DEPARTMENT OF BIOMEDICAl ENGINEERING

University of Strathclyde Engineering

Design of a controller for a rehabilitation robotic system with one degree of freedom

This thesis is submitted to the

Department of Biomedical Engineering

of the Strathclyde University by

Claudio Accarino

M.Sc. in Biomedical Engineering

YEAR 2013-2014

‘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: 23rd September 2014

Acknowledgment

I would like to thank my supervisors Dr Wei Yao and Dr Mario Giardini. My discussions with Dr Yao were of fundamental importance in the early stage of the project planning. I am grateful for his support and availability both in the project making and in the thesis writing.

Dr Giardini provided me a electrooptic muscle activity signal that was the key to test my designed system in a real operating condition. Furthermore, he was very helpful in the late discussion on the signal and the system too.

I cannot forget to thank all my friends for their help and encouragement.

A special thanks is due to my family and my girlfriend for their continuous support and encouragement. They helped me a lot over the year and especially in the end. Their support was without any doubts the incentive that let me finish this wonderful year.

Contents

List of figures

List of tables

Abstract

Post stroke impairments are one of the pathologies most demanding for rehabilitation. Regular rehabilitation is very expensive both economically and socially and it requires physiotherapists to work very hard. Robotic rehabilitation is seen as an excellent substitute to the regular rehabilitation.

This work deals with the controller of the actuators of a generic exoskeleton system for the upper limb rehabilitation.

The exoskeleton actuator control is performed via the Simulink software in combination with the CPU of a normal computer. The use of Simulink ensures a full, easy and straightforward customisation of the system response. So the patient/physiotherapist’s needs can be fulfilled without subverting the system stability.

The actuator control was tested with signals related to one person’s muscle activity, simulating a rehabilitation session with an active role of the patient. Furthermore, many tests were performed with signals generated in Simulink and thought to be used as exercise on a passive patient.

Hence, the designed controller showed to successfully control an exoskeleton actuator with great accuracy, repeatability and with a good timing. These results were confirmed both in patient in charge and robot in charge tasks.

Introduction

In the last decades robotic systems have undergone a wide expansion in the rehabilitation field due both to the initial success they have had and the following increasing demand.

There are a wide number of pathologies and conditions requiring rehabilitation and in developed countries both post stroke impairments and the increased expectative of life represent the two most common of all.

The purpose of the rehabilitation process is to improve both muscle tone and motion capacity, minimizing or eliminating the functional impairment at the same time. This process is performed through specific movements leading to increase the muscle plasticity.

Normal rehabilitation has a very high economic and social cost while hard work is required to the physiotherapists.

Robotic system rehabilitation is seen as an excellent improvement of normal rehabilitation since a greater number of repetition of therapeutic exercises can be done. Furthermore, robots can reduce rehabilitation costs and give quantitative feedback, like muscle force and angle of movement, on one patient’s status that can be used for a better follow up [1].

Aim of the work

The aim of this work is to design a controller for an exoskeleton system using the Simulink software. The controller must allow the exoskeleton to move, via its actuator control, according to a given input signal. The exoskeleton actuator control is made using the CPU of a normal computer. In this work the connection between the computer and the actuator is realized through an Arduino board.

This exoskeleton system is thought for biomedical rehabilitation purpose. So the whole system must be triggered by a signal related to the muscle activity of the patient and perform the same movement of the limb of the patient. Both the timing and the amplitude of the movement must be respected. Obviously the system does not have to follow the input signal of the patient’s muscular activity, but instead it must respond to that signal like the real limb does.

Anyway, a kind of rehabilitation, with the patient’s passive role rather than active, was thought to be used with this system too.

The controller is thought for the simplest robotic system having only one degree of freedom, such as that used to flex or extend upper and lower limbs or fingers.

1 Robot in charge and Patient in charge

In a robotic exoskeleton system used for rehabilitation purpose there are two possible options on who has the control: the robot or the patient.

When patients have the control the system is known as “patient in charge” while “robot in charge” obviously means the patient is passive during the movement[2].

1.1 Patient in charge

In order to let the “patient in charge” condition work properly there is the need of a signal related somehow to the patient movement. Several signals suit this task such as surface electromyography (sEMG), electroencephalography (EEG), electrooptical signals and so on.

Usually these signals cannot be used as they are sampled because of the noise so they must be filtered, amplified, and must undergo another processing to allow them to be suitable for a robotic system controlled either in position, speed, torque or whatever is the controlled variable.

In particular, the robotic system must use these kinds of signals as trigger rather than follow them. In order to achieve this there must be a sort of conversion between the signal and the controlled variable of the robotic system. For example, to simulate a patient in charge task in this work I assumed, without any evidence, that the highest peak of the electrooptical signal correspond to a contraction of the muscle causing the forearm(the thigh or a finger) to flex of 120°. Furthermore, I assumed the relationship between the signal and the angle position to be linear, just because it is the easiest case possible.

Obviously further developments and testing should be done on electrooptical signals before a real usage on an exoskeleton. Anyway, they are not part of this work.

1.2 Robot in charge

The “robot in charge” mode is much easier than the “Patient in charge” one. In fact, the signal the robot system must perform is decided prior to the performance. The signal should be accurately created according to the need and it can be changed time to time to fit best the requirements.

For example, it is possible to design a signal to flex the forearm (or a finger or the shank) of 90° and then to extend it back to the null position. A triangular waveform signal will perfectly suit this task.

The same signal can be replicated more times during the same trial if needed.

Furthermore, there are several parameters can be set according to the needs:

- Required time for the flexion\extension to be completed;
- Maximum velocity;
- Repetition time [2].

2 Literature review

2.1 Digits rehabilitation robotic systems

Robotic rehabilitation has recently increased to provide patients full of range of motion after several vascular accidents, or strokes. This causes neurological deficit and the most common affected domain is the motor system. Rehabilitation robotics benefits include incorporation of interactive virtual reality systems, a large number of quantitative data to optimize therapy and assess patient outcomes, the availability of precise and repeatable therapeutic exercises, reduced number of therapists. Robot-assisted therapy is proven to improve proximal arm function [3–8]. So a number of robots has been developed for hand motor therapy, including a regained range of motion and the ability to grasp objects. All these devices vary for degrees of freedom (DOFs), range of motion and design.

One kind of devices is based on an endpoint control strategy, whereby forces are applied to the distal segments of the digits.

HandCARE uses cable loops attached to the ends of each digit [9,10].

Rutgers Hand Master II is a force-feedback glove powered by pneumatic pistons placed in the palm of the hand [11] that proved to be successful with chronic stroke patients in clinical and functional gains [12,13].

AMADEO is another commercially available device that provides endpoint control of each digit along fixed trajectories.

Another kind of device is represented by “actuated objects” that can expand or contract. The ”haptic knob” uses an actuated parallelogram structure with 2 movable surfaces the subject grips [14]. The InMotion Hand Robot uses a double crank and slider mechanism driven by an electric motor, that controls the radius of the cylinder and allows grasping motions [15].

But these kinds of endpoint control and actuated objects show limited control of the proximal joints of the fingers , with possible physiologically inaccurate joint kinematics, especially in subjects with abnormal increase of flexor tone. Applying torque to each joint of the finger in a fixed ratio can be an alternate approach. Two cable-driven devices have been developed for individual control of the fingers and thumb with pulley system on the back of the hand [16,17]. Overall weight is reduced by remote location of motor so it is possible to use these devices in conjunction with arm movements.

Another approach uses a glove with an air bladder and channels along the palmar side of the digits. Air pressure regulated by an electro-pneumatic servo valve helps digit extension.

But this device proved to give modest functional gains [18]. These approaches, however proved to provide a not adjustable ratio of torques applied to the joints in a digit with a possible abnormal joint kinematics.

Exoskeletons represent another kind of devices. The characteristic of exoskeleton is that its joint are aligned with the anatomical ones, allowing for proper inter-joint coordination between anatomical joints.

HAND WRIST ASSISTIVE REHABILITATION DEVICE (HWARD) is a robot that controls finger rotation about the metacarpophalangeal joint(MCP), thumb abduction\adduction and wrist extension\flexion [19].

HAND Mentor (Kinematic Muscle Inc., Tempe, AZ) is an exoskeleton device based on the simultaneously extension and flexion of the fingers and the wrist by an artificial muscle, but the motion does not include the thumb.

Many of these devices have given significant clinical and functional gains.

A new kind of exoskeleton is the Hand Exoskeleton Rehabilitation Robot HEXORR that uses low-friction geartrains and electric motors.

It allows for implementation of both position and torque control with enough torque capacity to open a hand with high flexor tone. The advantages of HEXORR are the physiologically accurate grasping and nearly full ROM for every digit of the hand. HERROR is controlled with only two actuators and the thumb activator allows for variable thumb plane of motion to incorporate different degrees of extension\flexion and adduction\abduction. A force assistance mode is available to provide extension assistance according to user’s needs. So HEXORR represents the most complete device to date. It is based on these criteria:

1. Digits full ROM;
2. Emulating physiologically accurate kinematics trajectories;
3. Comfortably fitting different hand sizes [20].

The exoskeleton must incorporate hardware and software safety mechanisms because of users’ impaired hands. Safety stops are positioned so that the fingers and the thumb cannot be hyper-extended, a kill switch allows to shut down both motors at any time. HEXORR has software ROM stops too.

Before each training session, it is possible to limit the device’s ROM via the graphical user interface and the velocity of the linkages through software controls.

The motor command too is always under control by saturation levels.

The pilot study shows that HEXORR was able of moving the hand digits through the whole ROM following physiologically accurate trajectories. In fact, stroke patients were at least able to extend their fingers although they had mild to moderate motor function impairment. They could extend and flex their digits inside of HEXORR which succeeded in increasing the subject’s ROM and promoting active participation[20].

2.2 Lower limb robotic rehabilitation [1]

Rehabilitation aims to improve the motor plasticity of the patient through motor recovery and lower functional deficits. Rehabilitation is based on specific exercises of limbs. A study about lower limb rehabilitation divides the process in three phases :

1. The chair as soon as possible;
2. Restoration of gait;
3. Improvement of gait (i.e. training of free walking if possible).

Because of the enormous economic load and a higher quality of technics robotics for rehabilitation is an emerging solution.

Passive robotics devices are less complex and cheaper but cannot supply energy to the impaired limbs whilst activity devices can.

The last lower limb rehabilitation robots can be grouped according the principle they are based on:

1. Treadmill gait trainers
2. Foot-plate-based gait trainers
3. Overground gait trainers

4. Stationary gait trainers
5. Ankle robotic systems
    a. Stationary systems
    b. Active foot orthoses

This rehabilitation technique consists in particular bodyweight support treadmill training (PBWSTT). Three therapists assist the patient. Many robots have been designed to improve this technique through automation systems such as exoskeleton type robots beside a treadmill. Only 3 of these subsystems are commercially available: the LOKOMAT, the LOKOHELP, the REO AMBULATOR.

The first one consists of a robotic gait orthoses and an advanced bodyweight support system, combined with a treadmill [21].

It uses drives (controlled motors) integrated in the gait orthoses of each hip and knee joint. The drives are precisely synchronized with the treadmill speed. It is the most clinically evaluated system until now.

The second is an electromechanical device to improve gait after brain injury [22]. It improves the same gait ability as manual locomotor training, but few therapists are required and much comfort to the patient is assured.

The third is based on a robotic arm strapped to the patient's legs at the thigh and ankle.

Other robotic systems are at a research state or under development such as those for locomotor training after spinal cord injury at California University: ARTHUR (Ambulation Assisting Robotic Tool for Human Rehabilitation) designed to measure and manipulate human stepping on a treadmill [23] reducing trainers manual assistance; POGO

(Pneumatically Operator Gait Orthosis) an improved leg robot design; PAM (Pelvic Assist Manipulator) to control and accommodate naturalistic pelvic motion [24].

ALEX (Active Leg Exoskeleton) is a powered leg orthosis with linear actuators at the hip and knee joint with a force-field controller using the assist-as-needed approach [25]. It showed to improve patient’s gait pattern and walking speed on the treadmill.

LOPES (Lower-extremity Powered Exoskeleton) moves in parallel with the leg of a person walking on a treadmill, connected at pelvis height. There is already a first clinical trial that tests the efficacy of this system.

ALTRACCO, RGR and String-Man are three robotic systems under research, ALTRACCO uses a lightweight, pneumatic actuator [26]. It consists of a unilateral exoskeleton and a supportive arm to passively gravity-balance the device.

The RGR was built to target secondary gait deviations in stroke patients. Force fields applied to the pelvis generate corrective forces to balance deviations from normal pelvic motion [27].

The String-Man has a particular kinematic structure with 7 wires attached to the patient’s trunk.

2.2.1 Foot-plate based

These robotic systems control movements off patient’s feet positioned on separate foot plates to simulate different gait patterns. The only system on the market is the GTI (Gang Trainer) pioneer of robotic rehabilitation. It is at least as effective as the manual therapy but undemanding for the therapist.

The GT5 simulates walking and climbing moving the user’s foot forward or up and down.

A 6 degree of freedom (DOF) allows walking velocity updates on various terrains. It has been tested with hemiplegic patients [28].

