btn90 = tk.Button(self, text="Record", width = 8, command = click_decrease_ch5_0kg_record)
btn90.place(x=490,y=380)
btn91 = tk.Button(self, text="Calculate", width = 8, command= click_decrease_ch5_0kg_calmean)
btn91.place(x=560,y=380)

```

```

btn92 = tk.Button(self, text="Display", width = 8, command= click_decrease_ch5_0kg_showmean)
btn92.place(x=630,y=380)
lb93 = tk.Label(self, text="v/v.")
lb93.place(x=770,y=380)

class MainView(tk.Frame):
    def __init__(self, *args, **kwargs):
        tk.Frame.__init__(self, *args, **kwargs)
        p1 = Page1(self)
        p2 = Page2(self)
        p3 = Page3(self)
        p4 = Page4(self)
        p5 = Page5(self)
        p6 = Page6(self)

        buttonframe = tk.Frame(self)
        container = tk.Frame(self)
        buttonframe.pack(side="top", fill="x", expand=False)
        container.pack(side="top", fill="both", expand=True)

        p1.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
        p2.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
        p3.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
        p4.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
        p5.place(in_=container, x=0, y=0, relwidth=1, relheight=1)
        p6.place(in_=container, x=0, y=0, relwidth=1, relheight=1)

        b1 = tk.Button(buttonframe, text="Channel 81/0", command=p1.lift)
        b2 = tk.Button(buttonframe, text="Channel 81/1", command=p2.lift)
        b3 = tk.Button(buttonframe, text="Channel 81/2", command=p3.lift)
        b4 = tk.Button(buttonframe, text="Channel 81/3", command=p4.lift)
        b5 = tk.Button(buttonframe, text="Channel 18/0", command=p5.lift)
        b6 = tk.Button(buttonframe, text="Channel 18/1", command=p6.lift)

        b1.pack(side="left", padx=3, pady =5)
        b2.pack(side="left", padx=3, pady =5)
        b3.pack(side="left", padx=3, pady =5)
        b4.pack(side="left", padx=3, pady =5)
        b5.pack(side="left", padx=3, pady =5)
        b6.pack(side="left", padx=3, pady =5)

        p1.show()

if __name__ == "__main__":
    root = tk.Tk()
    root.title('Load Cell Calibration Program')
    main = MainView(root)
    main.pack(side="top", fill="both", expand=True)
    root.wm_geometry("810x500")

    # Button for closing
    exit_button = Button(root, text="Exit", width = 5, font = "Helvetica 12 bold", command=root.destroy)
    exit_button.place(x=735,y=450)

    root.mainloop()

```

### Appendix 3 Coding for force measuring program for GUI (Chapter 5)

```

import tkinter as tk
from PIL import Image, ImageTk
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,NavigationToolbar2Tk)
from matplotlib.backend_bases import key_press_handler
from matplotlib.figure import Figure
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import style
import csv
import pyautogui
from tkinter import filedialog

```

```

from tkinter import *
from tkinter import ttk
import force_measuring

class Page(tk.Frame):
    def __init__(self, *args, **kwargs):
        tk.Frame.__init__(self, *args, **kwargs)
    def show(self):
        self.lift()

class Page1(Fernández-González et al.):
    def __init__(self, *args, **kwargs):
        Page.__init__(self, *args, **kwargs)
        lb1 = tk.Label(self, text="Force Measuring Program", font = "Helvetica 24 bold")
        lb1.place(x=650,y=20)

        self.img = tk.PhotoImage(file = 'test.png')
        self.trunk = tk.Label(self, image=self.img)
        self.trunk.place(x=290,y=70)
        self.img2 = tk.PhotoImage(file = 'logo.png')
        self.logo = tk.Label(self, image=self.img2)
        self.logo.place(x=1550,y=30)
        #---Record all sensors in the same time-----
        #Click to Record load cells
        def click_all_record():
            force_measuring.all_record()
        btn32 = tk.Button(self, text="All Record", width = 12, font = "Helvetica 12 bold", command = click_all_record)
        btn32.place(x=700,y=790)

#-----Left Axillary 81_0-----
#Click to Record load cells
def click_left_axilla_record():
    force_measuring.left_axilla_record()
#Click to calculate mean
def click_left_axilla_calmean():
    force_measuring.left_axilla_calmean()
#Click to display mean
def click_left_axilla_showmean():
    with open("left_axilla_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=6, height=1,text=row, relief=tk.RIDGE)
                label.place(x=226, y=374)
lb2 = tk.Label(self, text="Left Axillary Force", font = "Helvetica 10 bold")
lb2.place(x=20,y=348)
btn3 = tk.Button(self, text="Record", width = 6, command = click_left_axilla_record)
btn3.place(x=20,y=370)
btn4 = tk.Button(self, text="Calculate", width = 8, command= click_left_axilla_calmean)
btn4.place(x=83,y=370)
btn5 = tk.Button(self, text="Display", width = 6, command= click_left_axilla_showmean)
btn5.place(x=162,y=370)
lb6 = tk.Label(self, text="N.")
lb6.place(x=280,y=374)

#-----Right Axillary 81_1-----
#Click to Record load cells
def click_right_axilla_record():
    force_measuring.right_axilla_record()
#Click to calculate mean
def click_right_axilla_calmean():
    force_measuring.right_axilla_calmean()
#Click to display mean
def click_right_axilla_showmean():
    with open("right_axilla_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=6, height=1,text=row, relief=tk.RIDGE)
                label.place(x=735, y=374)
lb7 = tk.Label(self, text="Right Axillary Force", font = "Helvetica 10 bold")

```

```

lb7.place(x=815,y=345)
btn8 = tk.Button(self, text="Record", width = 6, command = click_right_axilla_record)
btn8.place(x=815, y=370)
btn9 = tk.Button(self, text="Calculate", width = 8, command= click_right_axilla_calmean)
btn9.place(x=878,y=370)
btn10 = tk.Button(self, text="Display", width = 6, command= click_right_axilla_showmean)
btn10.place(x=957,y=370)
lb11 = tk.Label(self, text="N.")
lb11.place(x=795,y=374)
#-----Left TL 81_2-----
#Click to Record load cells
def click_left_TL_record():
    force_measuring.left_TL_record()
#Click to calculate mean
def click_left_TL_calmean():
    force_measuring.left_TL_calmean()
#Click to display mean
def click_left_TL_showmean():
    with open("left_TL_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=6, height=1,text=row, relief=tk.RIDGE)
                label.place(x=226, y=514)
lb12 = tk.Label(self, text="Left Corrective Force", font = "Helvetica 10 bold")
lb12.place(x=20,y=490)
btn13 = tk.Button(self, text="Record", width = 6, command = click_left_TL_record)
btn13.place(x=20, y=510)
btn14 = tk.Button(self, text="Calculate", width = 8, command= click_left_TL_calmean)
btn14.place(x=83,y=510)
btn15 = tk.Button(self, text="Display", width = 6, command= click_left_TL_showmean)
btn15.place(x=162,y=510)
lb16 = tk.Label(self, text="N.")
lb16.place(x=280,y=514)
#-----Right TL 81_3-----
#Click to Record load cells
def click_right_TL_record():
    force_measuring.right_TL_record()
#Click to calculate mean
def click_right_TL_calmean():
    force_measuring.right_TL_calmean()
#Click to display mean
def click_right_TL_showmean():
    with open("right_TL_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=6, height=1,text=row, relief=tk.RIDGE)
                label.place(x=735, y=514)
lb17 = tk.Label(self, text="Right Corrective Force", font = "Helvetica 10 bold")
lb17.place(x=815, y=485)
btn18 = tk.Button(self, text="Record", width = 6, command = click_right_TL_record)
btn18.place(x=815, y=510)
btn19 = tk.Button(self, text="Calculate", width = 8, command= click_right_TL_calmean)
btn19.place(x=878,y=510)
btn20 = tk.Button(self, text="Display", width = 6, command= click_right_TL_showmean)
btn20.place(x=957,y=510)
lb21 = tk.Label(self, text="N.")
lb21.place(x=795,y=514)
#-----Left GT 18_0-----
#Click to Record load cells
def click_left_GT_record():
    force_measuring.left_GT_record()
#Click to calculate mean
def click_left_GT_calmean():
    force_measuring.left_GT_calmean()
#Click to display mean
def click_left_GT_showmean():

```

```

with open("left_GT_mean.csv", newline="") as file:
    reader = csv.reader(FileZilla)
    r = 0
    for col in reader:
        c = 0
        for row in col:
            label = tk.Label(self, width=6, height=1, text=row, relief=tk.RIDGE)
            label.place(x=226, y=634)
lb22 = tk.Label(self, text="Left GM Force", font = "Helvetica 10 bold")
lb22.place(x=20, y=610)
btn23 = tk.Button(self, text="Record", width = 6, command = click_left_GT_record)
btn23.place(x=20, y=630)
btn24 = tk.Button(self, text="Calculate", width = 8, command= click_left_GT_calmean)
btn24.place(x=83, y=630)
btn25 = tk.Button(self, text="Display", width = 6, command= click_left_GT_showmean)
btn25.place(x=162, y=630)
lb26 = tk.Label(self, text="N.")
lb26.place(x=280, y=634)
#-----Right GT 18_1-----
#Click to Record load cells
def click_right_GT_record():
    force_measuring.right_GT_record()
#Click to calculate mean
def click_right_GT_calmean():
    force_measuring.right_GT_calmean()
#Click to display mean
def click_right_GT_showmean():
    with open("right_GT_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=6, height=1, text=row, relief=tk.RIDGE)
                label.place(x=735, y=634)
lb27 = tk.Label(self, text="Right GM Force", font = "Helvetica 10 bold")
lb27.place(x=815, y=605)
btn28 = tk.Button(self, text="Record", width = 6, command = click_right_GT_record)
btn28.place(x=815, y=630)
btn29 = tk.Button(self, text="Calculate", width = 8, command= click_right_GT_calmean)
btn29.place(x=878, y=630)
btn30 = tk.Button(self, text="Display", width = 6, command= click_right_GT_showmean)
btn30.place(x=957, y=630)
lb31 = tk.Label(self, text="N.")
lb31.place(x=795, y=634)
#----Record all sensors in the same time-----
#Click to Record load cells
def click_all_record():
    force_measuring.right_GT_record()
#-----Abdominal Force_Using Rt Corrective or Rt Axillar-----
#Click to Record load cells
def click_right_axilla_record():
    force_measuring.right_axilla_record()
#Click to calculate mean
def click_right_axilla_calmean():
    force_measuring.right_axilla_calmean()
#Click to display mean
def click_right_axilla_showmean():
    with open("right_axilla_mean.csv", newline="") as file:
        reader = csv.reader(FileZilla)
        r = 0
        for col in reader:
            c = 0
            for row in col:
                label = tk.Label(self, width=6, height=1, text=row, relief=tk.RIDGE)
                label.place(x=1390, y=484)
lb32 = tk.Label(self, text="Abdominal Force (using RT Axilla)", font = "Helvetica 10 bold")
lb32.place(x=1415, y=455)
btn33 = tk.Button(self, text="Record", width = 6, command = click_right_axilla_record)
btn33.place(x=1480, y=480)
btn34 = tk.Button(self, text="Calculate", width = 8, command= click_right_axilla_calmean)
btn34.place(x=1548, y=480)

```



```

ch0.setDeviceSerialNumber(468981)
ch0.setChannel(0)
ch0.openWaitForAttachment(5000)
with open('left_axilla.csv', 'w', newline= ") as f:
    write = csv.writer(f, delimiter = ' ')
    for row in range (0, 100):
        var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
        newvar2 = '{:.2f}'.format(var2)
        print("Force of left axilla [SN:468981, Ch:0]: " + newvar2)
        fieldnames = ['newvar2']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'newvar2': newvar2})
ch0.close()

ch0 = VoltageRatioInput()
ch0.setDeviceSerialNumber(468981)
ch0.setChannel(1)
ch0.openWaitForAttachment(5000)
with open('right_axilla.csv', 'w', newline= ") as f:
    write = csv.writer(f, delimiter = ' ')
    for row in range (0, 100):
        var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
        newvar2 = '{:.2f}'.format(var2)
        print("Force of right axilla [SN:468981, Ch:1]: " + newvar2)
        fieldnames = ['newvar2']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'newvar2': newvar2})
ch0.close()

ch0 = VoltageRatioInput()
ch0.setDeviceSerialNumber(468981)
ch0.setChannel(2)
ch0.openWaitForAttachment(5000)
with open('left_TL.csv', 'w', newline= ") as f:
    write = csv.writer(f, delimiter = ' ')
    for row in range (0, 100):
        var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
        newvar2 = '{:.2f}'.format(var2)
        print("Force of left TL [SN:468981, Ch:2]: " + newvar2)
        fieldnames = ['newvar2']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'newvar2': newvar2})
ch0.close()

ch0 = VoltageRatioInput()
ch0.setDeviceSerialNumber(468981)
ch0.setChannel(Payne et al.)
ch0.openWaitForAttachment(5000)
with open('right_TL.csv', 'w', newline= ") as f:
    write = csv.writer(f, delimiter = ' ')
    for row in range (0, 100):
        var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
        newvar2 = '{:.2f}'.format(var2)
        print("Force of right TL [SN:468981, Ch:3]: " + newvar2)
        fieldnames = ['newvar2']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'newvar2': newvar2})
ch0.close()

ch0 = VoltageRatioInput()
ch0.setDeviceSerialNumber(473718)
ch0.setChannel(0)
ch0.openWaitForAttachment(5000)
with open('left_GT.csv', 'w', newline= ") as f:
    write = csv.writer(f, delimiter = ' ')
    for row in range (0, 100):
        var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
        newvar2 = '{:.2f}'.format(var2)
        print("Force of left GT [SN:473718, Ch:0]: " + newvar2)
        fieldnames = ['newvar2']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'newvar2': newvar2})

```

```

ch0.close()

ch0 = VoltageRatioInput()
ch0.setDeviceSerialNumber(473718)
ch0.setChannel(1)
ch0.openWaitForAttachment(5000)
with open('right_GT.csv', 'w', newline= ") as f:
    write = csv.writer(f, delimiter = ' ')
    for row in range (0, 100):
        var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
        newvar2 = '{:.2f}'.format(var2)
        print("Force of right GT [SN:473718, Ch:1]: " + newvar2)
        fieldnames = ['newvar2']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'newvar2': newvar2})
ch0.close()

#(voltage ratio - y-intercept)/slop
#Left axilla 81_0
def left_axilla_record():
    ch0 = VoltageRatioInput()
    ch0.setDeviceSerialNumber(468981)
    ch0.setChannel(0)
    ch0.openWaitForAttachment(5000)
    with open('left_axilla.csv', 'w', newline= ") as f:
        write = csv.writer(f, delimiter = ' ')
        for row in range (0, 100):
            var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
            newvar2 = '{:.2f}'.format(var2)
            print("Force of left axilla [SN:468981, Ch:0]: " + newvar2)
            fieldnames = ['newvar2']
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            writer.writerow({'newvar2': newvar2})
    ch0.close()

def left_axilla_calmean():
    df = pd.read_csv (r'left_axilla.csv')
    df.columns = ['A']
    mean_ch0 = df['A'].mean()
    newmean_ch0 = '{:.2f}'.format(mean_ch0)
    print('Mean Left Axilla: ' + str(newmean_ch0))

    with open('left_axilla_mean.csv', 'w', newline= ") as f:
        write = csv.writer(f, delimiter = ' ')
        result = str(newmean_ch0)
        fieldnames = ['result']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'result': result})

#Right axilla 81_1
def right_axilla_record():
    ch0 = VoltageRatioInput()
    ch0.setDeviceSerialNumber(468981)
    ch0.setChannel(1)
    ch0.openWaitForAttachment(5000)
    with open('right_axilla.csv', 'w', newline= ") as f:
        write = csv.writer(f, delimiter = ' ')
        for row in range (0, 100):
            var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
            newvar2 = '{:.2f}'.format(var2)
            print("Force of right axilla [SN:468981, Ch:1]: " + newvar2)
            fieldnames = ['newvar2']
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            writer.writerow({'newvar2': newvar2})
    ch0.close()

def right_axilla_calmean():
    df = pd.read_csv (r'right_axilla.csv')
    df.columns = ['A']
    mean_ch0 = df['A'].mean()
    newmean_ch0 = '{:.2f}'.format(mean_ch0)

```



```

print('Mean right Axilla: ' + str(newmean_ch0))

with open('right_axilla_mean.csv', 'w', newline= "") as f:
    write = csv.writer(f, delimiter = ' ')
    result = str(newmean_ch0)
    fieldnames = ['result']
    writer = csv.DictWriter(f, fieldnames = fieldnames)
    writer.writerow({'result': result})

#Left TL 81_2
def left_TL_record():
    ch0 = VoltageRatioInput()
    ch0.setDeviceSerialNumber(468981)
    ch0.setChannel(2)
    ch0.openWaitForAttachment(5000)
    with open('left_TL.csv', 'w', newline= "") as f:
        write = csv.writer(f, delimiter = ' ')
        for row in range (0, 100):
            var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
            newvar2 = '{:.2f}'.format(var2)
            print("Force of left TL [SN:468981, Ch:2]: " + newvar2)
            fieldnames = ['newvar2']
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            writer.writerow({'newvar2': newvar2})
    ch0.close()

def left_TL_calmean():
    df = pd.read_csv (r'left_TL.csv')
    df.columns = ['A']
    mean_ch0 = df['A'].mean()
    newmean_ch0 = '{:.2f}'.format(mean_ch0)
    print('Mean left_TL: ' + str(newmean_ch0))

    with open('left_TL_mean.csv', 'w', newline= "") as f:
        write = csv.writer(f, delimiter = ' ')
        result = str(newmean_ch0)
        fieldnames = ['result']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'result': result})

#Right TL 81_3
def right_TL_record():
    ch0 = VoltageRatioInput()
    ch0.setDeviceSerialNumber(468981)
    ch0.setChannel(Payne et al.)
    ch0.openWaitForAttachment(5000)
    with open('right_TL.csv', 'w', newline= "") as f:
        write = csv.writer(f, delimiter = ' ')
        for row in range (0, 100):
            var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
            newvar2 = '{:.2f}'.format(var2)
            print("Force of right TL [SN:468981, Ch:3]: " + newvar2)
            fieldnames = ['newvar2']
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            writer.writerow({'newvar2': newvar2})
    ch0.close()

def right_TL_calmean():
    df = pd.read_csv (r'right_TL.csv')
    df.columns = ['A']
    mean_ch0 = df['A'].mean()
    newmean_ch0 = '{:.2f}'.format(mean_ch0)
    print('Mean right TL: ' + str(newmean_ch0))

    with open('right_TL_mean.csv', 'w', newline= "") as f:
        write = csv.writer(f, delimiter = ' ')
        result = str(newmean_ch0)
        fieldnames = ['result']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'result': result})

#Left GT 18_0

```

```

def left_GT_record():
    ch0 = VoltageRatioInput()
    ch0.setDeviceSerialNumber(473718)
    ch0.setChannel(0)
    ch0.openWaitForAttachment(5000)
    with open('left_GT.csv', 'w', newline=" ") as f:
        write = csv.writer(f, delimiter = ' ')
        for row in range (0, 100):
            var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
            newvar2 = '{:.2f}'.format(var2)
            print("Force of left GT [SN:473718, Ch:0]: " + newvar2)
            fieldnames = ['newvar2']
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            writer.writerow({'newvar2': newvar2})
    ch0.close()

def left_GT_calmean():
    df = pd.read_csv (r'left_GT.csv')
    df.columns = ['A']
    mean_ch0 = df['A'].mean()
    newmean_ch0 = '{:.2f}'.format(mean_ch0)
    print('Mean left_GT: ' + str(newmean_ch0))

    with open('left_GT_mean.csv', 'w', newline=" ") as f:
        write = csv.writer(f, delimiter = ' ')
        result = str(newmean_ch0)
        fieldnames = ['result']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'result': result})

#Right GT 18_1
def right_GT_record():
    ch0 = VoltageRatioInput()
    ch0.setDeviceSerialNumber(473718)
    ch0.setChannel(1)
    ch0.openWaitForAttachment(5000)
    with open('right_GT.csv', 'w', newline=" ") as f:
        write = csv.writer(f, delimiter = ' ')
        for row in range (0, 100):
            var2 = (((ch0.getVoltageRatio()*1000000)-(Y-intercept))/(Slope)) * 9.81
            newvar2 = '{:.2f}'.format(var2)
            print("Force of right GT [SN:473718, Ch:1]: " + newvar2)
            fieldnames = ['newvar2']
            writer = csv.DictWriter(f, fieldnames = fieldnames)
            writer.writerow({'newvar2': newvar2})
    ch0.close()

def right_GT_calmean():
    df = pd.read_csv (r'right_GT.csv')
    df.columns = ['A']
    mean_ch0 = df['A'].mean()
    newmean_ch0 = '{:.2f}'.format(mean_ch0)
    print('Mean right GT: ' + str(newmean_ch0))

    with open('right_GT_mean.csv', 'w', newline=" ") as f:
        write = csv.writer(f, delimiter = ' ')
        result = str(newmean_ch0)
        fieldnames = ['result']
        writer = csv.DictWriter(f, fieldnames = fieldnames)
        writer.writerow({'result': result})

```

## Appendix 4 Coding for xy-value calculation on Raspberry Pi (Chapter 6)

```
import numpy as np
import cv2
import socket
import time
import os

camnum=1
camconnect=0 # 1 is connected
automode=0 #1 is on

cap =cv2.VideoCapture(0)
background=cv2.imread('blankmask.jpg',cv2.IMREAD_GRAYSCALE)
storedmask=cv2.imread('maskframe.jpg',cv2.IMREAD_GRAYSCALE)
ret2, storedmask=cv2.threshold(storedmask,200,255,cv2.THRESH_BINARY)

mtx = np.array(...)
dist = np.array(...)

while (True):
    ret, frame = cap.read()
    cv2.imshow('frame', frame)

    while(camconnect==0):
        s = socket.socket()
        host = '192.168.0.103' # ip of Raspberry Pi
        port = 27015
        err=999
        while (err!=0):
            err=s.connect_ex((host, port))
            #if (automode!=1) : print ("connection error ",err)
            time.sleep(1)
        stcamnum=str(camnum)
        if(camnum<10):
            stcamnum="0"+ str(camnum)
        string=stcamnum
        #if (automode!=1) : print("rec string ", string)
        outarr=bytes(string, 'utf-8')
        #if (automode!=1) : print(outarr)
        s.send(outarr)
        camconnect=1

    inarr=s.recv(1024)

    sentnum=((inarr[0]-48)*10)+ ((inarr[1]-48))
    mode=inarr[2]-48
    framenum=((inarr[3]-48)*1000)+((inarr[4]-48)*100)+((inarr[5]-48)*10)+((inarr[6]-48)*1)
    maxsize=((inarr[7]-48)*100)+((inarr[8]-48)*10)+((inarr[9]-48))
    minsize=((inarr[10]-48)*100)+((inarr[11]-48)*10)+((inarr[12]-48))
    thru=((inarr[13]-48)*100)+((inarr[14]-48)*10)+((inarr[15]-48))
    thr1=((inarr[16]-48)*100)+((inarr[17]-48)*10)+((inarr[18]-48))
    top=((inarr[19]-48)*100)+((inarr[20]-48)*10)+((inarr[21]-48))
    bottom=((inarr[22]-48)*100)+((inarr[23]-48)*10)+((inarr[24]-48))
    left=((inarr[25]-48)*100)+((inarr[26]-48)*10)+((inarr[27]-48))
    right=((inarr[28]-48)*100)+((inarr[29]-48)*10)+((inarr[30]-48))

    string=""
    # print(top," ",left, " ", bottom, " ", right)
    for i in range (0, 30):
        string=string+str(inarr[i]-48)
    for i in range (len(string), 200):
        string=string+"."

    ret, frame1 = cap.read()
    frame1 = cv2.undistort(frame1, mtx, dist, None)
    cv2.imshow('frame1', frame1)
    frame2= cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
    cv2.imshow('frame2', frame2)
    ret2, frame3=cv2.threshold(frame2,thr1,thru,cv2.THRESH_BINARY)
    cv2.imshow('frame3', frame3)
    frame4= cv2.bitwise_and(cv2.bitwise_not(storedmask),frame3)
```

```

cv2.imshow('frame4', frame4)
frame5=frame4[top:bottom,left:right]
contours , hierarchy = cv2.findContours(frame5, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE,offset=(left,top))
cv2.imshow('frame5', frame5)

mycount=0
for contour in contours:
    mycount=mycount+1
if(mycount>999):
    mycount=999
stcontours=str(mycount)
if(mycount<100):
    stcontours="0"+ str(mycount)
if(mycount<10):
    stcontours="00"+ str(mycount)
string=string+stcontours
#print(stcontours)
string=string+stcontours
content=0
strxy=""
for contour in contours:
    content=content+1
    cx=999
    cy=999
    x=999
    y=999
    w=999
    h=999
    area=999
    scx="999"
    scy="999"
    sx="999"
    sy="999"
    sw="999"
    sh="999"
    sarea="999"
    if (content<11):
        M = cv2.moments(contour)
        if (M['m00']>0):
            cx= int(M['m10']/M['m00'])
            cy= int(M['m01']/M['m00'])
            scx=str(cx)
            scy=str(cy)
            if(cx<100) :
                scx="0"+str(cx)
            if(cx<10) :
                scx="00"+str(cx)
            if(cy<100) :
                scy="0"+str(cy)
            if(cy<10) :
                scy="00"+str(cy)
            strxy=strxy+ " "+scx+" " "+scy

    if (cx!= 999):
        if (cy!=999):
            x,y,w,h =cv2.boundingRect(contour)
            area=w*h
            if (area > maxsize):
                area=999
            if (area < minsize):
                area=999
            sarea=str(area)
            if (area < 100):
                sarea= "0" + str(area)
            if (area < 10):
                sarea= "00" + str(area)

            sx=str(x)
            if (x < 100):
                sx= "0" + str(x)
            if (x < 10):
                sx= "00" + str(x)

```

```

sy=str(y)
if (y < 100):
    sy= "0" + str(y)
if (y < 10):
    sy= "00" + str(y)

sw=str(w)
if (w < 100):
    sw= "0" + str(w)
if (w < 10):
    sw= "00" + str(w)

sh=str(h)
if (h < 100):
    sh= "0" + str(h)
if (h < 10):
    sh= "00" + str(h)

if(area>100): #my adding
    string = string + scx + scy + sarea +sx + sy + sh + sw #my adding
#string = string + scx + scy + sarea +sx + sy + sh + sw #my hiding
#print(scx + scy + sarea +sx + sy + sh + sw) #my hiding

background=cv2.imread('blankmask.jpg',cv2.IMREAD_GRAYSCALE)
contoursframe = cv2.drawContours(background, contours,-1, (255,255,255),1)
cv2.rectangle(contoursframe,(left,top),(right,bottom),(255,255,255),2)
cv2.imshow('contours frame',contoursframe)

print(strxy)

for i in range (len(string), 1024):
    string=string+" "
outarr=bytes(string, 'utf-8')
myerr = s.send(outarr)

if cv2.waitKey(1) == 27:
    cv2.destroyAllWindows()
    break

if(inarr[2]==0+48):

    if (automode!=1) : print(" sending ")

if (inarr[2]==1+48):
    newmask=cv2.add(storedmask, threshframe)
    cv2.imwrite('maskframe.jpg',newmask)
    cv2.show('newmask',newmask) #changing
    storedmask=newmask
    if (automode!=1) : print (" masking")

if(inarr[2]==2+48):
    storedmask=cv2.imread('blankmask.jpg',cv2.IMREAD_GRAYSCALE)
    cv2.imwrite('maskframe.jpg',storedmask)
    if (automode!=1) : print (" reset performed")

```

## Appendix 5 Coding for xy-value calculation on computer (Chapter 6)

```
import numpy as np
import cv2
import time
import os

#Loading camera calibration result
mtx = np.array(...)
dist = np.array(...)

