University of Strathclyde

Department of Bioengineering

Msc Thesis

Shared Control in Brain Computer Interfaces

Author:
Caroline Davie

Supervisor:
Dr. Heba Lakany

September 11, 2012

Declaration

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.

Signed:

Date:

Abstract

Patients with spinal cord injuries (SCIs) can have extremely restricted movement and independent transport functionality. Brain computer interfaces (BCIs) offer the potential to enable robotic wheelchair control through the monitoring and categorisation of brain activity signals. This enables users to control the wheelchair through the use of predefined thought patterns. However, the process of continually executing commands can be tiring, and errors could have potentially dangerous consequences.

Shared control interfaces aim to address these issues by inserting an intelligent computer interface between the user's brain activity signal commands and the output actuated by the wheelchair. The computer should be able to appraise both user commands and the surrounding environment to compute appropriate wheelchair behaviour. Information regarding the surrounding environment can be either collected from sensors attached to the chair, or be made available through stored map data.

The Bioengineering Department at the University of Strathclyde has developed a virtual reality (VR) model of an electric wheelchair that can be used to navigate in a virtual environment. The VR wheelchair can be used as a platform to develop and test the wheelchair's performance. A shared control system has been developed which is able to perform several desirable functions, however can be augmented and improved upon through further research and development.

This project involves reviewing the current system and upgrading or adding new functionality as required. In addition, both pre-existing and additional functionality has been documented and quantized. Furthermore, new maps have been developed to facilitate in the testing of the system.

The major outcomes of this project include implementing significant improvements

and testing facilities into the shared control system, and providing documentation to allow continual development.

Acknowledgements

I would like to extend my thanks to everyone who has supported me during my studies and through the completion of this thesis. My primary thanks go to my supervisor, Dr. Heba Lakany, for her invaluable support and advice throughout this project. Additionally, I am thankful to all of my lecturers, whose teaching and enthusiasm have developed my skills and kindled my passion for a variety of subjects.

I would like to extend my sincere appreciation to all of my colleagues and friends in the Bioengineering department. Their support and friendship has made it a pleasure to work with them. A special mention should be made towards my lab colleges, particuarily Niall, who was there to encourage me and assist with simulator setup.

Particular thanks must be made towards Bartek and Radhika, who had developed the simulator and initial shared control system. And thanks again to Radhika, for always being available to answer any questions.

I would like to express my gratitude to my family, in particular my mother, for being there whenever I needed her. Finally, I give my thanks to my partner, James, for his love and companionship, and for strengthening me in times of self-doubt.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the United Kingdom alone, there are approximately 1,200 people paralysed each year due to spinal cord injury [1]. Paralysis can be roughly categorised depending on the position and extent of the injury. Paraplegia involves paralysis of the lower extremities, with full functionality retained in the upper limbs. Tetraplegia (or quadriplegia) involves paralysis to some extent in all four limbs, and is caused by damage to the spinal cord between segments C1-C7. The cause of injury determines the extent of damage; with partial paralysis, some motor or sensory function may be retained, and the chances of curative rehabilitation are much higher than with total paralysis, which generally involves complete severance of the spinal cord. Approximately 20% of SCIs are classified as complete quadriplegia.

The annual cost to the UK due to SCI is estimated at £1 billion per year [2]. This figure includes hospital costs, rehabilitation costs, long-term care and compensation. There is very often some degree of long-term aftercare needed with all forms of quadriplegia, due to very much reduced mobility. Developing solutions for independent mobility would reduce the degree of care needed in addition to providing a degree of independence to the user, thus improving their quality of life. The use of an electric wheelchair is a well-established means to provide mobility, however this usually requires a degree of hand motor control (for example, through use of a joystick). Therefore, it is desirable to create a method of controlling an electric powered wheelchair which does not require motor control. A BCI could accomplish this by receiving wheelchair control commands from the user's brain.

1.2 Brain Computer Interfaces

Brain activity involves nerve cells, or neurons, conducting and transmitting electrical impulses. Surface electroencephalogram (EEG) is a relatively portable and low-cost method of measuring brain activity which uses multiple electrodes to measure the difference in voltages induced over various regions of the scalp. EEG data has been used to diagnose a variety of conditions, for example epilepsy, and is the primary method of signal acquisition for non-invasive BCIs. Through signal processing and classification techniques it is possible to identify mental states or thought patterns of the subject. Practical implementations can use techniques such as training the computer to recognise a certain thought by averaging multiple recordings of the pattern being induced, which can provide a sample that can be compared to live signals.

A primary example of this technique involved teaching Rhesus monkeys to control a cursor through moving a pole with their hand and recording the EEG signals produced from invasive electrodes [3]. The signals could be classified into each type of movement, and over time the pole input could be disconnected, allowing the monkeys to control the cursor using cortical activity only. This is an example of an asynchronous protocol, as the signals can be received and processed at any time. An alternative method involves using a synchronous protocol, which records the subject's neural reactions to an external stimulus. An example of this uses the P300 signal to measure reactions to flashing target words [4]. When the desired word is emphasised (Figure 1.1), a P300 wave is produced, and through several iterations it is possible for the BCI to interpret the desired command. This process is used to identify directional commands from a subject, allowing them to control a device such as an electric wheelchair with no mobility requirements.

The BCI developed at the Motor Control Laboratory in the University of Strathclyde uses an asynchronous protocol involving wrist movements in the desired direction of travel [5]. The user participates in a training session during which they are required to move their wrist in response to certain visual directional stimuli. This then enables them to execute directional commands using comparisons to the extracted patterns from

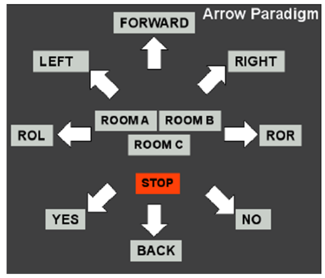

Figure 1.1: User display designed to induce P300 signal when desired command is emphasised (C. Lopes)

EEG signals obtained during the training session. The user can then use wrist movements to control a virtual wheelchair in a simulated environment. Studies conducted on the Strathclyde BCI system have shown that user performance feedback has a positive effect on learning rates, indicating the benefits of the virtual environment simulator [6].

1.3 Robotic Wheelchairs

Powered wheelchairs provide mobility to people who may not have the strength or mobility required to use a conventional wheelchair. The most traditional control mechanism is the use of a joystick to control forward, reverse, left and right movements. However, this method still requires adequate hand motor control, which can restrict access to many people with muscular disorders or tetraplegia. Single switch control, which involves using a single button to select highlighted commands, can be implemented for users who have restricted but functional motor control in their hand, however this type of system is generally slow to use and can be frustrating.

When there is little-to-no hand motor control a similar button system or complex positional apparatus can be activated by the person's head [7], however this requires constant head movement and position awareness and could cause muscle fatigue. Modern alternative control techniques involve sip-n-puff mechanisms which translate breathing patterns into directional commands, tongue controlled devices (Figure 1.2), speech recognition and eye tracking systems.

While these have been shown to be capable of controlling a wheelchair, all the above systems interfere with normal human behaviour to some extent, such as speech and sight. A BCI driven chair should be able to be controlled in the absence of external visual/spoken cues or stimuli.

Figure 1.2: TongueTouch Keypad - an example of hands-free motion control

1.4 Environment Input Methods

Shared control mechanisms rely not only on directional information from the user, but also require information regarding the surrounding environment. In order to receive this information the wheelchair must be equipped with sensor apparatus or be pre-programmed with map data.

A primitive form of navigation involves use of set tracks placed on the floor [8], usually made of ferromagnetic material, which the wheelchair can sense and follow. This technique is commonly used for autonomous robot navigation in environments such as factories, but has limited use for wheelchair control due to the limited area and paths available and the need for environmental modification. The most desirable and practical systems should be able to detect objects in the local environment, and need no previous modifications or information. There are many different sensor systems which can be used to achieve this, by using a mix of the technologies described in the following paragraphs.

Sonar sensors are used in a wide variety of fields, such as science, military, and navigation, as they are cheap (see Table 1.1 for pricing information), easily attainable and have low power requirements. Active sonar sensors record the time taken for an emitted sound pulse to be reflected and received, and can therefore calculate the distance to a nearby object for a known speed of sound in a medium. As they only use sonic energy they are not affected by light interference, and can therefore operate in bright or dark conditions; however, cross-talk interference between multiple sonar arrays can be an issue. Additionally sonar sensors perform best when the emitted pulse strikes the obstacle at a 90°angle, which reflects the greatest amount of energy back towards the receiver. Because of this, for a fail-safe system it is necessary that there be multiple sonar sensors at various positions, adding to the cross-talk issue. It is possible for sonar techniques to be used to detect the relative speeds of surrounding objects by using the Doppler effect to calculate the speed from a known frequency change. However, applications involving robotic wheelchairs have been unrecorded, and with a high sensor sampling rate objects should be detected at an appropriate distance regardless.

Infrared (IR) sensors operate in a similar manner to conventional sonar, using a low frequency light wave as opposed to sound. This means that they are susceptible to problems such as dark or transparent objects, however they are commonly used in conjunction with sonar to provide comprehensive sensing data, such as with the Strathclyde system. Passive infrared sensing can be used to detect heat sources, which may be useful in determining dangerously hot or cold environments. However, in a shared control system the user should be able to provide an overall observational control, therefore such applications are not widely considered.

Laser rangefinders (LRFs) use a sweeping laser beam to calculate distances to nearby objects and, unlike sonar or IR, can detect steep drops around the wheelchair (known as drop-off detection). LRFs are also not commonly affected by interference, however they are both more expensive than the other sensors described here and power-hungry, which could affect the practicality for their use in a portable low-cost system.

Vision systems use visible-light cameras to receive pictures of the surrounding area, and can use vision processing techniques to determine the surrounding topography. For example, straight edge detection can be used to determine the edges of paths or corridors, and object detection can be accomplished by examining regions of certain colours or textures. However, calculating object distance can be difficult [9]. With increased availability in commercial cameras and computer processing power vision systems are becoming relatively affordable, however the system complexity and processing power needed remains significantly higher than a system using traditional distance sensors. Additionally, it would

Table 1.1: Example sensor pricing information.

Sensor type	Specifications	Example cost
Sonar	Range 0.01 - 4.00 m, Resolution 0.025 m	£11.50 [10]
IR	Range 0.2 - 1.5 m	£15.79 [11]
Laser Rangefinder	Range 5.6m, Scan field 240°	£1044.00 [12]
Vision sensor (camera)	Resolution 1300x1040	£10.34 [13]

be more complicated to produce a 100% fail-safe system using vision techniques, due to lighting considerations and low colour variations in certain environments.

Odometry techniques include using rotational sensors on wheels to calculate the distance travelled, and are useful for keeping track of the wheelchair's current position. They provide no information about the environment, but can be constructed from basic materials, such as mechanical counters, which facilitates a low sensor cost.

Physical bump sensors can be attached to a wheelchair to detect if a collision has been made, however they usually require actual contact with an object, and are therefore not suitable for collision avoidance. However they can be used to prevent a user from continuously attempting to travel in the direction of an unseen obstacle, for example at foot-level.