2.2.2 Overground gait trainers

These kinds of robots allow patients move under their own control:

- The Kine Assist is a device for gait and balance training [29].
- The Walk Trainer is a robot system used for rehabilitation. It consists of a deambulator, a pelvis orthosis, a body weight support, 2 leg orthoses and a real-time controlled electro stimulator [30].
- Rewalk is a quasi-robotic suit. It is wearable and motorised. It is made by ARGO Medical Technologies ltd. and upper-body movement of patients are detected too to initiate and maintain walking process. It is undergoing clinical test at the Moss Rehabilitation Hospital in Philadelphia.
- Hybrid Assistive Limb (HAL) is a wearable robot used for rehabilitation or heavy works support. There are two versions of the device, a full body and a 2 legs one. A single-leg version is also available for hemiplegic subjects [31].
- WHERE I and II are two mobile gait systems for rehabilitation that allow overground gait, training. They are undergoing clinical test [32].

2.2.3 Stationary gait trainers

These are robotic systems based on guided movements of limbs to optimize therapeutic and functional use. They aim to efficient strengthening of the muscles and the development of endurance, joint mobility and movement coordination.

MotionMaker (Swortec SA) allows fitness exercises with active participation of the paralyzed limbs [33]. Its advantage is its real-time sensor-controlled exercises, combined with the controlled electrostimulation. Patients showed to have improved their voluntary force during a leg-press movement according to first clinical trials. Two other robotic systems: Lambda and a wire-driven leg rehabilitation system, were developed with a similar working principle.

2.2.4 Stationary systems

Stationary systems are robotics mechanism designed to exercise the motions of the ankle and the knee of the target limb without walking.

The Italian Institute of Technology (IIT) has developed a High Performance Ankle Rehabilitation Robot [34]. It allows plantar/dorsiflexion and inversion/eversion.

The Active Knee R. Orthotic Devices (AKROD) represent a recent system that provides variable damping at the knee joint, useful to facilitate motor recovery in post stroke and neurological disease patients. It is now a stationary system but an AKROD system during walking is going to be actuated.

2.2.5 Active foot orthoses

These are actuated exoskeletons wearable while walking overground or in a treadmill. They aim to control the position and the motion of the ankle, compensate for weakness, correct deformities.

Till date the only commercialized system for rehabilitation is the Anklebot developed at the MIT for ankle rehabilitation after stroke. A normal range of motion in all 3 DOF of the foot relative to the shank while walking overground or on a treadmill is possible [35,36].

Finally, clinical studies still show little superior effectiveness of robotic therapy, although time and cost are at least reduced.

2.2.6 Challenges

Robotic systems are believed to be the best future tools for rehabilitation.

However, standard protocols and procedures to obtain assessment data are still being developed. Currently, patient recovery of walking ability is measurable using Barthel index [37]. Gait velocity and walking distance, ROM, and many other dynamic measures have been used for assessment, but there is not a universal accepted method. So clinical criteria and trials are expected in the near future because clinical studies still show too little superior effectiveness of robotic therapy, although time and cost are at least reduced. Furthermore, they could be used independently at home by patients to support and optimize therapy rehabilitation program. However, higher benefits should be pursued in rehabilitation program from robotic system.

2.3 NEUROEXOS

NEUROEXOS is a new exoskeleton used for elbow rehabilitation. It is characterized by (three innovative elements): the double-shelled links, the 4 DOF passive mechanism and a compliant antagonist actuation system. It has undoubtedly some advantages such as a wide human-robot physical interface able to transmit interaction torque, the kinematic compatibility between the human and the exoskeleton with a proper torque transmission to the human joint without any risk for patient's articulations because of overloading, a safe and effective actuation system, which can allow the execution of both robot-in-charge and patient-in-charge rehabilitation exercises. It is desirable in the near future to spread NEUROEXOS use in clinics in post-stroke rehabilitation trials and to explore design solutions for developing a more compact actuation and control system, based on electromagnetic motors and embedded control units to increase the performance of the system and its acceptability[2].

3 Control theory

In a generic dynamic system, robot is an example, there is the need for a control in order to obtain a system with the desired properties, such as stability, performance, accuracy, etc.

The simplest system possible known as single input single output (SISO), shown in Figure 1, perfectly represents the system used in this project.

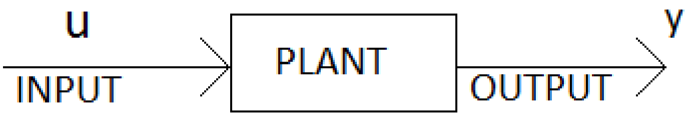


Figure 1 SISO system

The input and the output of a plant are linked by a physical relationship, for example a differential equation, whose complexity varies according to the complexity of the plant itself.

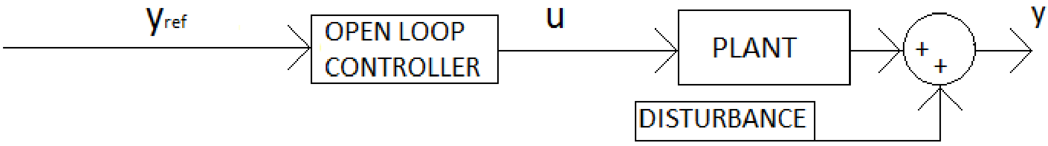


Figure 2 Open loop system

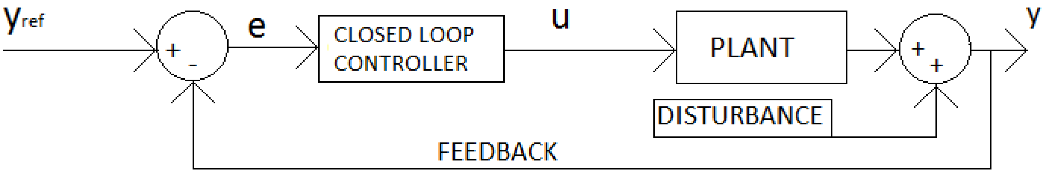


Figure 3 Closed loop system

There are two possible kinds of control configuration: open loop and closed loop (also known as feedback). In the first configuration, shown in Figure 2, the controller has information only about the required control variable (y_{ref}) and in some cases the disturbance. In this case the configuration is called open loop with disturbance measure. In the other configuration, shown in Figure 3, the controller is provided with information on the output controlled variable (y) too.

As it can be noticed the controller is always inserted between the required controlled variable value (y_{ref}) and the plant in order to have a direct effect on it.

Furthermore, the required controlled variable (y_{ref}) and the plant output controlled variable (y) must have the same physical characteristic otherwise the difference between these two signal values (y_{ref}-y=e) would not have any physical meaning.

The resulting value represents the difference of the values between the actual state of the plant and the desired state.

Another important consideration regards the characteristics of the system input signal (y_{ref}). For example, it is possible to have a constant input reference signal [y_{ref}(t)=constant] or a periodic or semi-periodic one [y_{ref}(t)=cos(t)] or in general a non-periodic non-constant signal.

Sometimes it is possible an actuator is inserted between the controller and the plant, but this case will not be discussed in this work.

Open loop controllers can be used only in the case of a well-established knowledge and they are used with very simple process, but closed loop controllers are the most used ones.

For the purpose of this project the main goal of the controller is to bring the value of the plant output controlled variable as close as possible to the required controlled variable value and maintain this value over the whole period.

Actually, this is the main purpose of all the controllers when a desired output is required by the system.

Another possible goal of the controller is to maximize or minimize the output plant but this kind of controllers are mainly applied for economical purpose, so they are not discussed in this work [38].

3.1 Motor control

According to the literature there are two ways to control a motor using either the angle position or the angular speed of the motor shaft. In both cases sensor providing information on angle position or angular speed or both is required as a feedback provider[39].

Optical encoder are a good choice as feedback for both angle position and angular speed [39], while potentiometers, especially the precision rotary potentiometers, are a good choice as position feedback provider [40,41].

Regardless what is the controlled variable the signal sent to the motor it is usually a Pulse Width Modulation (PWM) signal, see section 3.3.1 for PWM description. So a conversion between either speed or angle position and PWM must be set somehow.

The figure below shows the block diagram of a generic motor control system:

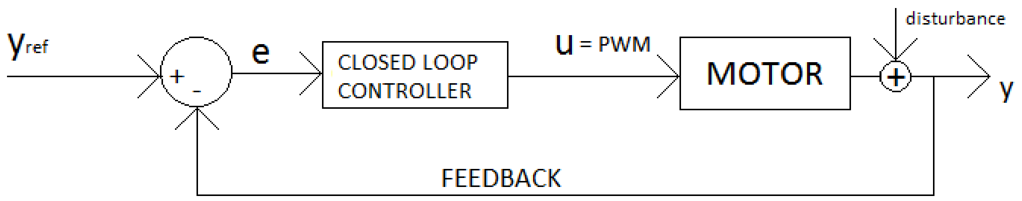


Figure 4 Closed loop Motor system

3.1.1 PWM

Pulse Width Modulation (PWM) is a well-established technique used to control analogic circuit using digital signals generated via a digital microprocessor. This technique is mainly used in robotic, motor control, power applications and sometimes conversion fields.

The reason why a digital signal is preferred to an analogical one is mainly due to economic reason of power saving.

The digital output of the microprocessor is a digital one having only two values: null (down) and full (up) and a period “T”. To convert the analogical signal in the digital one, the microprocessor controls the duty-cycle of the period “T” of the digital output signal.

This way the analogic circuit will read the average value of the digital signal in the “T” period.

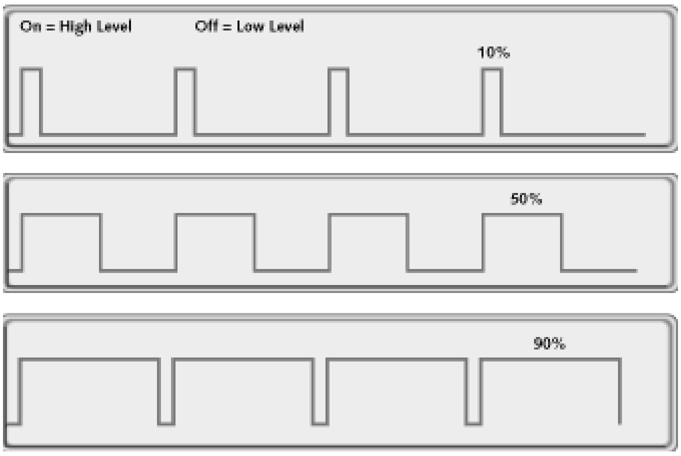


Figure 5 Example of PWM [42]

Below it is reported an example considering a waveform period of 1 s.

The analogic signal is ranged between 0 and 5 V hence the digital signal will have only two values: 0 and 5V.

If the analogic signal assumes a value of 5 V the digital output will be a waveform assuming the full (up) value, that is 5 V, for all the period (1 s). In fact, this way the average of the signal along the period is exactly 5 V.

If the analogical value is 3 V the digital output waveform must assume the full (up) value , that is 5 V, for 3/5 of the period so the average value will be equal to 3 V [42].

3.2 PID controller

A PID controller has 3 different gains (proportional, integral, derivative) that can be used in any possible configuration.

Looking at Figure 3 it can be noticed that:

$$e(t) = y_{ref}(t) - y(t) \tag{1}$$

The normal equation of a PID controller is:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \dot{e}(t) \tag{2}$$

where:

e(t) : error, it is the difference between the required controlled variable value [y_{ref}(t)] and the output controlled variable value [y(t)] as defined by (1)];

u(t) : input signal of the plant;

k_p, k_i, k_d : gains of the PID controller.

The typical response (waveform) of a PID controller and its characteristics are shown in Figure 6 below.

In a more general term the waveform represented below and its features are common of all the closed loop dynamic systems and they are not an exclusive of a PID controller.

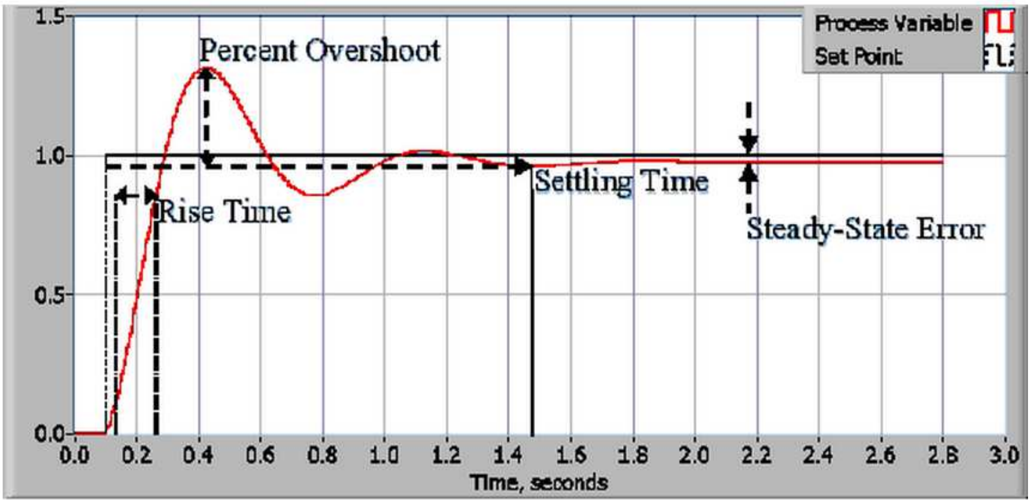


Figure 6 PID controller waveform [43]

"Rise Time is the amount of time the system takes to go from 10% to 90% of the steady-state, or final, value. Percent Overshoot is the amount that the process variable overshoots the final value, expressed as a percentage of the final value. Settling time is the time required for the process variable to settle to within a certain percentage (commonly 5%) of the final value. Steady-State Error is the final difference between the process variable and set point" according to the article PID Theory Explained – National Instruments, 2014 [43].