#Reading image
img = cv2.imread("image/ball.jpg") #Image of markers

background = cv2.imread('blankmask.jpg',cv2.IMREAD_GRAYSCALE)
storedmask = cv2.imread('maskframe.jpg',cv2.IMREAD_GRAYSCALE)
ret2, storedmask = cv2.threshold(storedmask, 180, 255, cv2.THRESH_BINARY)

while (True):
    maxsize = 50
    minsize = 0
    thru = 255
    thrl = 120
    string = ""

    for i in range (0, 30):
        string = string + str(i)

    for i in range (len(string), 200):
        string = string + "."

    frame1 = cv2.undistort(img, mtx, dist, None) #Undistorted the image
    cv2.imshow('frame1', img)
    frame2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #Convert to gray scale image
    cv2.imshow('frame2', frame2)
    ret2, frame3 = cv2.threshold(frame2, thrl, thru, cv2.THRESH_BINARY) #Thresholding the image
    cv2.imshow('frame3', frame3)
    frame4 = cv2.bitwise_and(cv2.bitwise_not(storedmask), frame3) #Masking
    contours, hierarchy = cv2.findContours(frame4, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE) #Find the contour of
    detected object
    cv2.imshow('frame4', frame4)

    mycount = 0
    for contour in contours:
        mycount = mycount+1
    if(mycount > 999):
        mycount = 999
    stcontours = str(mycount)
    if(mycount < 100):
        stcontours = "0" + str(mycount)
    if(mycount < 10):
        stcontours = "00" + str(mycount)
    string = string + stcontours
    print(stcontours) #Total number of marker detected
    content = 0

    strxy = ""
    for contour in contours:
        content = content+1
        cx = 999
        cy = 999
        x = 999
        y = 999
        w = 999
        h = 999
        area = 999
        scx = "999"
        scy = "999"
        sx = "999"
        sy = "999"
        sw = "999"
        sh = "999"
```

```

sarea = "999"

#Calculating the center of detected object
if (content < 11):
    M = cv2.moments(contour)
    if (M['m00'] > 0):
        cx = int(M['m10']/M['m00'])
        cy = int(M['m01']/M['m00'])
        scx = str(cx)
        scy = str(cy)
        if(cx < 100):
            scx = "0" + str(cx)
        if(cx < 10):
            scx = "00" + str(cx)
        if(cy < 100):
            scy = "0" + str(cy)
        if(cy < 10):
            scy = "00" + str(cy)
        strxy = strxy + " " + scx + " " + scy

    if (cx != 999):
        if (cy != 999):
            x,y,w,h =cv2.boundingRect(contour)
            area = w*h
            if (area > maxsize):
                area = 999
            if (area < minsize):
                area = 999
            sarea = str(area)
            if (area < 100):
                sarea = "0" + str(area)
            if (area < 10):
                sarea = "00" + str(area)

            sx = str(x)
            if (x < 100):
                sx = "0" + str(x)
            if (x < 10):
                sx = "00" + str(x)

            sy = str(y)
            if (y < 100):
                sy = "0" + str(y)
            if (y < 10):
                sy = "00" + str(y)

            sw = str(w)
            if (w < 100):
                sw = "0" + str(w)
            if (w < 10):
                sw = "00" + str(w)

            sh=str(h)
            if (h < 100):
                sh= "0" + str(h)
            if (h < 10):
                sh= "00" + str(h)

        if(area > 100):
            string = string + scx + scy + sarea +sx + sy + sh + sw
            print(scx + scy + sarea +sx + sy + sh + sw)

background = cv2.imread('blankmask.jpg', cv2.IMREAD_GRAYSCALE)
frame5 = cv2.drawContours(background, contours, -1, (255,255,255), 1)
cv2.imshow('frame5', frame5)
print(strxy)

if cv2.waitKey(1) == 27:
    cv2.destroyAllWindows()
    break

```

## Appendix 6 Coding for camera calibration on Raspberry Pi (Chapter 6)

```
import cv2
import numpy as np
import os
import glob

CHECKERBOARD = (6,9)
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 22, 0.001)

objpoints = []
imgpoints = []

objp = np.zeros((1, CHECKERBOARD[0] * CHECKERBOARD[1], 3), np.float32)
objp[0,::2] = np.mgrid[0:CHECKERBOARD[0], 0:CHECKERBOARD[1]].T.reshape(-1, 2)
prev_img_shape = None

images = glob.glob('./picamera1/*.jpg')
for fname in images:
    img = cv2.imread(fname)
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    ret, corners = cv2.findChessboardCorners( gray, CHECKERBOARD, cv2.CALIB_CB_ADAPTIVE_THRESH +
cv2.CALIB_CB_FAST_CHECK + cv2.CALIB_CB_NORMALIZE_IMAGE)

    if ret == True:
        objpoints.append(objp)
        corners2 = cv2.cornerSubPix(gray, corners, (11,11),(-1,-1), criteria)
        imgpoints.append(corners2)
        img = cv2.drawChessboardCorners(img, CHECKERBOARD, corners2, ret)

    cv2.imshow('img',img)
    cv2.waitKey(0)

cv2.destroyAllWindows()

h,w = img.shape[:2]

ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[:-1], None, None)

print("Camera matrix : \n")
print(mtx)
print("dist : \n")
print(dist)
print("rvecs : \n")
print(rvecs)
print("tvecs : \n")
print(tvecs)

np.savetxt("picamera1_CM.txt", mtx)
np.savetxt("picamera1_D.txt", dist)
```



## Appendix 7 Coding for camera calibration, stereo camera calibration, and marker calculation (Chapter 7)

### 7.1. Coding for taking images for camera calibration (getpicameraimages program)

```
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <stdlib.h>

// Need to link with Ws2_32.lib
#pragma comment (lib, "Ws2_32.lib")

// for console
#define _WIN32_WINNT 0x0500

//included for opencv
#include <opencv2/opencv.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>

//for phidgets
#include <stdlib.h>
#include "phidget22.h"
//other included
#include <conio.h> //for key hit
#include <string>
#include <iostream>
#include <fstream>
#include <shellapi.h>
#include <filesystem>

//namespaces
using namespace cv;
namespace fs = std::filesystem;

void delete_dir_content(const fs::path& dir_path)
{
    for (auto& path : fs::directory_iterator(dir_path))
    {
        fs::remove_all(path);
    }
}

int __cdecl main(void)
{
    //-----
    // start of main program
    //-----

    //get label to console window
    HWND mainwin = GetForegroundWindow();
    //move console window to left side
    MoveWindow(mainwin, 1, 1, 600, 500, TRUE);
    Sleep(1000);

    //-----
    // set up of program
    //-----

    //program settings
    // number of cameras in system up to 8 starting at ip address 192.168.0.101:8000
    int firstcam = 0;
    std::cout << std::endl;
    std::cout << std::endl;
    std::cout << " enter the camera number" << std::endl;
```

```

std::string camnum;
getline(std::cin, camnum);
if (camnum == "")
{
    return 1;
}
if (camnum == "1") { firstcam = 1; }
if (camnum == "2") { firstcam = 2; }
if (camnum == "3") { firstcam = 3; }
if (camnum == "4") { firstcam = 4; }
if (camnum == "5") { firstcam = 5; }
if (camnum == "6") { firstcam = 6; }
if (camnum == "7") { firstcam = 7; }
if (camnum == "8") { firstcam = 8; }
//-----
// global variables
//-----
Mat currentframe1;
VideoCapture cam1;
String videoStreamAddress[9];

//for phidgets
PhidgetReturnCode res;
PhidgetVoltageInputHandle ch;
double voltage;

//setup phidget switch
PhidgetVoltageInput_create(&ch);
Phidget_setChannel((PhidgetHandle)ch, 7); // 7 is channel for switch may change
res=Phidget_openWaitForAttachment((PhidgetHandle)ch, PHIDGET_TIMEOUT_DEFAULT);
if (res != EPHIDGET_OK)
{
    std::cout << " Phidget board not found terminate program" << std::endl;
    return 1; // Exit in error
}

res = PhidgetVoltageInput_getVoltage(ch, &voltage);
if (res != EPHIDGET_OK)
{
    std::cout << " Phidget switch not found terminate program" << std::endl;
    return 1; // Exit in error
}

//-----
// open the pi video camera channels from the ip addresses
//-----

std::cout << "opening video channels" << std::endl;
videoStreamAddress[1] = "http://192.168.0.101:8000/stream.mjpg";
videoStreamAddress[2] = "http://192.168.0.102:8000/stream.mjpg";
videoStreamAddress[3] = "http://192.168.0.103:8000/stream.mjpg";
videoStreamAddress[4] = "http://192.168.0.104:8000/stream.mjpg";
videoStreamAddress[5] = "http://192.168.0.105:8000/stream.mjpg";
videoStreamAddress[6] = "http://192.168.0.106:8000/stream.mjpg";
videoStreamAddress[7] = "http://192.168.0.107:8000/stream.mjpg";
videoStreamAddress[8] = "http://192.168.0.108:8000/stream.mjpg";

if (!cam1.open(videoStreamAddress[firstcam]))
{
    std::cout << "Error opening video stream camera " << firstcam << std::endl;
    return -1;
}

std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << "Done - opening video channel" << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;

std::cout << " " << std::endl;
std::cout << " " << std::endl;
std::cout << " camera " << firstcam << std::endl;

```

```

std::cout << " are you sure you want to delete the current images" << std::endl;
std::cout << " and collect a new set ?" << std::endl;
std::string response;
getline(std::cin, response);
if (response != "y")
{
    return 1;
}
std::cout << " continue" << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;

int stop = 0;
int keynum = 0;

char s0[50] = "C:\\tum\\picamerainages\\picamera";
char s1[50] = "";
_itoa_s(firstcam, s1, 10);
char s4[50] = "";
strcat_s(s4, s0);
strcat_s(s4, s1);
std::cout << s4 << std::endl;

delete_dir_content(s4);
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << " removed old pictures" << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
char r0[50] = "C:\\tum\\picamerainages\\picamera";
char r1[50] = "";
_itoa_s(firstcam, r1, 10);
char r3[50] = "\\";
char r4[50] = "list.txt";
char r5[50] = "";
strcat_s(r5, r0);
strcat_s(r5, r1);
strcat_s(r5, r3);
strcat_s(r5, r4);

std::cout << "storing picture names in file " << r5 << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::ofstream myfile(r5);
if (!myfile.is_open())
{
    std::cout << "Unable to open file" << r5 << std::endl;
    return 1;
}
int mycount = 0;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << "press switch to capture or ESC to end" << std::endl;

while (stop == 0)
{
    // update picture
    cam1.open(videoStreamAddress[firstcam]);
    cam1 >> currentframe1;
    cv::imshow("Image1", currentframe1);
    waitKey(1);
    cam1.release();

    res = PhidgetVoltageInput_getVoltage(ch, &voltage);
    if (voltage > 2.5)
    {
        mycount = mycount + 1;
        //write out current frame
        char t0[50] = "C:\\tum\\picamerainages\\picamera";
        char t1[50] = "";
        _itoa_s(firstcam, t1, 10);
        char t3[50] = "\\picamera";
    }
}

```

```

char t4[50] = "";
_itoa_s(mycount, t4, 10);
char t6[50] = ".jpg";
char t7[50] = "";
strcat_s(t7, t0);
strcat_s(t7, t1);
strcat_s(t7, t3);
strcat_s(t7, t4);
strcat_s(t7, t6);
myfile << t7 << std::endl;
imwrite(t7, currentframe1);
std::cout << "frame captured is " << t7 << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
Sleep(1000);
}
SetForegroundWindow(mainwin);
if (_kbhit()) // if key pressed
{
    keynum = _getch();
    if (keynum == 27) //if escape pressed break loop and end
    {
        stop = 1;
    }
}
myfile.close();

// close phidget
PhidgetVoltageInput_delete(&ch);

// Closes all the frames
cv::destroyAllWindows();
std::cout << "Done - all " << std::endl;
return 0;
}

```

## 7.2. Coding for camera calibration (calibpicamera program)

```

//for sockets
#undef UNICODE
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>

// Need to link with Ws2_32.lib
#pragma comment (lib, "Ws2_32.lib")

// #pragma comment (lib, "Mswsock.lib")
// for console
#define _WIN32_WINNT 0x0500

//included for opencv
#include <opencv2/opencv.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>

//other included
#include <conio.h> //for key hit
#include <string>
#include <iostream>
#include <fstream>
#include <shellapi.h>

```

```

//namespaces
using namespace cv;

int __cdecl main(void)
{
    //-----
    // start of main program
    //-----

    //get label to console window
    HWND mainwin = GetForegroundWindow();
    //move console window to left side
    MoveWindow(mainwin, 1, 1, 700, 500, TRUE);
    Sleep(1000);

    //-----
    // set up of program
    //-----
    //program settings
    int numcameras = 8; // number of cameras in system up to 8 starting at ip address 192.168.0.101:8000
    int review = 1; // 0 is no review of pictures 1 is review picture for best grid
    int calculate = 1; // 0 is dont calculate matrix
    int observe = 1; //observe initial image 0 no 1 yes
    int observeresult = 1; //observe final image 0 no 1 yes
    int observeiterations = 1; // observe threshold interations 0 no 1 yes

    //-----
    // global variables
    //-----
    float squaresize = 26.4;
    int numsuccessfull;
    int mykey1 = 0;
    int mykey2 = 0;
    int cnum = 0;
    int huns, tens, units;
    Mat currentframe;
    int checkerboardheight;
    int checkerboardwidth;
    int CHECKERBOARD[2]{ 1, 1 };
    bool success;
    Mat gray, bwimage;

    int cameranum = 0;
    std::cout << std::endl;
    std::cout << std::endl;
    std::cout << " enter the camera number" << std::endl;
    std::string camnum;
    getline(std::cin, camnum);
    if (camnum == "")
    {
        return 1;
    }
    if (camnum == "1") { cameranum = 1; }
    if (camnum == "2") { cameranum = 2; }
    if (camnum == "3") { cameranum = 3; }
    if (camnum == "4") { cameranum = 4; }
    if (camnum == "5") { cameranum = 5; }
    if (camnum == "6") { cameranum = 6; }
    if (camnum == "7") { cameranum = 7; }
    if (camnum == "8") { cameranum = 8; }

    //-----
    // set calibration board type and points needed
    //-----

    // Defining the dimensions of checkerboard
    checkerboardheight = 6;
    checkerboardwidth = 9;
    CHECKERBOARD[0] = checkerboardheight;
    CHECKERBOARD[1] = checkerboardwidth;

    // Creating vector to store vectors of 3D points for each checkerboard image

```

```

std::vector<std::vector<cv::Point3f>> objpoints;

// Creating vector to store vectors of 2D points for each checkerboard image
std::vector<std::vector<cv::Point2f>> imgpoints;

// Defining the world coordinates for 3D points
std::vector<cv::Point3f> objp;

for (int i{ 0 }; i < CHECKERBOARD[1]; i++)
{
    for (int j{ 0 }; j < CHECKERBOARD[0]; j++)
    {
        objp.push_back(cv::Point3f(j*squaresize, i*squaresize, 0));
    }
}

// vector to store the pixel coordinates of detected checker board corners
std::vector<cv::Point2f> corner_pts;

//-----
// review images and store best points
//-----

std::vector<cv::String> images;
std::string path;

if (review == 1)
{
    std::cout << "review images " << std::endl;
    // review pictures for each camera
    // Extracting path of individual image stored in a given directory

    // Path of the folder containing checkerboard images
    if (camer anum == 1) { path = "c://tum//picameraimages//picamera1//*.jpg"; }
    if (camer anum == 2) { path = "c://tum//picameraimages//picamera2//*.jpg"; }
    if (camer anum == 3) { path = "c://tum//picameraimages//picamera3//*.jpg"; }
    if (camer anum == 4) { path = "c://tum//picameraimages//picamera4//*.jpg"; }
    if (camer anum == 5) { path = "c://tum//picameraimages//picamera5//*.jpg"; }
    if (camer anum == 6) { path = "c://tum//picameraimages//picamera6//*.jpg"; }
    if (camer anum == 7) { path = "c://tum//picameraimages//picamera7//*.jpg"; }
    if (camer anum == 8) { path = "c://tum//picameraimages//picamera8//*.jpg"; }

    cv::glob(path, images);
    std::cout << "images" << images.size() << std::endl;
    numsuccessful = 0;
    bool success;
    bool anysuccess;
    // Looping over all the images in the directory
    for (int j = 0; j < images.size(); j++)
    {
        success = false;
        anysuccess = false;
        // get frame
        currentframe = cv::imread(images[j]);
        //and show ?
        if (observe == 1)
        {
            cv::imshow("Image", currentframe);
            cv::waitKey(1);
        }
        SetForegroundWindow(mainwin);
        std::cout << " processing " << j + 1 << " of " << images.size() << " for camera " << camer anum << std::endl;

        float mymaxerrorvalarray[256];
        for (int h = 0; h < 256; h++)
        {
            std::cout << "iteration " << h << "\r";
            // turn to gray and threshold
            cv::cvtColor(currentframe, gray, cv::COLOR_BGR2GRAY);
            cv::threshold(gray, bwimage, h, 255, cv::THRESH_BINARY);
            // Finding checker board corners
            // If desired number of corners are found in the image then success = true

```

```

//success = cv::findChessboardCorners(bwimage, cv::Size(CHECKERBOARD[0], CHECKERBOARD[1]),
corner_pts, cv::CALIB_CB_ADAPTIVE_THRESH | cv::CALIB_CB_FAST_CHECK |
cv::CALIB_CB_NORMALIZE_IMAGE);
success = cv::findChessboardCorners(bwimage, cv::Size(CHECKERBOARD[0], CHECKERBOARD[1]),
corner_pts, cv::CALIB_CB_FAST_CHECK);

if (success == true)
{
cv::TermCriteria criteria(cv::TermCriteria::EPS | cv::TermCriteria::MAX_ITER, 30, 0.001);

// refining pixel coordinates for given 2d points.
//cv::cornerSubPix(bwimage, corner_pts, cv::Size(11, 11), cv::Size(-1, -1), criteria);

// Displaying the detected corner points on the checker board
cv::drawChessboardCorners(bwimage, cv::Size(CHECKERBOARD[0], CHECKERBOARD[1]), corner_pts,
success);
//std::cout << corner_pts << std::endl;
}

if (observeiterations == 1)
{
cv::imshow("Image", bwimage);
cv::waitKey(1);
SetForegroundWindow(mainwin);
}
mymaxerrorvalarray[h] = 999;
if (success == true)
{
//calculate maximum inter circle length as a % of average length
int cntlines = 0;
int outlines = 0;
float lenlines[94];
float lineaverage[94];
float percenterror[94];
//columns first
for (int l = 0; l < 9; l++)
{
outlines = l * 5;
cntlines = l * 6;
lenlines[1 + outlines] = pow(pow((corner_pts[0 + cntlines].x - corner_pts[1 + cntlines].x), 2) +
pow((corner_pts[0 + cntlines].y - corner_pts[1 + cntlines].y), 2), 0.5);
lenlines[2 + outlines] = pow(pow((corner_pts[1 + cntlines].x - corner_pts[2 + cntlines].x), 2) +
pow((corner_pts[1 + cntlines].y - corner_pts[2 + cntlines].y), 2), 0.5);
lenlines[3 + outlines] = pow(pow((corner_pts[2 + cntlines].x - corner_pts[3 + cntlines].x), 2) +
pow((corner_pts[2 + cntlines].y - corner_pts[3 + cntlines].y), 2), 0.5);
lenlines[4 + outlines] = pow(pow((corner_pts[3 + cntlines].x - corner_pts[4 + cntlines].x), 2) +
pow((corner_pts[3 + cntlines].y - corner_pts[4 + cntlines].y), 2), 0.5);
lenlines[5 + outlines] = pow(pow((corner_pts[4 + cntlines].x - corner_pts[5 + cntlines].x), 2) +
pow((corner_pts[4 + cntlines].y - corner_pts[5 + cntlines].y), 2), 0.5);
lineaverage[1 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines]) / 5;
lineaverage[2 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines]) / 5;
lineaverage[3 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines]) / 5;
lineaverage[4 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines]) / 5;
lineaverage[5 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines]) / 5;
percenterror[1 + outlines] = 100 * (abs(lenlines[1 + outlines] - lineaverage[1 + outlines]) / lineaverage[1 +
outlines]);
percenterror[2 + outlines] = 100 * (abs(lenlines[2 + outlines] - lineaverage[2 + outlines]) / lineaverage[2 +
outlines]);
percenterror[3 + outlines] = 100 * (abs(lenlines[3 + outlines] - lineaverage[3 + outlines]) / lineaverage[3 +
outlines]);
percenterror[4 + outlines] = 100 * (abs(lenlines[4 + outlines] - lineaverage[4 + outlines]) / lineaverage[4 +
outlines]);
percenterror[5 + outlines] = 100 * (abs(lenlines[5 + outlines] - lineaverage[5 + outlines]) / lineaverage[5 +
outlines]);
}
//now rows
for (int l = 0; l < 6; l++)

```