Chapter 2

Previous Studies

2.1 Overview

Shared control, also known as shared autonomy, can be applied to many different fields. Many safety systems are a form of basic shared control, where the user can operate a system within certain boundary conditions. If an unsafe operating environment is detected the system can automatically be shut down, or safety protocols can beimplemented. A sophisticated example of this is the Mako Rio® surgical device [14], which uses CAT scan data to determine dangerous drilling regions during total hip arthroplasty, and by providing tactile impedance it can warn a surgeon if the drill starts to move close to these regions. During the process the surgeon still retains complete control of the position of the drill, with the automation simply providing additional safety features.

More complicated situations could involve human and machine contributing equally to the operation; or indeed allowing an artificial intelligence primary control with human supervision, such as in factory control lines. This approach would typically have a greater processing demand, as the system would be the primary decision-maker. Additionally, it would be of vital importance to ensure that all possible situations are accounted for during the programming of such a system, and that appropriate safety measures are accounted for.

The field of shared control for robotic wheelchairs provides several unique challenges. One consideration to take into account is the user comfort and confidence in the system. As a robotic wheelchair could be the only means of transportation for a subject it is important that they do not feel out of control at any point. For this reason the patient should still retain overall control of the machine, with additional safety features being implemented by the shared control system. Exceptions to this may include path-finding implementation, which would enable the user to select a room and have the wheelchair manoeuvre there automatically. However, even for high-level applications a direct "emergency stop" command should be able to be initiated by the user at any time.

While realising the limitations and safety requirements of such a system there must also be importance placed on sophistication and functionality. The ideal system should be able to provide a completely safe operation of the wheelchair while simultaneously not restricting mobility. Safety protocols involve object detection/avoidance and drop-off detection, and can be expanded to include complex real-life situations such as manoeuvring through a door, "docking" the wheelchair in a specific place, or the detection of an unsafe ground environment for the wheelchair. In addition to basic safety features it is also practical to include features which make control of the wheelchair easier and less time-consuming, such as wall-following functionality or path-finding through a cluttered room. The concentration required for EEG control can involve considerable mental effort and is affected by distractions and state of mind [15], therefore it is desirable to simplify the process as much as possible for the user.

An additional consideration involves the success rate of feature extraction and classification of EEG. Different classification methods and different BCIs have varied success rates [16], with the Strathclyde wrist-movement implementation achieving correct classification rates of 68-77%. Although this could be improved upon by additional system calibration and complexity a shared control interface would ensure that any possible misclassification did not result in a possibly harmful movement.

2.2 Review of Shared Control Interfaces

There have been several studies undertaken regarding shared control interfaces explicitly regarding BCI-controlled wheelchairs. One such example, reported in 2011 [4], involves the previously discussed synchronous paradigm. The system uses sensor and odometry data to calculate the position of the wheelchair on a known map of the environment. This is accomplished by using statistical analysis to compare the local area features obtained through the sensors to the known possible map locations. This enables path-finding techniques to be used such as those required for high-level room-to-room navigation. Additionally, the system includes a local planner which is able to detect and avoid an object which is new to the environment (i.e. is not recorded on the known map).

Different levels of control are allocated to different classes of user ability, which introduce limitations for basic and average users in certain situations, namely choosing a path around an obstacle and manoeuvring out of a deadlock scenario. For example, basic users cannot reverse from a deadlock but are required to use left and right pure rotations and forward movements only. This may aid in avoiding rear collisions for patients with hearing or visual problems. Currently the system is unable to account for moving objects such as people, however there are plans to include such functionality. The limitations of this system include the partial dependence on a predefined global map, and it is not stated if there are map size limitations before the system cannot determine the chair position using Markov Localisation.

The shared control interface developed by J. Philips *et al* [17] focuses on determining appropriate situations where the user requires assistance to ensure that the interface provides minimum unnecessary restriction. The BCI system receives signals asynchronously from the user and calculates the statistical probability of the intent matching a right, left or forward command. This data is used in combination with obstacle detection obtained via laser sensors, allowing the system to calculate what assistance, if any, is required. Currently implemented techniques involve collision avoidance braking mechanisms, object avoidance steering assistance, and orientation recovery, which calculates the next most suitable direction of travel if the first is blocked. The latter protocol aims to allow the user to reach their intended goal regardless of steering mistakes or feature misclassifications. The resultant paths taken to reach a goal are shown in Figure 2.1, and show that when the system is activated there are less sharp bends and course anomalies. It was also found that the course was more likely to include loops with the system inactivated. The total distance travelled when orientation recovery was activated is found to be significantly less than when deactivated. In addition, the time required per trial is also reduced, but to a

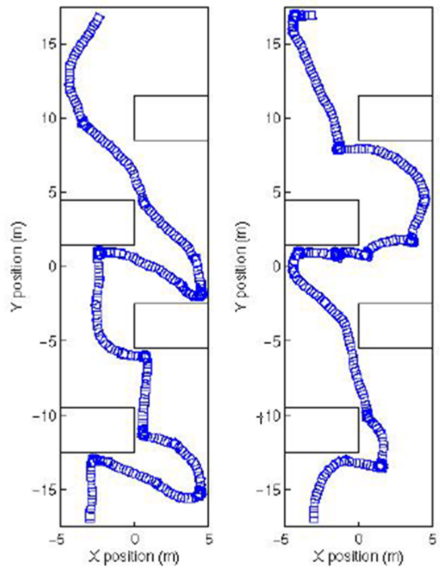

Figure 2.1: Assisted (left) vs unassisted (right) navigation paths, showing the influence of the Philips shared control interface.

lesser extent than the distance. This is due to the methodology the system uses to scan the environment; by physically rotating the wheelchair allowing the laser to record the local environment. This introduces a large time delay to the system, and may also be frustrating to the user.

Based on trials by 2 different subjects, the system enabled users to reach a goal despite a relatively low correct classification rate of 53.69%. This demonstrates that a shared control system can assist in nullifying the anomalies caused by misclassification, to a certain extent. The future scope of this project would involve implementing a user ability-rating system to determine the amount of assistance required by different users.

Before BCI developments, shared control systems had been developed for several "traditional" joystick-controlled wheelchairs, in addition to some of the novel techniques outlined above. These systems are highly relevant, as basic mobility requirements and safety features of a motorised wheelchair remain consistent independent of the command input method used. A literature review by Dr. R Simpson [18] illustrates many different smartchair implementations. There is a large diversity in the method which the chair receives information from the outside world, including track-following techniques [8], various sensor configurations [19] and vision systems [20].

The following systems have been selected to demonstrate the wide range of shared control systems and physical systems, in addition to providing examples of the capabilities of existing implementations.

The robotic wheelchair system "Wheelesley" [21] obtains user information via eye tracking functionality or single switch scanning. It utilises a shared control system designed to differentiate assistance between inside and outdoors, which it can detect using a sophisticated range of sensors and algorithms including temperature and light detection. The system results in an 71%/88% reduction in user effort and 25%/27% reduction in time required to complete a course, in inside and outdoor trials respectively. Consideration has been placed on the complexity of the user interface and setup of the system, such as the automatic indoor/outside detection. The wheelchair has 5 commands; forward, reverse, left, right, and stop, which are designed to be used for high-level directional control only with the scared control system using "common sense constraints" to determine appropriate directional commands.

Indoors, the system implements sonar and infrared sensors to achieve object collision avoidance and detection in addition to hall-centreing mechanisms, which were reported to aid greatly in wheelchair drift over long distances. Outdoors, a vision system is used to receive data regarding the surrounding data, due to the infrared interference and low sonar sensor coverage. The vision system uses image processing techniques and filters to determine the boundaries of obstacles and pathways, allowing the implementation of path-following algorithms similar to corridor-following above. Drawbacks to the system include the difficulty in differentiating path edges with straight shadows; it is suggested that the use of a colour camera would aid in solving this issue.

When reviewed by physiotherapists, the system received praise and interest, however not without issue. The necessity of the laptop used for eye-tracking was questioned, as the placement currently sits over the user's lap area, potentially taking up too much space and/or causing mobility problems getting in and out of the chair. In addition, it was mentioned that the eye-tracking electrodes may be proved unpopular due to aesthetic reasons. This may also be an issue in a BCI-based system due to the electrodes needed to obtain neurological signals, however this is a difficult problem to eliminate due to the necessity of the signal acquisition apparatus. The physiotherapists also wished that the system could be adapted to run on any motorised wheelchair to allow them to test the platform on equipment that they and their patients are already familiar with.

Finally, the added complexity of the vision system and indoor/outside detectors may add both additional cost and complexity to the system, however if the wheelchair came in a complete pre-setup package this would solve most complexity issues for the user.

The NavChair system [19] possesses similar functionality to the Strathclyde system (see below); namely obstacle avoidance, door passage and wall following. The environmental input is attained through 12 sonar sensors mounted on the front of the chair in addition to wheel motion sensors, with a joystick being used to provide user input. Information from the sensors is used to produce a certainty grid which determines the probability that an object is present at a certain position. This method of object detection is designed to circumvent any erroneous data received from the sensors. Direction of travel is then calculated to be as close to the user's command as possible while avoiding probable object locations.

A recent study [22] performs an in-depth review of the different methods of shared control functionality and semi-autonomous robotics. Using a mixture of these methods an SCI has been developed which is designed to be used regardless of input manner. The architect is split into 3 layers; the safeguard layer, reactive layer, and deliberative layer, where each layer represents a different level of system autonomy. The deliberative layer, for example, provides high-level point-to-point navigation and is designed to be active only for users with cognitive disabilities. The safeguard layer can be considered to be a low-level safety system and is active at all times, however it only initiates wheelchair command control to prevent a collision.

The main focus is placed on the reactive layer, which provides the majority of the shared control influence. Here a Potential Fields Approach (PFA) [23] is utilised, which models obstacles and goals and repulsors and attractors respectively. This enables the system to determine the desired direction through calculating a motion vector through the "field" produced, allowing the wheelchair to travel along a path free of obstacles.

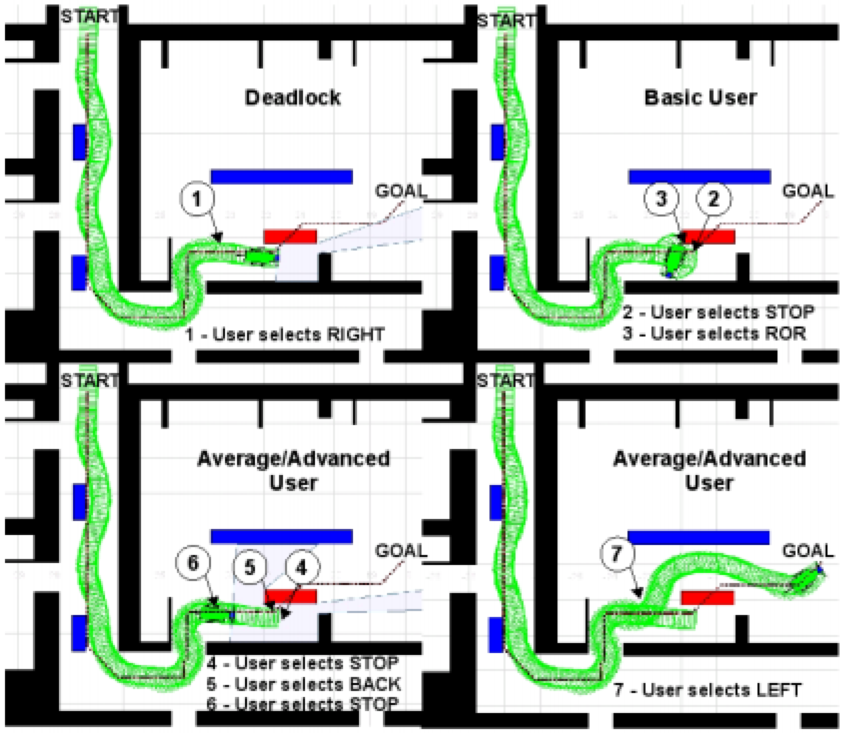

Figure 2.2: The experimental results presented by C. Lopes *et al*, showing the wheelchair route data for a deadlock scenario.