Now it is possible to consider how the response curve characteristics are linked to the PID terms. Where the word term means the gain multiplied by the error as specified for each term.

The proportional term [k_p*e(t)] is responsible for the raising time since it governs the ratio of the output controlled variable of the system response to the error (u(t) according to (2)). The more is the value of k_p and less time is required to reach the reference point so the rise time is reduced. On the other hand, if k_p is too big it leads the system to oscillate until for a critical value of the proportional gain, known as k_{pc}, the system becomes instable.

The integral term $\left[k_i \int_0^t e(\tau)d\tau\right]$ makes the summation of the error during the whole work period. Since the summation of even a tiny error will generate a considerable error the only way to not introduce this error is to have no error at all. So the effect of the integral term is to bring the steady-state error to zero. The bigger is the integral gain and the greater is the error introduced.

The derivative term $[k_d \dot{e}(t)]$ determines the slope of the error over the time and it responds to the rate of change of the controlled variable (see (1)). So the more is the rate of change and stronger is the effect of the derivative term. Hence, the effect of this term is to smooth the slope of the waveform. Noise is a big problem when using the derivative term because this one can increase the effect of the noise [43—45].

3.2.1 PI controller demonstration

As an example it is possible to see what happens to a generic system when the plant is controlled by a PI controller.

The block diagram of the system is shown in Figure 7 below:

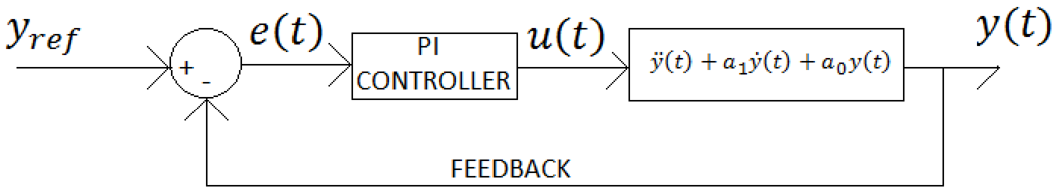


Figure 7 Generic system controlled by a PI controller

$$u(t) = k_pe(t) + k_i\int_0^t e(\tau)d\tau \tag{3}$$

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0y(t) = k_p\{y_{ref} - y(t)\} + k_i\int_0^t\{y_{ref} - y(\tau)\}d\tau \tag{4}$$

Taking the first differential to eliminate the integration (4) becomes

$$\dddot{y}(t) + a_1\ddot{y}(t) + a_o\dot{y}(t) = -k_p\dot{y}(t) + k_i\{y_{ref} - y(t)\} \tag{5}$$

So considering the steady-state response

$$k_i\{y_{ref} - y_\infty\} = 0 \tag{6}$$

hence

$$y_\infty = y_{ref} \ \ \forall k_i \neq 0 \tag{7}$$

3.2.2 ANTI wind-up

A common problem that can happen when using a PID controller is due to the non-linear behaviour of the actuator. Usually actuators have a limited range for both input and output so if the plant reaches the saturation state the system will switch from closed loop to open loop with all the problem discussed in section 3.0.

Actually in the system created for this work there is not a real actuator, but the *continuous servo write block* has a limited range for both its input and output signals, as discussed in section 4.2. So if the output of the PID controller exceeds the range of value accepted by the *continuous servo write block* this will cause its saturation.

Once this block is saturated the PID controller will lose the control of the system until the *continuous servo write block* will come back to the non- saturated region.

Since this problem is mainly due to the integral gain of the PID controller the most used and efficient remedy is to switch off the integral gain when the output of the PID controller reaches the saturation region of the plant.

So in the case of the system used in this work the output of the PI controller (the differential gain was not used in this work) was limited to the range of the *continuous servo write block input values* [46].

4 Simulink models

4.1 Arduino board

When implementing the real model one of the first problem was the way to interface the software(Simulink) with the hardware (motor and sensor). Actually there are several input\output boards (I\O boards) compatible with Simulink but I choose to use an Arduino Board, in particular Arduino Mega 2560, since Arduino environment presents the following characteristics as reported in the Arduino web site [47]:

- Inexpensive compared to other microcontroller platforms;
- The Arduino software is designed to run on all the major operative systems, such as Windows, Macintosh OSX, and Linux;
- Simple and clear especially for beginners;
- Open source and extensible software;
- Full compatible with Simulink [47].

Moreover, the Arduino library for Simulink is very straightforward and intuitive to use.

According to the web site [48] the Arduino MEGA 2560 board has the following technical features :

Table 1 Arduino MEGA 2560 technical features [48]

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
ADC	10 bit (1024 levels)
	Arduino Motor Shield compatible

4.2 Real model

Figure 8 shows the system block diagram for this work. The motor is controlled in position but the signal given to the motor is an angular speed one. This configuration seems to be in contradiction with the discussion made in section 3.2 “Motor Control” but it indeed is correct. In fact, the angular speed signal is used because, as explained below, the Simulink block providing the signal to the motor converts a dimensionless speed in PWM.

The conversion from angle to angular speed and all the other conversions are explained in section 4.3.1

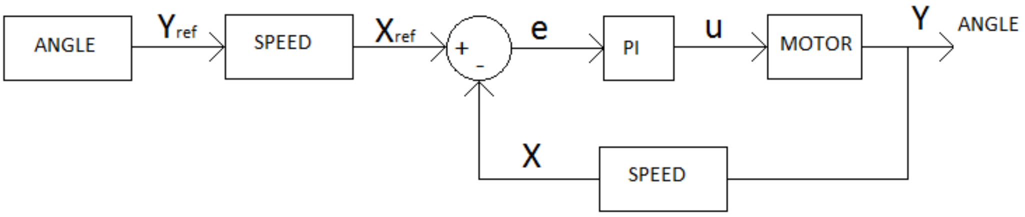


Figure 8 Model block diagram

The additional Simulink library for Arduino was required to create this model.

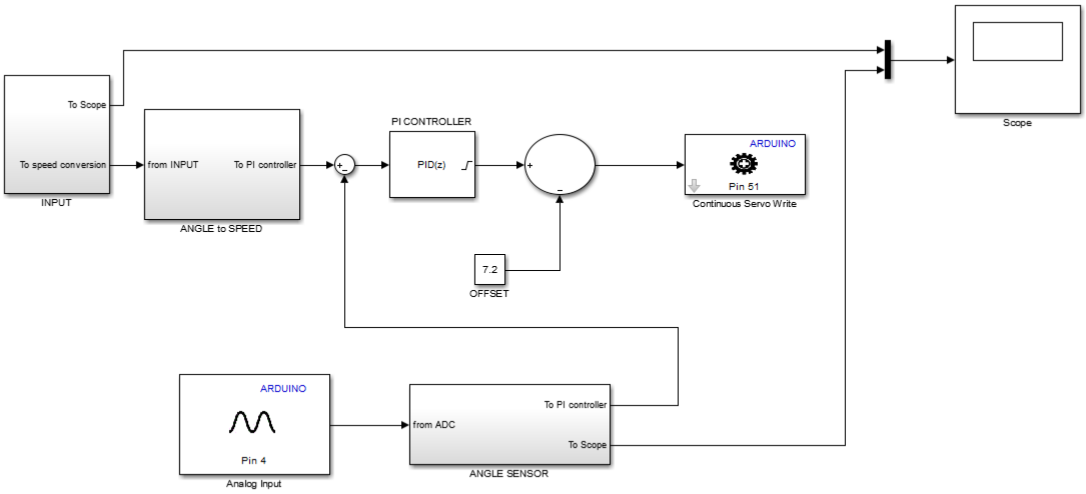


Figure 9 Whole system Simulink project

The Simulink model, shown in Figure 9, is composed by the following blocks:

- A *continuous servo write* ;
- *An analog input*;
- A *summation\subtracting*;
- A *discrete PID controller*;
- A *scope*;
- A *mux*;
- A *constant block*;
- A *subsystem* used for data conversion of the input, called “Angle to Speed”;

- A *subsystem* used for data conversion of the sensor, named “Angle sensor”;
- A *subsystem* for the input.

The subsystem for the input is discussed in section 4.2.2, 4.2.3 and 4.2.4.

The signal to the motor is given through the SIMULINK block *continuous servo write*. This block accepts an input value in the scale from -90 to 90 (dimensionless) and sets the angular speed value of the motor generating a PWM signal, see section 3.3.1 for PWM. The input scale is normally symmetrical around 0 that is the value corresponding to 0 angular speed, so the values of 90 and -90 represent the maximum motor angular speed in clockwise and counter clockwise direction respectively. Instead in this work there is an off-set of -7,2 because of the required technical modification on the motor, see Section 5.4. It must be added after the PI controller in order to restore the null condition in the *continuous servo write* block.

The sign “Pin 51” written in this block means the active pin on the Arduino board is 51 and this one must be connected to the motor.

The *analog input* block can read, via the analogic pins on the board, any tension between 0 and 12 V and it gives a digital value between 0 and 1023 in Simulink. The Arduino board pin used for this block was the 4th analogic.

The subtracting block deducts the feedback signal from the input signal generating the signal known in section 3 as “e(t)” and marked as “e” in Figure 8 .

This signal is the input for the *discrete PI controller* that elaborates e(t) in order to obtain the required waveform characteristics. The output of the *discrete PI controller* is sent to the *continuous servo write* that converts it in PWM and sends it to the real servo motor.

The *mux,* sometimes better known as multiplexer, is a device used to send more than one signal on one channel. In this project it was required to plot the input signal and the sensor signal on the same *Scope*.

4.2.1 Conversion values and conversion subsystems

The Angle to Speed subsystem, shown in Figure 10, performs the conversion from angle to speed. This way the signal coming from the Input subsystem is converted and sent to the PI controller.

The conversion is performed via a *gain block* whose value is set through a mask, shown in Figure 11, in order to allow a faster and easier management.

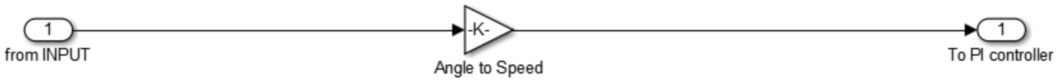
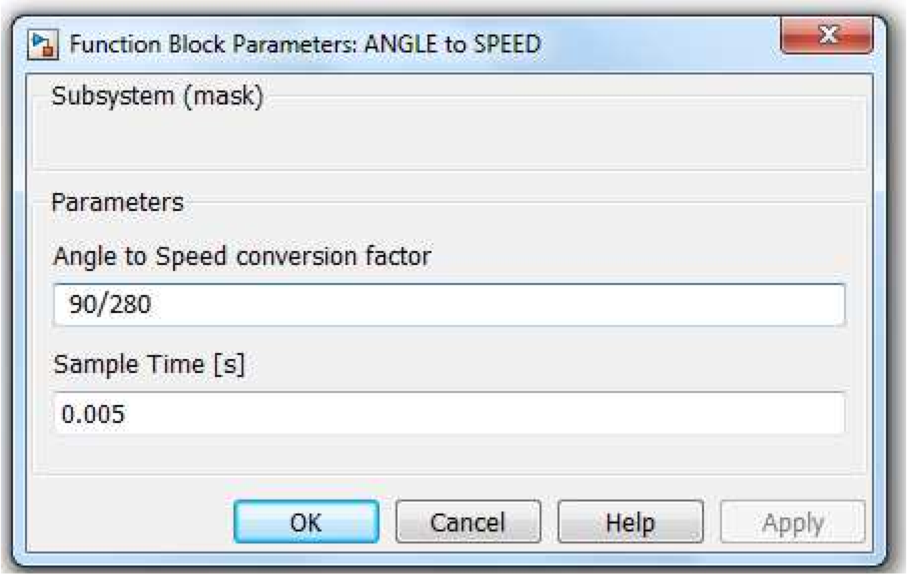


Figure 10 Angle to Speed subsystem


Function Block Parameters: ANGLE to SPEED

Subsystem (mask)

Parameters

Angle to Speed conversion factor

90/280

Sample Time [s]

0.005

Figure 11 Angle to Speed subsystem mask

The *Angle sensor subsystem*, shown in Figure 12 below, performs a double conversion: the first conversion is from voltage to angle while the second one is from angle to speed.

The angle signal is sent to the *Scope* and the speed signal goes to the *subtraction block*.

The conversions are performed via two *gain blocks*, one for each conversion respectively.

A mask reporting the values of the *gain blocks* and the sampling time was created for this subsystem. This way the management of the subsystem is far much easier since all the parameters can be changed via just one mask rather than using the single masks of each component of the subsystem. The mask is shown in Figure 13.