```

    {
        outlines = (5 * 9) + (1 * 8);
        cntlines = (1 * 6);
        lenlines[1 + outlines] = pow(pow((corner_pts[0 + 1].x - corner_pts[6 + 1].x), 2) + pow((corner_pts[0 + 1].y -
corner_pts[6 + 1].y), 2), 0.5);
        lenlines[2 + outlines] = pow(pow((corner_pts[6 + 1].x - corner_pts[12 + 1].x), 2) + pow((corner_pts[6 + 1].y -
corner_pts[12 + 1].y), 2), 0.5);
        lenlines[3 + outlines] = pow(pow((corner_pts[12 + 1].x - corner_pts[18 + 1].x), 2) + pow((corner_pts[12 + 1].y -
corner_pts[18 + 1].y), 2), 0.5);
        lenlines[4 + outlines] = pow(pow((corner_pts[18 + 1].x - corner_pts[24 + 1].x), 2) + pow((corner_pts[18 + 1].y -
corner_pts[24 + 1].y), 2), 0.5);
        lenlines[5 + outlines] = pow(pow((corner_pts[24 + 1].x - corner_pts[30 + 1].x), 2) + pow((corner_pts[24 + 1].y -
corner_pts[30 + 1].y), 2), 0.5);
        lenlines[6 + outlines] = pow(pow((corner_pts[30 + 1].x - corner_pts[36 + 1].x), 2) + pow((corner_pts[30 + 1].y -
corner_pts[36 + 1].y), 2), 0.5);
        lenlines[7 + outlines] = pow(pow((corner_pts[36 + 1].x - corner_pts[42 + 1].x), 2) + pow((corner_pts[36 + 1].y -
corner_pts[42 + 1].y), 2), 0.5);
        lenlines[8 + outlines] = pow(pow((corner_pts[42 + 1].x - corner_pts[48 + 1].x), 2) + pow((corner_pts[42 + 1].y -
corner_pts[48 + 1].y), 2), 0.5);
        lineaverage[1 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[2 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[3 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[4 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[5 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[6 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[7 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        lineaverage[8 + outlines] = (lenlines[1 + outlines] + lenlines[2 + outlines] + lenlines[3 + outlines] + lenlines[4
+ outlines] + lenlines[5 + outlines] + lenlines[6 + outlines] + lenlines[7 + outlines] + lenlines[8 + outlines]) / 8;
        percenterror[1 + outlines] = 100 * (abs(lenlines[1 + outlines] - lineaverage[1 + outlines]) / lineaverage[1 +
outlines]);
        percenterror[2 + outlines] = 100 * (abs(lenlines[2 + outlines] - lineaverage[2 + outlines]) / lineaverage[2 +
outlines]);
        percenterror[3 + outlines] = 100 * (abs(lenlines[3 + outlines] - lineaverage[3 + outlines]) / lineaverage[3 +
outlines]);
        percenterror[4 + outlines] = 100 * (abs(lenlines[4 + outlines] - lineaverage[4 + outlines]) / lineaverage[4 +
outlines]);
        percenterror[5 + outlines] = 100 * (abs(lenlines[5 + outlines] - lineaverage[5 + outlines]) / lineaverage[5 +
outlines]);
        percenterror[6 + outlines] = 100 * (abs(lenlines[6 + outlines] - lineaverage[6 + outlines]) / lineaverage[6 +
outlines]);
        percenterror[7 + outlines] = 100 * (abs(lenlines[7 + outlines] - lineaverage[7 + outlines]) / lineaverage[7 +
outlines]);
        percenterror[8 + outlines] = 100 * (abs(lenlines[8 + outlines] - lineaverage[8 + outlines]) / lineaverage[8 +
outlines]);
    }
    //for (int l = 1; l < 94; l++)
    //{
    //    std::cout << lenlines[l] << " of " << lineaverage[l] << " percent error = " << percenterror[l] << std::endl;
    //}
    //get maxerror for this iteration of this frame
    float mymaxerrorval = 0;
    for (int n = 0; n < 94; n++)
    {
        if (percenterror[n] >= mymaxerrorval)
        {
            mymaxerrorval = percenterror[n];
        }
    }
    mymaxerrorvalarray[h] = mymaxerrorval;
    anysuccess = 1;
}
} //next h iteration
std::cout << std::endl;

int bestiteration = 999;

```



```

float bestiterationval = 999;
// now if any success then pick one with lowest maxerror
if (anysuccess == true)
{
    for (int h = 0; h < 256; h++)
    {
        if (mymaxerrorvalarray[h] < bestiterationval)
        {
            bestiterationval = mymaxerrorvalarray[h];
            bestiteration = h;
        }
    }
    std::cout << "best iteration is number " << bestiteration << " with a threshold of " << bestiterationval << " and with an
error of " << bestiterationval << std::endl;

    //ok now redo with that threshold and store the values and show
    // turn to gray and threshold
    cv::cvtColor(currentframe, gray, cv::COLOR_BGR2GRAY);
    cv::threshold(gray, bwimage, bestiteration, 255, cv::THRESH_BINARY);
    // Finding checker board corners
    // If desired number of corners are found in the image then success = true
    //success = cv::findChessboardCorners(bwimage, cv::Size(CHECKERBOARD[0], CHECKERBOARD[1]),
corner_pts, cv::CALIB_CB_ADAPTIVE_THRESH | cv::CALIB_CB_FAST_CHECK |
cv::CALIB_CB_NORMALIZE_IMAGE);
    success = cv::findChessboardCorners(bwimage, cv::Size(CHECKERBOARD[0], CHECKERBOARD[1]),
corner_pts, cv::CALIB_CB_ADAPTIVE_THRESH | cv::CALIB_CB_FAST_CHECK |
cv::CALIB_CB_NORMALIZE_IMAGE);

    if (success == true)
    {
        cv::TermCriteria criteria(cv::TermCriteria::EPS | cv::TermCriteria::MAX_ITER, 22, 0.001);

        // refining pixel coordinates for given 2d points.
        //cv::cornerSubPix(bwimage, corner_pts, cv::Size(11, 11), cv::Size(-1, -1), criteria);

        // Displaying the detected corner points on the checker board
        cv::drawChessboardCorners(bwimage, cv::Size(CHECKERBOARD[0], CHECKERBOARD[1]), corner_pts,
success);
        //std::cout << corner_pts << std::endl;
    }

    if (observeresult == 1)
    {
        cv::imshow("Image", bwimage);
        cv::waitKey(1);
        SetForegroundWindow(mainwin);
        Sleep(2000);
    }
    objpoints.push_back(objp);
    imgpoints.push_back(corner_pts);
}
//on a successful frame write out and store
if (success == true)
{
    std::cout << " frame successful" << std::endl;
    //write out frame data
    numssuccessfull = numssuccessfull + 1;

    // Path of the file containing points data for frame
    if (j < 10)
    {
        char s0[50] = "c:\\tum\\picamerapoints\\picamera";
        char s1[50] = "";
        _itoa_s(cameranum, s1, 10);
        char s2[50] = "\\calpoints00";
        char s3[50] = "";
        _itoa_s(j, s3, 10);
        char s4[50] = ".txt";
        char s5[50] = "";
        strcat_s(s5, s0);
        strcat_s(s5, s1);
        strcat_s(s5, s2);
    }
}

```

```

strcat_s(s5, s3);
strcat_s(s5, s4);
std::cout << "writing points file " << s5 << std::endl;
std::ofstream myfile(s5);
if (!myfile.is_open())
{
    std::cout << "Unable to open file" << std::endl;
    return 1;
}
int numpoints;
myfile << " points" << std::endl;
numpoints = 54;
for (int k = 0; k < numpoints; k++)
{
    myfile << corner_pts[k].x << " , " << corner_pts[k].x << std::endl;
}
myfile.close();
}
if (j >= 10)
{
    if (j < 100)
    {
        char s0[50] = "c:\\tum\\picamerapoints\\picamera";
        char s1[50] = "";
        _itoa_s(cameranum, s1, 10);
        char s2[50] = "\\calpoints0";
        char s3[50] = "";
        _itoa_s(j, s3, 10);
        char s4[50] = ".txt";
        char s5[50] = "";
        strcat_s(s5, s0);
        strcat_s(s5, s1);
        strcat_s(s5, s2);
        strcat_s(s5, s3);
        strcat_s(s5, s4);
        std::cout << "writing points file " << s5 << std::endl;
        std::ofstream myfile(s5);
        if (!myfile.is_open())
        {
            std::cout << "Unable to open file" << std::endl;
            return 1;
        }
        int numpoints;
        myfile << " points" << std::endl;
        numpoints = 54;
        for (int k = 0; k < numpoints; k++)
        {
            myfile << corner_pts[k].x << " , " << corner_pts[k].x << std::endl;
        }
        myfile.close();
    }
}
if (j >= 100)
{
    char s0[50] = "c:\\tum\\picamerapoints\\picamera";
    char s1[50] = "";
    _itoa_s(cameranum, s1, 10);
    char s2[50] = "\\calpoints";
    char s3[50] = "";
    _itoa_s(j, s3, 10);
    char s4[50] = ".txt";
    char s5[50] = "";
    strcat_s(s5, s0);
    strcat_s(s5, s1);
    strcat_s(s5, s2);
    strcat_s(s5, s3);
    strcat_s(s5, s4);
    std::cout << "writing points file " << s5 << std::endl;
    std::ofstream myfile(s5);
    if (!myfile.is_open())
    {
        std::cout << "Unable to open file" << std::endl;
        return 1;
    }
}

```

```

    }
    int numpoints;
    myfile << " points" << std::endl;
    numpoints = 54;
    for (int k = 0; k < numpoints; k++)
    {
        myfile << corner_pts[k].x << " , " << corner_pts[k].y << std::endl;
    }
    myfile.close();
}
}
else
{
    std::cout << " frame rejected" << std::endl;
}
}
std::cout << numsuccessfull << " out of " << images.size() - 1 << " for camera " << cameranum << std::endl;
}
std::cout << "Done - review " << std::endl;

//-----
// calculate and store matrices for cameras
//-----

std::cout << "calculating matrices " << std::endl;

std::cout << "calculating for camera " << cameranum << std::endl;
cv::Mat cameraMatrix, distCoeffs, R, T;
cv::calibrateCamera(objpoints, imgpoints, cv::Size(gray.rows, gray.cols), cameraMatrix, distCoeffs, R, T);

std::string outpath;
// Path of the file containing camera data
if (cameranum == 1) { outpath = "c://tum//picameraimages//picamera1.xml"; }
if (cameranum == 2) { outpath = "c://tum//picameraimages//picamera2.xml"; }
if (cameranum == 3) { outpath = "c://tum//picameraimages//picamera3.xml"; }
if (cameranum == 4) { outpath = "c://tum//picameraimages//picamera4.xml"; }
if (cameranum == 5) { outpath = "c://tum//picameraimages//picamera5.xml"; }
if (cameranum == 6) { outpath = "c://tum//picameraimages//picamera6.xml"; }
if (cameranum == 7) { outpath = "c://tum//picameraimages//picamera7.xml"; }
if (cameranum == 8) { outpath = "c://tum//picameraimages//picamera8.xml"; }

std::cout << "cameraMatrix : " << cameraMatrix << std::endl;
std::cout << "distCoeffs : " << distCoeffs << std::endl;
std::cout << "Rotation vector : " << R << std::endl;
std::cout << "Translation vector : " << T << std::endl;

FileStorage fs(outpath, FileStorage::WRITE);
fs << "cameraMatrix" << cameraMatrix;
fs << "distCoeffs" << distCoeffs;
fs << "R" << R;
fs << "T" << T;
fs.release();

std::cout << "Done - calculating matrices " << std::endl;

cv::destroyAllWindows();
std::cout << "Done - all " << std::endl;
return 0;
}

```

### 7.3. Coding for taking stereo images for stereo camera calibration (getpistereimages program)

```
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <stdlib.h>

//#include <filesystem>
// Need to link with Ws2_32.lib
#pragma comment (lib, "Ws2_32.lib")

// for console
#define _WIN32_WINNT 0x0500
//included for opencv
#include <opencv2/opencv.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>

//for phidgets
#include <stdlib.h>
#include "phidget22.h"

//other included
#include <conio.h> //for key hit
#include <string>
#include <iostream>
#include <fstream>
#include <shellapi.h>

//namespaces
using namespace cv;
namespace fs = std::filesystem;
#include <filesystem>
namespace fs = std::filesystem;

void delete_dir_content(const fs::path& dir_path)
{
    for (auto& path : fs::directory_iterator(dir_path))
    {
        fs::remove_all(path);
    }
}

int __cdecl main(void)
{
    //-----
    // start of main program
    //-----

    //get label to console window
    HWND mainwin = GetForegroundWindow();
    //move console window to left side
    MoveWindow(mainwin, 1, 1, 500, 500, TRUE);
    Sleep(1000);

    //-----
    // set up of program
    //-----

    //program settings
    // number of cameras in system up to 8 starting at ip address 192.168.0.101:8000
    int firstcam = 0;
    int secondcam = 0;
    std::string camnum;
    std::cout << std::endl;
    std::cout << std::endl;
    std::cout << " enter the first camera number" << std::endl;
    getline(std::cin, camnum);
    if (camnum == "")
    {

```

```

    return 1;
}
if (camnum == "1") { firstcam = 1; }
if (camnum == "2") { firstcam = 2; }
if (camnum == "3") { firstcam = 3; }
if (camnum == "4") { firstcam = 4; }
if (camnum == "5") { firstcam = 5; }
if (camnum == "6") { firstcam = 6; }
if (camnum == "7") { firstcam = 7; }
if (camnum == "8") { firstcam = 8; }

std::cout << std::endl;
std::cout << std::endl;
std::cout << " enter the second camera number" << std::endl;
getline(std::cin, camnum);
if (camnum == "")
{
    return 1;
}
if (camnum == "1") { secondcam = 1; }
if (camnum == "2") { secondcam = 2; }
if (camnum == "3") { secondcam = 3; }
if (camnum == "4") { secondcam = 4; }
if (camnum == "5") { secondcam = 5; }
if (camnum == "6") { secondcam = 6; }
if (camnum == "7") { secondcam = 7; }
if (camnum == "8") { secondcam = 8; }

//-----
// global variables
//-----
Mat currentframe1;
Mat undcurrentframe1;
Mat currentframe2;
Mat undcurrentframe2;
VideoCapture cam1;
VideoCapture cam2;
String videoStreamAddress[9];
cv::Mat cameraMatrix1, distCoeffs1;
cv::Mat cameraMatrix2, distCoeffs2;

//for phidgets
PhidgetReturnCode res;
PhidgetVoltageInputHandle ch;
double voltage;

//setup phidget switch
PhidgetVoltageInput_create(&ch);
Phidget_setChannel((PhidgetHandle)ch, 7); // 7 is channel for switch may change
res = Phidget_openWaitForAttachment((PhidgetHandle)ch, PHIDGET_TIMEOUT_DEFAULT);
if (res != EPHIDGET_OK)
{
    std::cout << " Phidget board not found terminate program" << std::endl;
    return 1; // Exit in error
}

res = PhidgetVoltageInput_getVoltage(ch, &voltage);
if (res != EPHIDGET_OK)
{
    std::cout << " Phidget switch not found terminate program" << std::endl;
    return 1; // Exit in error
}

//-----
// open the pi video camera channels from the ip addresses
//-----

std::cout << "opening video channels" << std::endl;
videoStreamAddress[1] = "http://192.168.0.101:8000/stream.mjpg";
videoStreamAddress[2] = "http://192.168.0.102:8000/stream.mjpg";
videoStreamAddress[3] = "http://192.168.0.103:8000/stream.mjpg";
videoStreamAddress[4] = "http://192.168.0.104:8000/stream.mjpg";

```

```

videoStreamAddress[5] = "http://192.168.0.105:8000/stream.mjpg";
videoStreamAddress[6] = "http://192.168.0.106:8000/stream.mjpg";
videoStreamAddress[7] = "http://192.168.0.107:8000/stream.mjpg";
videoStreamAddress[8] = "http://192.168.0.108:8000/stream.mjpg";

if (!cam1.open(videoStreamAddress[firstcam]))
{
    std::cout << "Error opening video stream camera " << firstcam << std::endl;
    return -1;
}
std::cout << "opened video stream camera " << firstcam << std::endl;
if (!cam1.open(videoStreamAddress[secondcam]))
{
    std::cout << "Error opening video stream camera " << secondcam << std::endl;
    return -1;
}
std::cout << "opened video stream camera " << secondcam << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << "Done - opening video channels" << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;

//-----
// get camera calibration data
//-----

std::string inpath1;
// Path of the file containing camera data
if (firstcam == 1) { inpath1 = "c://tum//picameraimages//picamera1.xml"; }
if (firstcam == 2) { inpath1 = "c://tum//picameraimages//picamera2.xml"; }
if (firstcam == 3) { inpath1 = "c://tum//picameraimages//picamera3.xml"; }
if (firstcam == 4) { inpath1 = "c://tum//picameraimages//picamera4.xml"; }
if (firstcam == 5) { inpath1 = "c://tum//picameraimages//picamera5.xml"; }
if (firstcam == 6) { inpath1 = "c://tum//picameraimages//picamera6.xml"; }
if (firstcam == 7) { inpath1 = "c://tum//picameraimages//picamera7.xml"; }
if (firstcam == 8) { inpath1 = "c://tum//picameraimages//picamera8.xml"; }

std::string inpath2;
// Path of the file containing camera data
if (secondcam == 1) { inpath2 = "c://tum//picameraimages//picamera1.xml"; }
if (secondcam == 2) { inpath2 = "c://tum//picameraimages//picamera2.xml"; }
if (secondcam == 3) { inpath2 = "c://tum//picameraimages//picamera3.xml"; }
if (secondcam == 4) { inpath2 = "c://tum//picameraimages//picamera4.xml"; }
if (secondcam == 5) { inpath2 = "c://tum//picameraimages//picamera5.xml"; }
if (secondcam == 6) { inpath2 = "c://tum//picameraimages//picamera6.xml"; }
if (secondcam == 7) { inpath2 = "c://tum//picameraimages//picamera7.xml"; }
if (secondcam == 8) { inpath2 = "c://tum//picameraimages//picamera8.xml"; }

cv::Mat cameraMatrix;
cv::Mat distCoeffs;
Mat_<double> M1(3, 3);
Mat_<double> D1(5, 1);

FileStorage fs1(inpath1, FileStorage::READ);
fs1["cameraMatrix"] >> cameraMatrix;
fs1["distCoeffs"] >> distCoeffs;
fs1.release();
M1 = cameraMatrix;
D1 = distCoeffs;

Mat_<double> M2(3, 3);
Mat_<double> D2(5, 1);
FileStorage fs2(inpath2, FileStorage::READ);
fs2["cameraMatrix"] >> cameraMatrix;
fs2["distCoeffs"] >> distCoeffs;
fs2.release();
M2 = cameraMatrix;
D2 = distCoeffs;

std::cout << "Done - getting calibration data " << std::endl;
std::cout << "" << std::endl;

```

```

std::cout << " " << std::endl;
std::cout << " camera " << firstcam << " and camera " << secondcam << std::endl;
std::cout << " are you sure you want to delete the current stereo images" << std::endl;
std::cout << " and collect a new set ?" << std::endl;
std::string response;
getline(std::cin, response);
if (response != "y")
{
    return 1;
}
std::cout << " continue" << std::endl;
//-----
// scan cameras
//-----
int stop = 0;
int keynum = 0;

char s0[50] = "C:\\tum\\picameraimages\\picamerastereo";
char s1[50] = "";
_itoa_s(firstcam, s1, 10);
char s2[50] = "";
_itoa_s(secondcam, s2, 10);
char s3[50] = "\\";

char s4[50] = "";
strcat_s(s4, s0);
strcat_s(s4, s1);
strcat_s(s4, s2);
strcat_s(s4, s3);
std::cout << s4 << std::endl;

delete_dir_content(s4);
std::cout << " removed old pictures" << std::endl;
char r0[50] = "C:\\tum\\picameraimages\\picamerastereo";
char r1[50] = "";
_itoa_s(firstcam, r1, 10);
char r2[50] = "";
_itoa_s(secondcam, r2, 10);
char r3[50] = "\\";
char r4[50] = "list.txt";

char r5[50] = "";
strcat_s(r5, r0);
strcat_s(r5, r1);
strcat_s(r5, r2);
strcat_s(r5, r3);
strcat_s(r5, r4);

std::cout << r5 << std::endl;
std::ofstream myfile(r5);
if (!myfile.is_open())
{
    std::cout << "Unable to open file" <<r5<< std::endl;
    return 1;
}

std::cout << std::endl;
std::cout << std::endl;
std::cout << " press switch to capture frames or esc to finish" << std::endl;
int mycount = 0;
while (stop == 0)
{
    // update pictures and undistort
    cam1.open(videoStreamAddress[firstcam]);
    cam2.open(videoStreamAddress[secondcam]);
    cam1 >> currentframe1;
    cam2 >> currentframe2;
    cv::undistort(currentframe1, undcurrentframe1, M1, D1, cv::noArray());
    cv::undistort(currentframe2, undcurrentframe2, M2, D2, cv::noArray());
    cv::imshow("Image1", undcurrentframe1);
    //cv::imshow("Image1", currentframe1);
    waitKey(1);
}

```

```

cv::imshow("Image2", undcurrentframe2);
//cv::imshow("Image2", currentframe2);
waitKey(1);
cam1.release();
cam2.release();

res = PhidgetVoltageInput_getVoltage(ch, &voltage);
if (voltage > 0.5) //2.5
{
    mycount = mycount + 1;
    //write out current frames
    //first camera
    char t0[50] = "C:\\tum\\picamerainages\\picamerastereo";
    char t1[50] = "";
    _itoa_s(firstcam, t1, 10);
    char t2[50] = "";
    _itoa_s(secondcam, t2, 10);
    char t3[50] = "\\";
    char t4[50] = "";
    _itoa_s(mycount, t4, 10);
    char t5[50] = "";
    _itoa_s(firstcam, t5, 10);
    char t6[50] = ".jpg";
    char t7[50] = "";
    strcat_s(t7, t0);
    strcat_s(t7, t1);
    strcat_s(t7, t2);
    strcat_s(t7, t3);
    strcat_s(t7, t4);
    strcat_s(t7, t5);
    strcat_s(t7, t6);
    myfile << t7 << std::endl;
    //imwrite(t7, undcurrentframe1);
    imwrite(t7, currentframe1);

    //secondcamera
    char u0[50] = "C:\\tum\\picamerainages\\picamerastereo";
    char u1[50] = "";
    _itoa_s(firstcam, u1, 10);
    char u2[50] = "";
    _itoa_s(secondcam, u2, 10);
    char u3[50] = "\\";
    char u4[50] = "";
    _itoa_s(mycount, u4, 10);
    char u5[50] = "";
    _itoa_s(secondcam, u5, 10);
    char u6[50] = ".jpg";
    char u7[50] = "";
    strcat_s(u7, u0);
    strcat_s(u7, u1);
    strcat_s(u7, u2);
    strcat_s(u7, u3);
    strcat_s(u7, u4);
    strcat_s(u7, u5);
    strcat_s(u7, u6);
    myfile << u7 << std::endl;
    //imwrite(u7, undcurrentframe2);
    imwrite(u7, currentframe2);
    std::cout << "next pair of frames captured" << std::endl;
    Sleep(1000);
}
SetForegroundWindow(mainwin);
if (_kbhit()) // if key pressed
{
    keynum = _getch();
    if (keynum == 27) //if escape pressed break loop and end
    {
        stop = 1;
    }
}
}
myfile.close();

```



```

// close phidget
PhidgetVoltageInput_delete(&ch);

// Closes all the frames
cv::destroyAllWindows();
std::cout << "Done - all " << std::endl;
return 0;
}

```

## 7.4. Coding for stereo camera calibration (calibpistereocameras program)

```

#pragma warning(disable : 4996)

//included for opencv
#include <opencv2/opencv.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include "opencv2/imgcodecs.hpp"
#include "opencv2/core/utility.hpp"
#include <windows.h>
#include <sstream>
#include <iostream>
#include <fstream>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
using namespace std;
using namespace cv;

int main()
{
    const char* imageList = "C:\\tum\\picameraimages\\picamerastereo12\\list.txt";
    int board_w = 9, board_h = 6;
    //-----
    // set up of program
    //-----
    //program settings
    int camsparallel = 0;
    int allnframes = 0;

    // number of cameras in system up to 8 starting at ip address 192.168.0.101:8000
    int firstcam = 1;
    int secondcam = 2;
    //-----
    // global variables
    //-----
    int nx = board_w;
    int ny = board_h;
    bool useUncalibrated = false;
    bool displayCorners = true;
    bool showUndistorted = true;
    bool isVerticalStereo = false; // horiz or vert cams
    const int maxScale = 1;
    float squaresize = 26.4f;
    //-----
    // get camera calibration data
    //-----

    std::string inpath1;
    // Path of the file containing camera data
    if (firstcam == 1) { inpath1 = "c://tum//picameraimages//picamera1.xml"; }
    if (firstcam == 2) { inpath1 = "c://tum//picameraimages//picamera2.xml"; }
    if (firstcam == 3) { inpath1 = "c://tum//picameraimages//picamera3.xml"; }
}

```

```

if (firstcam == 4) { inpath1 = "c://tum//picameraimages//picamera4.xml"; }
if (firstcam == 5) { inpath1 = "c://tum//picameraimages//picamera5.xml"; }
if (firstcam == 6) { inpath1 = "c://tum//picameraimages//picamera6.xml"; }
if (firstcam == 7) { inpath1 = "c://tum//picameraimages//picamera7.xml"; }
if (firstcam == 8) { inpath1 = "c://tum//picameraimages//picamera8.xml"; }

std::string inpath2;
// Path of the file containing camera data
if (secondcam == 1) { inpath2 = "c://tum//picameraimages//picamera1.xml"; }
if (secondcam == 2) { inpath2 = "c://tum//picameraimages//picamera2.xml"; }
if (secondcam == 3) { inpath2 = "c://tum//picameraimages//picamera3.xml"; }
if (secondcam == 4) { inpath2 = "c://tum//picameraimages//picamera4.xml"; }
if (secondcam == 5) { inpath2 = "c://tum//picameraimages//picamera5.xml"; }
if (secondcam == 6) { inpath2 = "c://tum//picameraimages//picamera6.xml"; }
if (secondcam == 7) { inpath2 = "c://tum//picameraimages//picamera7.xml"; }
if (secondcam == 8) { inpath2 = "c://tum//picameraimages//picamera8.xml"; }