2.3 Assessment and testing

The Lopes system [4] used purely observational and recorded route data to perform analysis over the system performance. While this provides an overall idea of the system capabilities it does not enable a measure of performance and system success rate. The route diagrams attained from experimental results are mainly used as a demonstration of the system's decision making performance and difference between allotted user levels (see Figure 2.2. However, a team of physiotherapists were invited to review the system in person, with their feedback being used to help implement further improvements.

The Philips wheelchair [17] used two subjects to each compare shared control vs manual control. The trial time and distance was recorded and compared and the route taken was plotted to enable visual comparison (See Figure 2.1). These measures were sufficient enough to demonstrate an advantage with shared control activated, which resulted in a reduced time and distance travelled. Additionally, through inspection of the visual log it is possible to see an improved performance in terms of less looping and path deviation with the system activated.

The Wheelesely system [21] was tested under indoor and outdoor conditions and trials were repeated to compare individual user performance with and without assisted control. Users were required to navigate a pre-defined course under both conditions. As a single switch scanner was used as the input device measured parameters consisted of number of clicks required to navigate the course, the scan time required to choose each command, the time taken for the wheelchair to execute manoeuvres, and the total time taken to complete the course.

The results taken under assistive control were greatly improved over those under manual control only. Although no official user opinion survey was taken it is mentioned that users preferred to use the assistive control.

During testing of the NavChair system [19], four quantitative measures were collected; average speed, ride "wobble", obstacle clearance and risk of collision. All three shared control functions were tested, each being used to compare 2 different methods of direction calculation (vector field histogram (VHF) and minimal VHF), alongside an unmodified control system tested by an experienced system user. The testing environment was inclusive of multiple types of obstacle in addition to multiple surrounding material types which may affect sensor readings.

The testing measures suggested that the best performance was achieved by the experienced user using the basic control system, however this may be due to the limited recorded parameters. It is possible that other more subjective measures may have indicated favourable user effort and stress levels with the shared control system active. Additionally, a subject who has experience using the shared control system would be expected to perform better than someone unfamiliar with the control method and may demonstrate an advantage over the experienced user with the basic system.

The PFA [22] approach of object avoidance uses several methods to provide a comprehensive description of system performance. Performance measurement is split into two main fields; task metrics, which describe wheelchair/user performance during a trial, and psycho metrics, which measure user parameters such as fatigue and attention. Trails involved independent navigation as opposed to following a set track to ensure that the participant is focused on controlling the wheelchair efficiently as opposed to concentrating on a pre-defined path.

Performance measurands included closeness to objects, trial time, distance taken and sharpness of direction changes (noted as “smoothness”), in addition to user-based parameters which are calculated indirectly, such as disagreement, which represents the difference in output between the user and shared control system. Measures of success revolved around the successful cooperation between the subject and the system. Using these measures the shared control system was found to improve both human and robot performance.

Chapter 3

Existing Strathclyde System

3.1 Virtual Environment

The development of any new system requires constant modification and testing until an optimal result is achieved. Due to this fact, testing the system using a physical implementation would be time-consuming, expensive, and inefficient. Therefore, a virtual navigation simulator (see Figure 3.1) has been created at the University of Strathclyde to produce a practical testing environment [24]. This allows continuous modification of the testing environment, the sensor placement, and the wheelchair in addition to ensuring minimal safety considerations for test subjects.

The simulator system is based on an Urban Search and Rescue simulator (USARsim) [25], which contains many useful features for robotic simulation including native incorporation of many different sensor types. This allows the rapid development and alteration of a sensor array for a shared control system in addition to providing the basic physics engine on which to run the simulator.

The USARsim package has been developed using the UnrealEngine [26]. This is a platform mainly used for game development, however can be used to develop 3D navigational scenarios.

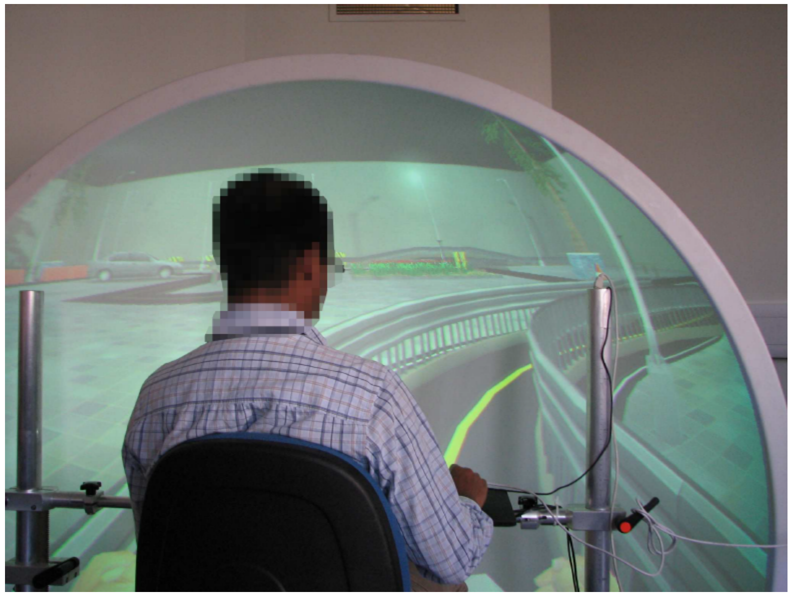

Figure 3.1: A user demonstrating the virtual wheelchair environment simulator at the University of Strathclyde.

3.2 Shared Control Implementation

The existing shared control system had been developed by a past student at the University of Strathclyde [27] and employs a sonar/infrared sensor array (see Figure 3.2) as an input method. Several shared control functions had been implemented in C++ and integration with USARsim has facilitated the addition of and communication with the virtual sensors. Collision detection and avoidance algorithms calculate the direction and proximity of nearby obstacles and either alter the wheelchair direction to avoid collisions or stop movement completely. Wall following functionality operates by attempting to guide the wheelchair on a straight line parallel to a wall. This is essential for travel down a corridor where the user may have to make continuous and small adjustments to keep the chair on a straight path, which is impractical with a BCI-driven system.

Although the existing system provided a framework on which to build, many aspects were required to be improved upon in order to provide a smooth, reliable user experience. Areas requiring improvement included collision avoidance as the system would occasionally fail to detect obstacles, usually resulting in wheelchair collision. Additionally, wall following algorithms would often make the wheelchair travel on a meandering path. Improving these

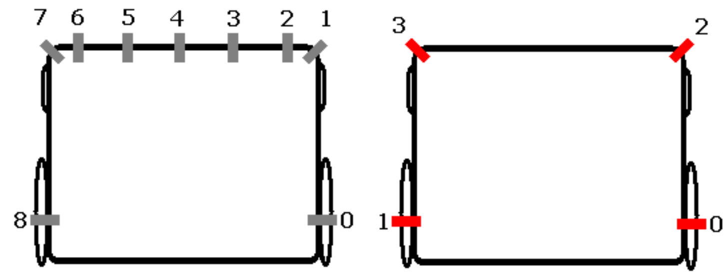

Figure 3.2: The sonar (left) and IR (right) sensor arrangement on the Strathclyde wheelchair.

features would also make it easier to implement other mechanisms such as door and narrow corridor passage.

Additionally, many aspects of the current system have not been quantified and documented, particularity in terms of the size/shape of obstacles which could be detected, precise stopping distances, turning circles, and success rates in terms of number of collisions prevented. Additionally, sensors have been placed heuristically and therefore information on their coverage is not available. Consequently, in addition to implementing further shared control functionality it was necessary to upgrade, test, and quantify existing features.

3.2.1 Basic Movement Controls

An important difference between a BCI controlled vs a traditional joystick is the control type. A joystick continually gives direction information based on the position of the stick at any one time, e.g. when the user desires to go forward the joystick should be kept in the "forward" position. A BCI system may work through discrete controls, e.g. a "forward" command is given, and the chair must move forward for a continuous amount of time after the command. Therefore, the shared control system in place utilised the part of the existing control system which responded to discrete keypad presses corresponding to forward, reverse, left, right, and stop. When stopped the wheelchair can perform left and right pure rotations and initiate movement at the slowest speeds in the forward and back directions through use of the corresponding controls. When in movement the left and right controls guide the user left and right while moving in a forward or backward direction, and additional forward/back commands change the speed of travel. This is a simple means of ensuring that the user is always control of their speed and direction.

3.2.2 Obstacle and Collision Avoidance

These functions use a range of sonar sensors places around the front of the chair, and are designed to minimise collisions with objects. If the wheelchair detects obstacles nearby it modifies its course in order to avoid a collision and continues moving in the previous user-defined direction once the collision has been averted. If the sensors detect that there is no way to avoid the obstacle the wheelchair gently stops.

Unfortunately, these routines occasionally failed to react and collisions to the front and corners would occur. This was thought to be mainly due to the sensor scan interval; occasionally the sensors would not alert the system to an obstacle's presence until the wheelchair had already collided. As would be expected this effect was greatly amplified when the wheelchair was moving rapidly as there is a larger effective distance travelled between readings.

3.2.3 Wall Following

The existing wall following functionality allowed the wheelchair to successfully detect and travel along walls, however it would greatly meander along the route. This is due to the implementation method; once the chair exceeded a maximum distance threshold it would steer towards the wall. The chair would eventually pass the minimum distance threshold and reverse steering direction again. This method, although successful in maintaining wall following, would presumably be uncomfortable or frustrating for the user or any other persons in the vicinity. Therefore great importance was placed on significantly improving this feature.

Chapter 4

Version Upgrade Modifications

4.1 Aims and Objectives

The main objectives are as follows:

- To create new maps which provide suitable testing scenarios for both old and new functionality.
- To review the current shared control system, amending it where needed in order to increase efficiency and reliability.
- To perform a review of the current sensor arrangement and to modify or add/remove sensors as need be.
- To introduce further shared control functionality as required.

4.2 Maps

In order to test the shared control system it is important to select the most appropriate test environment with suitable obstacles and challenges that users would expect to encounter in the real world. Several maps had been previously developed for the system, however these had not been specifically designed for the shared control system and there was therefore a need to review the existing maps and make adjustments and new developments as required.

The maps described below were created in UnreadEd 2.0 which is included with the version of the game required to run the system. The objects within the office map were textured and edited using 3D Studio Max® and imported into the level as static meshes.

4.2.1 Existing Maps

Before the commencement of this project two maps were available for testing. The "Hall" map is a 10 x 10x 3m empty room originally intended for user orientation with the system. However this map is also ideal for testing basic navigational controls such as turning circle, acceleration, etc. Additionally, due to the straight exposed walls it is suitable to test one-sided wall following using this map and has therefore been used as the main testing ground for this purpose. However, this does not provide testing functionality for environments such as corridors with walls on both sides of the chair, or curved-wall scenarios which may be present in modern buildings.