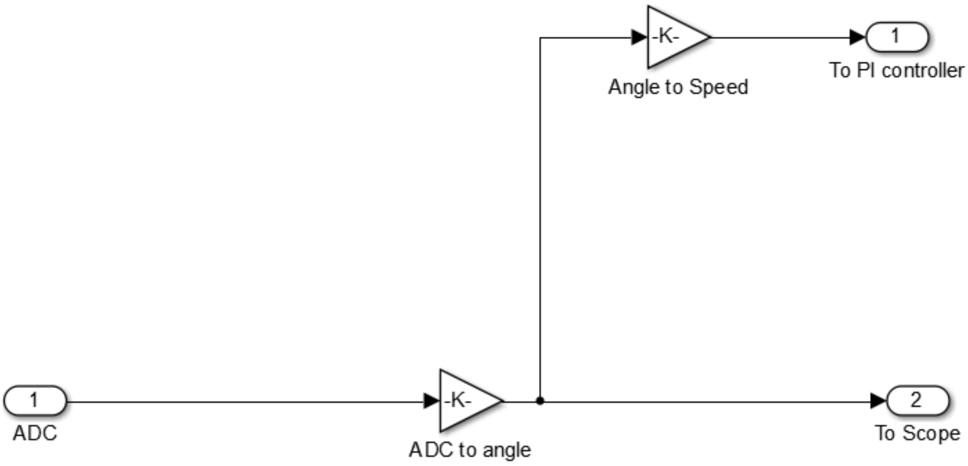
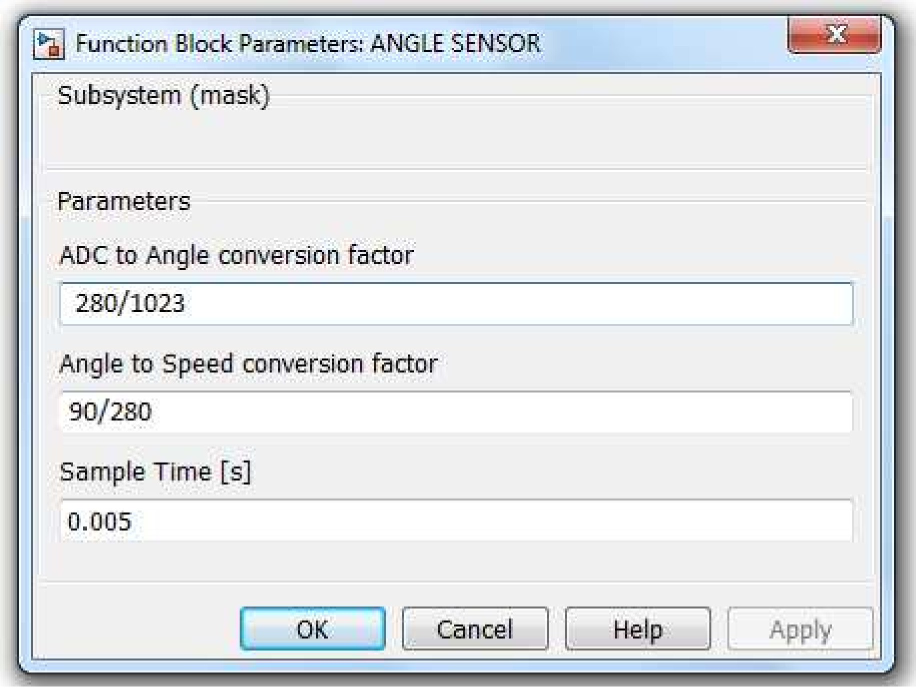


Figure 12 Angle sensor subsystem


Function Block Parameters: ANGLE SENSOR

Subsystem (mask)

Parameters

ADC to Angle conversion factor

280/1023

Angle to Speed conversion factor

90/280

Sample Time [s]

0.005

Figure 13 Angle sensor subsystem mask

The used potentiometer can rotate up to 280° so when connected to the 5V power source provided by the ARDUINO board, it is possible to create a direct linear relation between angle and voltage. For example, 280° corresponds to 5V while when the shaft is set to 0° there is no voltage read.

For a description of how a potentiometer works please refers to section 5.2

So the conversion factor is equal to:

$$a = \frac{280}{5} [=] \frac{deg}{V} \quad (8)$$

While the conversion formula is the following:

$$\delta = a * V \quad (9)$$

where:

a : conversion factor from voltage to degree, expressed in °/V;

δ : angle position value of the potentiometer, expressed in °;

V : voltage output value of the potentiometer, expressed in V.

Once this conversion is settled it is very easy to read the output of the potentiometer and convert this value in the angle position of the potentiometer shaft.

In this work the motor controlled variable is the angle position of the motor shaft, but the motor input signal is in speed. So there was the need to create a conversion between the angular speed and the angle position. This is necessary because as reminded in section 3.0 “CONTROL THEORY” the required control variable and the controlled output variable must be expressed in the same physical characteristic.

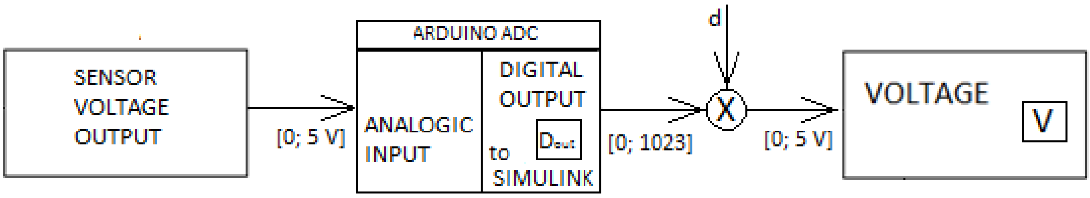


Figure 14 Arduino ADC scheme and voltage to voltage conversion

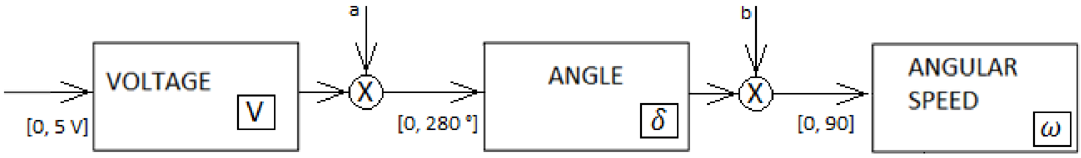


Figure 15 Voltage to angle and angle to speed conversion scheme

The conversion value between angle and angular speed is:

$$b = \frac{90}{280} \quad [=] \frac{1}{deg} \tag{10}$$

$$\omega = b * \delta \tag{11}$$

where:

b : conversion factor from angle to angular speed (dimensionless), expressed in 1/°;

δ : angle position value of the potentiometer, expressed in °;

ω : angular speed, dimensionless.

The value "90" is dimensionless since according to the *continuous servo write* block of SIMULINK the input is dimensionless.

The ARDUINO board analogic input channels accept an analogic signal only in the range [0; 5 V] and give as output to the Simulink project a value in the range [0;1023].

Therefore in order to restore the sensor scale a conversion must be set dividing the maximum values of both the ranges as shown below:

$$d = \frac{5}{1023} [=] \frac{V}{1} \tag{12}$$

$$V = c * D_{out} \tag{13}$$

where:

d : conversion factor from ADC output to degree, expressed in V/1;

D_{out} : output of Arduino ADC , dimensionless ;

V : voltage output value of the potentiometer, expressed in V.

4.2.2 Rectangular input

An ideal input signal was created using several *step blocks* and a *summation block* as shown in Figure 16 below.

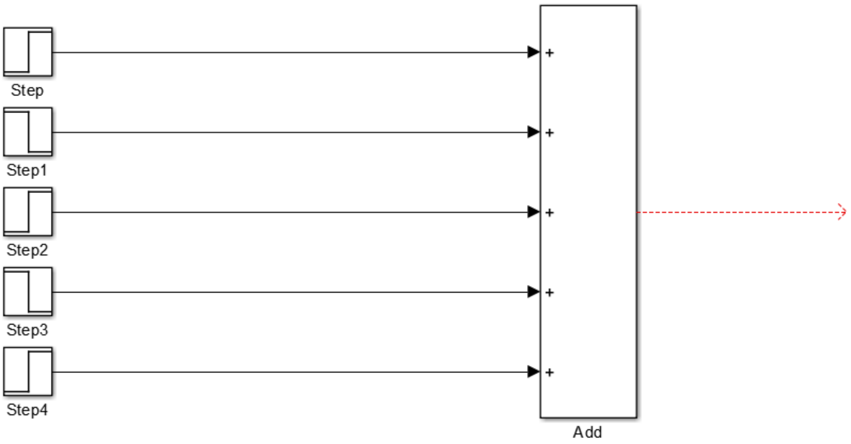
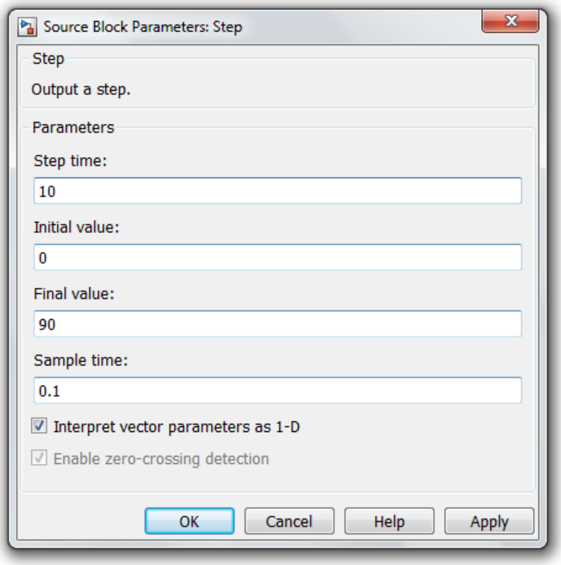


Figure 16 Step blocks and summation blocks



Source Block Parameters: Step

Step

Output a step.

Parameters

Step time:

10

Initial value:

0

Final value:

90

Sample time:

0.1

☑ Interpret vector parameters as 1-D

☑ Enable zero-crossing detection

Figure 17 Step block mask

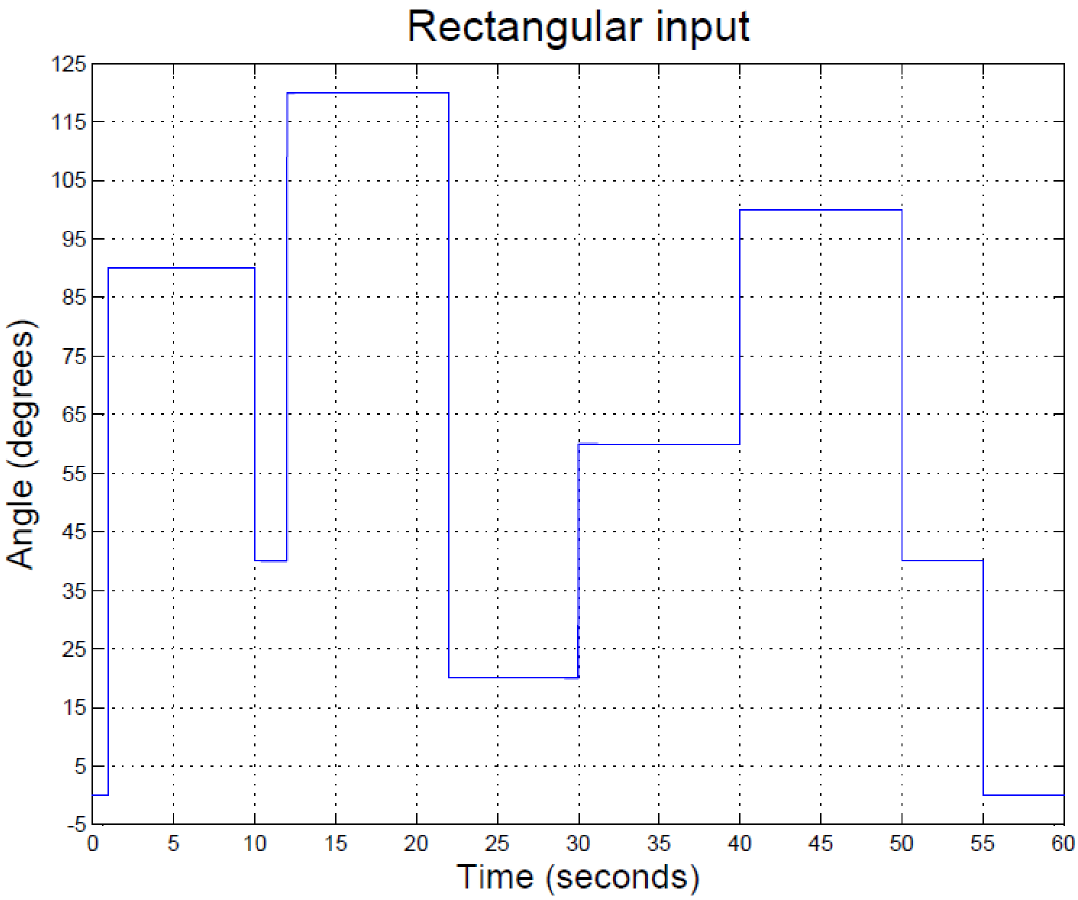


Figure 18 Rectangular input

Each Step block is characterised by 4 parameters: step time, initial value, final value and sample time, as shown in Figure 17.

A step signal is represented by the following equation:

$$step = \begin{cases} \alpha & t < \tau \\ \beta & t \geq \tau \end{cases} \tag{14}$$

where:

τ is the step time

α is the initial value

β is the final value

So by adding several step blocks together it is possible to create a signal with different amplitude and lasting for any required period of time.

This signal is not very suitable for a direct use on a robotic system because of the several discontinuities involved in it. Anyway this signal can be used to make an estimation on the reactivity of the controller.

An ideal signal that can be used in practice on a robotic system is discussed in section 4.2.4.

4.2.3 Triangular input

In order to eliminate the discontinuities the signal in section 4.2.3 has, it is possible to use a *ramp block* rather than a *step block*.

A ramp block is characterised by 3 parameters: slope, start time and initial output. An example of a ramp mask is shown in Figure 19.