Mat M1, M2, D1, D2;
cv::Mat R, T;
cv::Mat cameraMatrix, distCoeffs;

FileStorage fs1(inpath1, FileStorage::READ);
fs1["cameraMatrix"] >> cameraMatrix;
M1 = cameraMatrix;
fs1["distCoeffs"] >> distCoeffs;
D1 = distCoeffs;
fs1["R"] >> R;
fs1["T"] >> T;
fs1.release();

FileStorage fs2(inpath2, FileStorage::READ);
fs2["cameraMatrix"] >> cameraMatrix;
M2 = cameraMatrix;
fs2["distCoeffs"] >> distCoeffs;
D2 = distCoeffs;
fs2["R"] >> R;
fs2["T"] >> T;
fs2.release();

std::cout << "Done - getting calibration data " << std::endl;
/// actual square size
FILE* f = fopen(imageList, "rt");
int i, j, lr;
int N = nx * ny;
cv::Size board_sz = cv::Size(nx, ny);
vector<string> imageNames[2];
vector<cv::Point3f> boardModel;
vector<vector<cv::Point3f>> objectPoints;
vector<vector<cv::Point2f>> points[2];
vector<cv::Point2f> corners[2];

vector<cv::Point2f> corners1;
vector<cv::Point2f> corners2;

vector<cv::Point3f> xyd;
vector<cv::Point3f> xyz;
bool found[2] = { false, false };
cv::Size imageSize;

// READ IN THE LIST OF CIRCLE GRIDS:
//
if (!f) {
    cout << "Cannot open file " << imageList << endl;
    return 0;
}
for (i = 0; i < ny; i++)
    for (j = 0; j < nx; j++)
        boardModel.push_back(
            cv::Point3f((float)(i * squaresize), (float)(j * squaresize), 0.f));
i = 0;
for (;) {
    char buff[1024];

```

```

    lr = i % 2;
    if (lr == 0)
        found[0] = found[1] = false;
    if (!fgets(buf, sizeof(buf) - 3, f))
        break;
    size_t len = strlen(buf);
    while (len > 0 && isspace(buf[len - 1]))
        buf[--len] = '\0';
    if (buf[0] == '#')
        continue;
    cv::Mat img = cv::imread(buf, 0);
    if (img.empty())
        break;
    imageSize = img.size();
    imageNames[lr].push_back(buf);
    i++;

    // If we did not find board on the left image,
    // it does not make sense to find it on the right.
    //
    if (lr == 1 && !found[0])
        continue;

    // Find circle grids and centers therein:
    for (int s = 1; s <= maxScale; s++) {
        cv::Mat timg = img;
        if (s > 1)
            resize(img, timg, cv::Size(), s, s, cv::INTER_CUBIC);

        found[lr] = cv::findChessboardCorners(timg, board_sz, corners[lr]);

        if (found[lr] || s == maxScale) {
            cv::Mat mcorners(corners[lr]);
            mcorners *= (1. / s);
        }
        if (found[lr])
            break;
    }
    if (displayCorners) {
        cout << buf << endl;
        cv::Mat cimg;
        cv::cvtColor(img, cimg, cv::COLOR_GRAY2BGR);

        // draw chessboard corners works for circle grids too
        cv::drawChessboardCorners(cimg, cv::Size(nx, ny), corners[lr], found[lr]);
        cv::imshow("Corners", cimg);
        waitKey(100);
        if ((cv::waitKey(1) & 255) == 27) // Allow ESC to quit
            exit(-1);
    }
    else
        cout << '\n';
    if (lr == 1 && found[0] && found[1]) {
        objectPoints.push_back(boardModel);
        points[0].push_back(corners[0]);
        points[1].push_back(corners[1]);
    }
}
fclose(f);
cv::destroyWindow("Corners");

// CALIBRATE THE STEREO CAMERAS
cv::Mat RS, TS, E, F;
cout << "\nRunning stereo calibration ... \n";
cv::stereoCalibrate(
    objectPoints, points[0], points[1], M1, D1, M2, D2, imageSize, RS, TS, E, F,
    cv::CALIB_FIX_INTRINSIC //USE ESTABLISHED INTRINSIC MATRICIES
| cv::CALIB_FIX_ASPECT_RATIO //FIX THE ASPECT RATIO
| cv::CALIB_FIX_PRINCIPAL_POINT //FIX THE CAMERAS PRINCIPAL POINTS
| cv::CALIB_ZERO_TANGENT_DIST //REMOVE THE DISTORTIONS
| cv::CALIB_FIX_FOCAL_LENGTH //FIX THE FOCAL LENGTH TO THE ONES IN THE MATRICIES
,

```

```

cv::TermCriteria(cv::TermCriteria::COUNT | cv::TermCriteria::EPS, 30,
1e-6);

cout << endl;
cout << imageSize << endl;
cout << endl;
cout << endl;
cout << R << endl;
cout << endl;
cout << T << endl;
cout << endl;
cout << endl;
double a = T.at<double>(0, 0);
double b = T.at<double>(1, 0);
double c = T.at<double>(2, 0);
double Tlength = pow(pow(a, 2) + pow(b, 2) + pow(c, 2), 0.5);
cout << "Tlength = " << Tlength << endl;
cout << endl;
cv::destroyWindow("corners");
//cout << "Done! Press any key to step through images, ESC to exit\n\n";

// CALIBRATION QUALITY CHECK
// because the output fundamental matrix implicitly
// includes all the output information,
// we can check the quality of calibration using the
// epipolar geometry constraint:  $m_2^t * F * m_1 = 0$ 
vector<cv::Point3f> lines[2];
double avgErr = 0;
int nframes = (int)objectPoints.size();
for (i = 0; i < nframes; i++) {
    vector<cv::Point2f>& pt0 = points[0][i];
    vector<cv::Point2f>& pt1 = points[1][i];
    cv::undistortPoints(pt0, pt0, M1, D1, cv::Mat(), M1);
    cv::undistortPoints(pt1, pt1, M2, D2, cv::Mat(), M2);
    cv::computeCorrespondEpilines(pt0, 1, F, lines[0]);
    cv::computeCorrespondEpilines(pt1, 2, F, lines[1]);

    for (j = 0; j < N; j++) {
        double err = fabs(pt0[j].x * lines[1][j].x + pt0[j].y * lines[1][j].y +
            lines[1][j].z) +
            fabs(pt1[j].x * lines[0][j].x + pt1[j].y * lines[0][j].y +
            lines[0][j].z);
        avgErr += err;
    }
}
cout << "avg err = " << avgErr / (nframes * N) << endl;

// COMPUTE AND store maps
//

cv::Mat Q;
cv::Mat R1, R2, P1, P2, map11, map12, map21, map22;

std::cout << "BOUGUET'S METHOD" << std::endl;
if (camsparallel == 0)
{
    stereoRectify(M1, D1, M2, D2, imageSize, RS, TS, R1, R2, P1, P2,
        Q, 0);
}
else
{
    stereoRectify(M1, D1, M2, D2, imageSize, RS, TS, R1, R2, P1, P2,
        Q, cv::CALIB_ZERO_DISPARITY);
}

isVerticalStereo = fabs(P2.at<double>(1, 3)) > fabs(P2.at<double>(0, 3));
// Precompute maps for cvRemap()
initUndistortRectifyMap(M1, D1, R1, P1, imageSize, CV_16SC2, map11,
    map12);
initUndistortRectifyMap(M2, D2, R2, P2, imageSize, CV_16SC2, map21,
    map22);

```

```

std::cout << "writing location file " << std::endl;
FileStorage fs3("c:\\tum\\picameraimages\\picamerastereo12\\location.xml", FileStorage::WRITE);
fs3 << "RS" << RS;
fs3 << "TS" << TS;
fs3.release();

//by tum added
//std::cout << "writing TUM file " << std::endl;
//FileStorage fs4("c:\\tum\\picameraimages\\picamerastereo12\\tum.xml", FileStorage::WRITE);
//fs4 << "R1" << R1;
//fs4 << "R2" << R2;
//fs4 << "P1" << R1;
//fs4 << "P2" << R2;
//fs4.release();
//

std::cout << "writing maps file " << std::endl;
FileStorage fs("c:\\tum\\picameraimages\\picamerastereo12\\maps.xml", FileStorage::WRITE);

fs << "map11" << map11;
fs << "map12" << map12;
fs << "map21" << map21;
fs << "map22" << map22;
fs << "Q" << Q;
fs.release();

std::cout << " done writing maps file " << std::endl;

std::cout << "rectifying and finding disparity maps and displaying" << std::endl;
// RECTIFY THE IMAGES AND FIND DISPARITY MAPS
//
cv::Mat pair;
if (!isVerticalStereo)
    pair.create(imageSize.height, imageSize.width * 2, CV_8UC3);
else
    pair.create(imageSize.height * 2, imageSize.width, CV_8UC3);

// Setup for finding stereo correspondences
//
cv::Ptr<cv::StereoSGBM> stereo = cv::StereoSGBM::create(
    -64, 128, 11, 100, 1000, 32, 0, 15, 1000, 16, StereoSGBM::MODE_SGBM);

if (allnframes != 0)
{
    int numframes = nframes;
    if (allnframes == 1) { numframes = 1; }
    for (i = 0; i < numframes; i++) {
        cv::Mat img1 = cv::imread(imageNames[0][i].c_str(), 0);
        cv::Mat img2 = cv::imread(imageNames[1][i].c_str(), 0);
        cv::Mat img1r, img2r, disp, vdisp, my3dimage;
        if (img1.empty() || img2.empty())
            continue;
        cv::remap(img1, img1r, map11, map12, cv::INTER_LINEAR);
        cv::remap(img2, img2r, map21, map22, cv::INTER_LINEAR);
        if (!isVerticalStereo || !useUncalibrated) {

            //When the stereo camera is oriented vertically,
            //Hartley method does not transpose the
            //image, so the epipolar lines in the rectified
            //images are vertical. Stereo correspondence
            //function does not support such a case.
            stereo->compute(img1r, img2r, disp);
            cv::normalize(disp, vdisp, 0, 256, cv::NORM_MINMAX, CV_8U);
            cv::imshow("disparity", vdisp);
        }
        if (!isVerticalStereo)
        {
            cv::Mat part = pair.colRange(0, imageSize.width);
            cv::cvtColor(img1r, part, cv::COLOR_GRAY2BGR);
            part = pair.colRange(imageSize.width, imageSize.width * 2);
            cv::cvtColor(img2r, part, cv::COLOR_GRAY2BGR);
            for (j = 0; j < imageSize.height; j += 16)

```

```

        cv::line(pair, cv::Point(0, j), cv::Point(imageSize.width * 2, j),
                cv::Scalar(0, 255, 0));
    }
    else
    {
        cv::Mat part = pair.rowRange(0, imageSize.height);
        cv::cvtColor(img1r, part, cv::COLOR_GRAY2BGR);
        part = pair.rowRange(imageSize.height, imageSize.height * 2);
        cv::cvtColor(img2r, part, cv::COLOR_GRAY2BGR);
        for (j = 0; j < imageSize.width; j += 16)
            cv::line(pair, cv::Point(j, 0), cv::Point(j, imageSize.height * 2),
                    cv::Scalar(0, 255, 0));
    }
    cv::imshow("rectified", pair);
    cv::waitKey();
    cv::destroyWindow("rectified");
}
}
return 0;
}

```

## 7.5. Coding for taking images for marker calculation (getimagepairs program)

```

#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#include <stdlib.h>
// #include <filesystem>

// Need to link with Ws2_32.lib
#pragma comment (lib, "Ws2_32.lib")

// for console
#define _WIN32_WINNT 0x0500

// included for opencv
#include <opencv2/opencv.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>

// for phidgets
#include <stdlib.h>
#include "phidget22.h"

// other included
#include <conio.h> // for key hit
#include <string>
#include <iostream>
#include <fstream>
#include <shellapi.h>

// namespaces
using namespace cv;
namespace fs = std::filesystem;
#include <filesystem>
namespace fs = std::filesystem;

void delete_dir_content(const fs::path& dir_path)
{
    for (auto& path : fs::directory_iterator(dir_path))
    {
        fs::remove_all(path);
    }
}

int __cdecl main(void)

```

```

{
//-----
// start of main program
//-----

//get label to console window
HWND mainwin = GetForegroundWindow();
//move console window to left side
MoveWindow(mainwin, 1, 1, 500, 500, TRUE);
Sleep(1000);

//-----
// set up of program
//-----
//program settings
// number of cameras in system up to 8 starting at ip address 192.168.0.101:8000
int firstcam = 0;
int secondcam = 0;
std::string camnum;
std::cout << std::endl;
std::cout << std::endl;
std::cout << " enter the first camera number" << std::endl;
getline(std::cin, camnum);
if (camnum == "")
{
return 1;
}
if (camnum == "1") { firstcam = 1; }
if (camnum == "2") { firstcam = 2; }
if (camnum == "3") { firstcam = 3; }
if (camnum == "4") { firstcam = 4; }
if (camnum == "5") { firstcam = 5; }
if (camnum == "6") { firstcam = 6; }
if (camnum == "7") { firstcam = 7; }
if (camnum == "8") { firstcam = 8; }

std::cout << std::endl;
std::cout << std::endl;
std::cout << " enter the second camera number" << std::endl;
getline(std::cin, camnum);
if (camnum == "")
{
return 1;
}
if (camnum == "1") { secondcam = 1; }
if (camnum == "2") { secondcam = 2; }
if (camnum == "3") { secondcam = 3; }
if (camnum == "4") { secondcam = 4; }
if (camnum == "5") { secondcam = 5; }
if (camnum == "6") { secondcam = 6; }
if (camnum == "7") { secondcam = 7; }
if (camnum == "8") { secondcam = 8; }

//-----
// global variables
//-----
Mat currentframe1;
Mat undcurrentframe1;
Mat currentframe2;
Mat undcurrentframe2;
VideoCapture cam1;
VideoCapture cam2;
String videoStreamAddress[9];
cv::Mat cameraMatrix1, distCoeffs1;
cv::Mat cameraMatrix2, distCoeffs2;

//for phidgets
PhidgetReturnCode res;
PhidgetVoltageInputHandle ch;
double voltage;

//setup phidget switch

```

```

PhidgetVoltageInput_create(&ch);
Phidget_setChannel((PhidgetHandle)ch, 7); // 7 is channel for switch may change
res = Phidget_openWaitForAttachment((PhidgetHandle)ch, PHIDGET_TIMEOUT_DEFAULT);
if (res != EPHIDGET_OK)
{
    std::cout << " Phidget board not found terminate program" << std::endl;
    return 1; // Exit in error
}

res = PhidgetVoltageInput_getVoltage(ch, &voltage);
if (res != EPHIDGET_OK)
{
    std::cout << " Phidget switch not found terminate program" << std::endl;
    return 1; // Exit in error
}

//-----
// open the pi video camera channels from the ip addresses
//-----

std::cout << "opening video channels" << std::endl;
videoStreamAddress[1] = "http://192.168.0.101:8000/stream.mjpg";
videoStreamAddress[2] = "http://192.168.0.102:8000/stream.mjpg";
videoStreamAddress[3] = "http://192.168.0.103:8000/stream.mjpg";
videoStreamAddress[4] = "http://192.168.0.104:8000/stream.mjpg";
videoStreamAddress[5] = "http://192.168.0.105:8000/stream.mjpg";
videoStreamAddress[6] = "http://192.168.0.106:8000/stream.mjpg";
videoStreamAddress[7] = "http://192.168.0.107:8000/stream.mjpg";
videoStreamAddress[8] = "http://192.168.0.108:8000/stream.mjpg";

if (!cam1.open(videoStreamAddress[firstcam]))
{
    std::cout << "Error opening video stream camera " << firstcam << std::endl;
    return -1;
}
std::cout << "opened video stream camera " << firstcam << std::endl;
if (!cam1.open(videoStreamAddress[secondcam]))
{
    std::cout << "Error opening video stream camera " << secondcam << std::endl;
    return -1;
}
std::cout << "opened video stream camera " << secondcam << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << "Done - opening video channels" << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;

//-----
// get camera calibration data
//-----

std::string inpath1;
//// Path of the file containing camera data
if (firstcam == 1) { inpath1 = "c://tum//picameraimages//picamera1.xml"; }
if (firstcam == 2) { inpath1 = "c://tum//picameraimages//picamera2.xml"; }
if (firstcam == 3) { inpath1 = "c://tum//picameraimages//picamera3.xml"; }
if (firstcam == 4) { inpath1 = "c://tum//picameraimages//picamera4.xml"; }
if (firstcam == 5) { inpath1 = "c://tum//picameraimages//picamera5.xml"; }
if (firstcam == 6) { inpath1 = "c://tum//picameraimages//picamera6.xml"; }
if (firstcam == 7) { inpath1 = "c://tum//picameraimages//picamera7.xml"; }
if (firstcam == 8) { inpath1 = "c://tum//picameraimages//picamera8.xml"; }

std::string inpath2;
// Path of the file containing camera data
if (secondcam == 1) { inpath2 = "c://tum//picameraimages//picamera1.xml"; }
if (secondcam == 2) { inpath2 = "c://tum//picameraimages//picamera2.xml"; }
if (secondcam == 3) { inpath2 = "c://tum//picameraimages//picamera3.xml"; }
if (secondcam == 4) { inpath2 = "c://tum//picameraimages//picamera4.xml"; }
if (secondcam == 5) { inpath2 = "c://tum//picameraimages//picamera5.xml"; }
if (secondcam == 6) { inpath2 = "c://tum//picameraimages//picamera6.xml"; }
if (secondcam == 7) { inpath2 = "c://tum//picameraimages//picamera7.xml"; }

```



```

if (secondcam == 8) { inpath2 = "c://tum//picameraimages//picamera8.xml"; }

cv::Mat cameraMatrix;
cv::Mat distCoeffs;
Mat_<double> M1(3, 3);
Mat_<double> D1(5, 1);

FileStorage fs1(inpath1, FileStorage::READ);
fs1["cameraMatrix"] >> cameraMatrix;
fs1["distCoeffs"] >> distCoeffs;
fs1.release();
M1 = cameraMatrix;
D1 = distCoeffs;

Mat_<double> M2(3, 3);
Mat_<double> D2(5, 1);
FileStorage fs2(inpath2, FileStorage::READ);
fs2["cameraMatrix"] >> cameraMatrix;
fs2["distCoeffs"] >> distCoeffs;
fs2.release();
M2 = cameraMatrix;
D2 = distCoeffs;

std::cout << "opened video stream camera " << secondcam << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << "Done - opening video channels" << std::endl;
std::cout << "" << std::endl;
std::cout << "" << std::endl;
std::cout << " camera " << firstcam << " and camera " << secondcam << std::endl;
std::cout << " are you sure you want to delete the current images" << std::endl;
std::cout << " in c://tumcal and collect a new set ?" << std::endl;
std::cout << " " << std::endl;
std::cout << " if not enter n copy the existing set to " << std::endl;
std::cout << " a new location and rerun program" << std::endl;
std::cout << " " << std::endl;

std::string response;
getline(std::cin, response);
if (response != "y")
{
    return 1;
}
std::cout << " continue" << std::endl;

//-----
// scan cameras
//-----
int stop = 0;
int keynum = 0;

char s4[50] = "c://tumcal";
delete_dir_content(s4);
std::cout << " removed old pictures" << std::endl;

std::cout << " press switch to capture frames or esc to finish" << std::endl;
int mycount = 0;

while (stop == 0)
{
    // update pictures and undistort
    cam1.open(videoStreamAddress[firstcam]);
    cam2.open(videoStreamAddress[secondcam]);
    cam1 >> currentframe1;
    cam2 >> currentframe2;
    cv::undistort(currentframe1, undcurrentframe1, M1, D1, cv::noArray());
    cv::undistort(currentframe2, undcurrentframe2, M2, D2, cv::noArray());
    cv::imshow("Image1", undcurrentframe1);
    //cv::imshow("Image1", currentframe1);
    waitKey(1);
    cv::imshow("Image2", undcurrentframe2);
    //cv::imshow("Image2", currentframe2);
}

```

```

waitKey(1);
cam1.release();
cam2.release();

res = PhidgetVoltageInput_getVoltage(ch, &voltage);
if (voltage > 0.1) //2.5
{
    mycount = mycount + 1;
    //write out current frames
    //first camera
    char t0[50] = "C:\\tumcal\\";
    char t4[50] = "";
    _itoa_s(mycount, t4, 10);
    char t5[50] = "";
    _itoa_s(firstcam, t5, 10);
    char t6[50] = ".jpg";
    char t7[50] = "";
    strcat_s(t7, t0);
    strcat_s(t7, t4);
    strcat_s(t7, t5);
    strcat_s(t7, t6);
    imwrite(t7, undcurrentframe1);
    //imwrite(t7, currentframe1);
    //secondcamera
    char u0[50] = "C:\\tumcal\\";
    char u4[50] = "";
    _itoa_s(mycount, u4, 10);
    char u5[50] = "";
    _itoa_s(secondcam, u5, 10);
    char u6[50] = ".jpg";
    char u7[50] = "";
    strcat_s(u7, u0);
    strcat_s(u7, u4);
    strcat_s(u7, u5);
    strcat_s(u7, u6);
    imwrite(u7, undcurrentframe2);
    //imwrite(u7, currentframe2);
    std::cout << "next pair of frames captured" << std::endl;
    Sleep(1000);
}

//SetForegroundWindow(mainwin);
if (_kbhit() // if key pressed
{
    keynum = _getch();
    if (keynum == 27) //if escape pressed break loop and end
    {
        stop = 1;
    }
}
}

// close phidget
PhidgetVoltageInput_delete(&ch);

// Closes all the frames
cv::destroyAllWindows();
std::cout << "Done - all " << std::endl;
return 0;
}

```

## 7.6. Coding for marker calculation (reconstructfromimages program)

```
#pragma once
#pragma warning(disable : 4996)

//included for opencv
#include <opencv2/opencv.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/core/core.hpp>
#include "opencv2/imgcodecs.hpp"
#include "opencv2/core/utility.hpp"
#include <windows.h>
#include <sstream>
#include <iostream>
#include <fstream>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include </opengl/glm/vec3.hpp>
#include </opengl/glm/vec4.hpp>
#include </opengl/glm/mat4x4.hpp>
#include </opengl/glm/gtc/matrix_transform.hpp>
#include </opengl/glm/gtx/transform.hpp>
#include <conio.h> //for key hit

//glew
#define GLEW_STATIC
#include <GL/glew.h>

//glfw
#include <GLFW/glfw3.h>
// window dimensions
#define _USE_MATH_DEFINES
GLFWwindow* mywindow;

using namespace std;
using namespace cv;

int main()
{
    //-----
    // program settings
    //-----
    int drawframes = 0;
    int drawrectified = 0;

    //-----
    // global variables
    //-----
    Mat currentframe1;
    Mat undcurrentframe1;
    Mat currentframe2;
    Mat undcurrentframe2;
    Mat remapframe1, remapframe2;
    Mat gframe1, gframe2, tframe1, tframe2;

    int thresh1 = 180;
    int thresh2 = 180;
    vector< vector< cv::Point> > contours1;
    vector< vector< cv::Point> > contours2;
    float pts1[500][100][3];
    int box1[5];
    int box2[5];
    int currentcam = 0;

    box1[1] = 1;
    box1[2] = 639;
    box1[3] = 1;
    box1[4] = 479;
```

```

box2[1] = 1;
box2[2] = 639;
box2[3] = 1;
box2[4] = 479;
cv::Mat ibox1, ibox2;

//get label to console window
HWND mainwin = GetForegroundWindow();
//move console window to left side
MoveWindow(mainwin, 200, 1, 500, 500, TRUE);
Sleep(1000);

int firstcam = 0;
int secondcam = 0;
SetForegroundWindow(mainwin);
std::string camnum;
std::cout << std::endl;
std::cout << std::endl;
std::cout << " enter the first camera number" << std::endl;
getline(std::cin, camnum);
if (camnum == "")
{
    return 1;
}
if (camnum == "1") { firstcam = 1; }
if (camnum == "2") { firstcam = 2; }
if (camnum == "3") { firstcam = 3; }
if (camnum == "4") { firstcam = 4; }
if (camnum == "5") { firstcam = 5; }
if (camnum == "6") { firstcam = 6; }
if (camnum == "7") { firstcam = 7; }
if (camnum == "8") { firstcam = 8; }

std::cout << std::endl;
std::cout << std::endl;
std::cout << " enter the second camera number" << std::endl;
getline(std::cin, camnum);
if (camnum == "")
{
    return 1;
}
if (camnum == "1") { secondcam = 1; }
if (camnum == "2") { secondcam = 2; }
if (camnum == "3") { secondcam = 3; }
if (camnum == "4") { secondcam = 4; }
if (camnum == "5") { secondcam = 5; }
if (camnum == "6") { secondcam = 6; }
if (camnum == "7") { secondcam = 7; }
if (camnum == "8") { secondcam = 8; }