Another more complex map named "Track" 4.1 had also been developed. This map provides a more diverse testing environment by allowing the user to follow a marked track on the floor, encountering numerous obstacles along the way. The obstacles encountered include narrow passageways, turns, ramps and obstructions. As such this map is ideal for testing many features, and the pre-existing obstacle avoidance functionality was tested using this course. Advantages of the Track map include a random layout of various obstacles, which ensures a wide variation in test variables as would be encountered in the real world. Additionally the continuous track provides a pre-defined testing environment and track times are quantifiable.

However, it is difficult to test individual scenarios with this method as there is no measurable difficulty difference between obstacles. Additionally, the track does not provide a realistic testing scenario and does not test certain necessary aspects of wheelchair functionality such as manoeuvring in small spaces.

The limitations and desired features of the current maps were taken into account in order to develop new maps which overcome any existing restrictions in addition to adding new testing functionality. As such it was decided that there were two distinct areas for development. The first involves development of quantifiable scenarios ranging in difficulty

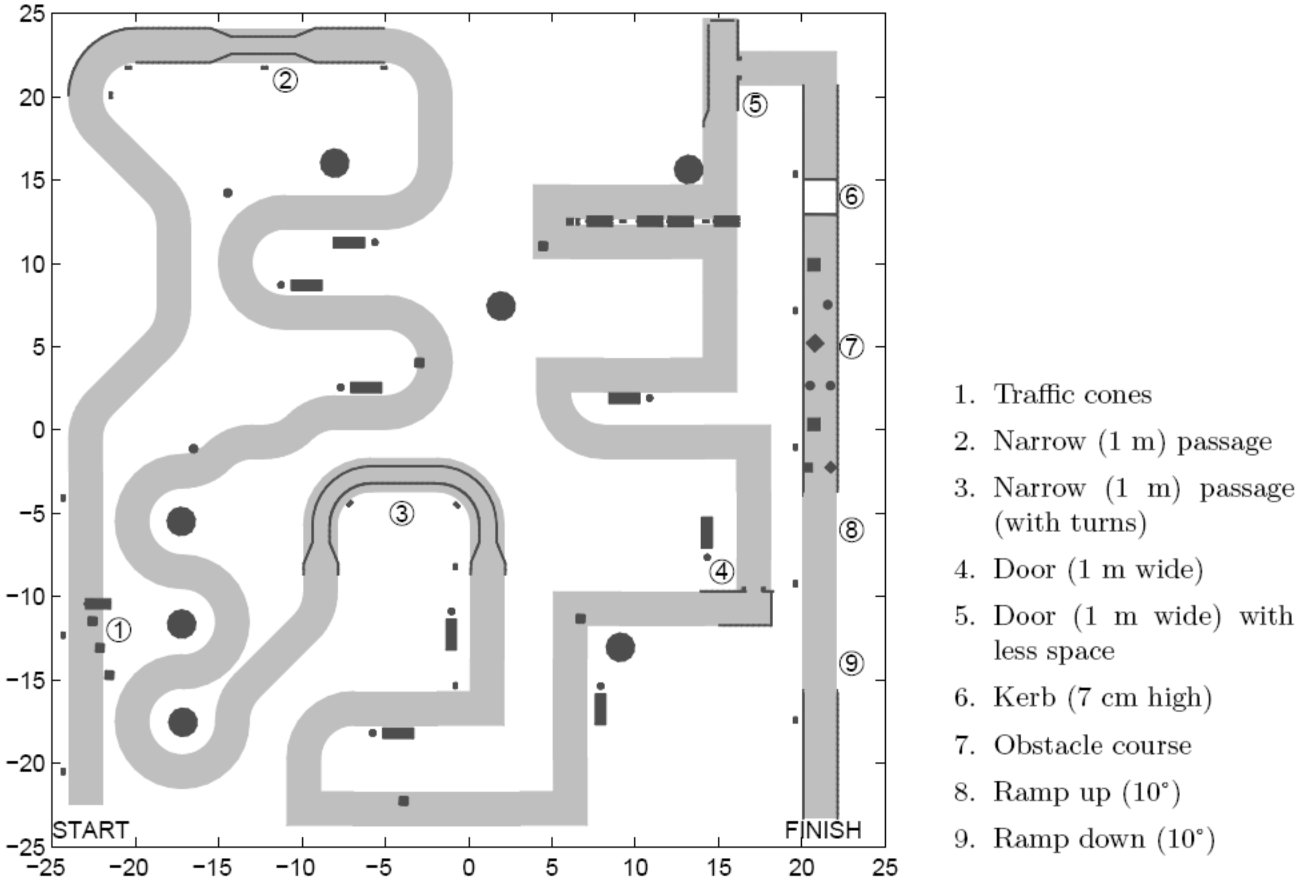

Figure 4.1: Layout of the existing Track map.

levels (Table 4.1) which can be used in the development and testing of shared control features. The second would provide a more natural immersive environment for testing by implementing room layouts and obstacles that a wheelchair user may commonly experience in the real world.

4.2.2 Accessibility Legislation

In order to provide realistic environmental scenario, maps were built according to the UK Building Regulations [28]. These outline the many requirements which modern buildings must fulfil in order to be licensed for public use. In particular corridors, doors and sanitary facilities must meet certain specifications which are designed with wheelchair access in mind. These specifications should meet the minimum requirements needed to allow wheelchair passage throughout the building in addition to providing advice for the easy implementation of basic comfort and safety features. Particular care must be taken when designing a wheelchair accessible toilet facilities as there are many different factors to take into consideration. An office scenario was developed in accordance with these specifications in order to provide a realistic testing experience.

4.2.3 Development Map

This map was created to test discrete obstacles that may provide a navigational challenge and which can be measured and compared. It is designed to be edited as features from the shared controlled system need to be tested, i.e. it is intended to be dynamically updated. Each type of obstacle is described in detail in Table 4.1. See Figure 4.2 for an overview of the map layout. Figures 4.3 and 4.4 show the “doors” and “drops” obstacles respectively.

Table 4.1: Obstacle quantification data.

Obstacle	Description	Quantifiable levels
Doorways	Straight doorways with varying widths designed to test door passage	Width: 2.0 m 1.6 m 1.0 m 0.8 m
Drops	Sheer drops designed for drop-off detection	Drop size: 0.05 m 0.07 m 0.10 m
Kerbs	Sharp inclines designed for kerb detection incorporated into depressions with drops	Kerb size: 0.05 m 0.07 m 0.10 m
Corridors	Passage between two walls designed to test double-sided wall-following	Width: 2.0 m 1.8 m 1.4 m 1.2 m 1.0 m

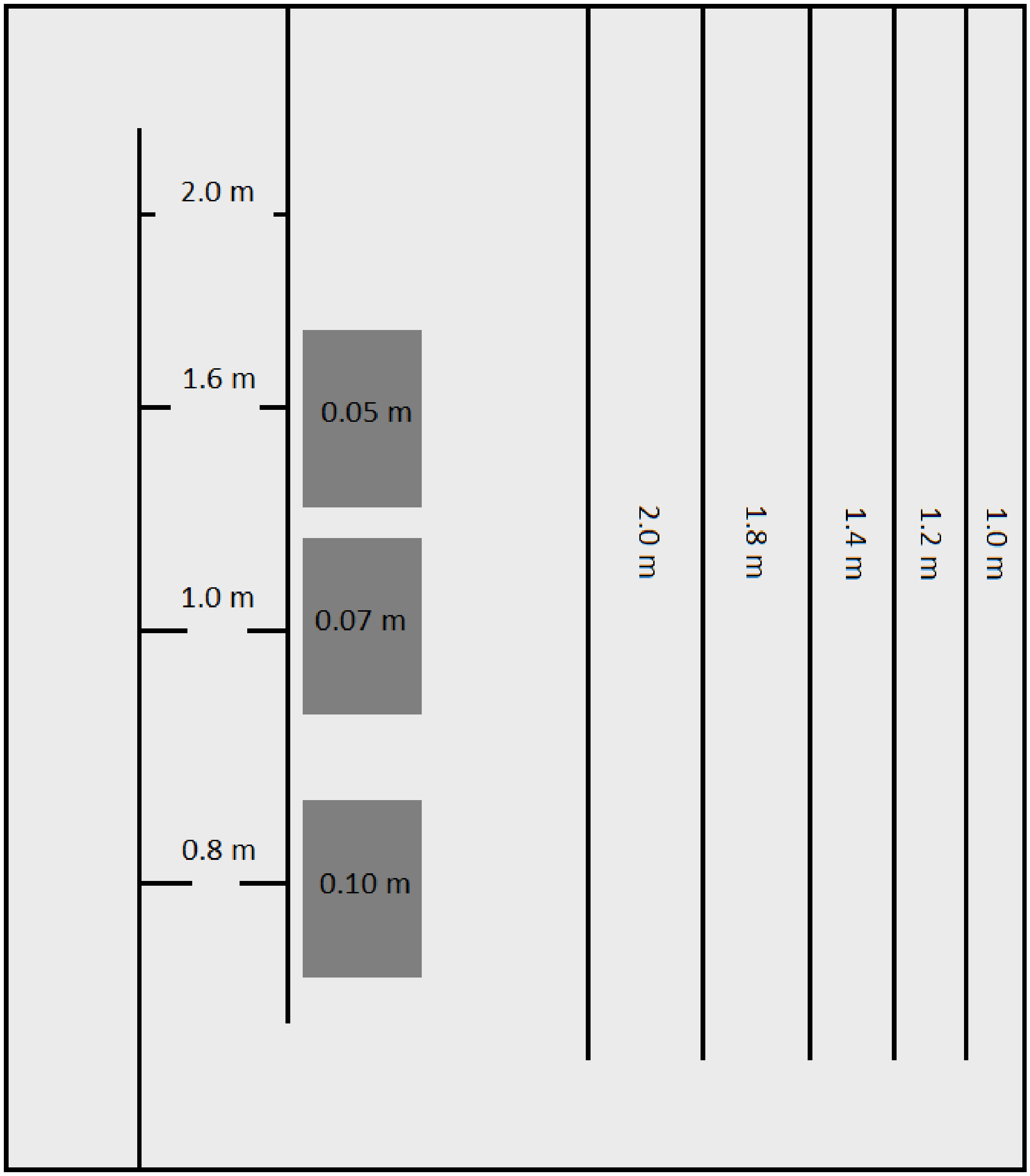

Figure 4.2: Layout of the development map showing door, drop and corridor obstacles.

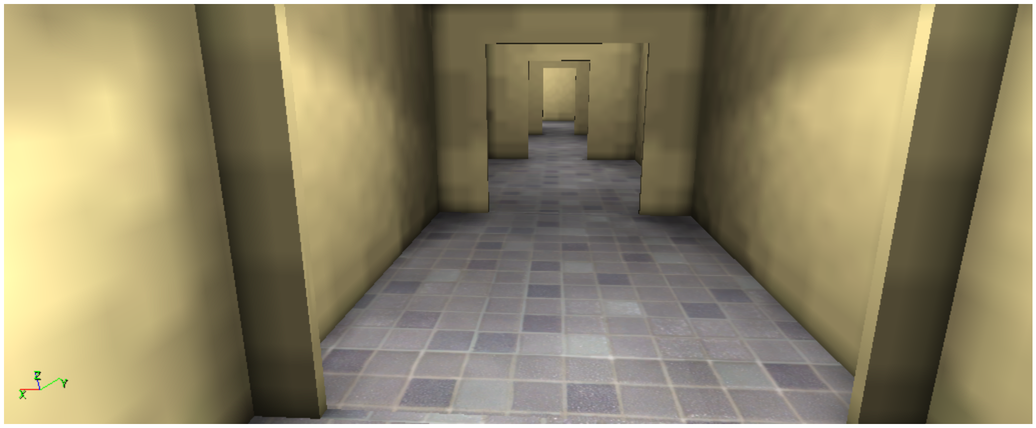

Figure 4.3: Screenshot showing the doors obstacles in the development map.