The mathematic expression of a ramp signal is the following one:

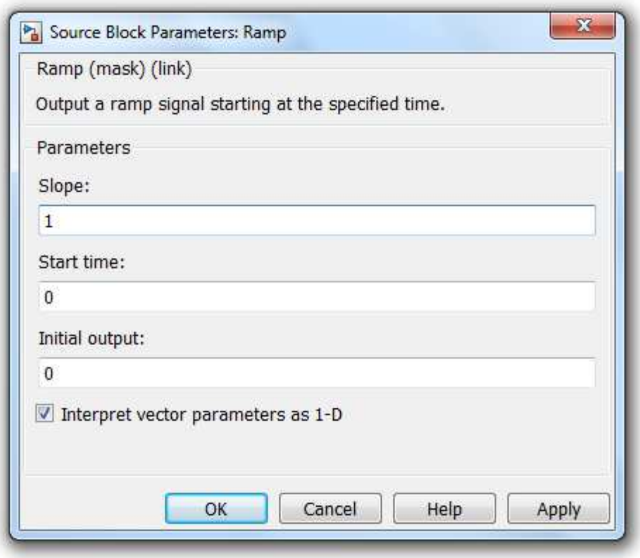
$$ramp = \begin{cases} a * t + a_0 & t > \tau \\ 0 & elsewhere \end{cases} \tag{15}$$

where:

a is the slope

a_0 is the initial output

τ is the start time



Source Block Parameters: Ramp

Ramp (mask) (link)

Output a ramp signal starting at the specified time.

Parameters

Slope:

1

Start time:

0

Initial output:

0

☑ Interpret vector parameters as 1-D

Figure 19 Ramp block mask

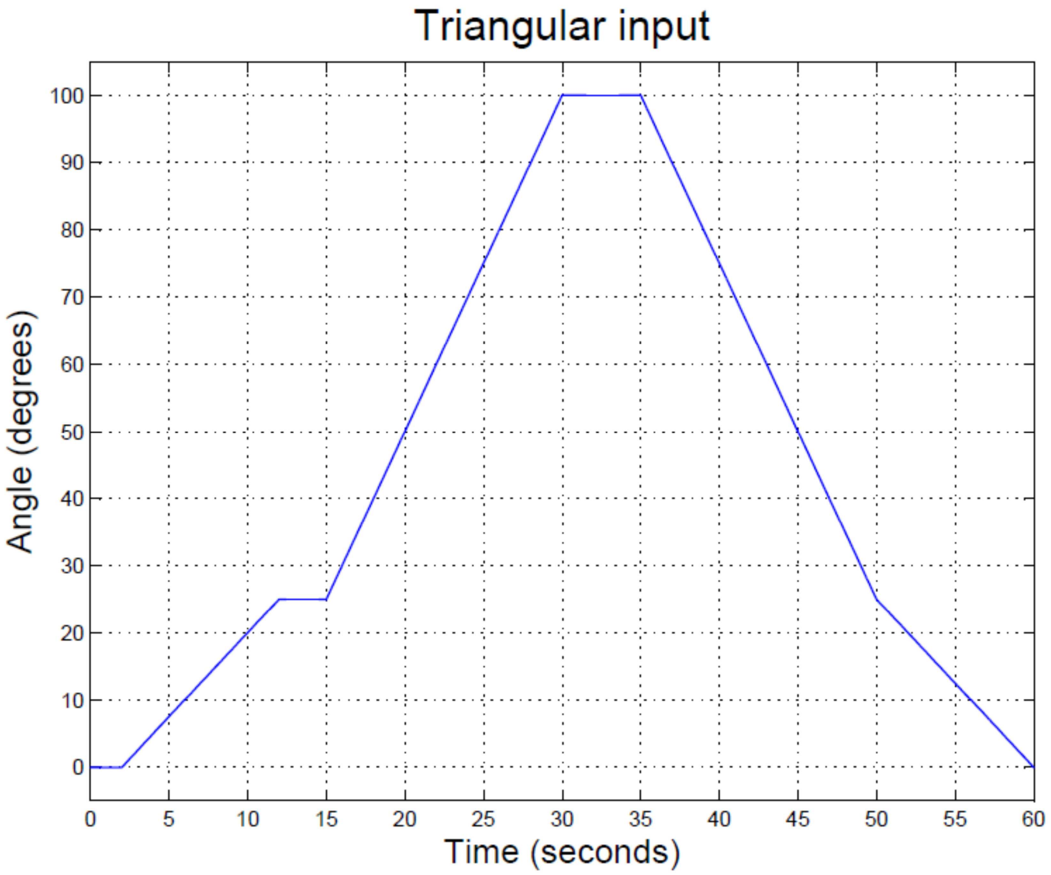


Figure 20 Triangular input

A signal obtained joining together several ramp blocks, represented in Figure 20, showed to have still some discontinuities even if they could be accepted by the system. In fact such discontinuities did not affect too much either the motor or the whole system.

Hence, this kind of signal can be used to check how the system reacts in a robot in charge task.

4.2.4 Electrooptical input signal

In order to test the system in a human in charge task there was the need for a signal connected to the muscular activity. For this purpose a signal sampled using an electrooptical muscle contraction sensor was used.

The signal was recorded by Dr. Mario Giardini who kindly provided me with it to allow me to test the system in a real operating condition. The signal had been published previously [49], and therefore can be used without further formal ethical approval.

The signal represents an isotonic contraction of the biceps. An isotonic contraction implies the tension of the muscle does not change while its length is varying causing a movement of a body part.

Some tiny modifications were necessary, mainly because of some ripples, to the real signals and they are discussed below.

Due to elaboration necessity the manipulation of the signal was divided in several parts as shown in Table 2. Each part is divided in several steps according to the needs. For each step is reported the note M for Matlab or S for Simulink according to the program used for that step.

Actually, it must be pointed out that this manipulation was possible because the used signal was recorded.

In the case of a real time sampling with a direct usage of the signal on the motor other manipulations will be necessary, but they are not part of this project.

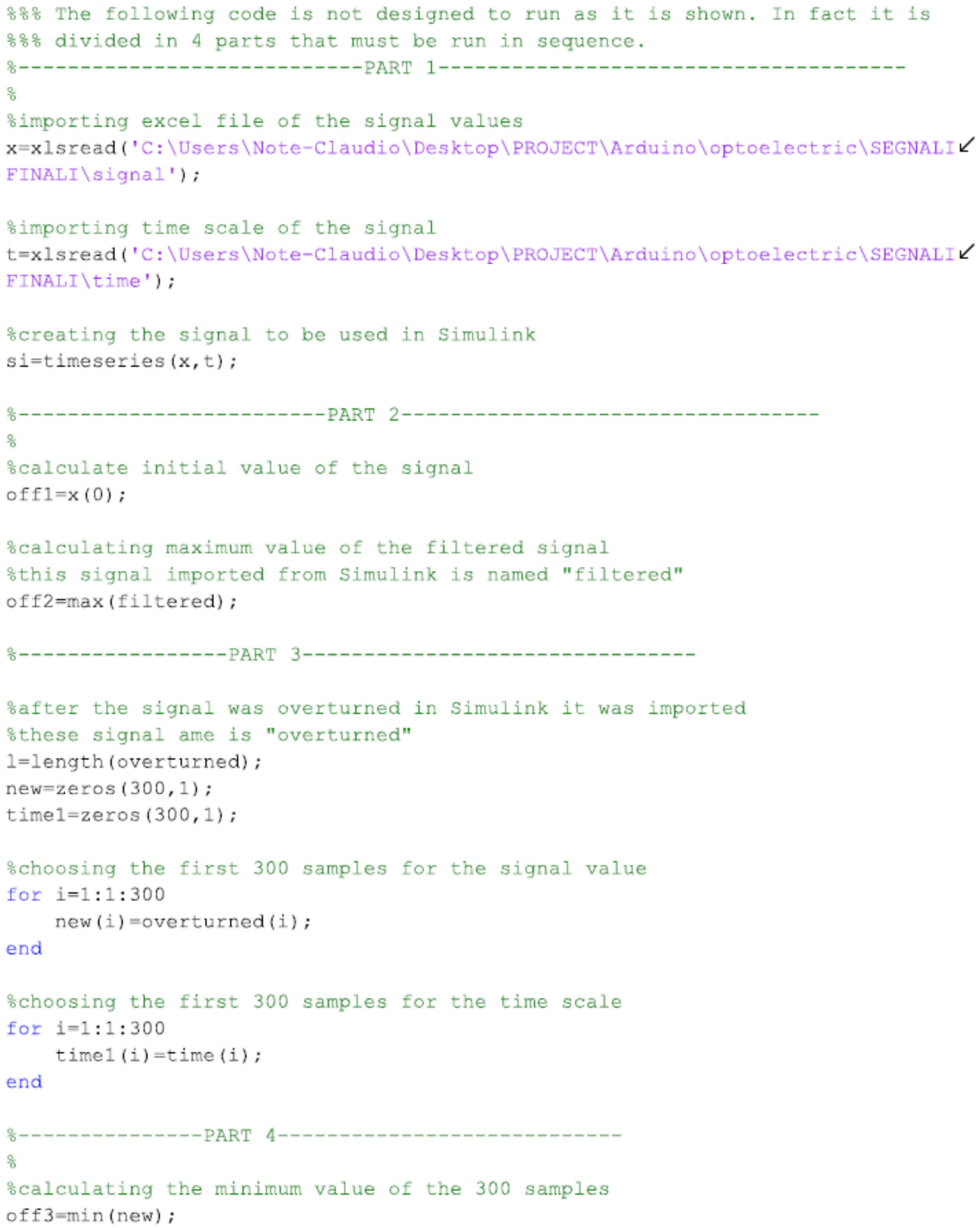
When using a signal related to the muscle activity of a patient the exoskeleton must use the signal as a trigger rather than follow the waveform of the signal. This implies that the correlation between the signal value and the exoskeleton controlled variable must be known. With regard to this work since the controlled variable is the angular position this means that a relation between the amplitude of the electrooptical signal and the patient limb angular position must be set. Once this relation is known the exoskeleton will react to the muscle signal as the real limb does if both the relation and the exoskeleton control are well established.

Table 2 Electrooptical signal modification process

Part 1	Step 1	M	The excel files with the electrooptical signal and the time scale were both imported in Matlab, combined together as a *timeseries* signal named “si”.
	Step 2	M	Calculate “off1” that is the initial value of the “si” signal.
	Step 3	S	Import the “si” signal from Matlab to Simulink.
Part 2	Step1	S	An offset value, named “off1”, was applied to the “si” signal to bring its initial value to zero. This step was necessary to avoid problem in the origin after the filtering. The original signal is shown in Figure 22.
	Step 2	S	The “si” signal was low-pass filtered to eliminate the ripple. a small delay was introduced by the filter. This new signal named “filtered” was exported to Matlab.
	Step 3	M	Calculate the maximum value of the “filtered” signal. This value was named “off2”.
Part 3	Step 1	S	The offset value named “off2” was applied to the “filtered” signal so its maximum value was equal to zero.
	Step 2	S	The “filtered” signal was overturned along the x-axis and exported to Matlab. This new signal was named “overturned”.
	Step 3	M	The “overturned” signal obtained was too wide to be handled by the Arduino Board . So just one peak was selected to create a new signal named “new”.
	Step 4	M	Calculate the maximum value of the “new” signal. This value was named “off3”.
Part 4	Step 1	S	The offset value named “off3” was applied to bring the minimum value of the signal to zero. The final modified signal is shown in Figure 23.
	Step 2	S	A gain block was used to convert the electrooptical signal in an angular signal. This signal is shown in Figure 24.

Hence without a real assessed relation between the electrooptical signal and the limb angular flexion\extension there was the need to suppose a relation in order to test the exoskeleton control system in a simulated patient in charge task.