//get boxes from file

char s0[50] = "C://tum//picameraimages//picamerastereo";
char s1[50] = "";
_itoa_s(firstcam, s1, 10);
char s2[50] = "";
_itoa_s(secondcam, s2, 10);
char s3[50] = "//boxes.txt";
char s4[80] = "";
strcat_s(s4, s0);
strcat_s(s4, s1);
strcat_s(s4, s2);
strcat_s(s4, s3);
std::ifstream imyfile(s4);
if (!imyfile.is_open())
{
    std::cout << "Unable to open file" << std::endl;
    return 1;
}
imyfile >> box1[1];
imyfile >> box1[2];

```

```

imyfile >> box1[3];
imyfile >> box1[4];
imyfile >> box2[1];
imyfile >> box2[2];
imyfile >> box2[3];
imyfile >> box2[4];
imyfile.close();
//-----
// get camera location data
//-----

std::string inpath1;
// Path of the file containing camera data
if (firstcam == 1) { inpath1 = "c://tum//picameraimages//locationpicamera1.xml"; }
if (firstcam == 2) { inpath1 = "c://tum//picameraimages//locationpicamera2.xml"; }
if (firstcam == 3) { inpath1 = "c://tum//picameraimages//locationpicamera3.xml"; }
if (firstcam == 4) { inpath1 = "c://tum//picameraimages//locationpicamera4.xml"; }
if (firstcam == 5) { inpath1 = "c://tum//picameraimages//locationpicamera5.xml"; }
if (firstcam == 6) { inpath1 = "c://tum//picameraimages//locationpicamera6.xml"; }
if (firstcam == 7) { inpath1 = "c://tum//picameraimages//locationpicamera7.xml"; }
if (firstcam == 8) { inpath1 = "c://tum//picameraimages//locationpicamera8.xml"; }

std::string inpath2;
// Path of the file containing camera data
if (secondcam == 1) { inpath2 = "c://tum//picameraimages//locationpicamera1.xml"; }
if (secondcam == 2) { inpath2 = "c://tum//picameraimages//locationpicamera2.xml"; }
if (secondcam == 3) { inpath2 = "c://tum//picameraimages//locationpicamera3.xml"; }
if (secondcam == 4) { inpath2 = "c://tum//picameraimages//locationpicamera4.xml"; }
if (secondcam == 5) { inpath2 = "c://tum//picameraimages//locationpicamera5.xml"; }
if (secondcam == 6) { inpath2 = "c://tum//picameraimages//locationpicamera6.xml"; }
if (secondcam == 7) { inpath2 = "c://tum//picameraimages//locationpicamera7.xml"; }
if (secondcam == 8) { inpath2 = "c://tum//picameraimages//locationpicamera8.xml"; }

cv::Mat Rnew, Tnew;
cv::Mat Rnew1, Tnew1;
cv::Mat Rnew2, Tnew2;

FileStorage fs1(inpath1, FileStorage::READ);
fs1["Rnew"] >> Rnew1;
//Rnew1 = Rnew;
fs1["Tnew"] >> Tnew1;
//Tnew1 = Tnew;
fs1.release();
Mat cam1rot;
cv::Rodrigues(Rnew1, cam1rot);
FileStorage fs2(inpath2, FileStorage::READ);
fs2["Rnew"] >> Rnew2;
//Rnew2 = Rnew;
fs2["Tnew"] >> Tnew2;
//Tnew2 = Tnew;
fs2.release();
Mat cam2rot;
cv::Rodrigues(Rnew2, cam2rot);

std::cout << " " << std::endl;
std::cout << "Done - getting location and rotation data " << std::endl;
std::cout << " " << std::endl;
std::cout << " " << std::endl;

std::string inpath1;
// Path of the file containing camera data
if (firstcam == 1) { inpath1 = "c://tum//picameraimages//picamera1.xml"; }
if (firstcam == 2) { inpath1 = "c://tum//picameraimages//picamera2.xml"; }
if (firstcam == 3) { inpath1 = "c://tum//picameraimages//picamera3.xml"; }
if (firstcam == 4) { inpath1 = "c://tum//picameraimages//picamera4.xml"; }
if (firstcam == 5) { inpath1 = "c://tum//picameraimages//picamera5.xml"; }
if (firstcam == 6) { inpath1 = "c://tum//picameraimages//picamera6.xml"; }
if (firstcam == 7) { inpath1 = "c://tum//picameraimages//picamera7.xml"; }
if (firstcam == 8) { inpath1 = "c://tum//picameraimages//picamera8.xml"; }

std::string inpath2;
// Path of the file containing camera data

```

```

if (secondcam == 1) { inpath2 = "c://tum//picameraimages//picamera1.xml"; }
if (secondcam == 2) { inpath2 = "c://tum//picameraimages//picamera2.xml"; }
if (secondcam == 3) { inpath2 = "c://tum//picameraimages//picamera3.xml"; }
if (secondcam == 4) { inpath2 = "c://tum//picameraimages//picamera4.xml"; }
if (secondcam == 5) { inpath2 = "c://tum//picameraimages//picamera5.xml"; }
if (secondcam == 6) { inpath2 = "c://tum//picameraimages//picamera6.xml"; }
if (secondcam == 7) { inpath2 = "c://tum//picameraimages//picamera7.xml"; }
if (secondcam == 8) { inpath2 = "c://tum//picameraimages//picamera8.xml"; }

Mat M1, M2, D1, D2;
cv::Mat R, T;
cv::Mat cameraMatrix, distCoeffs;

FileStorage fs3(inpath1, FileStorage::READ);
fs3["cameraMatrix"] >> cameraMatrix;
M1 = cameraMatrix;
fs3["distCoeffs"] >> distCoeffs;
D1 = distCoeffs;
fs3["R"] >> R;
fs3["T"] >> T;
fs3.release();

FileStorage fs4(inpath2, FileStorage::READ);
fs4["cameraMatrix"] >> cameraMatrix;
M2 = cameraMatrix;
fs4["distCoeffs"] >> distCoeffs;
D2 = distCoeffs;
fs4["R"] >> R;
fs4["T"] >> T;
fs4.release();

cv::Mat map11, map12, map21, map22, Q;
std::cout << "reading maps file " << std::endl;

FileStorage fs5("c://tum//picameraimages//picamerastereo12//maps.xml", FileStorage::READ);
fs5["map11"] >> map11;
fs5["map12"] >> map12;
fs5["map21"] >> map21;
fs5["map22"] >> map22;
fs5["Q"] >> Q;
fs5.release();

std::cout << "processing pairs of images" << std::endl;
std::cout << "use key 1 to select camera 1 " << std::endl;
std::cout << "use key 2 to select camera 2 " << std::endl;
std::cout << "use key f1/f2 to move left edge of box " << std::endl;
std::cout << "use key f3/f4 to move right edge of box " << std::endl;
std::cout << "use key f5/f6 to move top edge of box " << std::endl;
std::cout << "use key f7/f8 to move bottom edge of box " << std::endl;
std::cout << "press space bar to accept " << std::endl;

int mycount = 0;
int stop = 0;
//get latest filename from store file and open data file
//open file
char r0[60] = "C://tumprogs//reconstructfromimages//filename.txt";
std::ifstream myfilein(r0);
if (!myfilein.is_open())
{
    std::cout << "Unable to open file" << r0 << std::endl;
    return 1;
}
int filenum = 0;
myfilein >> filenum;
myfilein.close();
filenum = filenum + 1;

std::ofstream myfileout(r0);
if (!myfileout.is_open())
{
    std::cout << "Unable to open file" << r0 << std::endl;
    return 1;
}

```

```

}
myfileout << filename << endl;
myfileout.close();

char t0[70] = "C:\\tum\\picameradata\\reconstructdatafromimages";
char t1[70] = "";
_itoa_s(filename, t1, 10);
char t2[70] = ".txt";
char t3[70] = "";
strcat_s(t3, t0);
strcat_s(t3, t1);
strcat_s(t3, t2);

std::ofstream myfilestore(t3);
if (!myfilestore.is_open())
{
    std::cout << "Unable to open file" << r0 << std::endl;
    return 1;
}

std::vector<cv::String> images;
std::string imgpath = "c://tumcal//";
cv::glob(imgpath, images);
std::cout << "image pairs found is " << int(images.size() / 2) << std::endl;
for (int i = 1; i <= int(images.size() / 2); i++)
{
    int nogo = 0;
    int keynum1 = 0;
    int keynum2 = 0;
    char s0[50] = "c:\\tumcal\\";
    char s1[50] = "";
    _itoa_s(i, s1, 10);
    char s2[50] = "1.jpg";
    char s3[50] = "";
    strcat_s(s3, s0);
    strcat_s(s3, s1);
    strcat_s(s3, s2);
    char s4[50] = "c:\\tumcal\\";
    char s5[50] = "";
    _itoa_s(i, s5, 10);
    char s6[50] = "2.jpg";
    char s7[50] = "";
    strcat_s(s7, s4);
    strcat_s(s7, s5);
    strcat_s(s7, s6);
    std::cout << s3 << endl;
    std::cout << s7 << endl;
    while (nogo == 0)
    {
        //get images
        currentframe1 = cv::imread(s3);
        currentframe2 = cv::imread(s7);
        //add box 1
        cv::line(currentframe1, cv::Point(box1[1], box1[3]), cv::Point(box1[2], box1[3]), cv::Scalar(0, 0, 255));
        cv::line(currentframe1, cv::Point(box1[2], box1[3]), cv::Point(box1[2], box1[4]), cv::Scalar(0, 0, 255));
        cv::line(currentframe1, cv::Point(box1[2], box1[4]), cv::Point(box1[1], box1[4]), cv::Scalar(0, 0, 255));
        cv::line(currentframe1, cv::Point(box1[1], box1[4]), cv::Point(box1[1], box1[3]), cv::Scalar(0, 0, 255));
        //add box 2
        cv::line(currentframe2, cv::Point(box2[1], box2[3]), cv::Point(box2[2], box2[3]), cv::Scalar(0, 0, 255));
        cv::line(currentframe2, cv::Point(box2[2], box2[3]), cv::Point(box2[2], box2[4]), cv::Scalar(0, 0, 255));
        cv::line(currentframe2, cv::Point(box2[2], box2[4]), cv::Point(box2[1], box2[4]), cv::Scalar(0, 0, 255));
        cv::line(currentframe2, cv::Point(box2[1], box2[4]), cv::Point(box2[1], box2[3]), cv::Scalar(0, 0, 255));

        SetForegroundWindow(mainwin);
        keynum1 = 0;
        keynum2 = 0;
        if (_kbhit()) // if key pressed
        {
            keynum1 = _getch();
            if (keynum1 == 0)
            {
                keynum2 = _getch();
            }
        }
    }
}

```

```

    }
    //std::cout << keynum1 << " " << keynum2 << std::endl;
}

if (keynum1 == 32)
{
    nogo = 1;
}
if (keynum1 == 49)
{
    currentcam = 1;
    std::cout << "camera 1 selected" << std::endl;
}
if (keynum1 == 50)
{
    currentcam = 2;
    std::cout << "camera 2 selected" << std::endl;
}

//left
if (keynum2 == 59)
{
    if (currentcam == 1)
    {
        //std::cout << " change box 1" << //std::endl;
        box1[1] = box1[1] - 10;
        if (box1[1] < 1)
        {
            box1[1] = 1;
        }
        //std::cout << box1[1] << //std::endl;
    }
    if (currentcam == 2)
    {
        //std::cout << " change box 2" << //std::endl;
        box2[1] = box2[1] - 10;
        if (box2[1] < 1)
        {
            box2[1] = 1;
        }
        //std::cout << box2[1] << //std::endl;
    }
}
if (keynum2 == 60)
{
    if (currentcam == 1)
    {
        //std::cout << " change box 1" << //std::endl;
        box1[1] = box1[1] + 10;
        if (box1[1] > 639)
        {
            box1[1] = 639;
        }
        if (box1[1] >= box1[2])
        {
            box1[1] = box1[2] - 10;
        }
        //std::cout << box1[1] << //std::endl;
    }
    if (currentcam == 2)
    {
        //std::cout << " change box 2" << //std::endl;
        box2[1] = box2[1] + 10;
        if (box2[1] < 1)
        {
            box2[1] = 1;
        }
        if (box2[1] >= box2[2])
        {
            box2[1] = box2[2] + 10;
        }
        //std::cout << box2[1] << //std::endl;
    }
}

```



```

    }
}

//right
if (keynum2 == 61)
{
    if (currentcam == 1)
    {
        //std::cout << " change box 1" << //std::endl;
        box1[2] = box1[2] - 10;
        if (box1[2] < 1)
        {
            box1[2] = 1;
        }
        if (box1[2] <= box1[1])
        {
            box1[2] = box1[1] + 10;
        }
        //std::cout << box1[2] << //std::endl;
    }
    if (currentcam == 2)
    {
        //std::cout << " change box 2" << //std::endl;
        box2[2] = box2[2] - 10;
        if (box2[2] < 1)
        {
            box2[2] = 1;
        }
        if (box2[2] <= box2[1])
        {
            box2[2] = box2[1] + 10;
        }
        //std::cout << box2[2] << //std::endl;
    }
}
if (keynum2 == 62)
{
    if (currentcam == 1)
    {
        //std::cout << " change box 1" << //std::endl;
        box1[2] = box1[2] + 10;
        if (box1[2] > 639)
        {
            box1[2] = 639;
        }
        //std::cout << box1[2] << //std::endl;
    }
    if (currentcam == 2)
    {
        //std::cout << " change box 2" << //std::endl;
        box2[2] = box2[2] + 10;
        if (box2[2] > 639)
        {
            box2[2] = 639;
        }
        //std::cout << box2[2] << //std::endl;
    }
}

//top
if (keynum2 == 63)
{
    if (currentcam == 1)
    {
        //std::cout << " change box 3" << //std::endl;
        box1[3] = box1[3] - 10;
        if (box1[3] < 1)
        {
            box1[3] = 1;
        }
        //std::cout << box1[3] << //std::endl;
    }
}

```

```

if (currentcam == 2)
{
//std::cout << " change box 4" << //std::endl;
box2[3] = box2[3] - 10;
if (box2[3] < 1)
{
box2[3] = 1;
}
//std::cout << box2[3] << //std::endl;
}
}
if (keynum2 == 64)
{
if (currentcam == 1)
{
//std::cout << " change box 3" << //std::endl;
box1[3] = box1[3] + 10;
if (box1[3] > 639)
{
box1[3] = 639;
}
if (box1[3] >= box1[4])
{
box1[3] = box1[4] - 10;
}
//std::cout << box1[3] << //std::endl;
}
if (currentcam == 2)
{
//std::cout << " change box 4" << //std::endl;
box2[3] = box2[3] + 10;
if (box2[3] < 1)
{
box2[3] = 1;
}
if (box2[3] >= box2[4])
{
box2[3] = box2[4] + 10;
}
//std::cout << box2[3] << //std::endl;
}
}
}
//bottom
if (keynum2 == 65)
{
if (currentcam == 1)
{
//std::cout << " change box 3" << //std::endl;
box1[4] = box1[4] - 10;
if (box1[4] < 1)
{
box1[4] = 1;
}
if (box1[4] <= box1[3])
{
box1[4] = box1[3] + 10;
}
//std::cout << box1[4] << //std::endl;
}
if (currentcam == 2)
{
//std::cout << " change box 4" << //std::endl;
box2[4] = box2[4] - 10;
if (box2[4] < 1)
{
box2[4] = 1;
}
if (box2[4] <= box2[3])
{
box2[4] = box2[3] + 10;
}
}
}
}

```

```

        //std::cout << box2[4] << //std::endl;
    }
}
if (keynum2 == 66)
{
    if (currentcam == 1)
    {
        //std::cout << " change box 3" << //std::endl;
        box1[4] = box1[4] + 10;
        if (box1[4] > 479)
        {
            box1[4] = 479;
        }
        //std::cout << box1[4] << //std::endl;
    }
    if (currentcam == 2)
    {
        //std::cout << " change box 4" << //std::endl;
        box2[4] = box2[4] + 10;
        if (box2[4] > 479)
        {
            box2[4] = 479;
        }
        //std::cout << box2[4] << //std::endl;
    }
}
cv::Mat scurrentframe1;
cv::resize(currentframe1, scurrentframe1, cv::Size(), 1, 1);
cv::imshow("currentframe1", scurrentframe1);
cv::moveWindow("currentframe1", 0, 0);
waitKey(1);
cv::Mat scurrentframe2;
cv::resize(currentframe2, scurrentframe2, cv::Size(), 1, 1);
cv::imshow("currentframe2", scurrentframe2);
cv::moveWindow("currentframe2", 650, 0);
waitKey(1);
}
cv::destroyAllWindows();

// reread frames without box
currentframe1 = cv::imread(s3);
currentframe2 = cv::imread(s7);

//mask frames
cv::rectangle(currentframe1, cv::Point(0, 0), cv::Point(640, box1[3]), cv::Scalar(0, 0, 0, -1);
cv::rectangle(currentframe1, cv::Point(0, 0), cv::Point(box1[1], 480), cv::Scalar(0, 0, 0, -1);
cv::rectangle(currentframe1, cv::Point(0, box1[4]), cv::Point(640, 480), cv::Scalar(0, 0, 0, -1);
cv::rectangle(currentframe1, cv::Point(box1[2], 0), cv::Point(640, 480), cv::Scalar(0, 0, 0, -1);

cv::rectangle(currentframe2, cv::Point(0, 0), cv::Point(640, box2[3]), cv::Scalar(0, 0, 0, -1);
cv::rectangle(currentframe2, cv::Point(0, 0), cv::Point(box2[1], 480), cv::Scalar(0, 0, 0, -1);
cv::rectangle(currentframe2, cv::Point(0, box2[4]), cv::Point(640, 480), cv::Scalar(0, 0, 0, -1);
cv::rectangle(currentframe2, cv::Point(box2[2], 0), cv::Point(640, 480), cv::Scalar(0, 0, 0, -1);

cv::imshow("maskframe1", currentframe1); // Tum added
cv::imshow("maskframe2", currentframe2); // Tum added

// undistort frames
cv::undistort(currentframe1, undcurrentframe1, M1, D1, cv::noArray());
cv::undistort(currentframe2, undcurrentframe2, M2, D2, cv::noArray());

//remap them
cv::Mat rframe1, rframe2;
cv::remap(undcurrentframe1, rframe1, map11, map12, cv::INTER_LINEAR);
cv::remap(undcurrentframe2, rframe2, map21, map22, cv::INTER_LINEAR);

//grayscale frames
cv::cvtColor(rframe1, gframe1, COLOR_RGB2GRAY, 0);
cv::cvtColor(rframe2, gframe2, COLOR_RGB2GRAY, 0);

// threshold greyscale frames
cv::threshold(gframe1, tframe1, thresh1, 255, THRESH_BINARY); //tum adjusted

```

```

cv::threshold(gframe2, tframe2, thresh2, 255, THRESH_BINARY); //tum adjusted
cv::imshow("tframe1", tframe1);
cv::imshow("tframe2", tframe2);
cv::imshow("gframe1", gframe1); //tum adjusted
cv::imshow("gframe2", gframe2); //tum adjusted

//find contours and draw on images"
cv::findContours(tframe1, contours1, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);
std::cout << " no contours 1 =" << contours1.size() << std::endl;
if (contours1.size() > 0)
{
    cv::drawContours(gframe1, contours1, -1, cv::Scalar(255, 255, 255), 2, 8, noArray(), INT_MAX);
    cv::imshow("draw1", tframe1); //tum adjusted
}

cv::findContours(tframe2, contours2, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);
std::cout << " no contours 2 =" << contours2.size() << std::endl;
if (contours2.size() > 0)
{
    cv::drawContours(gframe2, contours2, -1, cv::Scalar(255, 255, 255), 2, 8, noArray(), INT_MAX);
    cv::imshow("draw2", tframe2); //tum adjusted
}

//draw pair
cv::Mat pair;
pair.create(480, 640 * 2, CV_8UC3);
cv::Mat part = pair.colRange(0, 640);
cvtColor(gframe1, part, cv::COLOR_GRAY2BGR);
part = pair.colRange(640, 640 * 2);
cvtColor(gframe2, part, cv::COLOR_GRAY2BGR);
//add lines
for (int j = 0; j < 480; j += 16)
{
    cv::line(pair, cv::Point(0, j), cv::Point(640 * 2, j), cv::Scalar(0, 255, 0));
}
//show pair
if (drawrectified != 0)
{
    cv::Mat spair;
    cv::resize(pair, spair, cv::Size(), 0.5, 0.5);
    cv::imshow("rectified", spair);
    cv::moveWindow("rectified", 200, 200);
    waitKey(2000);
    destroyAllWindows();
}

if (contours1.size() == 0)
{
    std::cout << " no contours 1" << contours1.size() << std::endl;
}
if (contours2.size() == 0)
{
    std::cout << " no contours 2" << contours2.size() << std::endl;
}
if (contours1.size() > 99) //99
{
    std::cout << " too many contours 1" << contours1.size() << std::endl;
}
if (contours2.size() > 99) //99
{
    std::cout << " too many contours 2" << contours2.size() << std::endl;
}

if ((contours1.size() > 0) || (contours1.size() < 100))
{
    if ((contours2.size() > 0) || (contours2.size() < 100))
    {
        //more than one contour on each side so find them and disparity for each pair and xyz and store
        //setup for finding contours
        int myr1[100];
        int myc1[100];
        int myr2[100];
    }
}

```

```

int myc2[100];
vector<float>radius1(contours1.size());
vector<vector<Point>> contours1_poly(contours1.size());
vector<Rect> boundRect1(contours1.size());
vector<Point2f>centers1(contours1.size());
vector<float>radius2(contours2.size());
vector<vector<Point>> contours2_poly(contours2.size());
vector<Rect> boundRect2(contours2.size());
vector<Point2f>centers2(contours2.size());
int matchfound = 0;
//find x y for contours1
for (int k = 0; k < contours1.size(); k++)
{
    //find row myr1 and column myc1 of contour in frame 1
    approxPolyDP(contours1[k], contours1_poly[k], 3, true);
    boundRect1[k] = boundingRect(contours1_poly[k]);
    minEnclosingCircle(contours1_poly[k], centers1[k], radius1[k]);
    myc1[k] = (int)centers1[k].x;
    myr1[k] = (int)centers1[k].y;
    std::cout << " from left camera row is " << myr1[k] << " and col is " << myc1[k] << std::endl;
}
//find x y for contours2
for (int k = 0; k < contours2.size(); k++)
{
    //find row myr2 and column myc2 of contour in frame 1
    approxPolyDP(contours2[k], contours2_poly[k], 3, true);
    boundRect2[k] = boundingRect(contours2_poly[k]);
    minEnclosingCircle(contours2_poly[k], centers2[k], radius2[k]);
    myc2[k] = (int)centers2[k].x;
    myr2[k] = (int)centers2[k].y;
    std::cout << " from right camera row is " << myr2[k] << " and col is " << myc2[k] << std::endl;
}

int tolerance = 3;
//now find matching contours
for (int k = 0; k < contours1.size(); k++)
{
    int selected = 0;
    double selecteddiff = 999;
    double diff = 999;
    for (int n = 0; n < contours2.size(); n++)
    {
        //get the lowest difference between rows and store row as selected
        diff = abs(myr1[k] - myr2[n]);
        if (diff < selecteddiff)
        {
            selecteddiff = diff;
            selected = n;
        }
    }

    //if selected one has difference less than or equal to tolerance then use pair
    if (selecteddiff <= tolerance)
    {
        matchfound = matchfound + 1;
        if (matchfound == 1) { mycount = mycount + 1; }

        //disparity is difference in x
        int d1 = myc1[k] - myc2[selected];
        std::cout << "disparity " << k << " is " << d1 << std::endl;

        //now calculate x,y,z with perspective transform for d1 from method 1
        std::vector<Point3f> inpts1;
        std::vector<Point3f> outpts1;

        inpts1.push_back(Point3f(myc1[k], myr1[k], d1));
        //std::cout << "inpts1 " << std::endl;
        std::cout << inpts1 << std::endl;
        cv::perspectiveTransform(inpts1, outpts1, Q);
        float myx = outpts1[0].x;
        float myy = outpts1[0].y; //240 -
    }
}

```

```

float myz = outputs1[0].z;

//store them
pts1[mycount][k][0] = myx;
pts1[mycount][k][1] = myy;
pts1[mycount][k][2] = myz;
std::cout << "x y z " << myx << " " << myy << " " << myz << " " << std::endl;
}