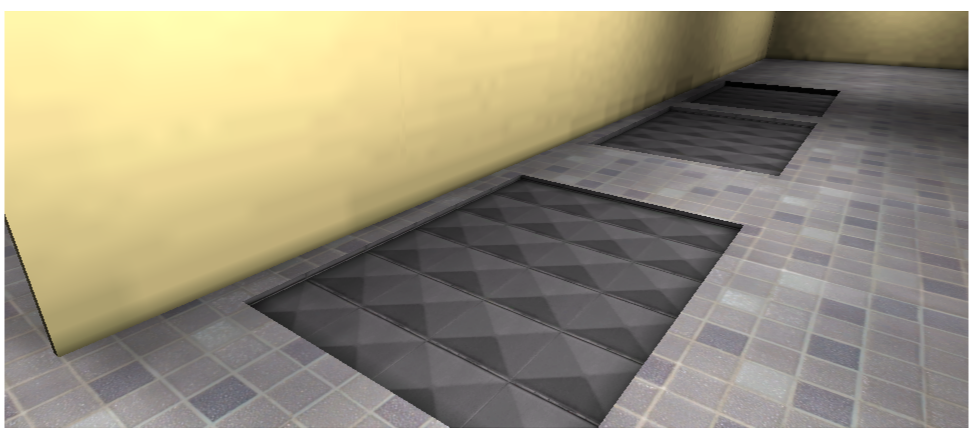

Figure 4.4: Screenshot showing the drops obstacles in the development map. Kerb detection was tested using the rising edge of the depressions.

4.2.4 Immersive Office Environment

This map is a fairly small, typical office environment built to minimum building standard regulations. The map consists of a hallway, bathroom, elevator, and main office with a kitchen area, workstations and side rooms. This was designed to provide a more natural testing location as the challenges it presents are based on those expected to be encountered in real life.

Figure 4.5: Screenshot showing the main office layout.

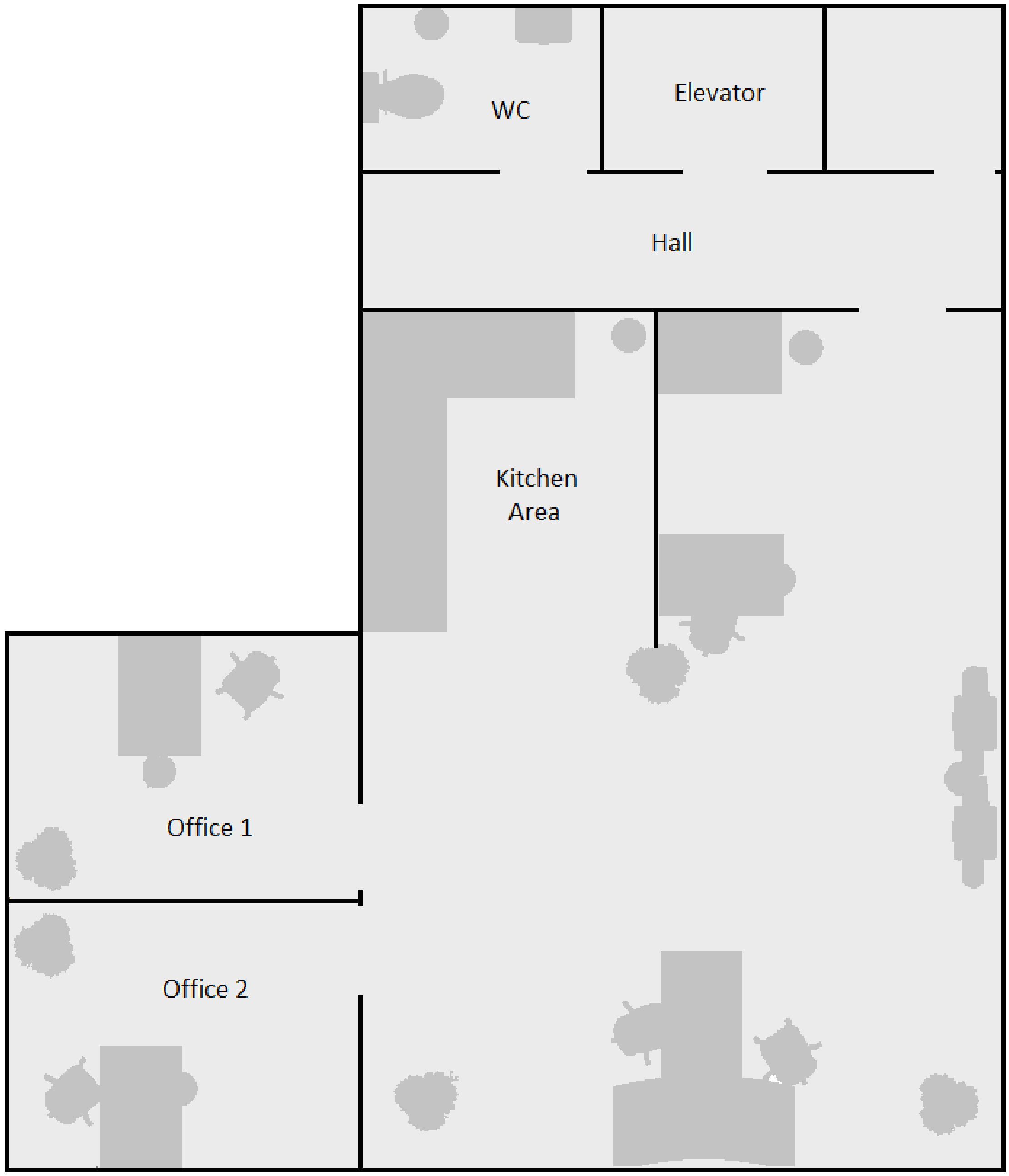

Figure 4.6: Layout of the office map showing different rooms, areas, and the obstacle layout.

4.3 Adjustments to Current Implementation

4.3.1 Sensor Update Frequency

A significant proportion of the issues associated with the current system where discovered to be due to the sensor update rate. It was calculated that the sensors were updated approximately once per second, which can be considered a low rate in regards to a moving object. Therefore it was greatly desirable to increase the sensor reading frequency. Upon examination of the code and USAR files it was found that this can be achieved through changing the sensor refresh rate and the rate at which the sensors send information to the wheelchair, both of which are configured in the USARBOT.ini file. This allowed the wheelchair to receive sensor data every 0.2 seconds which dramatically improved some functionality such as collision avoidance (see below). Additionally, this removed a random factor of the system's performance, e.g. the distance at which the sensors detected an object remained more or less constant as opposed to a function of speed and polling time.

4.3.2 Pure Rotations

A major amendment to the basic controls involved the ability to perform pure rotations. In the previous system, whenever the wheelchair detected an object nearby object avoidance routines prevented the wheelchair from moving without repeatedly sending commands, which would cause the wheelchair to perform small discrete movements. However a pure rotation involves the wheelchair rotating on the spot and would very rarely involve collision with obstacles. Additionally pure rotation is used to manoeuvre around many obstacles and so is performed often. Therefore repeatedly performing a rotation command is not suitable for a system designed to reduce user fatigue and aid navigation. As such, a special case was created for when the wheelchair was stopped or already performing a pure rotation which bypassed the obstacle avoidance routines and allowed the initiation of a “left” or “right” command to be executed unobstructed. This was expected to markedly reduce the number of commands required to navigate effectively with the system. As the wheelchair is longer than it is wide this does pose a risk of a corner collision while rotating. However, if the wheelchair has came to a stop due to collision avoidance routines there should be sufficient space to allow a rotation.

4.3.3 Obstacle Collision and Avoidance

The existing obstacle and collision avoidance routines have mainly been unaltered from the original implementation. However the execution of several functions has been amended to enhance code readability and to dispose of unnecessary variables and conditional statements. Additionally several function implementations, such as the corner obstacle avoidance routine, have been greatly optimised and annotated with additional comments.

As the sensor update frequency has been raised it was possible to lower the object detection distance from 1.2m to 0.6m. This is due to the necessary inclusion of a high margin of error with a low update frequency as the wheelchair may travel a far distance in between sensor pings. This has contributed to the constancy of the system as the margin of error for the stopping distance has been greatly reduced.

4.3.4 Wall Following

Initially the maximum and minimum thresholds were set to the same value in an attempt to minimise the wheelchair deviation, however due to the low sensor refresh rate this would still result in a meandering path. An additional disadvantage to this system was that the chair would automatically jump to the pre-configured wall distance, independent of the original desired distance.

Therefore, this implementation was re-written to use an angle-based algorithm (see Algorithm 1). The wheelchair now uses two sensor beams to calculate the angle of the chair with respect to the wall. If the chair is angled towards or away from the wall it is rotated until parallel, which should result in a straight path along the wall. This resulted in a path which still deviated from the centre however with a much less "sharp" turning angle and a more predictable pattern.

Upon raising the sensor refresh rate the deviation from a straight path was expected to be completely eliminated. On occasion this worked as expected, with the wheelchair initially making small adjustments until the correct orientation had been achieved. However, often the chair would turn in an erratic fashion at a lower, varying frequency than 5 Hz. By examining the system load this was found to be due to processor lag of the computer running the system. As the computer was needed to run the wheelchair simulator, operating system and background tasks this resulted in a higher load than if only the shared control system was active. This brought about the sensors being updated erratically as processing power became available.

```
switch wall side do
    case right
        if front sensors are blocked then
            stop wheelchair
            return
        end
        if angle donates wheelchair moving away from wall then
            turn right
        else if angle donates wheelchair moving towards wall then
            turn left
        else
            go forward
        end
    endsw
endsw
```

Algorithm 1: Pseudocode for wall following. Note only code for the right wall is shown; the left side is implemented in a similar case statement.

4.4 Additional Functionality

Although many potential dangers can be avoided through obstacle detection there are many other scenarios which may pose a danger to the wheelchair or others or result in the frustration of the wheelchair user. Therefore it was decided to conduct a review of desirable shared control functionality and implement appropriate features where possible.

4.4.1 Scenarios

In order to be able to choose the most desirable functions to implement it was decided to imagine several common scenarios and environments which wheelchair users may commonly encounter. The additional functionality could then be designed to combat the dangers or challenges introduced through these scenarios. Table 4.2 details the chosen scenarios

Table 4.2: Commonly encountered scenarios and challenges.