So I supposed a linear proportional relation between the electrooptical signal and the limb angular flexion. In this relationship the maximum value of the muscular signal corresponds to a flexion angle of about 120°. This explains the gain block used in Step 2 of Part 4 of Table 2.



```
%%% The following code is not designed to run as it is shown. In fact it is
%%% divided in 4 parts that must be run in sequence.
%----------------------------PART 1--------------------------------------
%
%importing excel file of the signal values
x=xlsread('C:\Users\Note-Claudio\Desktop\PROJECT\Arduino\optoelectric\SEGNALI↙
FINALI\signal');

%importing time scale of the signal
t=xlsread('C:\Users\Note-Claudio\Desktop\PROJECT\Arduino\optoelectric\SEGNALI↙
FINALI\time');

%creating the signal to be used in Simulink
si=timeseries(x,t);

%-------------------------PART 2---------------------------------
%
%calculate initial value of the signal
off1=x(0);

%calculating maximum value of the filtered signal
%this signal imported from Simulink is named "filtered"
off2=max(filtered);

%-----------------PART 3-------------------------------

%after the signal was overturned in Simulink it was imported
%these signal ame is "overturned"
l=length(overturned);
new=zeros(300,1);
time1=zeros(300,1);

%choosing the first 300 samples for the signal value
for i=1:1:300
    new(i)=overturned(i);
end

%choosing the first 300 samples for the time scale
for i=1:1:300
    time1(i)=time(i);
end

%---------------PART 4----------------------------
%
%calculating the minimum value of the 300 samples
off3=min(new);
```

Figure 21 Matlab code for the electrooptical signal modification

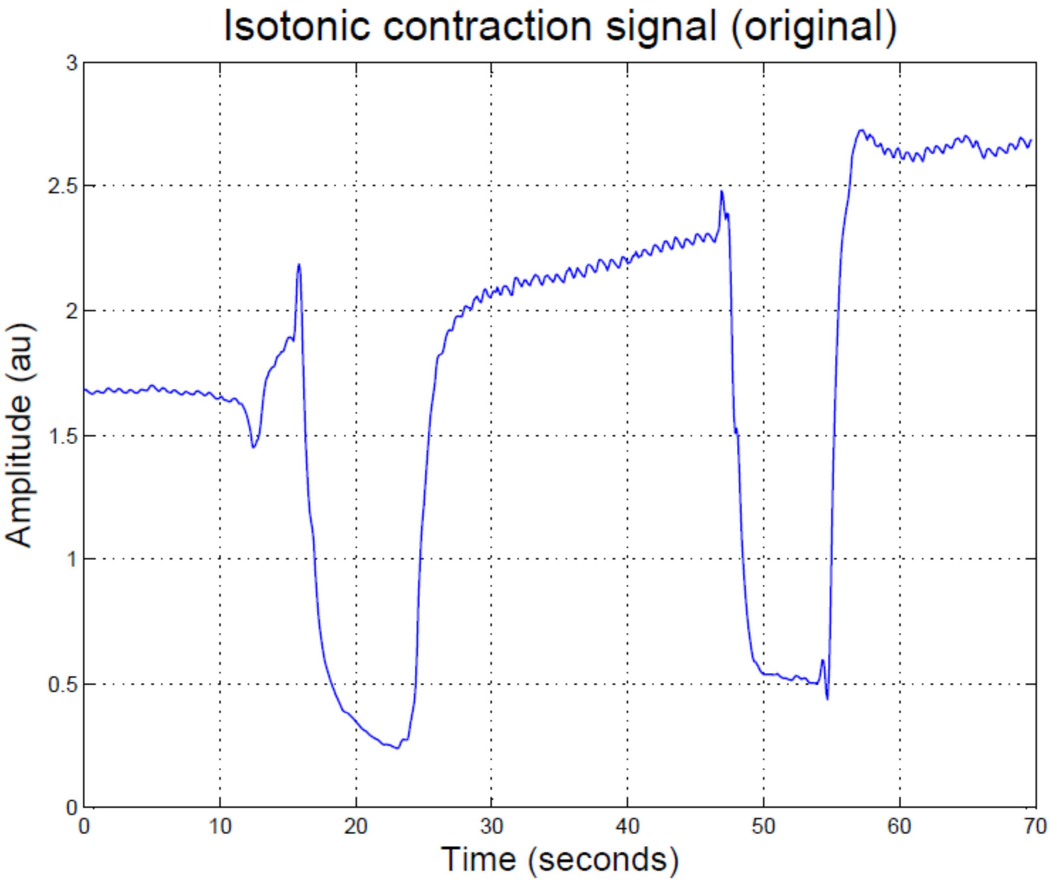


Figure 22 Isotonic contraction signal (original)

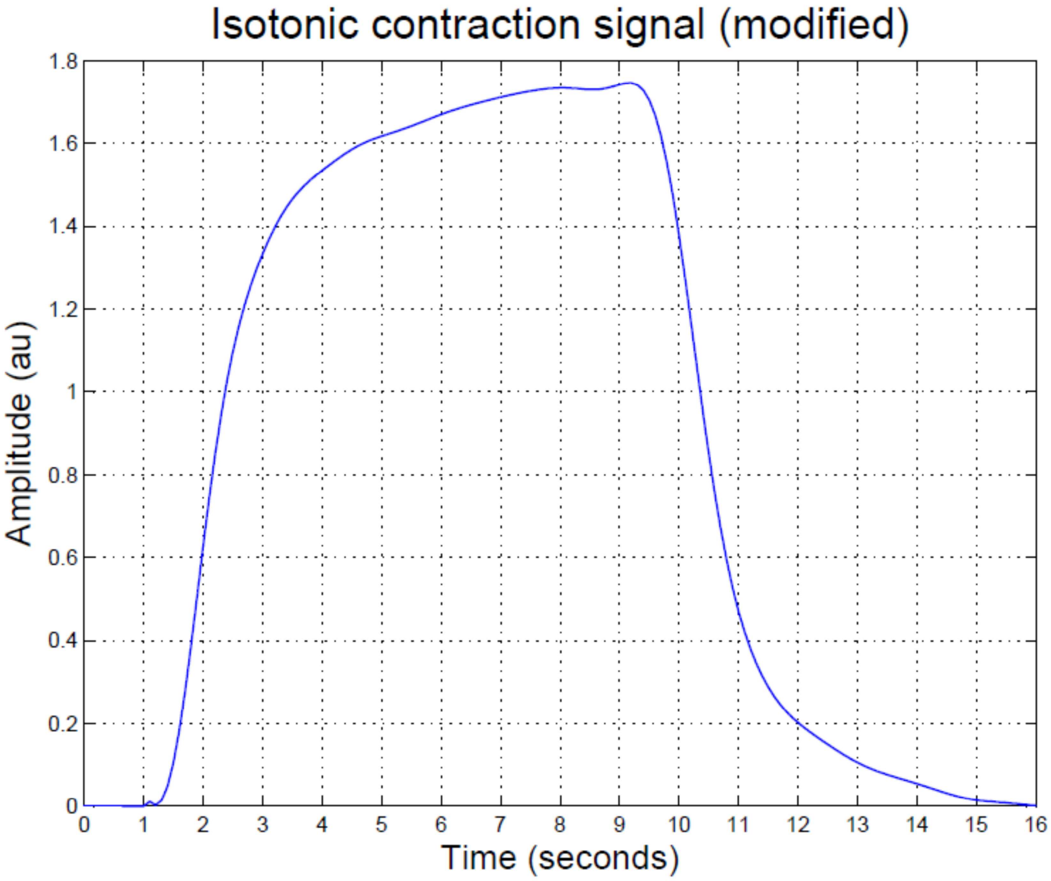


Figure 23 Isotonic contraction signal (modified)

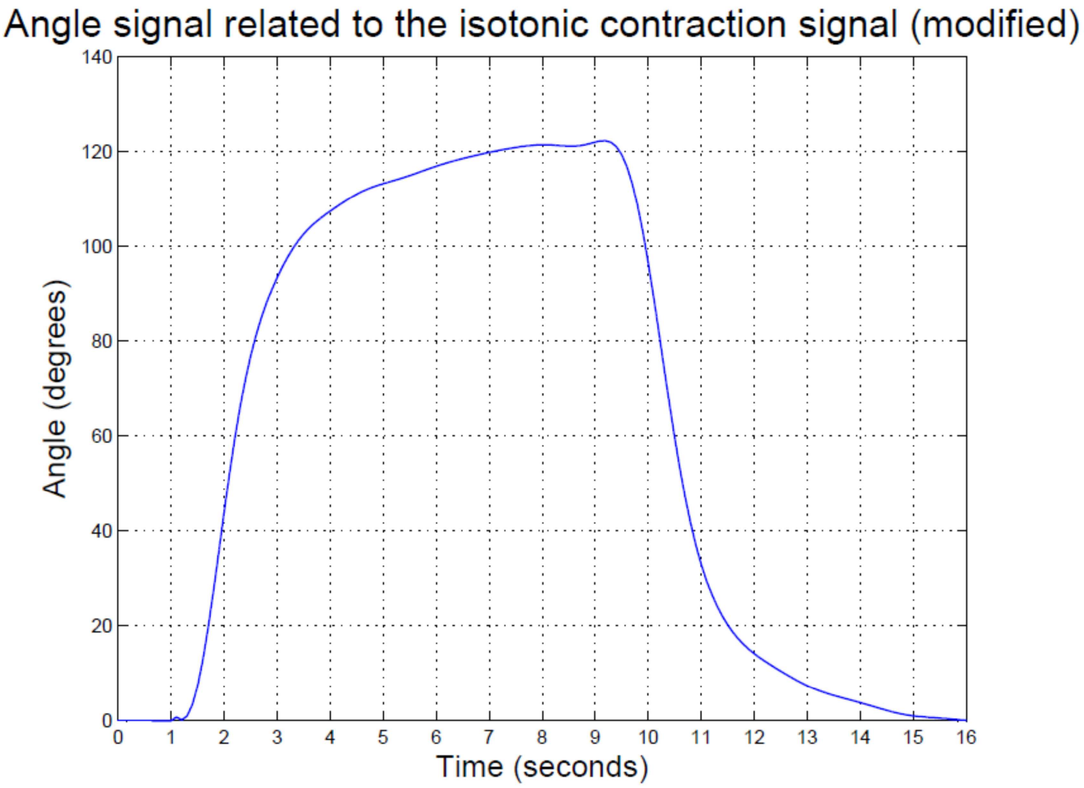


Figure 24 Angle signal related to the isotonic contraction signal (modified)

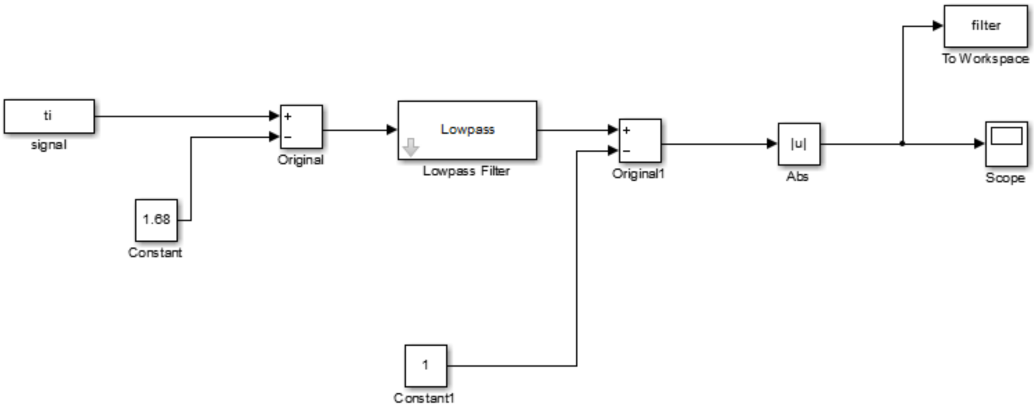


Figure 25 Simulink project for the electrooptical signal modification

The Simulink project shown in Figure 25 and the Matlab code did not work in real time, but they were divided into four parts as explained in Table 2.

The final signal was ready to be used in the simulation on the actuator.

4.3. Nyquist and sampling time

The voltage supply of the potentiometer is in Direct current. This means the frequency of the output signal of the potentiometer is ideally of 0 Hz.

The sample frequency of the Arduino board is 200 Hz so the Nyquist-Shannon theorem is respected and no aliasing should happen.

The Nyquist-Shannon theorem states that in order to avoid aliasing the sampling frequency must be at least the double of the highest frequency component of the sampled signal.

5 Project making

The real project was composed of the following components:

- A standard servo motor (Tower Pro GS-5010);
- A 10 kΩ potentiometer (used as a sensor);
- 2 gears of 2,5 cm radius;
- An ARDUINO MEGA 2560 board;
- A USB cable to connect ARDUINO to the laptop;
- A breadboard;
- Several wires;

5.1 Potentiometer as sensor

A potentiometer is a passive electrical component able to show a variable resistance according to its rotating shaft position.

Potentiometers are composed of a sliding contact that moves along a resistive layer. Both the resistive layer and the sliding contact are placed in a metallic housing. The resistive layer is usually shaped in two separated arches , as shown in Figure 26, the external one is connected to the ground and to a power source through two pins while the internal one has one pin representing the output. These two resistive layers are connected via a sliding contact, shown in Figure 27, and according to its position there is a different value of the potentiometer output [41].

The electric scheme of the potentiometer, reported in Figure 28, shows how it works using the basic principle of the voltage divider.

Figure 26 Resistive layers

Figure 27 Sliding contact

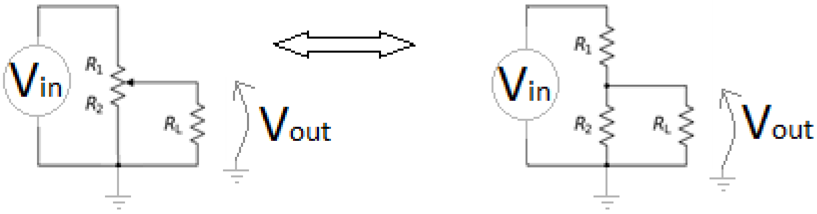


Figure 28 Potentiometer electric diagram

$$V_{out} = \frac{R_2 R_L}{R_1 R_L + R_2 R_L + R_1 R_2} V_{in} \tag{16}$$

If $R_L >> R_1, R_2$ (16) becomes

$$V_{out} = \frac{R_2}{R_1 R_2} V_{in} \tag{17}$$

Therefore once a conversion between voltage and angle is settled it is very easy to pass from voltage to angle. All the conversions used in the project are explained in section 4.3.1.

5.2 Servo motor technical sheet [41]

The motor used is a standard servo motor, a Tower Pro SG-5010, that was modified for the fulfilment of this work, see section 5.4 for the modification.

A standard servo motor is composed by the following components:

- a DC motor;
- a potentiometer used as a sensor
- the chip board used to control the motor
- several gears.

It has three wires connection: two are used for the power supply, voltage input (red wire) and ground (brown) respectively, while the last one (orange) is the PWM signal used to control the motor shaft speed.

The chip board, shown in Figure 33, performs two different tasks: the first one is to decode the control signal (orange wire), converting it to a voltage signal given to the motor. The second one is to adjust time by time the position of the motor shaft according to the difference between the PWM signal and the feedback position.