//now write out if matchfound for this pair of images
if (matchfound != 0)
{
    myfilestore << mycount;
    myfilestore << ", ";
    for (int n = 0; n < matchfound; n++)
    {
        myfilestore << n;
        myfilestore << ", ";
        myfilestore << pts1[mycount][n][0];
        myfilestore << ", ";
        myfilestore << pts1[mycount][n][1];
        myfilestore << ", ";
        myfilestore << pts1[mycount][n][2];
        myfilestore << ", ";
    }
    myfilestore << std::endl;
}
else
{
    std::cout << " no match found for this image pair smallest row difference is " << selecteddiff << std::endl;
}
}
}
}
}
myfilestore.close();

//write out boxes file
char u0[50] = "C://tum//picameraimages//picamerastereo";
char u1[50] = "";
_itoa_s(firstcam, u1, 10);
char u2[50] = "";
_itoa_s(secondcam, u2, 10);
char u3[50] = "//boxes.txt";
char u4[80] = "";
strcat_s(u4, u0);
strcat_s(u4, u1);
strcat_s(u4, u2);
strcat_s(u4, u3);

std::cout << "writing boxes file " << u4 << std::endl;
std::ofstream omyfile(u4);
if (!omyfile.is_open())
{
    std::cout << "Unable to open file" << std::endl;
    return 1;
}
omyfile << box1[1] << std::endl;
omyfile << box1[2] << std::endl;
omyfile << box1[3] << std::endl;
omyfile << box1[4] << std::endl;
omyfile << box2[1] << std::endl;
omyfile << box2[2] << std::endl;
omyfile << box2[3] << std::endl;
omyfile << box2[4] << std::endl;
omyfile.close();
return 0;
}

```

## 7.7. Coding on Raspberry Pi

```
# Web streaming example
# Source code from the official PiCamera package
# http://picamera.readthedocs.io/en/latest/recipes2.html#web-streaming

import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>Raspberry Pi - Surveillance Camera</title>
</head>
<body>
<center><h1>Raspberry Pi - Surveillance Camera</h1></center>
<center></center>
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
            except KeyboardInterrupt:
                self.wfile.write(b'--FRAME\r\n')
                self.send_header('Content-Type', 'image/jpeg')
                self.send_header('Content-Length', len(frame))
                self.end_headers()
                self.wfile.write(frame)
```

```
        self.wfile.write(b"\r\n")
    except Exception as e:
        logging.warning(
            'Removed streaming client %s: %s',
            self.client_address, str(e))
    else:
        self.send_error(404)
        self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    #Uncomment the next line to change your Pi's Camera rotation (in degrees)
    #camera.rotation = 90
    camera.start_recording(output, format='mjpeg')
    try:
        address = ("", 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```



## Appendix 8 Certificate of approval by Siriraj Institutional Review Board

2 WANGLANG Rd. BANGKOKNOI  
BANGKOK 10700



Tel. +66 2419 2667-72

Fax. +66 2411 0162

### Siriraj Institutional Review Board

#### Certificate of Approval

COA no. Si 289/2022

Protocol Title (English) : Development of a low-cost motion capture system and its 3-dimensional measurement of spinal parameters for scoliosis treatment  
Protocol Title (Thai) : การพัฒนาระบบจับการเคลื่อนไหวด้วยหุ่นยนต์และการวัดค่าพารามิเตอร์กระดูกสันหลังแบบ 3 มิติสำหรับการรักษากระดูกสันหลังคด  
SIRB Protocol No. : 151/2565(IRB2)  
Principal Investigator/Affiliation: Mr.Pitchaya Rayothee / Department of Sirindhorn School of Prosthetics and Orthotics  
Research site : Faculty of Medicine Siriraj Hospital  
Duration of research : 1 year 6 months  
Approval date : April 7, 2022  
Expired date : April 6, 2023

This is to certify that Siriraj Institutional Review Board is in full compliance with international guidelines for human research protection such as the Declaration of Helsinki, the Belmont Report, CIOMS Guidelines and the International Conference on Harmonization in Good Clinical Practice (ICH-GCP)

Handwritten signature of Assoc. Prof. Nisarat Opartkiattikul in blue ink.

(Assoc. Prof.Nisarat Opartkiattikul, M.D., Ph.D.)  
Chairperson

21 APR 2022

date

Handwritten signature of Prof. Prasit Watanapa in blue ink.

(Prof.Prasit Watanapa, M.D., Ph.D.)  
Dean of Faculty of Medicine Siriraj Hospital

22 APR 2022

date

#### Approval includes :

1. SIRB submission form, Version 2.0 date April 5, 2022
2. Proposal
3. Participant information sheet, Version 2.0 date April 4, 2022
4. Informed consent form, Version 2.0 date April 4, 2022
5. Assent form for subjects age 7-12 years, Version 2.0 date April 4, 2022
6. Assent form for subjects age 12-18 years, Version 2.0 date April 4, 2022
7. Case record form, Version 1.0 date February 18, 2022
8. Advertisement for recruitment
9. Curriculum vitae

2 WANGLANG Rd.  
BANGKOKNOI  
BANGKOK 10700



Tel. +66 2419 2667-72  
Fax. +66 2411 0162

Siriraj Institutional Review Board  
Certificate of Approval (Renewal)

COA no. Si 289/2022

Protocol Title (English) : Development of a low-cost motion capture system and its 3-dimensional measurement of spinal parameters for scoliosis treatment  
Protocol Title (Thai) : การพัฒนาระบบจับการเคลื่อนไหวต้นท่อน้ำและการวัดค่าพารามิเตอร์กระดูกสันหลังแบบ 3 มิติสำหรับการรักษากระดูกสันหลังคด  
SIRB Protocol No. : 151/2565(IRB2)  
Principal Investigator/Affiliation : Mr.Pitchaya Rayothee / Department of Sirindhorn School of Prosthetics and Orthotics  
Research site : Faculty of Medicine Siriraj Hospital  
Duration of research : 1 year 6 months  
Renewal date (1<sup>st</sup>) : April 7, 2023  
Expired date : April 6, 2024

This is to certify that Siriraj Institutional Review Board is in full compliance with international guidelines for human research protection such as the Declaration of Helsinki, the Belmont Report, CIOMS Guidelines and the International Conference on Harmonization in Good Clinical Practice (ICH-GCP)

(Prof.Naraporn Prayoonwiwat, M.D.)  
Chairperson

20 APR 2023

date

(Prof.Apichat Asavamongkolkul, M.D.)  
Dean of Faculty of Medicine Siriraj Hospital

21 APR 2023

date

Approval includes :

1. SIRB submission form, Version 2.0 date April 5, 2022
2. Proposal
3. Participant information sheet, Version 2.0 date April 4, 2022
4. Informed consent form, Version 2.0 date April 4, 2022
5. Assent form for subjects age 7-12 years, Version 2.0 date April 4, 2022
6. Assent form for subjects age 12-18 years, Version 2.0 date April 4, 2022
7. Case record form, Version 1.0 date February 18, 2022
8. Advertisement for recruitment
9. Curriculum vitae

## Appendix 9 Ethical approval from University of Strathclyde

UEC Endorsement of Ethical opinion - Mehidol University: UEC23/14 Rowe/Rayothee:  
Development of a low-cost motion capture system and its 3-dimensional  
measurement of spinal parameters for scoliosis treatment

Ethics <ethics@strath.ac.uk>

Tue 1/31/2023 8:30 PM

To: Pitchaya Rayothee <pitchaya.rayothee@strath.ac.uk>; Philip Rowe <philip.rowe@strath.ac.uk>

Cc: Ethics <ethics@strath.ac.uk>

Dear Phil, Pitchaya

I refer to the above study and can confirm that the Convener of the University Ethics Committee (UEC) has endorsed the decision of Mehidol University Research Ethics Committee and granted University ethics approval on behalf of the UEC. University insurance has also been confirmed.

I remind you that the Committee must be informed of any changes that are made to the research project, so that they have the opportunity to consider them. The Committee also expects you to report back on the progress and outcome of your project, (by providing to ethics@strath.ac.uk copies of Annual Progress Reports and an Outcome Report) and with an account of anything which may prompt ethical questions for any similar future project and with anything else that you feel the Committee should know.

Any adverse event that occurs during an investigation must be reported as quickly as possible to UEC and, within the required time frame, to Mehidol University or to any appropriate external agency.

On behalf of the Committee, I wish you success with this project.

Kind regards  
Angelique

Angelique Laverty  
University Ethics Committee Manager  
Research & Knowledge Exchange Services (RKES)  
University of Strathclyde  
Room 3.01, Graham Hills Building  
50 George Street  
Glasgow  
G1 1QE

[ethics@strath.ac.uk](mailto:ethics@strath.ac.uk)

<http://www.strath.ac.uk/rkes>



**THE QUEEN'S ANNIVERSARY PRIZES  
2019 & 2021**

For Higher and Further Education

## Appendix 10 Participant information sheet and Poster invitation

เอกสารหมายเลข 3ก (สำหรับผู้ปกครองของเด็กที่เข้าร่วมวิจัย)

### เอกสารชี้แจงผู้เข้าร่วมการวิจัย/อาสาสมัคร (Participant Information Sheet)

ในเอกสารนี้อาจมีข้อความที่เด็กในปกครองของท่านอ่านแล้วยังไม่เข้าใจ โปรดสอบถามหัวหน้าโครงการวิจัยหรือผู้แทนให้ช่วยอธิบายจนกว่าจะเข้าใจดี เด็กในปกครองของท่านอาจจะขอเอกสารนี้กลับไปอ่านที่บ้านเพื่อปรึกษา หรือกับญาติพี่น้อง เพื่อนสนิท แพทย์ประจำตัวของเด็กในปกครองของท่าน หรือแพทย์ท่านอื่น เพื่อช่วยในการตัดสินใจเข้าร่วมการวิจัย

ชื่อโครงการวิจัย การพัฒนาระบบจัดการเคลื่อนไหวต้นตูด้าและการวัดค่าพารามิเตอร์กระดูกสันหลังแบบ 3 มิติ สำหรับการรักษากระดูกสันหลังคด

ชื่อหัวหน้าโครงการวิจัย นายพิชญะ ระโยธี

สถานที่วิจัย ห้องวิจัยของโรงเรียนกายอุปกรณ์สิรินธร

สถานที่ทำงานและหมายเลขโทรศัพท์ของหัวหน้าโครงการวิจัยที่ติดต่อได้ทั้งในและนอกเวลาราชการ

โรงเรียนกายอุปกรณ์สิรินธร หมายเลขโทรศัพท์ 089-031-6810

ผู้สนับสนุนทุนวิจัย คณะแพทยศาสตร์ศิริราชพยาบาล

การมีส่วนได้ส่วนเสียกับแหล่งทุน  ไม่มี  มี ระบุ.....

ระยะเวลาในการวิจัย 1 ปี 6 เดือน

#### ที่มาของโครงการวิจัย

ภาวะกระดูกสันหลังคดโดยเฉพาะชนิดไม่ทราบสาเหตุที่เกิดขึ้นในวัยรุ่น เป็นภาวะกระดูกสันหลังคดที่พบบ่อยที่สุด ในคลินิก ส่งผลกระทบต่อคุณภาพและทำให้สูญเสียความมั่นใจในตนเอง ผู้ป่วยภาวะนี้จะมีการคดของกระดูกสันหลังเพิ่มขึ้นเร็วในช่วงที่กำลังเจริญเติบโตและมีภาวะหยุดคดเมื่อผู้ป่วยหยุดโต การรักษาประกอบด้วยการรักษาแบบผ่าตัด หากกระดูกสันหลังคดมาก (มุมกระดูกสันหลังคดมากกว่า 40 องศา) การรักษาแบบไม่ผ่าตัดโดยการใส่อุปกรณ์พยุงหลังเพื่อหยุดการคดของกระดูกสันหลัง (มุมกระดูกสันหลังคดอยู่ในช่วงระหว่าง 25 - 40 องศา) และการรักษาแบบสังเกตการณ์ (มุมกระดูกสันหลังคดน้อยกว่า 25 องศา) ด้วยการทำกายภาพบำบัด การออกกำลังกาย และการฝังเข็ม เป็นต้น

การวัดองศาของกระดูกสันหลังคดในปัจจุบัน ทำได้โดยให้ผู้ป่วยถ่ายภาพเอ็กซเรย์ เพื่อให้เห็นโครงร่างของกระดูกสันหลัง จากนั้นนำภาพที่ได้มาวัดองศากระดูกสันหลังคด โดยการวัดองศานี้จะทำตลอดระยะเวลาของการรักษา ตั้งแต่การตรวจประเมินร่างกายครั้งแรก ขณะใส่กายอุปกรณ์ การตรวจประเมินร่างกายตามรอบการนัดหมาย ทุก ๆ 6 เดือนถึง 1 ปี เป็นต้น ดังนั้น ผู้ป่วยจะต้องถ่ายภาพเอ็กซเรย์หลายครั้ง ซึ่งอาจจะส่งผลกระทบต่อสุขภาพของ

เอกสารชี้แจงผู้เข้าร่วมการวิจัย/อาสาสมัคร\_9 Aug 2021

	รูปของเอกสารการวิจัย ที่คณะแพทยศาสตร์ศิริราชพยาบาล	Version 2.0, Date April 4 <sup>th</sup> , 2022
รหัสโครงการ	151/2565	(RSZ)
COA no. S	289/2022	
รหัสประจำตัว	7 111 2565	

ผู้ป่วยในอนาคต ในปัจจุบันได้มีการนำเครื่องมือวัดที่ทันสมัยมาใช้ในการวัดองค์ประกอบของกระดูกสันหลัง เช่น การใช้ภาพถ่ายจากเครื่องอัลตราซาวด์ ภาพถ่ายจะสแกนเนอร์ 3 มิติ และระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติ

การวัดองค์ประกอบของกระดูกสันหลังด้วยระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติในปัจจุบันเป็นระบบที่มีมูลค่าสูง ยังอยู่ในขั้นตอนของการทำงานวิจัย และยังไม่ได้ถูกนำมาใช้จริงในคลินิก ดังนั้นผู้วิจัยจึงได้ทำการพัฒนาระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติที่มีต้นทุนต่ำ เพื่อใช้ในการวัดองค์ประกอบของกระดูกสันหลังคดขณะตรวจร่างกายและหล่อแบบ เพื่อนำข้อมูลที่ได้ไปประกอบการรักษาต่อไป

**ระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติต้นทุนต่ำ คืออะไร?** เป็นระบบที่ผู้วิจัยพัฒนาขึ้น โดยใช้หลักการของภาพถ่ายจากกล้องวิดีโอหลาย ๆ ตัว ถ่ายภาพวัตถุทรงกลมขึ้นเดียวกัน จากนั้นจะนำภาพที่ได้ไปคำนวณในโปรแกรมคอมพิวเตอร์ที่ผู้วิจัยพัฒนาขึ้น เพื่อคำนวณหาตำแหน่งของวัตถุทรงกลมนั้นในพื้นที 3 มิติ โดยระบบที่พัฒนาขึ้นนี้จะใช้กล้องวิดีโอจำนวน 8 ตัว ถ่ายภาพลำตัวของอาสาสมัคร โดยที่บนลำตัวของอาสาสมัครจะถูกติดวัตถุทรงกลมขนาดเล็กตามตำแหน่งต่าง ๆ ของร่างกาย จากนั้นระบบจะนำภาพที่ได้มาคำนวณหาตำแหน่งของวัตถุในพื้นที 3 มิติ และจะนำค่านี้ไปคำนวณหาค่ามุมของกระดูกสันหลังคดต่อไป

**วัตถุประสงค์ของโครงการวิจัย** เพื่อตรวจสอบความถูกต้องของระบบจับการเคลื่อนไหวต้นทุนต่ำเทียบกับค่าที่ได้จากภาพเอกซเรย์สำหรับการวัดค่าพารามิเตอร์กระดูกสันหลังในคนไข้กระดูกสันหลังคด

เด็กในปกครองของท่านได้รับเชิญให้เข้าร่วมการวิจัยนี้เนื่องจากเด็กในปกครองของท่านมีคุณสมบัติที่เหมาะสมตามเกณฑ์การรับเข้า ได้แก่

- เพศชายหรือหญิง ที่มีอายุระหว่าง 10 – 18 ปี
- มีภาวะกระดูกสันหลังคด
- มีองศาของกระดูกสันหลังคดระหว่าง 25 – 40 องศา
- มีค่าดัชนีมวลกายอยู่ในช่วง 18.5 ถึง 24.9
- ขึ้นและเดินได้ด้วยตนเอง
- ไม่เคยผ่าตัดเพื่อตัดแก้ไขกระดูกสันหลังคด
- ไม่มีปัญหาสุขภาพทางด้านหัวใจและระบบการหายใจ



จะมีผู้ร่วมวิจัย/อาสาสมัครนี้ทั้งสิ้นประมาณ 34 ราย



หากเด็กในปกครองของท่านตัดสินใจเข้าร่วมการวิจัยแล้ว จะมีขั้นตอนการวิจัยดังต่อไปนี้คือ.

เด็กในปกครองของท่านจะได้รับการนัดหมายทั้งหมด 2 ครั้ง ที่ห้องวิจัยของโรงเรียนกายอุปกรณ์สิรินธร ประกอบด้วยการซักประวัติ ตรวจประเมินร่างกาย และการหล่อแบบ โดยการนัดหมายแต่ละครั้งจะใช้เวลาประมาณ 3 ชั่วโมง

การนัดหมายครั้งที่ 1 ผู้วิจัยจะอธิบายรายละเอียดและวัตถุประสงค์ของโครงการวิจัยโดยละเอียด หากเด็กในการปกครองของท่านตัดสินใจเข้าร่วมโครงการวิจัย จะต้องเซ็นเอกสารแสดงเจตนายินยอมเข้าร่วมการวิจัย และเอกสารชี้แจง จากนั้นผู้วิจัยจะทำการสอบถามข้อมูลเด็กในปกครองของท่าน ได้แก่ อายุ อาชีพ/ระดับชั้นการศึกษา และประวัติการมีประจำเดือน นอกจากนี้ หากเด็กในปกครองของท่านไม่มีภาพถ่ายเอ็กซเรย์หรือภาพถ่ายนั้นนานกว่า 3 เดือน ผู้วิจัยจะให้เด็กในปกครองของท่านไปถ่ายภาพเอ็กซเรย์ที่โรงพยาบาลสิรินธร จากนั้นผู้วิจัยจะทำการนัดหมายครั้งที่ 2 ต่อไป

การนัดหมายครั้งที่ 2 เด็กในปกครองของท่านจะต้องปฏิบัติดังต่อไปนี้

1. ตรวจประเมินร่างกายแบบมาตรฐานด้วยนักกายอุปกรณ์ เพื่อประเมินความสมมาตรของร่างกาย
2. จากนั้นเด็กในปกครองของท่านจะถูกตรวจประเมินร่างกายอีกครั้งด้วยระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติต้นทุนต่ำ โดยนักกายอุปกรณ์
3. จากนั้นเด็กในปกครองของท่านจะถูกหล่อแบบด้วยเพือกปูนพลาสติก และเด็กในปกครองของท่านจะถูกตรวจประเมินร่างกายอีกครั้งด้วยระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติต้นทุนต่ำ โดยนักกายอุปกรณ์ (โดยทุก ๆ ขั้นตอนจะทำโดยนักกายอุปกรณ์คนเดียวกัน)

เมื่อเพือกปูนพลาสติกแข็งตัวและผู้วิจัยเก็บข้อมูลเรียบร้อยแล้ว เพือกปูนจะถูกแกะออกจากตัวเด็กในปกครองของท่านทันทีและเสร็จสิ้นกระบวนการ ข้อมูลวิจัยและเพือกปูนนี้จะไม่มีผลใด ๆ ต่อการรักษาในปัจจุบันและอนาคต

เด็กในปกครองของท่านสามารถเลือกเข้าร่วมหรือไม่เข้าร่วมโครงการวิจัยได้อย่างอิสระ เด็กในปกครองของท่านมีหน้าที่ให้ข้อมูลตามจริง เด็กในปกครองของท่านสามารถให้ข้อเสนอแนะหรือซักถามเพื่อคลายข้อสงสัยได้ตลอดเวลา อีกทั้งเด็กในปกครองของท่านยังสามารถตัดสินใจออกจากโครงการวิจัยได้ทุกเมื่อโดยไม่มีผลกระทบใด ๆ ต่อการรักษา

#### หมายเหตุ

- แบบมาตรฐาน หมายถึง การตรวจประเมินร่างกายโดยนักกายอุปกรณ์เพื่อประเมินภาวะกระดูกสันหลังคด เช่น ความสมมาตรของร่างกาย เป็นต้น นอกจากนี้ยังรวมถึงการวัดค่ากระดูกสันหลังจากภาพถ่ายเอ็กซเรย์ด้วย
- แบบที่พัฒนาขึ้น หมายถึง การตรวจประเมินร่างกายและการหล่อแบบโดยนักกายอุปกรณ์ โดยใช้ระบบวิเคราะห์การเคลื่อนไหวแบบ 3 มิติต้นทุนต่ำ ในการวัดค่ากระดูกสันหลัง

เอกสารชี้แจงผู้เข้าร่วมการวิจัย/อาสาสมัคร\_9 Aug 2021



### ความเสี่ยงที่อาจเกิดขึ้นเมื่อเข้าร่วมการวิจัย

- ขณะตรวจประเมินร่างกายและหล่อแบบ ผู้วิจัยจะออกแรงกดบนลำตัวเพื่อตัดกระดูกสันหลังให้ตรงขึ้น ดังนั้นแรงกดอาจทำให้มีผิวหนังมีรอยแดง รอยขีด หรือแผลผลอกได้ ในเบื้องต้นผู้วิจัยจะให้เด็กในปกครองของท่านสวมใส่ชุดคลุมอย่างน้อยหนึ่งชิ้นเพื่อช่วยลดแรงกดและความเจ็บปวดบนผิวหนัง แต่อย่างไรก็ตามเด็กในปกครองของท่านสามารถบอกผู้วิจัยได้ทันทีหากแรงกดนั้นมากจนเกินไป
- เมื่ออุปกรณ์พลาสติกจะถูกนำมาปกคลุมบนลำตัวของเด็กในปกครองของท่านขณะหล่อแบบ ในคนใช้บางรายอาจจะมีอาการแพ้พลาสติกเกิดขึ้น โดยจะเกิดรอยแดงหรือผื่นแดงขึ้นบนผิวหนัง ในเบื้องต้นผู้วิจัยจะปกคลุมบนลำตัวของเด็กในปกครองของท่านเพิ่มอีกชิ้นหนึ่งด้วยแผ่นพลาสติกใสแบบยืดหยุ่น เพื่อป้องกันผิวหนังสัมผัสกับพลาสติกโดยตรง

หากเด็กในปกครองของท่านไม่เข้าร่วมในโครงการวิจัยนี้ เด็กในปกครองของท่านก็จะได้รับการตรวจเพื่อการวินิจฉัยและรักษาโรคของท่านตามวิธีการที่เป็นมาตรฐานคือ ได้รับการรักษาด้วยกายอุปกรณ์เสริมลำตัวแบบมาตรฐานที่ใช้ในการรักษาทั่วไป อีกทั้งการวัดผลสัมฤทธิ์การรักษาด้วยวิธีมาตรฐานคือ ภาพถ่ายเอ็กซเรย์

หากมีข้อสงสัยที่จะสอบถามเกี่ยวข้องกับกรวิจัย หรือหากเกิดผลข้างเคียงที่ไม่พึงประสงค์จากการวิจัย เด็กในปกครองของท่านสามารถติดต่อ

นายพิษณุ ระบุโยธี                      โรงเรียนกายอุปกรณ์สิรินธร                      โทรศัพท์มือถือ 089-031-6810  
อาจารย์ ดร. มนัญญา สามาลา                      โรงเรียนกายอุปกรณ์สิรินธร                      โทรศัพท์มือถือ 090-983-6639

เด็กในปกครองของท่านจะได้รับการช่วยเหลือหรือดูแลรักษาการบาดเจ็บ/เจ็บป่วยอันเนื่องมาจากการวิจัยตามมาตรฐานทางการแพทย์ โดยผู้รับผิดชอบค่าใช้จ่ายในการรักษาคือ

คณะแพทยศาสตร์ศิริราชพยาบาล

### ประโยชน์ที่คิดว่าจะได้รับจากการวิจัย

ผู้เข้าร่วมวิจัยไม่ได้รับประโยชน์โดยตรงจากการวิจัยนี้ แต่โครงการนี้จะเป็นการพัฒนาเพื่อได้ระบบจับการเคลื่อนไหวแบบ 3 มิติต้นทุนต่ำ ที่ช่วยในการวัดกระดูกสันหลังคด โดยข้อมูลที่ได้จากระบบนี้จะช่วยให้แพทย์และนักกายอุปกรณ์ นำไปประกอบการตัดสินใจเพื่อให้การรักษาแม่นยำมากยิ่งขึ้น อีกทั้งยังช่วยลดจำนวนครั้งในการถ่ายภาพเอ็กซเรย์ ซึ่งจะส่งผลดีต่อสุขภาพของคนไข้ในอนาคต และระบบใหม่นี้ยังช่วยลดค่าใช้จ่ายในการถ่ายภาพเอ็กซเรย์ได้อีกด้วย