Scenario	Challenges
Home	Manoeuvring in tight places
	Obstacles at all heights
	Possible moving obstacles e.g. pets and children
	Door passage
Office	Manoeuvring in tight places
	Obstacles at all heights
	Door passage
	Corridors
	Uneven floor transitions e.g. into elevators
	Sharp drops and rises e.g stairs
Supermarket	Corridor passage
	Moving obstacles e.g. people
Street	Drop-off detection
	Kerb detection
	Kerb following
	Moving obstacles e.g. people, vehicles
	Uneven floors/surfaces
	Grip/Slip detection e.g. gravel, ice

and problem areas, and the following sections detail the functionality added to meet the specified requirements.

4.4.2 Drop-off and Kerb Detection

The current sonar sensor arrangement provides data about any obstacles at waist-level (see Table 4.3). However small obstacles at foot level such as kerbs were not accounted for. Additionally, no provision had been made for a possible sharp drop-off situation such as a falling kerb. During testing is was found that a relatively small drop (approximately

0.05 m) can cause the wheelchair to fall sideways, with the drop distance depending on the speed and angle of approach. Therefore it was of critical importance that some form of drop-off protection was included in this system.

An additional IR sensor has been added to the sensor array to allow both drop-off and kerb detection. This was accomplished by angling the sensor downwards 30° from parallel to the ground. At a sensor height of 0.2 m the expected ground sensor reading is 0.4 m (see Figure 4.7). A significant difference in this reading results in the wheelchair initiating stopping procedures. A tolerance of 0.03 m was included to account for the start and end of steep hills and to compensate for any small sensor inaccuracies.

An IR sensor was selected due to its narrow beam angle. Any deviation to this sensor reading can therefore be attributed to a change in height of a single "point" where the beam meets the ground. A sonar beam has a larger beam angle and returns the distance to the closest obstruction. This makes it more difficult to pinpoint the exact location of the obstruction and in this case the sensor would constantly detect the distance of the ground closest to the wheelchair, reducing the range for a given value. Additionally, sonar sensors respond best when the beam hits the obstacle perpendicularly and therefore is not suitable for this type of configuration.

Although IR sensors are affected by light interference and transparent surfaces this should not usually be an issue for a sensor directed at the ground. However there may need to be further provisions taken to allow the wheelchair to drive on transparent or highly luminous surfaces to prevent erroneous stopping measures.

Currently there is only a single sensor which detects drops directly in front of the chair. The function can be expanded to provide detection for multiple directions however this will require the inclusion of more sensors.

4.4.3 Door Passage

Doorways are one type of challenging obstacle which are commonly encountered by wheelchair users. In order to pass between two stationary obstacles the wheelchair must be perfectly aligned so as not to drift to either side and risk collision with the door frame. This presents a similar problem to the wall following scenario; the accuracy needed may require multiple

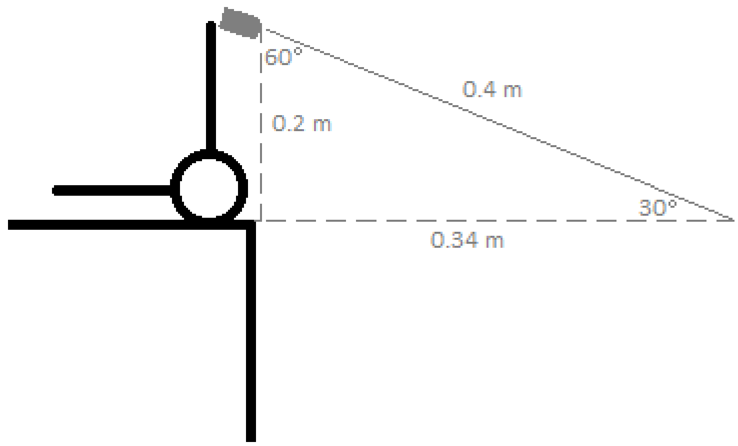

Figure 4.7: Kerb/drop-off detection sensor configuration.

small adjustments can be difficult and tedious to perform. Therefore automated door passage functionality has been added to the system. When the system identifies a door it runs a door passage function which automatically aligns the wheelchair's position and direction so as to guide the chair directly through the middle of the passage (see Algorithm 2). The algorithms classify the wheelchair into one of the four conditions shown in Figure 4.8. In order to reduce the amount of sensors required the pre-existing sonar sensors were used for this purpose.

A major issue with this current implementation is that the simulated wheelchair has large rear wheels which protrude significantly outwards from the main body of the chair. On occasion this can lead to the rear wheels catching on the door which in the worst case scenario can cause the wheelchair to tip. As this has potentially dangerous consequences it is important that this behaviour is eliminated, which may involve the addition of more sensors adjacent to the wheels. In addition to software safety checks it may be beneficial to install physical wheel guards for such scenarios.

The current implementation has been designed for situations where the wheelchair approaches the doorway straight-on however there are many situations where the doorway can only be approached from the side such as entryways in corridors. A basic function outline has been included to detect these doorways however is not currently implemented.

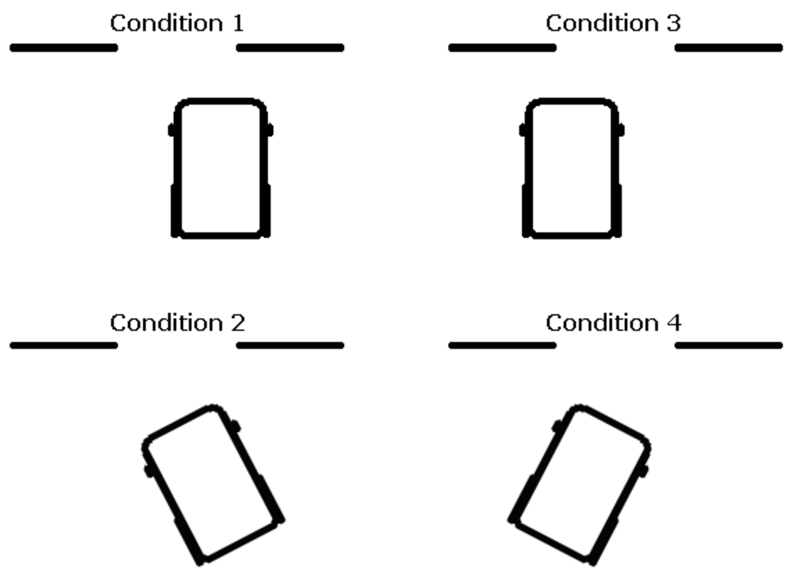

Figure 4.8: Diagram showing four different wheelchair positions with respect to the door. Each condition is evaluated by door passage algorithm 2.

```
if SonarSensor1<SonarSensor7 then
    if SonarSensor2<SonarSensor6 then
        turn left ;                      /* close to right side (Condition 1) */
    else
        turn right;                      /* tilted left (Condition 2 */
    end
else
    if SonarSensor6<SonarSensor2 then
        turn right ;                     /* close to left side (Condition 3) */
    else
        turn left;                       /* tilted right (Condition 4) */
    end
end
```

Algorithm 2: Pseudocode for door passage.

4.4.4 Further Functionality

The manoeuvrability of the wheelchair has been increased intrinsically due to the amendment of pure rotation commands. Additionally the reduction in collision avoidance stopping distances has made it possible to navigate in tighter spaces. Furthermore, the likelihood of detecting of moving obstacles has increased with the updated sensor frequency.

There is no current method for detecting slippery or unreliable floor surfaces however this may be possible by comparing the speed of the wheels compared to the speed of the wheelchair, which can be obtained from the existing GroundTruth sensor. Uneven floor surfaces may be detected by the kerb/drop detection functionality if drops and rises exceed 0.03 m, which would cause the wheelchair to stop. Specific ground surface detection has not been included primarily due to the difficulty in correcting such behaviour.

4.5 System Quantification

One major drawback of the pre-existing system is the lack of precise information on its abilities and limitations. It is essential to possess quantifiable information in order to fully understand the system and to be able to compare it with similar wheelchair realizations and upgrade modifications. The following tables describe functional aspects both before and after modifications where appropriate.

The sensor configuration data described in Table 4.3 was unchanged, excepting the update refresh rate which was changed from 1 Hz to 5 Hz. Additionally a further IR sensor has been added at a different height from the previous sensors (details in Figure 4.7).

The "sonar vertical range value" was calculated by taking into account the sonar sensor height, beam angle, and obstacle detection distance (0.6 m), as shown in Figure 4.9. In addition the "blind spot" distance of the sonar array was calculated as shown in Figures 4.10 and 4.11 below, for middle and corner sensors respectively.

Table 4.3: Sensor configuration data.

Sensor Configuration	
Measured	Value
Update rate	0.2 s
Sonar distance range	0.1 to 5 m
Sonar array height	0.43 m
Minimum vertical sonar range (at 0.6 m)	0.20 - 0.66 m
Sonar beam angle	45°
IR distance range	0.1 to 20 m
IR array height	0.05 m
IR beam angle	18°

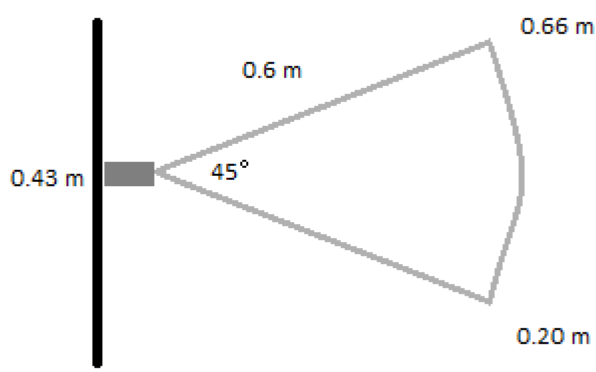

Figure 4.9: Diagram showing the vertical sonar sensor coverage.

Table 4.4: Quantized wall following capabilities.

Wall following		
Measured	Before	After
Wall detection distance	1m	1m
Wall following thresholds	0.35 to 0.65m	Constant at user's choice
Deviation from centre line	0.3m	no significant deviation
Obstacle detection	none	0.5 m

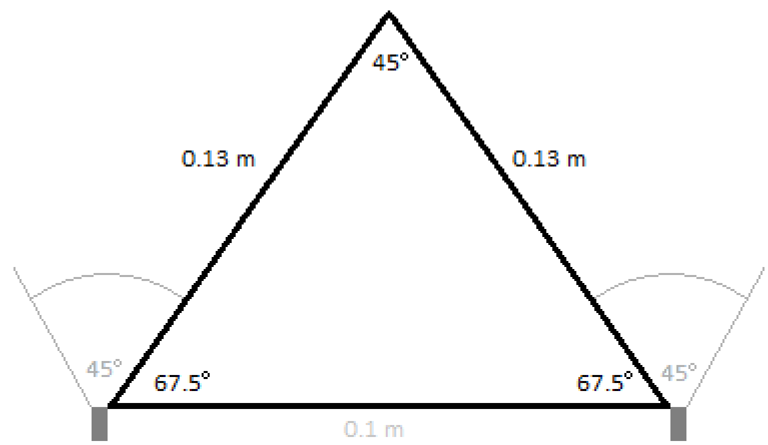

Figure 4.10: Diagram showing the middle sonar sensor blind spot.