The potentiometer welded on the board does not provide any feedback to the user, it is part of an inner closed loop of the servo motor not accessible from the outside. As a result the user has no real control on the motor because he has no chance to change neither the transient response parameter nor the steady state parameters (overshoot, rise time, etc..).

Furthermore, it is not possible to estimate the accuracy or the reliability of the system [41].

To overcome these difficulties the servo motor was modified as explained in section 5.4. Table 3 reports the technical features of the motor as reported from the manufacturer website[50] :

Table 3 Technical features of the servo motor Tower Pro GS-2010

Modulation:	Analog
Torque:	8,00 kg-cm (4,8 V) 11,00 kg-cm (6 V)
Speed:	0,17 sec/60° (4,8 V) 0,14 sec/60° (6,0 V)
Weight:	38,0 g
Dimensions:	Length: 40,1 mm Width:20,3 mm Height:43,2 mm
Motor Type:	3-pole
Gear Type:	Plastic
Rotation/Support:	Dual Bearings
Additional Specifications	
Rotational Range:	180°
Pulse Cycle:	20 ms
Pulse Width:	600-2400 µs

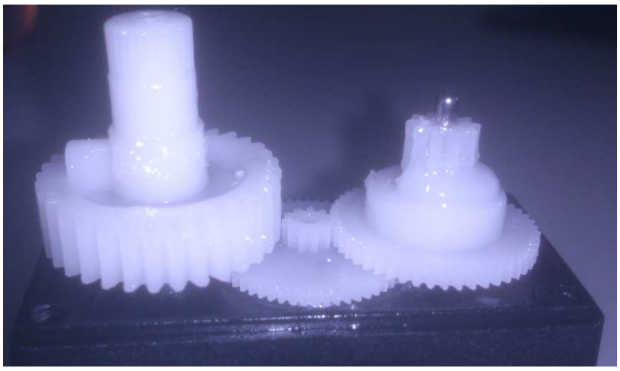


Figure 29 Gears of a servo motor

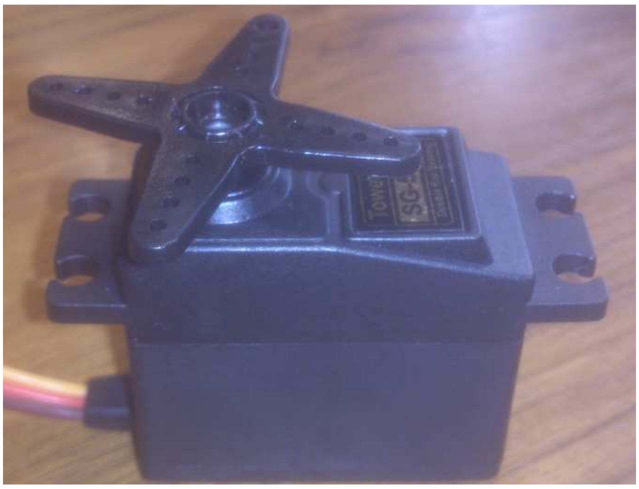


Figure 30 –Servo motor Tower Pro GS-5010

5.3 Motor modification

As explained in section 5.3, each servo motor has its own feedback circuit, but in this work this feedback is not well-accepted because the user wants the complete control of the motor. Therefor the inner feedback of the motor was removed by replacing the potentiometer with two resistors. One end of each resistor was soldered to the other one so the final structure had 2 resistors and 3 pins.

It is of fundamental importance to have a 3 pins structure because this one must replace the potentiometer, that has 3 pins as explained in section 5.2.

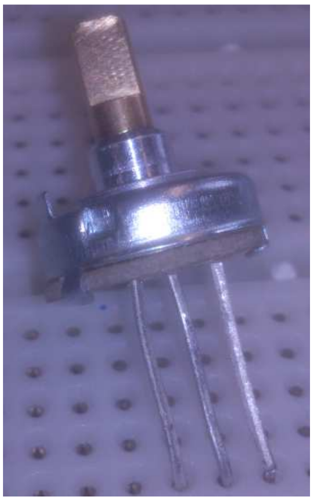


Figure 31 Removed potentiometer

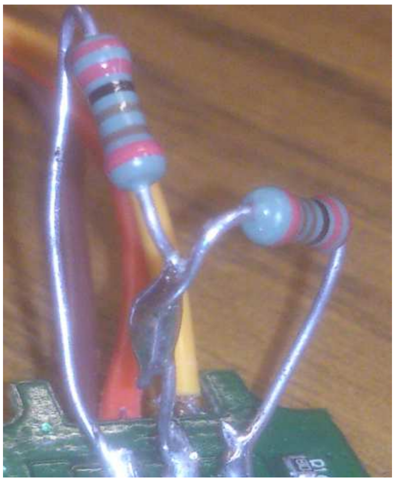


Figure 32 Replacing resistors structure

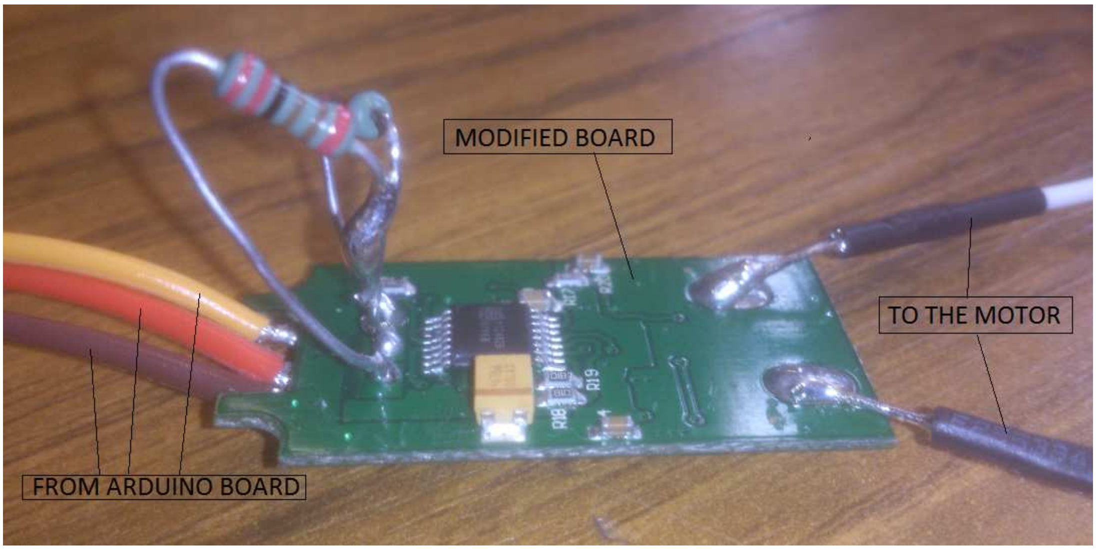


Figure 33 Modified servo motor board

In fact, this way the control chip board reads always the same position value of the motor and it is not able to perform any change on the motor neither in speed nor in position.

On the other hand, the user can change both position and speed of the motor shaft through the feedback provided by the sensor placed next to the motor.

5.4 Sensor and motor connection

Two gears having the same radius were used to connect the shaft of the servo motor to the shaft of the sensor. The equal radius of the gears ensures a 1:1 ratio in angle rotation between the components. This way the sensor shaft and the motor shaft must rotate of the same amount and the introduction of errors is avoided.

The shafts of the sensor and the motor were aligned as much as possible in order to obtain the best coupling possible and at the same time to avoid the mismatching of the gear tooth during the rotation.

The green gear, mounted on the sensor, was modified by enlarging the central hole where the sensor shaft was inserted. This modification was made using progressive larger drill bits until the suitable dimension was obtained.

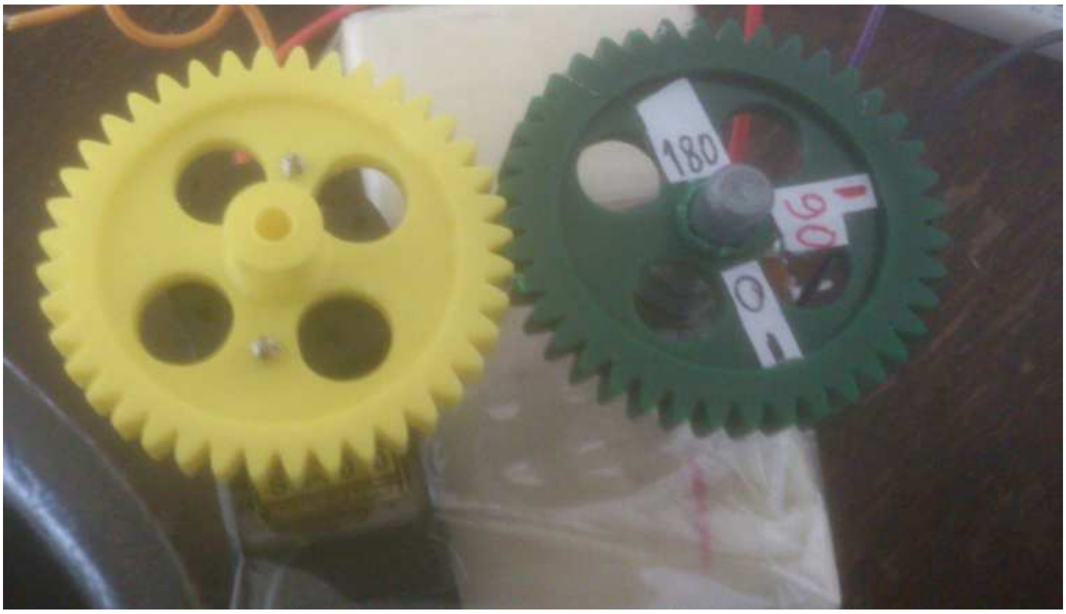


Figure 34 Gear system connection

On the other hand, the yellow gear , mounted on the motor, was screwed on the rounded horn provided with the motor. The central holes of both the horn and the gear were inserted on the same metallic shaft so they were screwed together in the best alignment possible.

5.5 Connection to the laptop

Three wires of different colours were welded on the potentiometer: black, violet and red representing respectively the ground, the output tension and the input tension.

These wires were connected to ARDUINO through a breadboard in the following configuration:

Table 4 Wires connection between Arduino board and Laptop

POTENTIOMETER	*BREADBOARD*	*ARDUINO*
RED	RED	5 V power supply
VIOLET	YELLOW	ANALOG INPUT CHANNEL PIN A04
BLACK	BLACK	GROUND

The motor was connected to the removed chip using two wires of different colours, blue and white respectively. The chip itself was connected to the breadboard rather than directly on the Arduino board.

This configuration was to be used because both the motor and sensor required the power supply provided by the Arduino board than has unfortunately only one 5 V power supply pin. So the wires for the power supply and the wires for the ground of both the potentiometer and the motor chip board were connected to the same Arduino pins.

The Arduino board is finally connected to the laptop via USB cable.

Table 5 Wires connection between the actuator and Arduino board

MOTOR CHIP BOARD	*BREADBOARD*	*ARDUINO*
GREY	BLACK	5 V power supply
BLUE	BLUE	DIGITAL OUTPUT CHANNEL PIN D51
VIOLET	RED	GROUND

The whole system is shown in Figure 35 below.

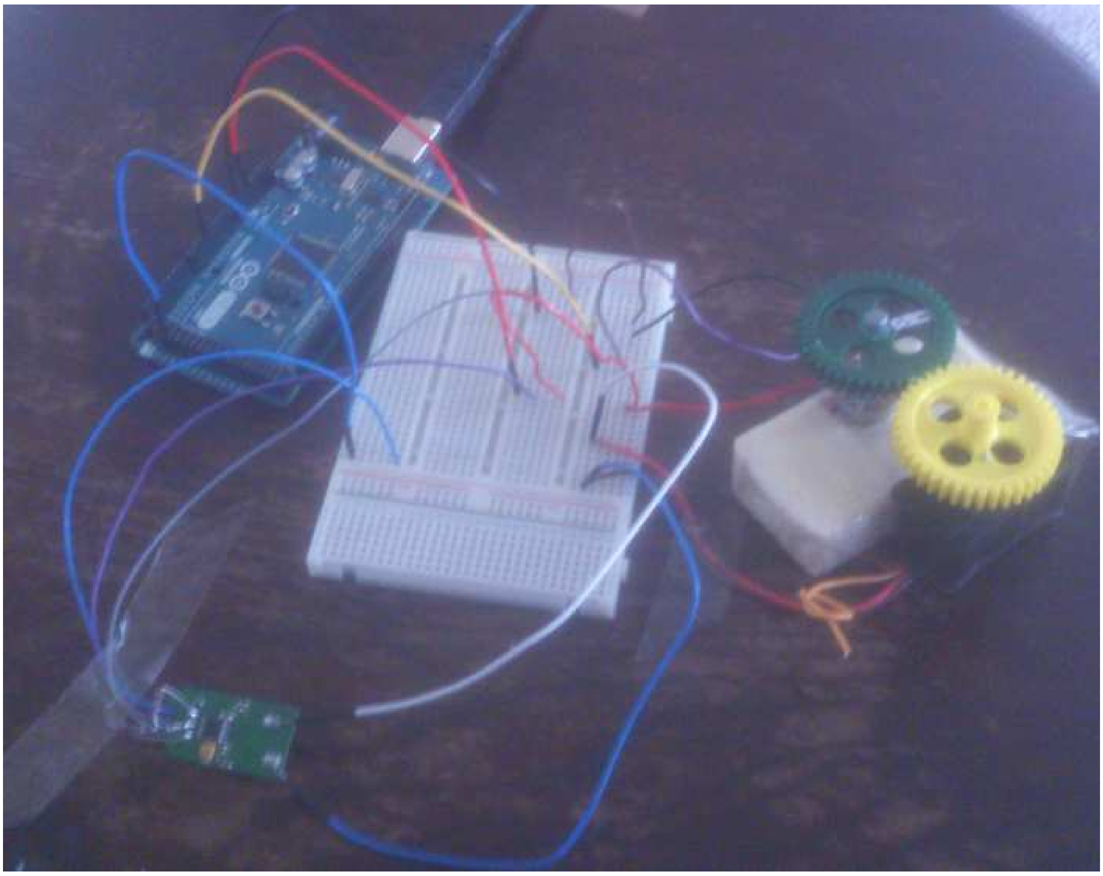


Figure 35 Image of the whole system

6 Empirical tuning of PID controller

6.1 Trial and error method

In this method the gains of a PID controller are chosen following different trials until the desired waveform characteristics are obtained.