คำตอบแทนที่ผู้ร่วมวิจัย/อาสาสมัครจะได้รับ  
เด็กในปกครองของท่านจะไม่ได้รับคำตอบแทนจากการวิจัยนี้ แต่จะได้รับค่าเดินทางเพื่อมาตามนัดหมาย  
จำนวน 800 บาทต่อครั้ง

ค่าใช้จ่ายที่ผู้ร่วมวิจัย/อาสาสมัครจะต้องรับผิดชอบเอง ไม่มี

หากมีข้อมูลเพิ่มเติมทั้งด้านประโยชน์และโทษที่เกี่ยวข้องกับการวิจัยนี้ ผู้วิจัยจะแจ้งให้ทราบโดย  
รวดเร็วและไม่ปิดบัง

ข้อมูลส่วนตัวของผู้ร่วมวิจัย/อาสาสมัคร จะถูกเก็บรักษาไว้เป็นความลับและจะไม่เปิดเผยต่อ  
สาธารณะเป็นรายบุคคล แต่จะรายงานผลการวิจัยเป็นข้อมูลส่วนรวมกรณีเป็นการวิจัยทางคลินิก  
ผลการวิจัยในภาพรวมนี้อาจดูได้จากเว็บไซต์ (<http://www.ClinicalTrials.gov/> <http://www.ClinicalTrials.in.th.>)  
ข้อมูลของผู้ร่วมวิจัย/อาสาสมัครเป็นรายบุคคลอาจมีคณะบุคคลบางกลุ่มเข้ามาตรวจสอบได้ เช่น ผู้ให้ทุน  
วิจัย ผู้กำกับดูแลการวิจัย สถาบันหรือองค์กรของรัฐที่มีหน้าที่ตรวจสอบ รวมถึงคณะกรรมการจริยธรรม  
การวิจัยในคน เป็นต้น โดยไม่ละเมิดสิทธิของผู้ร่วมวิจัย/อาสาสมัครในการรักษาความลับเกินขอบเขตที่  
กฎหมายอนุญาตไว้

ผู้ร่วมวิจัย/อาสาสมัครมีสิทธิ์ถอนตัวออกจากโครงการวิจัยเมื่อใดก็ได้ โดยไม่ต้องแจ้งให้ทราบ  
ล่วงหน้า และการไม่เข้าร่วมการวิจัยหรือถอนตัวออกจากโครงการวิจัยนี้ จะไม่มีผลกระทบต่อค่าบริการ  
และการรักษาที่สมควรจะได้รับตามมาตรฐานแต่ประการใด

ผู้ร่วมวิจัย/อาสาสมัครอาจหาความรู้ความเข้าใจเพิ่มเติมเกี่ยวกับการเข้าร่วมโครงการวิจัยของ  
คณะแพทยศาสตร์ศิริราชพยาบาลได้จากเว็บไซต์คณะกรรมการจริยธรรมการวิจัยในคน  
<http://www.si.mahidol.ac.th/sirb> ทั้งนี้ หากท่านได้รับการปฏิบัติที่ไม่ตรงตามที่ได้ระบุไว้ ท่านสามารถ  
ร้องเรียนไปยังประธานคณะกรรมการจริยธรรมการวิจัยในคนได้ที่เว็บไซต์ดังกล่าว หรือที่สำนักงาน  
คณะกรรมการจริยธรรมการวิจัยในคน อาคารเฉลิมพระเกียรติ ๘๐ พรรษา ๕ ธันวาคม ๒๕๕๐ ชั้น 2  
โทร. ๐ 2419 2667-72, 06 3903 4255 โทรสาร ๐ 2411 0162 ทั้งนี้ ข้าพเจ้าได้รับสำเนาเอกสารชี้แจง  
ผู้เข้าร่วมการวิจัยนี้ไว้แล้ว 1 ฉบับ

ลงชื่อ.....ผู้ร่วมวิจัย/อาสาสมัคร

(.....)

วันที่.....





ลงชื่อ.....ผู้ให้ข้อมูลและขอความยินยอม/หัวหน้าโครงการวิจัย

(.....)

วันที่.....





มหาวิทยาลัยมหิดล

โรงเรียนกายอุปกรณ์สิรินธร คณะแพทยศาสตร์ศิริราชพยาบาล

ขอเชิญชวน

# อาสาสมัคร

เข้าร่วมโครงการวิจัย

ขอเชิญชวนอาสาสมัครเข้าร่วมโครงการวิจัย

เรื่องการพัฒนาระบบจัดการเคลื่อนไหวต้นท่อน้ำและการวัดค่าพารามิเตอร์

กระดูกสันหลังแบบ 3 มิติ สำหรับการรักษากระดูกสันหลังคด

## คุณสมบัติอาสาสมัคร

- ชายหรือหญิง อายุระหว่าง 10 - 18 ปี
- มีภาวะกระดูกสันหลังคด
- ดัชนีมวลกายระหว่าง 18.5 - 24.9
- ยืนและเดินได้ด้วยตนเอง
- ไม่เคยผ่าตัดเพื่อตัดแก้ไขกระดูกสันหลังคด
- ไม่มีปัญหาสุขภาพทางด้านหัวใจและระบบการหายใจ

	รับตรงโดยคณะกรรมการการวิจัยในคน คณะแพทยศาสตร์ศิริราชพยาบาล
รหัสโครงการ	-151/2565-
COA no. SI	289/2022 ;
วันที่รับตรง	เม.ย. 2565

รับจำนวน 34 คน



ติดต่อสอบถามข้อมูลเพิ่มเติม:

นายพิษณุ ระโยธิ (หัวหน้าโครงการวิจัย) โทรศัพท์มือถือ 089-031-6810

อาจารย์ ดร. มนัชญา สามาลา (ผู้วิจัย) โทรศัพท์มือถือ 090-983-6639

โรงเรียนกายอุปกรณ์สิรินธร คณะแพทยศาสตร์ศิริราชพยาบาล มห. วิทยาลัยมหิดล



และยินยอมให้ผู้วิจัยใช้ข้อมูลส่วนตัวของข้าพเจ้าที่ได้รับจากการวิจัย แต่จะไม่เผยแพร่ต่อสาธารณะเป็นรายบุคคล โดยจะนำเสนอเป็นข้อมูลโดยรวมจากการวิจัยเท่านั้น

ข้าพเจ้าได้เข้าใจข้อความในเอกสารชี้แจงผู้เข้าร่วมการวิจัย และหนังสือแสดงเจตนายินยอมนี้ โดยตลอดแล้ว จึงลงลายมือชื่อไว้เป็นหลักฐาน ทั้งนี้ ข้าพเจ้าได้รับสำเนาหนังสือแสดงเจตนายินยอมเข้าร่วมในการวิจัยไว้แล้ว 1 ฉบับ

ลงชื่อ..... ผู้ร่วมวิจัย/อาสาสมัครหรือผู้แทนโดยชอบธรรม/วันที่.....  
(.....)

ลงชื่อ..... ผู้ให้ข้อมูลและขอความยินยอม/หัวหน้าโครงการวิจัย/วันที่.....  
(.....)

ในกรณีผู้ร่วมวิจัย/อาสาสมัคร อ่านหนังสือไม่ออก มีพยานซึ่งไม่มีส่วนได้เสียอยู่ด้วยตลอดระยะเวลาที่มีการให้ข้อมูลและผู้ร่วมวิจัย/อาสาสมัครหรือผู้แทนโดยชอบธรรมให้ความยินยอมเข้าร่วมการวิจัย พยานขอยืนยันว่าข้อมูลในหนังสือแสดงเจตนายินยอมหรือข้อมูลที่ได้รับและเอกสารอื่นได้รับการอธิบายอย่างถูกต้อง และผู้ร่วมวิจัย/อาสาสมัครหรือผู้แทนโดยชอบธรรมแสดงว่าเข้าใจรายละเอียดต่างๆ พร้อมทั้งให้ความยินยอมโดยสมัครใจ จึงได้ลงลายมือชื่อไว้เป็นพยาน

ลงชื่อ..... พยาน/วันที่.....  
(.....)

ลงชื่อ..... ผู้ร่วมวิจัย/อาสาสมัครหรือผู้แทนโดยชอบธรรม  
(.....)

วันที่.....

ลงชื่อ..... พยาน  
(.....)

วันที่.....



## Appendix 12 Case record form during clinical experiment

### Case Record Form

Research Title: Development of a low-cost motion capture system and its 3-dimensional measurement of spinal parameters for scoliosis treatment

Participant number:			
---------------------	--	--	--

**Participant screening** (Participant MUST follow this criteria):

- Male or female who is between 10 - 18 years old
- Person who was diagnosed with adolescent idiopathic scoliosis (AIS) by doctor
- Cobb angle is between 25 - 40 degrees and Risser's sign is between 0 - 5
- All spinal curve patterns
- Flexibility of curve: Flexible to semi-rigid curve
- Apical vertebra from T8 and below
- BMI is between 18.5 - 24.9
- Be able to walk
- NO** any history about spinal surgery
- NO** any underlying disease that affects the lung and heart function
- NO** Psychological and communication problems

Appointment time:  First  Second  Others.....

Date of Visit: \_\_\_\_/\_\_\_\_/\_\_\_\_

Participant information:		
Age:	Year:	Month:
Gender:	<input type="checkbox"/> Male	<input type="checkbox"/> Female
Occupation:	<input type="checkbox"/> Student	<input type="checkbox"/> Other (please specify):
History of menstruation:	<input type="checkbox"/> Not	<input type="checkbox"/> Yes (please specify the time):
	<input type="checkbox"/> N/A for male participant	



Version 1.0 Date: 18 February 2022



Appointment time:  First  Second  Others.....

Date of Visit: \_\_\_\_/\_\_\_\_/\_\_\_\_

Outcome measure (OM) from <b>X-ray image</b> :				
Sagittal OM	Single	<input type="checkbox"/> Thoracic Apex at: T____ Upper end at:____ Lower end at:____	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
	Double	<input type="checkbox"/> Thoracic Apex at: T____ Upper end at:____ Lower end at:____	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
		<input type="checkbox"/> Lumbar Apex at: L____ Upper end at:____ Lower end at:____	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
Coronal OM	Coronal decompensation	<input type="checkbox"/> No decompensation Decompensation to <input type="checkbox"/> Left or <input type="checkbox"/> Right side With _____ mm.		
	Thoracic Kyphosis	Curve angle: _____°		
	Lumbar Lordosis	Curve angle: _____°		
Risser sign		<input type="checkbox"/> 0 <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5		
Triradiate cartilage		<input type="checkbox"/> Open <input type="checkbox"/> Close		

Outcome measure (OM) from <b>physical examination</b> :	
Coronal decompensation	<input type="checkbox"/> No decompensation Decompensation to <input type="checkbox"/> Left or <input type="checkbox"/> Right side With _____ mm.
Shoulder	<input type="checkbox"/> Same level <input type="checkbox"/> Right is higher <input type="checkbox"/> Left is higher
Waist	<input type="checkbox"/> Same level <input type="checkbox"/> Right is higher <input type="checkbox"/> Left is higher
Pelvic level	<input type="checkbox"/> Same level <input type="checkbox"/> Right is higher <input type="checkbox"/> Left is higher
Axillary	<input type="checkbox"/> Same level <input type="checkbox"/> Right is higher <input type="checkbox"/> Left is higher
Flexibility of curve	<input type="checkbox"/> Flexible <input type="checkbox"/> Semi-rigid <input type="checkbox"/> Rigid
Rib hump	<input type="checkbox"/> Right <input type="checkbox"/> Left
Lumbar prominent	<input type="checkbox"/> Right <input type="checkbox"/> Left
True LLD	Right: _____ cm. Left: _____ cm.
Appearance LLD	Right: _____ cm. Left: _____ cm.
Arm gap	<input type="checkbox"/> Same among of gapping area <input type="checkbox"/> Right or <input type="checkbox"/> Left _____ is more gapping area



Version 1.0 Date: 18 February 2022

Appointment time:  First  Second  Others.....

Date of Visit: \_\_\_\_/\_\_\_\_/\_\_\_\_

Outcome measure (OM) from a developing system <b>(During assessment):</b>			
Sagittal OM	Single	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
	Double	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
		<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
Coronal OM	Thoracic Kyphosis	Curve angle: _____°	
	Lumbar Lordosis	Curve angle: _____°	

Coronal decompensation	<input type="checkbox"/> No decompensation Decompensation to <input type="checkbox"/> Left or <input type="checkbox"/> Right side With _____ mm.		
Shoulder	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher
Waist	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher
Axillary	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher

Force at T curve	Corrective: _____ N. Upper counter: _____ N. Lower counter : _____ N.
Force at L curve	Corrective: _____ N. Upper counter: _____ N. Lower counter : _____ N.
Force at TL curve	Corrective: _____ N. Upper counter: _____ N. Lower counter : _____ N.

Note:



Version 1.0 Date: 18 February 2022

Appointment time:  First  Second  Others.....

Date of Visit: \_\_\_\_/\_\_\_\_/\_\_\_\_

Outcome measure (OM) from a developing system <b>(During optimal correction):</b>			
Sagittal OM	Single	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
	Double	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
		<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
Coronal OM	Thoracic Kyphosis	Curve angle: _____°	
	Lumbar Lordosis	Curve angle: _____°	

Coronal decompensation	<input type="checkbox"/> No decompensation Decompensation to <input type="checkbox"/> Left or <input type="checkbox"/> Right side With _____ mm.		
Shoulder	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher
Waist	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher
Axillary	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher

Force at T curve	Corrective: _____ N. Upper counter: _____ N. Lower counter: _____ N.
Force at L curve	Corrective: _____ N. Upper counter: _____ N. Lower counter: _____ N.
Force at TL curve	Corrective: _____ N. Upper counter: _____ N. Lower counter: _____ N.

Note:



Version 1.0 Date: 18 February 2022



Appointment time:  First  Second  Others.....

Date of Visit: \_\_\_\_/\_\_\_\_/\_\_\_\_

Outcome measure (OM) from a developing system <b>(During casting):</b>			
Sagittal OM	Single	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
	Double	<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
		<input type="checkbox"/> Left <input type="checkbox"/> Right	Curve angle: _____°
Coronal OM	Thoracic Kyphosis	Curve angle: _____°	
	Lumbar Lordosis	Curve angle: _____°	

Coronal decompensation	<input type="checkbox"/> No decompensation Decompensation to <input type="checkbox"/> Left or <input type="checkbox"/> Right side With _____ mm.		
Shoulder	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher
Waist	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher
Axillary	<input type="checkbox"/> Same level	<input type="checkbox"/> Right is higher	<input type="checkbox"/> Left is higher

Force at T curve	Corrective: _____ N. Upper counter: _____ N. Lower counter : _____ N.
Force at L curve	Corrective: _____ N. Upper counter: _____ N. Lower counter : _____ N.
Force at TL curve	Corrective: _____ N. Upper counter: _____ N. Lower counter : _____ N.

Note:



Version 1.0 Date: 18 February 2022

### Appendix 13 Statistical analysis table

- Test of normality of coronal spinal angle from radiographs and the developed motion system in assessment, optimal correction, and casting

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
CUTA_x_ray	.188	10	.200 <sup>*</sup>	.914	10	.308
CUTA_assess	.205	10	.200 <sup>*</sup>	.894	10	.186
CLTA_x_ray	.250	10	.076	.941	10	.564
CLTA_assess	.213	10	.200 <sup>*</sup>	.929	10	.434
CULA_x_ray	.151	10	.200 <sup>*</sup>	.957	10	.747
CULA_assess	.177	10	.200 <sup>*</sup>	.891	10	.173
CLLA_x_ray	.214	10	.200 <sup>*</sup>	.926	10	.405
CLLA_assess	.141	10	.200 <sup>*</sup>	.977	10	.947
CUTA_optimal	.171	10	.200 <sup>*</sup>	.935	10	.497
CLTA_optimal	.229	10	.147	.762	10	.005
CULA_optimal	.144	10	.200 <sup>*</sup>	.963	10	.817
CLLA_optimal	.147	10	.200 <sup>*</sup>	.979	10	.962
CUTA_casting	.204	10	.200 <sup>*</sup>	.910	10	.284
CLTA_casting	.206	10	.200 <sup>*</sup>	.838	10	.041
CULA_casting	.236	10	.122	.940	10	.557
CLLA_casting	.175	10	.200 <sup>*</sup>	.869	10	.096

a. Lilliefors Significance Correction

\*. This is a lower bound of the true significance.

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
C_all_x_ray	.101	40	.200 <sup>*</sup>	.979	40	.635
C_all_assess	.135	40	.064	.958	40	.144
C_all_optimal	.081	40	.200 <sup>*</sup>	.982	40	.759
C_all_casting	.069	40	.200 <sup>*</sup>	.975	40	.519

a. Lilliefors Significance Correction

\*. This is a lower bound of the true significance.

- Test of normality of sagittal spinal angle from radiographs and the developed motion system in assessment, optimal correction, and casting

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
S_all_x_ray	.200	40	.000	.911	40	.004
S_all_assess	.138	40	.053	.952	40	.089
S_all_optimal	.162	40	.010	.937	40	.028
S_all_casting	.125	40	.116	.965	40	.249

a. Lilliefors Significance Correction

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
SUTA_x_ray	.172	10	.200 <sup>*</sup>	.946	10	.620
SUTA_assess	.258	10	.058	.802	10	.015
SLTA_x_ray	.176	10	.200 <sup>*</sup>	.923	10	.380
SLTA_assess	.226	10	.161	.941	10	.562
SULA_x_ray	.271	10	.036	.787	10	.010
SULA_assess	.144	10	.200 <sup>*</sup>	.953	10	.702
SLLA_x_ray	.200	10	.200 <sup>*</sup>	.878	10	.123
SLLA_assess	.179	10	.200 <sup>*</sup>	.946	10	.616
SUTA_optimal	.211	10	.200 <sup>*</sup>	.902	10	.230
SLTA_optimal	.200	10	.200 <sup>*</sup>	.900	10	.219
SULA_optimal	.211	10	.200 <sup>*</sup>	.911	10	.286
SLLA_optimal	.274	10	.032	.828	10	.032
SUTA_casting	.242	10	.101	.894	10	.186
SLTA_casting	.260	10	.055	.901	10	.223
SULA_casting	.123	10	.200 <sup>*</sup>	.959	10	.777
SLLA_casting	.260	10	.053	.899	10	.212

a. Lilliefors Significance Correction

\*. This is a lower bound of the true significance.

- Test of normality of 3-dimensional spinal angle from radiographs and the developed motion system in assessment, optimal correction, and casting

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
D3UTA_assess	.252	10	.071	.775	10	.007
D3UTA_optimal	.195	10	.200 <sup>*</sup>	.912	10	.293
D3UTA_casting	.234	10	.129	.893	10	.182
D3LTA_assess	.194	10	.200 <sup>*</sup>	.934	10	.488
D3LTA_optimal	.238	10	.114	.850	10	.059
D3LTA_casting	.320	10	.005	.841	10	.046
D3ULA_assess	.216	10	.200 <sup>*</sup>	.890	10	.170
D3ULA_optimal	.184	10	.200 <sup>*</sup>	.909	10	.274
D3ULA_casting	.208	10	.200 <sup>*</sup>	.927	10	.419
D3LLA_assess	.253	10	.069	.876	10	.116
D3LLA_optimal	.243	10	.097	.868	10	.095
D3LLA_casting	.260	10	.054	.901	10	.224

a. Lilliefors Significance Correction

\*. This is a lower bound of the true significance.

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
D3_all_assess	.215	40	.000	.923	40	.009
D3_all_optimal	.151	40	.022	.933	40	.020
D3_all_casting	.131	40	.083	.948	40	.066

a. Lilliefors Significance Correction

- Test of normality of trunk balance from radiographs and the developed motion system in assessment, optimal correction, and casting

**Tests of Normality**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Decom_x_ray	.164	10	.200*	.910	10	.280
Decom_assess	.160	10	.200*	.910	10	.279
Decom_optimal	.153	10	.200*	.908	10	.267
Decom_casting	.185	10	.200*	.949	10	.655

a. Lilliefors Significance Correction

\*. This is a lower bound of the true significance.

- Test of correlation of coronal spinal angle between radiographs and the developed motion system in assessment

**Correlations**

		CUTA_x_ray	CUTA_assess
CUTA_x_ray	Pearson Correlation	1	.958**
	Sig. (2-tailed)		.000
	N	10	10
CUTA_assess	Pearson Correlation	.958**	1
	Sig. (2-tailed)	.000	
	N	10	10

\*\* . Correlation is significant at the 0.01 level (2-tailed).

**Correlations**

		CLTA_x_ray	CLTA_assess
CLTA_x_ray	Pearson Correlation	1	.568
	Sig. (2-tailed)		.087
	N	10	10
CLTA_assess	Pearson Correlation	.568	1
	Sig. (2-tailed)	.087	
	N	10	10

**Correlations**

		CULA_x_ray	CULA_assess
CULA_x_ray	Pearson Correlation	1	.878**
	Sig. (2-tailed)		.001
	N	10	10
CULA_assess	Pearson Correlation	.878**	1
	Sig. (2-tailed)	.001	
	N	10	10

\*\* . Correlation is significant at the 0.01 level (2-tailed).

**Correlations**

		CLLA_x_ray	CLLA_assess
CLLA_x_ray	Pearson Correlation	1	.961**
	Sig. (2-tailed)		.000
	N	10	10
CLLA_assess	Pearson Correlation	.961**	1
	Sig. (2-tailed)	.000	
	N	10	10

\*\* . Correlation is significant at the 0.01 level (2-tailed).

**Correlations**

		C_all_x_ray	C_all_assess
C_all_x_ray	Pearson Correlation	1	.915**
	Sig. (2-tailed)		.000
	N	40	40
C_all_assess	Pearson Correlation	.915**	1
	Sig. (2-tailed)	.000	
	N	40	40

\*\* . Correlation is significant at the 0.01 level (2-tailed).

- Paired sample t-test of reduction of apical translation of the developed motion system in assessment VS optimal correction and assessment VS casting

**Paired Samples Test**

		Paired Differences				t	df	Sig. (2-tailed)	
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower				Upper
Pair 1	AssessDistance - OptimalDistance	6.56150	6.43652	1.01770	4.50300	8.62000	6.447	39	.000
Pair 2	AssessDistance - CastDistance	7.87950	6.71627	1.06194	5.73153	10.02747	7.420	39	.000

- Paired sample t-test of coronal spinal angle of the developed motion system in assessment VS optimal correction and assessment VS casting

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	CUTA_assess - CUTA_optimal	1.55600	2.44279	.77248	-.19147	3.30347	2.014	9	.075
Pair 2	CLTA_assess - CLTA_optimal	-.60900	4.02555	1.27299	-3.48871	2.27071	-.478	9	.644
Pair 3	CULA_assess - CULA_optimal	-1.45300	7.13751	2.25708	-6.55886	3.65286	-.644	9	.536
Pair 4	CLLA_assess - CLLA_optimal	5.45100	10.33857	3.26934	-1.94476	12.84676	1.667	9	.130
Pair 5	C_all_assess - C_all_optimal	1.23625	6.98910	1.10507	-.99897	3.47147	1.119	39	.270

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	CUTA_assess - CUTA_casting	1.13900	2.04876	.64788	-.32660	2.60460	1.758	9	.113
Pair 2	CLTA_assess - CLTA_casting	-.05500	2.73110	.86365	-2.00871	1.89871	-.064	9	.951
Pair 3	CULA_assess - CULA_casting	-2.23900	7.40461	2.34154	-7.53594	3.05794	-.956	9	.364
Pair 4	CLLA_assess - CLLA_casting	6.79000	11.08138	3.50424	-1.13714	14.71714	1.938	9	.085
Pair 5	C_all_assess - C_all_casting	1.40875	7.42195	1.17351	-.96490	3.78240	1.200	39	.237

- Paired t-test for equivalence of coronal spinal angle of the developed motion system in optimal correction VS casting, the margin of equivalence = 5

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-5 < [\text{Mean of (CUTA\_optimal)} - (\text{CUTA\_casting})] < 5$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -5	-0.417	0.4642079	9.8727	9	0.00000	Yes
Upper Boundary	Diff < 5	-0.417	0.4642079	-11.6693	9	0.00000	Yes
Equivalence	-5 < Diff < 5	-0.417				0.00000	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-5 < [\text{Mean of (CLTA\_optimal)} - (\text{CLTA\_casting})] < 5$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -5	0.554	0.9646384	5.7576	9	0.00014	Yes
Upper Boundary	Diff < 5	0.554	0.9646384	-4.6090	9	0.00064	Yes
Equivalence	-5 < Diff < 5	0.554				0.00064	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-5 < [\text{Mean of (CULA\_optimal)} - (\text{CULA\_casting})] < 5$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -5	-0.786	1.418205	2.9714	9	0.00783	Yes
Upper Boundary	Diff < 5	-0.786	1.418205	-4.0798	9	0.00138	Yes
Equivalence	-5 < Diff < 5	-0.786				0.00783	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-5 < [\text{Mean of (CLLA\_optimal)} - (\text{CLLA\_casting})] < 5$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -5	1.339	0.7858647	8.0663	9	0.00001	Yes
Upper Boundary	Diff < 5	1.339	0.7858647	-4.6586	9	0.00059	Yes
Equivalence	-5 < Diff < 5	1.339				0.00059	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-5 < [\text{Mean of (C\_all\_optimal)} - (\text{C\_all\_casting})] < 5$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -5	0.1725	0.4853404	10.6575	39	0.00000	Yes
Upper Boundary	Diff < 5	0.1725	0.4853404	-9.9466	39	0.00000	Yes
Equivalence	-5 < Diff < 5	0.1725				0.00000	Yes

† "Diff" refers to the Mean of the Paired Differences.