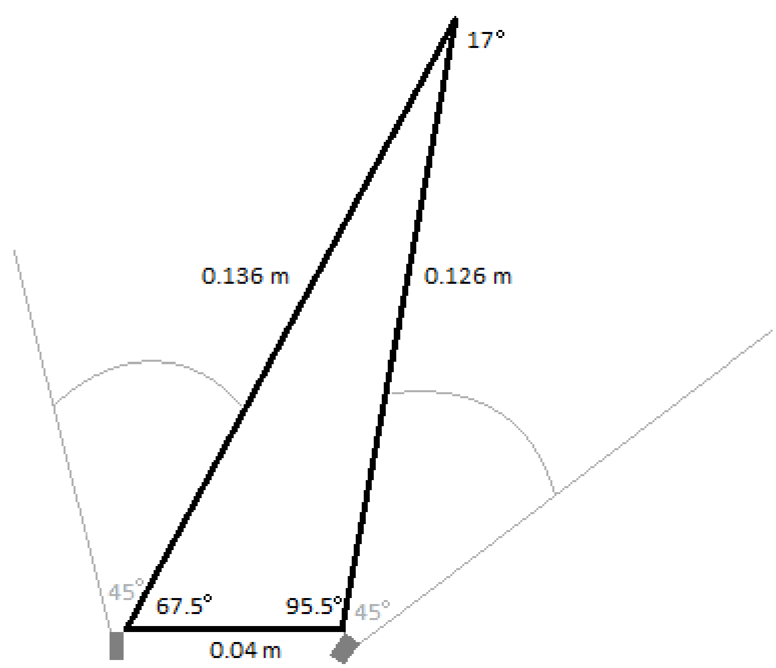

Figure 4.11: Diagram showing the corner sonar sensor blind spot.

Chapter 5

Testing and Evaluation

5.1 Previous Testing Methodology

As demonstrated by the wheelchair systems discussed previously, it is advantageous to test the system in both in a quantitative and qualitative fashion. During the creation of the simulator system wheelchair navigation logging was implemented detailing many quantitative variables [24], including the time required to complete a course, average and maximum speeds and number of collisions made. Additionally several Matlab programs were used to analyse the log files which would calculate the frequency of keystokes or joystick movements and various other parameters relating to the path taken by the subject.

A different approach was taken in order to test the initial shared control system [27]. Subjects were required to navigate through several obstacles in the course while performing mental calculations. Upon completion of the trial they were asked to fill out a survey relating to the NASA Taskload Index. The results were then used at a measure of how demanding the tasks were; both with and without shared control. This demonstrates a subjective measure of performance which is an appropriate measure for a system which aims to reduce user stress and frustration.

It was decided that it would be beneficial for any further testing to include both qualitative and quantitative elements to reflect the different nature of the benefits which can be obtained though the use of a shared control system.

5.2 Test Design

During a simulation session, log files are created which document several different simulator events:

SEN A sensor reading stating the type of sensor and the recording measured.

STA A message regarding the state of the wheelchair such as the position, direction, battery levels, etc.

INIT Indicates an initialisation command i.e. spawning a wheelchair.

TOUCH A type of sensor which indicates a collision with an obstacle along with the time and positional data.

DRIVE Indicates when a user command has been sent and displays the resulting left and right wheel speeds.

By examining these logs much information about an individual trial can be obtained. As previously mentioned, there have been several Matlab programs developed to automatically analyse these files and display useful information, such as the total number of keystrokes used, route taken and locations of collisions. However these programs have not been updated to account for the inclusion of sensor data and are therefore incompatible with the log files produced with the updated wheelchair model. Additionally the current logs do not indicate the number of specific keystrokes made, therefore a small tool has been developed using the AutoHotkey scripting language to allow this. This information is designed to be used in addition to data such as the number of collisions made and the trail time, which can be manually obtained from the log files.

In addition, a survey was constructed to receive descriptive information from each subject. This was developed to gain more subjective feedback which can be used to adapt the system to provide features which provide a smooth and comfortable user experience. The participant forms developed for this purpose are included in the Appendices.

5.3 Task

The office level map was used to provide a course for the wheelchair to navigate. The trial was designed in order to simulate typical movements which could be required in a day-to-day basis. This involved the subject navigating to fixed points in the map using their own navigational skills with emphasis made that participants should navigate the course at their own pace and comfort level. Figure 5.1 shows the map layout designed to be provided to participants, with the navigational landmarks circled.

Unfortunately, the trial implementation has been delayed due to a simulator malfunction. However there are current plans to correct the system fault and initiate testing.

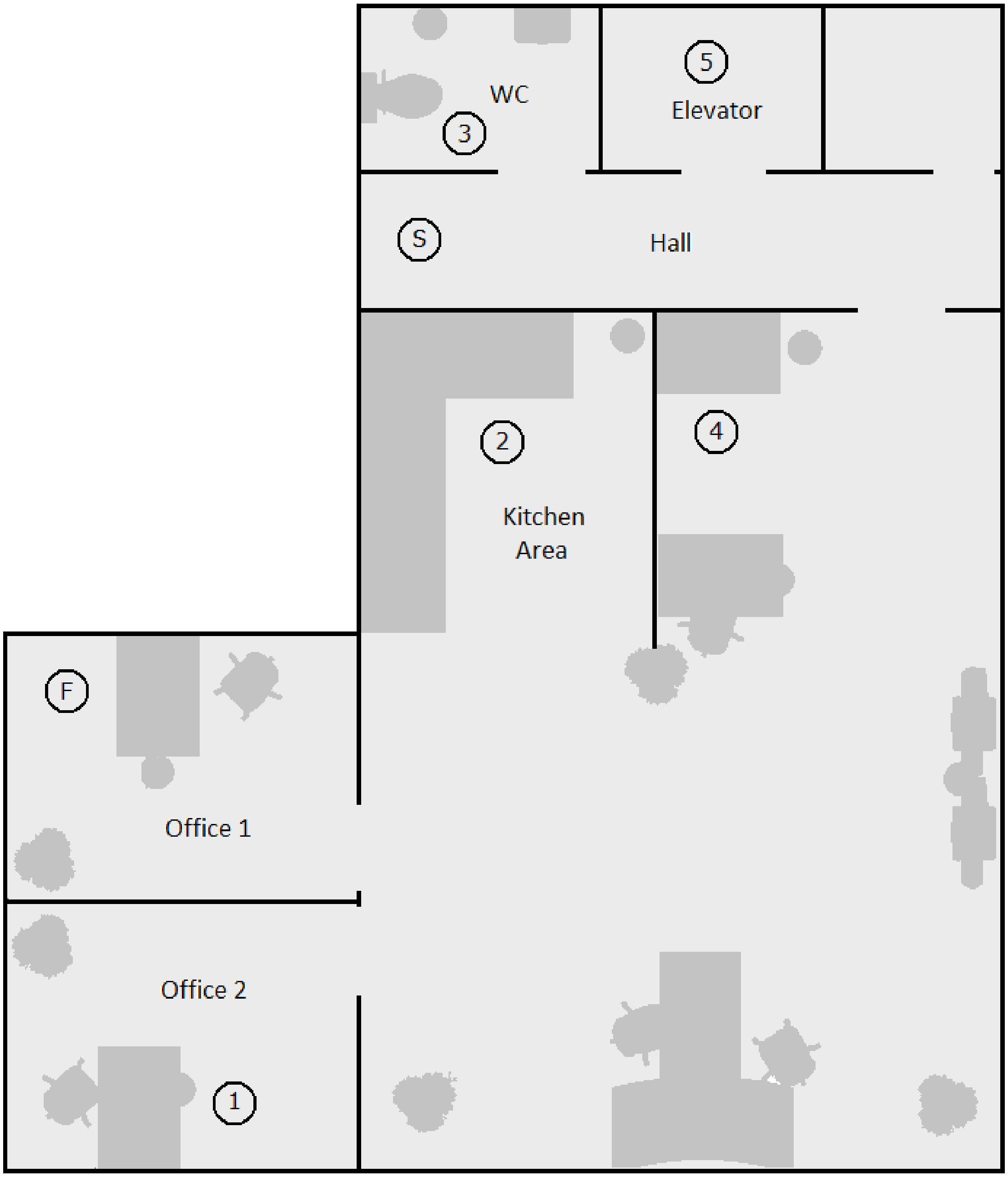

Figure 5.1: Map showing required routes for trial participants.

Chapter 6

Discussion

The aim of this project was to expand, test, and document the previously developed shared control implementation. The main objectives can be split into several sections as discussed below.

6.1 Map development

It became apparent that the maps initially designed during the creation of the simulator system did not provide all of the functionality required to test shared control functionality. Therefore a development map has been created which possesses several unique challenges which have been created to test specific functions e.g. door passage and drop-off detection. Focus has been placed on functionality rather than aesthetic appeal and the map is designed to be easily expanded to include more obstacles.

Additionally in order to achieve a more realistic environment for participant trials a simulated office environment was constructed, which has been designed to include various commonly-encountered challenges. More emphasis has been placed on aesthetic appeal and realistic effects in order to provide a more natural setting. During initial testing, the Office map containing high quality textures and models would cause the system to crash. The map has been greatly compressed in an attempt to compensate for this, however has not been sucessful, therefore there may be additional incompatibility errors with one or more elements of the map, which may be required to be replaced.

6.2 Documentation

A key factor in this project involved the documentation and quantification of the existing system and upgrades. The Strathclyde simulated wheelchair and shared control system has been developed by several students and it is expected to be further extended and improved in the future. Therefore it is of great importance that the system is sufficiently documented. This has been achieved by consolidating the system file information needed to further develop the system which can be found in the appendices.

Similarly the system capabilities and constraints must be adequately measured, documented and recorded. This allows future upgrades to be easily compared to previous versions and may help to identify any weak points or areas which require improvement. Additionally this enables the cross-comparison between various systems and allows the provision of a specifications file which can be administered to wheelchair users and distributors.

Such information was attained by analysing the configuration files for information regarding the position and nature of each sensor and by using trigonometric analysis to calculate sensor coverage.

6.3 Additional Functionality

Before commencing the addition of further wheelchair functionality it was important to determine the most useful shared control features. To achieve this a review was performed on several different environments which the wheelchair may encounter.

Consequently, two further features, door passage and drop/kerb detection, were developed in accordance with the challenges presented in this review. The former was designed to ease the navigational aspect of the wheelchair by adding a higher degree of system automation. As door passage can be considered a challenging move which may require multiple small adjustments to the wheelchair's heading this is expected to greatly reduce user effort. A series of doors decreasing in width has been included in the development map in order to test this. During development testing the wheelchair was able to correctly navigate these doorways using this functionality. However it was found that on occasion the protruding rear wheels could catch on the edge of the door, which can cause the wheelchair to tip if a forward command is maintained.

Drop-off detection was implemented as an additional safety feature. This required an additional IR sensor to be placed on the front of the wheelchair. This provides drop detection directly in front of the wheelchair, and will detect drops of more than 0.03m at a distance of 0.34 m. This may detect ramps with a gradient of 1:11 or steeper, causing the wheelchair to stop. UK building regulations allow short ramps of up to 1:12 gradient [28], therefore the detection of ramps greater than this declination may be desirable, in order to alert the user to a steep drop. This system should only be activated at the start and end of ramps as on the main body the ground is perpendicular relative to the wheelchair.

Kerb detection was implemented using the same sensor and structure as drop detection and is useful to detect small obstacles at ground level which would not be detected through the sonar array. Subsequently this allowed the detection of rising slopes in the same manor as described above. Both drop and kerb detection have been tested using the development map; depressions in the floor have been created for this purpose.