- Test of correlation of sagittal spinal angle between radiographs and the developed motion system in assessment

		SUTA_x_ray	SUTA_asses s
SUTA_x_ray	Pearson Correlation	1	.526
	Sig. (2-tailed)		.118
	N	10	10
SUTA_asses	Pearson Correlation	.526	1
	Sig. (2-tailed)	.118	
	N	10	10

		SLTA_x_ray	SLTA_assess
SLTA_x_ray	Pearson Correlation	1	.731*
	Sig. (2-tailed)		.016
	N	10	10
SLTA_assess	Pearson Correlation	.731*	1
	Sig. (2-tailed)	.016	
	N	10	10

\*. Correlation is significant at the 0.05 level (2-tailed).

		SULA_x_ray	SULA_asses s
SULA_x_ray	Pearson Correlation	1	.348
	Sig. (2-tailed)		.324
	N	10	10
SULA_asses	Pearson Correlation	.348	1
	Sig. (2-tailed)	.324	
	N	10	10

		SLLA_x_ray	SLLA_assess
SLLA_x_ray	Pearson Correlation	1	.935**
	Sig. (2-tailed)		.000
	N	10	10
SLLA_assess	Pearson Correlation	.935**	1
	Sig. (2-tailed)	.000	
	N	10	10

\*\* Correlation is significant at the 0.01 level (2-tailed).

		S_all_x_ray	S_all_assess
S_all_x_ray	Pearson Correlation	1	.912**
	Sig. (2-tailed)		.000
	N	40	40
S_all_assess	Pearson Correlation	.912**	1
	Sig. (2-tailed)	.000	
	N	40	40

\*\* Correlation is significant at the 0.01 level (2-tailed).

- Paired t-test for equivalence of sagittal spinal angle of the developed motion system in assessment VS optimal correction, the margin of equivalence = 9

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-9 < [\text{Mean of (SUTA\_asses)} - (\text{SUTA\_optimal})] < 9$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -9	-2.036	1.00519	6.9280	9	0.00003	Yes
Upper Boundary	Diff < 9	-2.036	1.00519	-10.9790	9	0.00000	Yes
Equivalence	-9 < Diff < 9	-2.036				0.00003	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-9 < [\text{Mean of (SLTA\_assess)} - (\text{SLTA\_optimal})] < 9$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -9	-2.243	1.632368	4.1394	9	0.00126	Yes
Upper Boundary	Diff < 9	-2.243	1.632368	-6.8875	9	0.00004	Yes
Equivalence	-9 < Diff < 9	-2.243				0.00126	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-9 < [\text{Mean of (SULA\_assess)} - (\text{SULA\_optimal})] < 9$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -9	3.017	1.222884	9.8268	9	0.00000	Yes
Upper Boundary	Diff < 9	3.017	1.222884	-4.8925	9	0.00043	Yes
Equivalence	-9 < Diff < 9	3.017				0.00043	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-9 < [\text{Mean of (SLLA\_assess)} - (\text{SLLA\_optimal})] < 9$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -9	-4.715	1.909073	2.2445	9	0.02573	Yes
Upper Boundary	Diff < 9	-4.715	1.909073	-7.1841	9	0.00003	Yes
Equivalence	-9 < Diff < 9	-4.715				0.02573	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-9 < [\text{Mean of (S\_all\_assess)} - (\text{S\_all\_optimal})] < 9$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -9	-1.49425	0.8432008	8.9015	39	0.00000	Yes
Upper Boundary	Diff < 9	-1.49425	0.8432008	-12.4457	39	0.00000	Yes
Equivalence	-9 < Diff < 9	-1.49425				0.00000	Yes

† "Diff" refers to the Mean of the Paired Differences.

- Paired t-test for equivalence of sagittal spinal angle of the developed motion system in assessment VS casting, the margin of equivalence = 8

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SUTA\_assess)} - (\text{SUTA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	-0.541	1.048559	7.1136	9	0.00003	Yes
Upper Boundary	Diff < 8	-0.541	1.048559	-8.1455	9	0.00001	Yes
Equivalence	-8 < Diff < 8	-0.541				0.00003	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SLTA\_assess)} - (\text{SLTA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	-0.035	1.078881	7.3826	9	0.00002	Yes
Upper Boundary	Diff < 8	-0.035	1.078881	-7.4475	9	0.00002	Yes
Equivalence	-8 < Diff < 8	-0.035				0.00002	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SULA\_assess)} - (\text{SULA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	4.066	1.77624	6.7930	9	0.00004	Yes
Upper Boundary	Diff < 8	4.066	1.77624	-2.2148	9	0.02701	Yes
Equivalence	-8 < Diff < 8	4.066				0.02701	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SLLA\_assess)} - (\text{SLLA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	-2.974	2.160969	2.3258	9	0.02253	Yes
Upper Boundary	Diff < 8	-2.974	2.160969	-5.0783	9	0.00033	Yes
Equivalence	-8 < Diff < 8	-2.974				0.02253	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (S\_all\_assess)} - (\text{S\_all\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	0.129	0.8637977	9.4108	39	0.00000	Yes
Upper Boundary	Diff < 8	0.129	0.8637977	-9.1121	39	0.00000	Yes
Equivalence	-8 < Diff < 8	0.129				0.00000	Yes

† "Diff" refers to the Mean of the Paired Differences.

- Paired t-test for equivalence of sagittal spinal angle of the developed motion system in optimal correction VS casting, the margin of equivalence = 8

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SUTA\_optimal)} - (\text{SUTA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.495	1.171194	8.1071	9	0.00001	Yes
Upper Boundary	Diff < 8	1.495	1.171194	-5.5542	9	0.00018	Yes
Equivalence	-8 < Diff < 8	1.495				0.00018	Yes

† "Diff" refers to the Mean of the Paired Differences.



**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SLTA\_optimal)} - (\text{SLTA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	2.208	1.564295	6.5256	9	0.00005	Yes
Upper Boundary	Diff < 8	2.208	1.564295	-3.7026	9	0.00245	Yes
Equivalence	-8 < Diff < 8	2.208				0.00245	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SULA\_optimal)} - (\text{SULA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.049	1.689973	5.3545	9	0.00023	Yes
Upper Boundary	Diff < 8	1.049	1.689973	-4.1131	9	0.00131	Yes
Equivalence	-8 < Diff < 8	1.049				0.00131	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (SLLA\_optimal)} - (\text{SLLA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.741	2.985273	3.2630	9	0.00490	Yes
Upper Boundary	Diff < 8	1.741	2.985273	-2.0966	9	0.03274	Yes
Equivalence	-8 < Diff < 8	1.741				0.03274	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (S\_all\_optimal)} - (\text{S\_all\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.62325	0.9506449	10.1229	39	0.00000	Yes
Upper Boundary	Diff < 8	1.62325	0.9506449	-6.7078	39	0.00000	Yes
Equivalence	-8 < Diff < 8	1.62325				0.00000	Yes

† "Diff" refers to the Mean of the Paired Differences.

- Paired sample t-test of 3-dimensional spinal angle of the developed motion system in assessment VS optimal correction and assessment VS casting

		Paired Differences				t	df	Sig. (2-tailed)	
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower				Upper
Pair 1	D3UTA_assess - D3UTA_optimal	-2.11400	2.90257	.91787	-4.19037	-.03763	-2.303	9	.047
Pair 2	D3LTA_assess - D3LTA_optimal	-1.73000	5.89267	1.86343	-5.94536	2.48536	-.928	9	.377
Pair 3	D3ULA_assess - D3ULA_optimal	2.72000	4.54232	1.43641	-.52938	5.96938	1.894	9	.091
Pair 4	D3LLA_assess - D3LLA_optimal	-9.11800	14.42348	4.56111	-19.43594	1.19994	-1.999	9	.077
Pair 5	D3_all_assess - D3_all_optimal	-2.56050	9.00726	1.42417	-5.44116	.32016	-1.798	39	.080

Paired Samples Test									
		Paired Differences				t	df	Sig. (2-tailed)	
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower				Upper
Pair 1	D3UTA_assess - D3UTA_casting	-1.22700	3.26870	1.03365	-3.56529	1.11129	-1.187	9	.266
Pair 2	D3LTA_assess - D3LTA_casting	-.00100	2.84374	.89927	-2.03529	2.03329	-.001	9	.999
Pair 3	D3ULA_assess - D3ULA_casting	4.64500	5.74786	1.81763	.53323	8.75677	2.556	9	.031
Pair 4	D3LLA_assess - D3LLA_casting	-7.16000	7.14737	2.26020	-12.27292	-2.04708	-3.168	9	.011
Pair 5	D3_all_assess - D3_all_casting	-.93575	6.47380	1.02360	-3.00617	1.13467	-.914	39	.366

- Paired t-test for equivalence of 3-dimensional spinal angle of the developed motion system in optimal correction VS casting, the margin of equivalence = 8

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (D3UTA\_optimal)} - (\text{D3UTA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	0.887	0.9108117	9.7572	9	0.00000	Yes
Upper Boundary	Diff < 8	0.887	0.9108117	-7.8095	9	0.00001	Yes
Equivalence	-8 < Diff < 8	0.887				0.00001	Yes

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (D3LTA\_optimal)} - (\text{D3LTA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.729	1.743141	5.5813	9	0.00017	Yes
Upper Boundary	Diff < 8	1.729	1.743141	-3.5975	9	0.00289	Yes
Equivalence	-8 < Diff < 8	1.729				0.00289	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (D3ULA\_optimal)} - (\text{D3ULA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.925	1.809324	5.4855	9	0.00019	Yes
Upper Boundary	Diff < 8	1.925	1.809324	-3.3576	9	0.00421	Yes
Equivalence	-8 < Diff < 8	1.925				0.00421	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (D3LLA\_optimal)} - (\text{D3LLA\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.958	3.268852	3.0463	9	0.00694	Yes
Upper Boundary	Diff < 8	1.958	3.268852	-1.8484	9	0.04880	Yes
Equivalence	-8 < Diff < 8	1.958				0.04880	Yes

† "Diff" refers to the Mean of the Paired Differences.

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-8 < [\text{Mean of (D3\_all\_optimal)} - (\text{D3\_all\_casting})] < 8$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -8	1.62475	1.016533	9.4682	39	0.00000	Yes
Upper Boundary	Diff < 8	1.62475	1.016533	-6.2716	39	0.00000	Yes
Equivalence	-8 < Diff < 8	1.62475				0.00000	Yes

† "Diff" refers to the Mean of the Paired Differences.

- Test of correlation of trunk balance between radiographs and the developed motion system in assessment

**Correlations**

		Decom_x_ray	Decom_assess
Decom_x_ray	Pearson Correlation	1	.907**
	Sig. (2-tailed)		.000
	N	10	10
Decom_assess	Pearson Correlation	.907**	1
	Sig. (2-tailed)	.000	
	N	10	10

\*\* . Correlation is significant at the 0.01 level (2-tailed).

- Paired sample t-test of trunk balance of the developed motion system in assessment VS optimal correction and assessment VS casting

**Paired Samples Test**

		Paired Differences				t	df	Sig. (2-tailed)	
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower				Upper
Pair 1	Decom_assess - Decom_optimal	-6.94000	10.58726	3.34798	-14.51367	.63367	-2.073	9	.068

**Paired Samples Test**

		Paired Differences				t	df	Sig. (2-tailed)	
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower				Upper
Pair 1	Decom_assess - Decom_casting	-7.47900	11.97210	3.78591	-16.04333	1.08533	-1.975	9	.080

- Paired t-test for equivalence of trunk balance of the developed motion system in optimal correction VS casting, the margin of equivalence = 3

**Paired T-Test for Equivalence using TOST (Two One-Sided Tests)**

Equivalence Hypothesis:  $-3 < [\text{Mean of (Decom\_optimal)} - (\text{Decom\_casting})] < 3$

Test	Alternative Hypothesis†	Mean Difference	Standard Error	T-Statistic	DF	Prob Level	Reject H0 at $\alpha = 0.050?$
Lower Boundary	Diff > -3	-0.539	1.079523	2.2797	9	0.02429	Yes
Upper Boundary	Diff < 3	-0.539	1.079523	-3.2783	9	0.00478	Yes
Equivalence	-3 < Diff < 3	-0.539				0.02429	Yes

† "Diff" refers to the Mean of the Paired Differences.

## Appendix 14 Abstract book for presentations at peer reviewed conferences

- Pitchaya Rayothee, Philip Rowe. (2021). A new low-cost motion capture system for assisting scoliosis treatment. Oral Presentation at International Society for Prosthetics and Orthotics (ISPO) 18<sup>th</sup> World Congress, Virtual presentation. November 2021.



### ISPO 18th WORLD CONGRESS virtual edition 1-4 November 2021

Digital Transformation in an Evolving World

### ABSTRACT BOOK



[www.ispo-congress.com](http://www.ispo-congress.com)

## 2.4.2.d

## A New Low-Cost Motion Capture System for Assisting Scoliosis Treatment

Pitchaya Rayothee<sup>1,2</sup>, Philip Rowe<sup>1</sup><sup>1</sup>University of Strathclyde, Glasgow, United Kingdom. <sup>2</sup>Sirindhorn School of Prosthetics and Orthotics (SSPO), Faculty of Medicine Siriraj Hospital, Mahidol University, Bangkok, Thailand

## BACKGROUND

Scoliosis is a three-dimensional spinal deformity and the radiographic result is a standard outcome measure. However, it only provides a two-dimensional result which is not enough to represent the whole deformity. Motion capture is an alternative method to measure the spinal outcome, however, the current system is expensive, not available in the clinic, and not specific for scoliosis application.

## AIM

To develop a low-cost motion capture system for assisting scoliosis treatment and to investigate its accuracy.

## METHOD

Raspberry pi with its camera was used for image processing and the main computer with developing software was used for calculating the camera calibration, camera position, and marker position. Raspberry pi communicated with the main computer wirelessly via a Wi-Fi router. In the current stage, three raspberry pi with its camera were initially used to capture the shape and investigate the system accuracy with a half-circle surface object. Multiple white markers were attached on the surface and the program calculated marker positions. Finally, the root-mean-square error (RMSE) was calculated by using the values from the calculation in the program and known marker position.

## RESULTS

For calculating marker position, these required at least one pair of cameras, and there will be 8 cameras in the whole system. However, the current stage started using three cameras to see the possibility of this development and it was enough to minimize the error by using 3 pair cameras (pair of cameras 1&2, 2&3, and 1&3). Moreover, the error was only focused on the marker position that could be seen by all pairs. Finally, the RMSE of X, Y, Z, and total error was 1.65mm, 1.44mm, 1.84mm, and 1.65mm respectively.

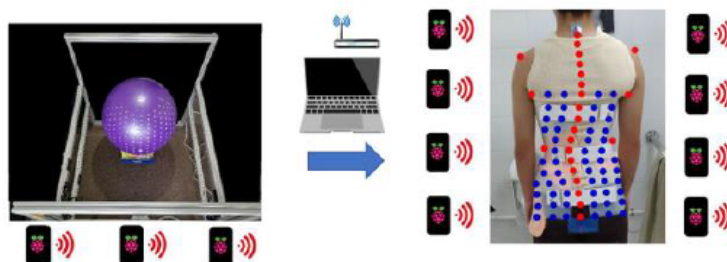


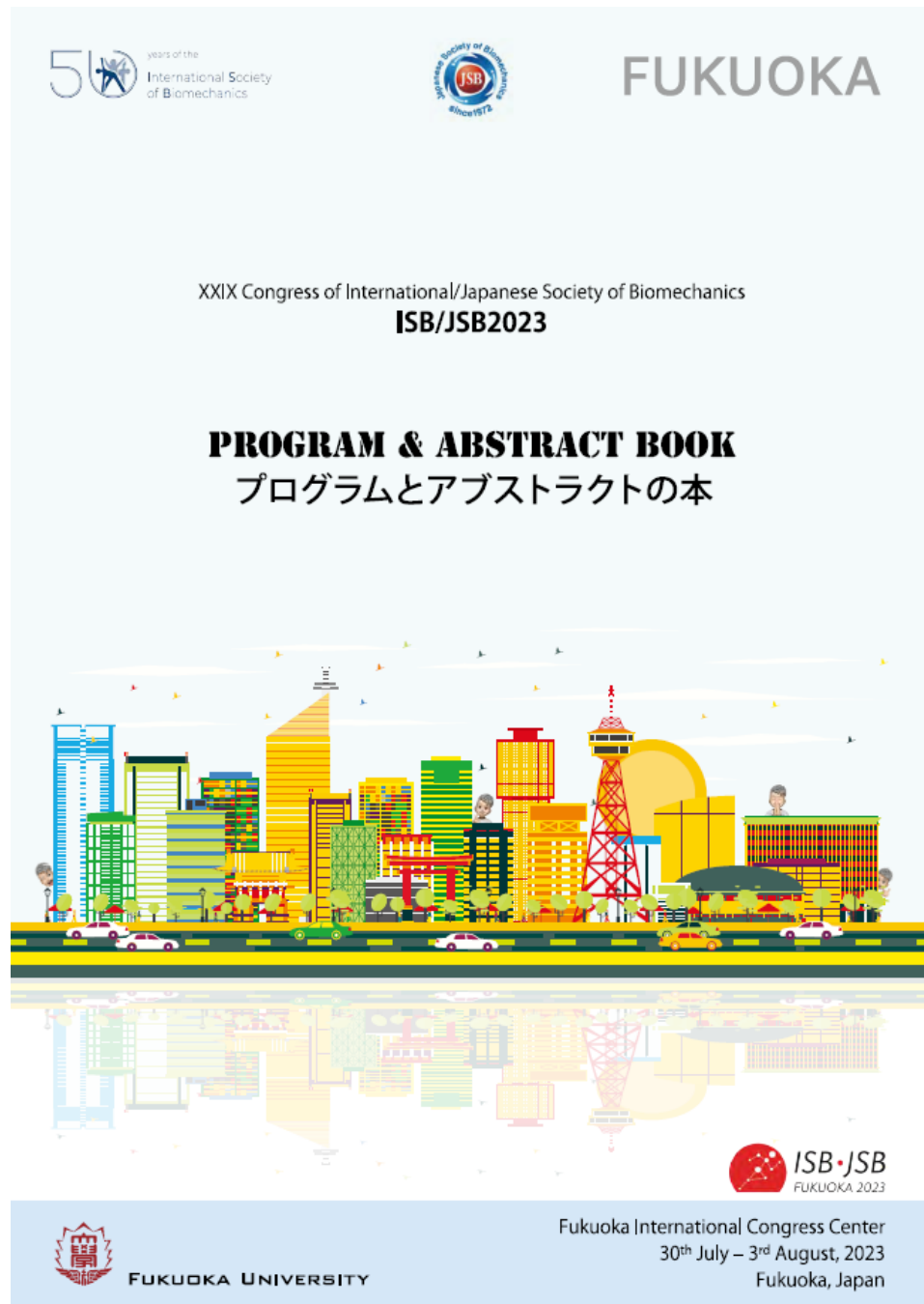
Figure 1: A new low-cost motion capture system, a). using 3 cameras to capture the surface and b) a conceptual idea by using 8 cameras to capture spinal parameters and shape

## DISCUSSION AND CONCLUSION

Comparing the accuracy with a high-cost motion system, the error was less than 1mm in static and 2mm in dynamic conditions. The accuracy of a low-cost system was between  $\pm 5\text{mm}$  to  $\pm 10\text{mm}$  and the suggested error was  $\pm 5\text{mm}$ . Therefore, our system showed an acceptable error ( $<5\text{mm}$ ). Next step, the study will increase more cameras and use them to capture the spinal parameters to assist the scoliosis treatment.

**ACKNOWLEDGEMENTS:** Thanks, Faculty of Medicine Siriraj Hospital for scholarship. Thanks, University of Strathclyde and supervisor who always support my Ph.D. study.

- Pitchaya Rayothee, Philip Rowe. (2023). Quantifying spinal parameters for scoliosis using a low-cost 3-dimensional motion capture system. Oral Presentation at International Society of Biomechanics (Shands and Eisberg) 2023, Fukuoka, Japan. August 2023.





**Quantifying Spinal Parameters for Scoliosis Using A Low-Cost 3-Dimensional Motion Capture System**  
Pitchaya Rayothee<sup>1,2</sup> and Professor Philip Rowe<sup>1</sup>

<sup>1</sup>Department of Biomedical Engineering, University of Strathclyde, Glasgow, Scotland.

<sup>2</sup>Faculty of Medicine Siriraj Hospital, Mahidol University, Bangkok, Thailand  
Email: Philip.rowe@strath.ac.uk

**INTRODUCTION**

A lateral deviation of the spine of more than 10° is considered scoliosis [1]. Adolescent idiopathic scoliosis accounts for 80% of cases and is more common in girls than boys [2]. Scoliosis treatment options include non-surgical and surgical correction. Spinal orthosis is recommended as a treatment option to delay the curve progression [1]. Radiography is a standard method for quantifying outcomes [1]; however, it can be harmful and cannot be described in three dimensions (3D). A motion capture system is another option; however, current systems are expensive and not available in clinics [3]. Therefore, the study objective was to develop an alternative method to quantify the spinal parameters (SP) for scoliosis using a low-cost 3D motion capture system.

**METHODS**

We developed a low-cost 3D motion capture system which utilised eight Raspberry Pis (3B+) each with a camera which streamed to a local network real time images wirelessly. A central, windows based computer running a purpose written C++ programme in Visual Studio 2019 collected and analysed the video frames from the cameras. White markers were used to track key anatomical landmarks. Twenty-five passive markers were attached using double sided wig tape to the spinous processes of C7/T7/T12/L2/L5; acromion processes; axilla level; inferior angle of scapula; waist level; at the same level as T12/L2/L5; PSIS; xyphoid process; umbilicus; middle distance between xyphoid and umbilicus; and middle distance between ASIS.

In the initial test phase, a scoliotic manikin was used to quantify relevant spinal parameters and check for accuracy and precision. After calibrating the camera and attaching the marker, the manikin was positioned in the center of the camera frame (Figure 1.a). The developing software then determined the locations of the markers prior to calculating the SP during the data processing method. The system was then used to measure before and after orthotic casting in a series of patients to determine the amount of change produced in spinal curvature during casting.

**RESULTS AND DISCUSSION**

The graphical user interface (GUI) displayed fourteen SP, including the level of shoulder, axilla, and waist; distance between left and right side to the central line at the axilla and waist level; thoracic kyphosis; lumbar lordosis; Cobb angle at T7 and L2; trunk horizontal

rotation at T7, T12, L2 and L5 level; and coronal decompensation (Figure 1.b). Moreover, there were another sixteen SP for further evaluation, including upper and lower thoracic, upper and lower lumbar in sagittal, coronal, and 3D angles with respect to the horizontal plan. The results displayed on the GUI were the most common SP observed by clinicians during the assessment process. Additionally, another set of SP were previously studied by Jane [3] and included in this study for further analysis.

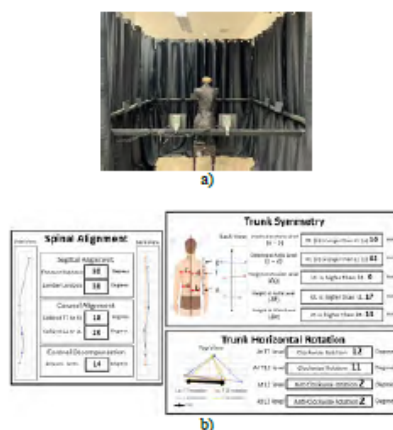


Figure 1 a) A low-cost 3D motion capture system and b) Results of spinal parameters and a graphical representation of the trunk.

**CONCLUSIONS**

A low-cost 3D motion capture system technique can be developed to quantify appropriate SP in clinics. In a clinical situation, the clinician normally observes by eye; however, the motion system developed could assist in quantifying and reporting the result and hence provide a scientific basis on which to develop future practice.

**ACKNOWLEDGEMENTS**

I would like to thank the University of Strathclyde for my Ph.D. study and thank the Faculty of Medicine Siriraj Hospital, Mahidol University, for my PhD scholarship.

**REFERENCES**

- [1] Negrini S et al. *J Scoliosis* 10:8, 2015.
- [2] Negrini S et al. *J Scoliosis and Spinal Disorders* 13:3, 2016.
- [3] Jang SH. *Thesis [Ph.D.]*