6.4 Amendments

In addition to providing new functionality existing configurations were amended to improve wheelchair performance. The increase in sensor polling frequency has improved many native functions. In particular wall following has been improved through both updating the sensor refresh rate and altering the appropriate algorithms. Furthermore, the difference between stopping distances before obstacles has been reduced as the higher sensor frequency means less distance is travelled in between checks.

During the testing of other features, performing pure rotations while near to obstacles would cause collision avoidance algorithms to stop the wheelchair moving. As this can cause much frustration, especially in navigating out of "dead-end" scenarios, additional checks were implemented in order to allow pure rotations independent of surrounding obstacles. This occasionally produced a collision of the wheelchair's corner therefore a future solution may be to allow the wheelchair to turn when sensors detect that there are no obstacles in very close proximity. However, the current implementation could be seen as acceptable as the collision speed is low, and small wheelchair bumps may be acceptable to certain users.

6.5 Future Developments

As this project simply aimed to improve the existing system it is expected that further enhancements may occur in future version upgrades. The following adjustments were not incorporated in this version, but are thought to be able to add to the system performance in the future.

6.5.1 Maps

As previously stated the development map has been designed to allow the constant modification and addition of obstacles as is needed to adequately test the wheelchair. Additional testing obstacles may include curved walls and corridor-facing doorways

Unfortunately, due to the limiting hardware capabilities the simulator system is not able to run the office map to an acceptable degree, which is thought to be due to the high quality models, textures and lighting. Therefore this map must be compressed before testing can proceed.

6.5.2 Shared Control Implementation

Future expansions of the system will likely involve implementing further navigational functionality. Additionally existing features should be improved and refined. This may involve the addition of more sensors including additional drop detection and rear collision sensors. An important requirement is that the sensors cover all of the surrounding area of the wheelchair in order to be able to detect thin or narrow obstacles at any position, such as narrow poles.

Currently many configurable values such as obstacle detection distances have been chosen from observation and estimation. It would be beneficial to test with a range of values to determine the best configuration for user comfort. As this is a subjective measure it would be best to calculate this using a large pool of subject data.

As currently implemented in other shared control systems [4] it may be advantageous to construct different system configurations for users with differing ability levels. This would aid in reducing frustration for higher-performing users while reducing stress and risk in users who are not familiar with the system.

6.5.3 Testing

It is hoped that some form of testing will commence with the repair of the simulator. It would be advantageous to test all existing features with an the upgraded system in order to test the shared control implementations specified sensor frequency. This will make it possible to determine if any undesirable behaviour is caused by the current implementations.

Furthermore comparison between the previous and upgraded shared control system should provide a measure of the value of the new and improved functionality. The recorded data should enable this comparison while also providing additional data which can be used for documentation and system quantisation purposes.

Chapter 7

Conclusions

The objectives of this project were to perform a review of the existing Strathclyde shared control interface and perform upgrade modifications as required. Subsequently, considerable changes have been implemented involving several different aspects of the system. Focus has been placed on the following elements:

- The upgrade of the existing functionality.
- The addition of new shared control algorithms, based on several situation appraisals.
- The development of new maps for testing the system.
- The quantisation and documentation of the sensor system.
- The creation of a sample testing procedure.

The outputs attained from this project include new map files, key-logging program and trial outline, which are designed to aid in recording interaction between the user and system during testing. Additionally, new code has been implemented to allow several new behaviour features to aid in the ease of control and safety of the wheelchair. Existing code and configurations have been edited to improve overall performance of the system, and some existing code has been optimised to increase efficiency and readability.

The changes implemented in this project have been demonstrated to improve the current functionality, and are expected to both increase subject performance when using the system and to promote the ease of developing and testing further version upgrades.

Bibliography

[1] Apparelyzed. Spinal cord injury statistics, June 2012. URL: http://www.apparelyzed.com/statistics.html.

[2] Spinal Research. Spinal cord injury - facts and figures, June 2012. URL: http://www.spinal-research.org/research-matters/spinal-cord-injury/facts-and-figures/.

[3] Mikhail A. Lebedev, Jose M. Carmena, Joseph E. O'Doherty, Miriam Zacksenhouse, Craig S. Henriquez, Jose C. Principe, and Miguel A. L. Nicolelis. Cortical ensemble adaptation to represent velocity of an artificial actuator controlled by a brain-machine interface. *The Journal of Neuroscience*, 25(19):4681–4693, May 2005. URL: http://www.jneurosci.org/content/25/19/4681.short.

[4] Ana C. Lopes, Gabriel Pires, Luis Vaz, and Urbano Nunes. Wheelchair navigation assisted by human-machine shared-control and a p300-based brain computer interface. In *IROS'2011*, pages 2438–2444, 2011.

[5] G. Valsan, B. Grychtol, H. Lakany, and B.A. Conway. The Strathclyde brain computer interface. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 606 –609, September 2009.

[6] B. Grychtol, H. Lakany, G. Valsan, and B.A. Conway. Human behavior integration improves classification rates in real-time bci. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(4):362 –368, aug. 2010.

[7] C. Mandel, T. Rofer, and U. Frese. Applying a 3dof orientation tracker as a human-robot interface for autonomous wheelchairs. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 52 –59, june 2007.

[8] H. Wakaumi, K. Nakamura, and T. Matsumura. Development of an automated wheelchair guided by a magnetic ferrite marker lane. *Journal of Rehabilitation Research and Development*, 29(1):27–34, 1992.

[9] Alexander Hornberg. *Handbook of Machine Vision*. Wiley-VCH, 2006.

[10] Active Robots. Sonar pricing example, 2012. URL: http://www.active-robots.com/sensors/object-detection/ultrasonic/srf05-ultrasonic-sensor.html.

[11] Active Robots. IR pricing example, 2012. URL: http://www.active-robots.com/sensors/object-detection/distance-measuring/sharp-gp2y0a02yk-sensor.html.

[12] Active Robots. Laser rangefinder pricing example, 2012. URL: http://www.active-robots.com/sensors/object-detection/laser-range-finder/hokuyo-laser-range-finder.html.

[13] Active Robots. Camera pricing example, 2012. URL:http://www.active-robots.com/sensors/camera-vision/cmos-camera-1300x1040.html.

[14] Mako Surgical Corp. MAKOPlasty Total Hip Arthroplasty, April 2012. URL: http://www.makosurgical.com/makoplasty/hip.html.

[15] Eleanor A. Curran and Maria J. Stokes. Learning to control brain activity: A review of the production and control of eeg components for driving brain-computer interface (bci) systems. *Brain and Cognition*, 51(3):326–336, April 2003. URL: http://www.sciencedirect.com/science/article/pii/S0278262603000368.

[16] F Lotte, M Congedo, A Lécuyer, F Lamarche, and B Arnaldi. A review of classification algorithms for eeg-based brain-computer interfaces. *Journal of Neural Engineering*, 4(2):R1, 2007. URL: http://stacks.iop.org/1741-2552/4/i=2/a=R01.

[17] J. Philips, J. del R. Millan, G. Vanacker, E. Lew, F. Galan, P.W. Ferrez, H. Van Brussel, and M. Nuttin. Adaptive shared control of a brain-actuated simulated wheelchair. In *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pages 408 –414, june 2007.

[18] Richard C Simpson. Smart wheelchairs: A literature review. *Journal of Rehabilitation Research & Development*, 42(4):423–436, 2005. URL: `http://www.rehab.research.va.gov/jour/05/42/4/pdf/simpson.pdf`.

[19] S.P. Levine, D.A. Bell, L.A. Jaros, R.C. Simpson, Y. Koren, and J. Borenstein. The navchair assistive wheelchair navigation system. *Rehabilitation Engineering, IEEE Transactions on*, 7(4):443 –451, December 1999.

[20] Y. Kuno, N. Shimada, and Y. Shirai. Look where you're going [robotic wheelchair]. *Robotics Automation Magazine, IEEE*, 10(1):26 – 34, March 2003.

[21] Holly A. Yanco. *Shared User-Computer Control of a Robotic Wheelchair System*. PhD thesis, Massachusetts Institute of Technology, September 2000.

[22] C. Urdiales, J. Peula, M. Fdez-Carmona, C. Barru, E. Prez, I. Snchez-Tato, J. del Toro, F. Galluppi, U. Corts, R. Annichiaricco, C. Caltagirone, and F. Sandoval. A new multi-criteria optimization strategy for shared control inwheelchair assisted navigation. *Autonomous Robots*, 30(2):179–197, 2011. 10.1007/s10514-010-9211-2.

[23] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. 1986. URL: `http://robotics.stanford.edu/~latombe/cs326/2002/khatib-pf.pdf`.

[24] B. Grychtol. *A Virtual Reality Electric Powered Wheelchair Simulator*. PhD thesis, Univeristy of Strathclyde, 2011.

[25] S. Carpi, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Usarsin: a robotic simulator for research and education. URL: `https://robotics.ucmerced.edu/Robotics/papers/usarsimpaper.pdf`.

[26] Epic Games. UnrealEngine. URL: `http://udn.epicgames.com/Two/WebHome.html`.

[27] R. Menon. Shared control between man and machine in brain computer interfaces. Master's thesis, Univeristy of Strathclyde, August 2011.

[28] HM Government. The building regulations 2010 - part m (the access to and use of buildings), 2010. URL: http://www.planningportal.gov.uk/buildingregulations/approveddocuments/partm/approved.

[29] Spinal Cord Injury Info Pages. Spinal cord injury facts & statistics, June 2012. URL: http://www.sci-info-pages.com/facts.html.

Appendix A

System Structure Documentation

The following files are contained within the *Wheelchair* Microsoft Visual Studio Solution which controls all aspects of the wheelchair functionality within the simulator.

The main shared control algorithms are included in the *SharedControl.cpp* file.

The *StateControl.cpp* file contains low-level command data relating to the wheelchair navigation. When shared control is activated the alternative SharedTransFun() function is implemented. The only alteration to this file is the inclusion of a flag used to allow pure rotations.

The *USARBot.ini* found within *C:\UT2004\System* contains the data needed to construct the robots for USARsim. This is where new sensors can be added and configured, including specific sensor configuration such as scan interval, beam angle, etc. After editing this file the *make.bat* file must be run to apply the changes.

The addition of new static meshes, textures or maps must be added to their respective file in the system folder.

Wheelchair logs can be found in the *C:\WheelchairSimulator* folder, presented in text files named after the date and time at which the simulation commenced. The matlab files needed to analyse these logs can be found in the *Matlab* folder of the Bartek Thesis CD.

Appendix B

AutoHotkey Script for Counting Keystrokes

```
; initialise variables
8count = 0
2count = 0
4count = 0
6count = 0
spacecount = 0

; increment on keypress without affecting function
~Numpad8::
8count++
return

~Numpad2::
2count++
return

~Numpad4::
```

```
4count++
return

~Numpad6::
6count++
return

~Numpad5::
~Space::
spacecount++
return

; display messagebox on Ctrl+F12
^F12::
MsgBox Forward: %8count%.`rBack: %2count%.`rLeft: %4count%.`rRight: %6count%.
`rStop: %spacecount%.
return
```

This code can be executed as an .ahk file on the simulator computer or any machine with AutoHotkey installed. Additionally, an executable file has been compiled which allows the script to be run on most computers.

Appendix C

Review Survey for Trial Participants

Although the system could not be tested due to a malfunction, several testing preparations has already been initiated. The following pages show an example consent form and survey which was designed to be distributed to trial participants.