Before explaining the method a table showing the effects the three gains of a PID controller have on the waveform features is shown below [51].

Table 6 Effects of the PID controller gains on the waveform [51]

Manual Tuning of a PID controller					
Gains	**Rise Time**	**Overshoot**	**Settling Time**	**Steady-state error**	**Stability**
k_p	Decrease	Increase	Small change	Decrease	Degrade
k_i	Decrease	Increase	Increase	Eliminate	Degrade
k_d	Minor change	Decrease	Decrease	No effect	Improve (small k_d only)

This method consists of the following steps:

1. Set both k_i and k_d equal to zero;
2. Choose a value of k_p, trial by trial, leading to acceptable overshoot, rise time and steady state error. Particularly attention must be paid to the stability of the system, since for a critic value of k_p, named k_{pc}, the system starts to oscillate. Usually k_p should be maximum the half of k_{pc}.
3. Once k_p is set it is possible to insert a derivative gain if needed. This term can reduce both the overshoot and the settling time and increases the system stability if its value is small. On the other hand k_d increases the effect of the noise so particularly attention must be paid in choosing this value when working with noisy

systems. In this work because of the noisy of the potentiometer the derivative gain was set to zero.

4. Once both k_p and k_d are set if there is the need to eliminate or reduce more the steady-state error k_i can be adjusted trial by trial. Since k_i leads to an increase in the overshoot there is a trade-off between the steady-state error and the overshoot.
5. 10 trials were performed for each value of k_p,k_i,k_d in order to have a statistic characterization and reduce the effect of random interference.

Because the potentiometer sensor showed to be noisy the derivative gain was set to zero in order to avoid the increase of the noise.

6.2 Trial testing report

Table 8 shows the results of all the different trials made to detect the desired PI gains. For each value of these gains 10 trials were made and it was determined both the average value and the standard deviation.

In each trial a step signal with the characteristics shown below was used as an input and the sensor response, that is equal to the motor response, was recorded and analysed.

$$step(t) = \begin{cases} 0^\circ\ t < 1 \\ 90^\circ\ 1 \leq t \leq 15 \end{cases}$$

Sampling time T_s=0,005 s

Lasting time 15 s

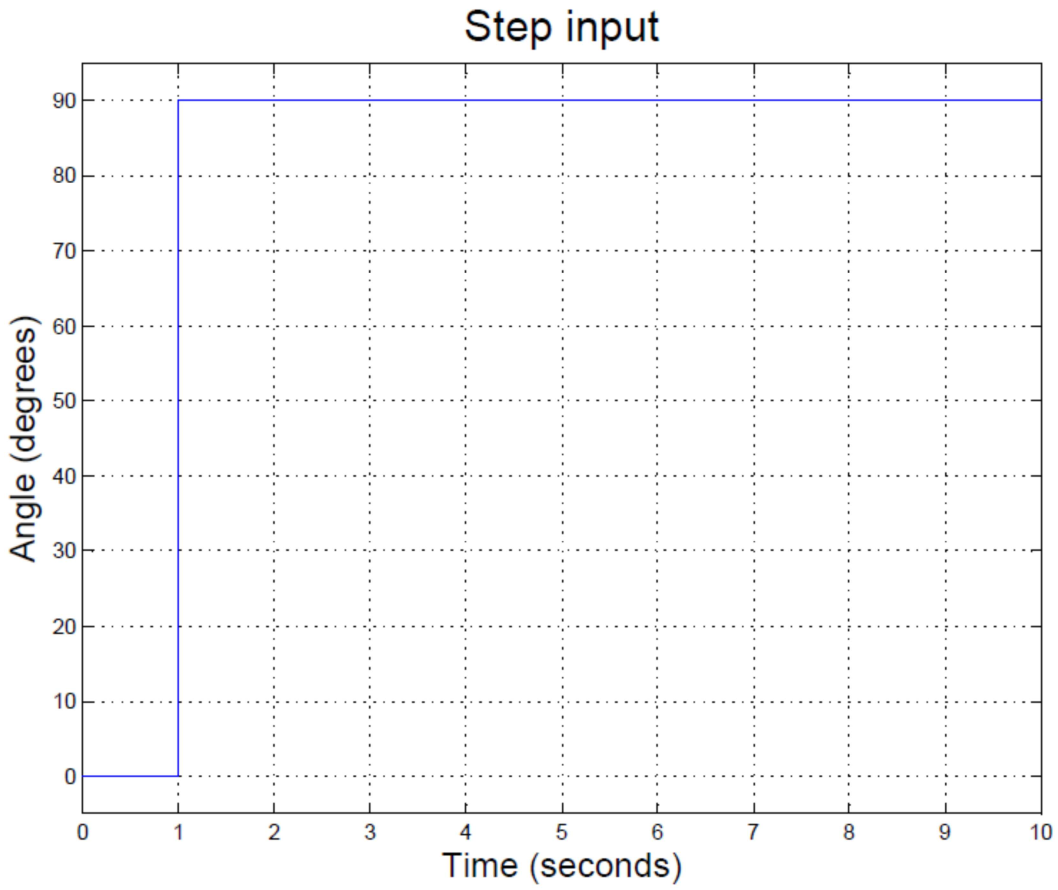


Figure 36 Step input

All the trials were organized according to the PI controller gains and both the average value and the standard deviation were calculated in Excel.

Table 7 shows as example the trials and the calculations for a set of PI controller gains.

From now on, the PI gains are reported as P and I rather than k_p and k_i.

Table 7 Trials and errors calculations for a set of PI controller gains (P=1 I=0,5)

sample	rise time	Overshoot		Settling time	Steady-state error	
	s	°	%	s	°	%
1	0,055	4,100	4,556	2,500	0,030	0,033
2	0,060	1,500	1,667	0,060	0,030	0,033
3	0,060	1,200	1,333	0,060	0,030	0,033
4	0,055	1,200	1,333	0,055	0,030	0,033
5	0,055	0,900	1,000	0,055	0,030	0,033
6	0,080	4,100	4,556	2,150	0,030	0,033
7	0,500	1,500	1,667	0,500	0,030	0,033
8	0,140	1,200	1,333	0,140	0,030	0,033
9	0,800	0,900	1,000	0,800	0,030	0,033
10	0,200	1,200	1,333	0,200	0,030	0,033
Average	0,201	1,78	1,978	0,652	90,03	0,033
Standard Deviation	0,239	1,175	1,306	0,871	1,42E-14	0

Table 8 Trials and errors calculations for all the sets of PI controller gains

PI Gains	P	2	1,5	1	1,2
	I	0,5	0,5	0,5	0,5
Rise time	(s)	0,047±0,004	0,052±0,005	0,201±0,239	0,110±0,043
Overshoot	(°)	3,350±0,694	1,532±0,898	1,780±1,175	1,720±1,057
	(%)	3,722±0,771	1,702±0,998	1,978±1,306	1,911±1,175
Settling time	(s)	0,301±0,082	0,482±0,859	0,652±0,871	0,562±0,880
Steady-state error	(°)	0,057±0,081	0,030±0,000	0,030±0,000	0,300±0,000
	(%)	0,063±0,090	0,033±0,000	0,033±0,000	0,333±0,000
PI Gains	**P**	**2**	**1,5**	**1**	**1,2**
	I	**0,8**	**0,8**	**0,8**	**0,8**
Rise time	(s)	0,052±0,004	0,053±0,002	0,060±0,006	0,110±0,064
Overshoot	(°)	3,490±0,804	3,040±1,328	1,900±1,084	1,360±1,825
	(%)	3,878±0,893	3,378±1,475	2,111±1,205	1,511±2,028
Settling time	(s)	0,436±0,307	0,391±0,466	0,389±0,656	0,409±0,579
Steady-state error	(°)	0,030±0,000	0,030±0,000	0,030±0,000	0,030±0,000
	(%)	0,033±0,000	0,033±0,000	0,033±0,000	0,033±0,000
PI Gains	**P**	**2**	**1,5**	**1**	**1,2**
	I	**0,2**	**0,2**	**0,2**	**0,2**
Rise time	(s)	0,047±0,002	0,051±0,007	0,403±0,295	0,077±0,027
Overshoot	(°)	3,210±1,067	1,780±1,155	1,360±0,871	1,250±1,009
	(%)	3,567±1,186	1,978±1,283	1,511±0,968	1,389±1,121
Settling time	(s)	0,587±0,707	0,306±0,431	1,122±1,286	0,851±1,540
Steady-state error	(°)	0,144±0,228	0,204±0,348	0,144±0,228	0,144±0,228
	(%)	0,160±0,253	0,227±0,387	0,160±0,253	0,160±0,253

All the reported PI gains showed to be good for this work purpose. In fact the highest *overshoot* is equal to 3,378±1,475 % (P=1,5 I=0,8) meaning that in the worst case this value was less than 5% anyway.

It must be considered *overshoot* is just one of the parameter to take into account but other important parameters are the *rise time* and the *steady-state error*. The other measured parameter is the *settling time,* that in most of this work cases, is more or less equal to the *rise time*.

Therefor when the gains of a PI controller are chosen there is a trade-off between all the aforementioned parameters.

For example, when the PI gains of the controller are equal to 1,2 and 0,2, respectively the proportional one and the integral one, the system has a rise *time* of 0,077±0,027 s, an *overshoot* of 1,389±1,121 % and a *steady-state error* of 0,160±0,253 %. Furthermore, even the *settling time* seems to be acceptable as it is less than 1 s.

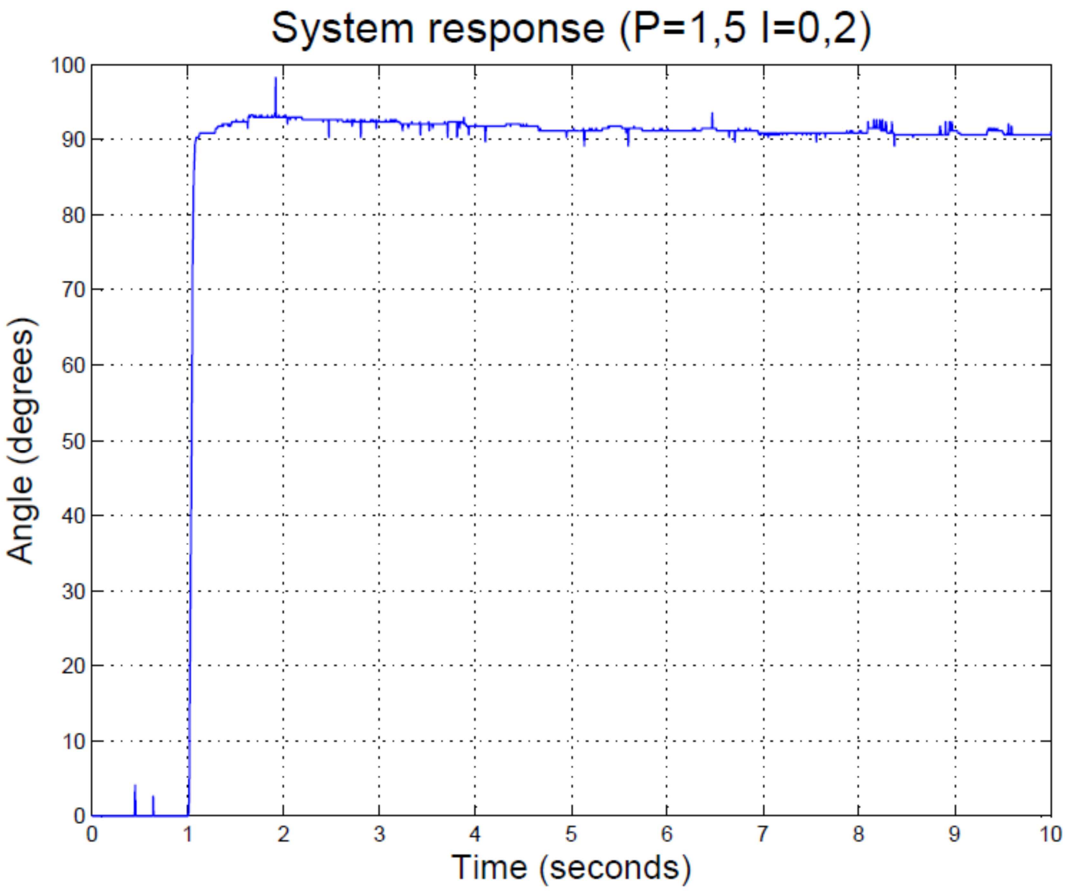


Figure 37 System response to a step input

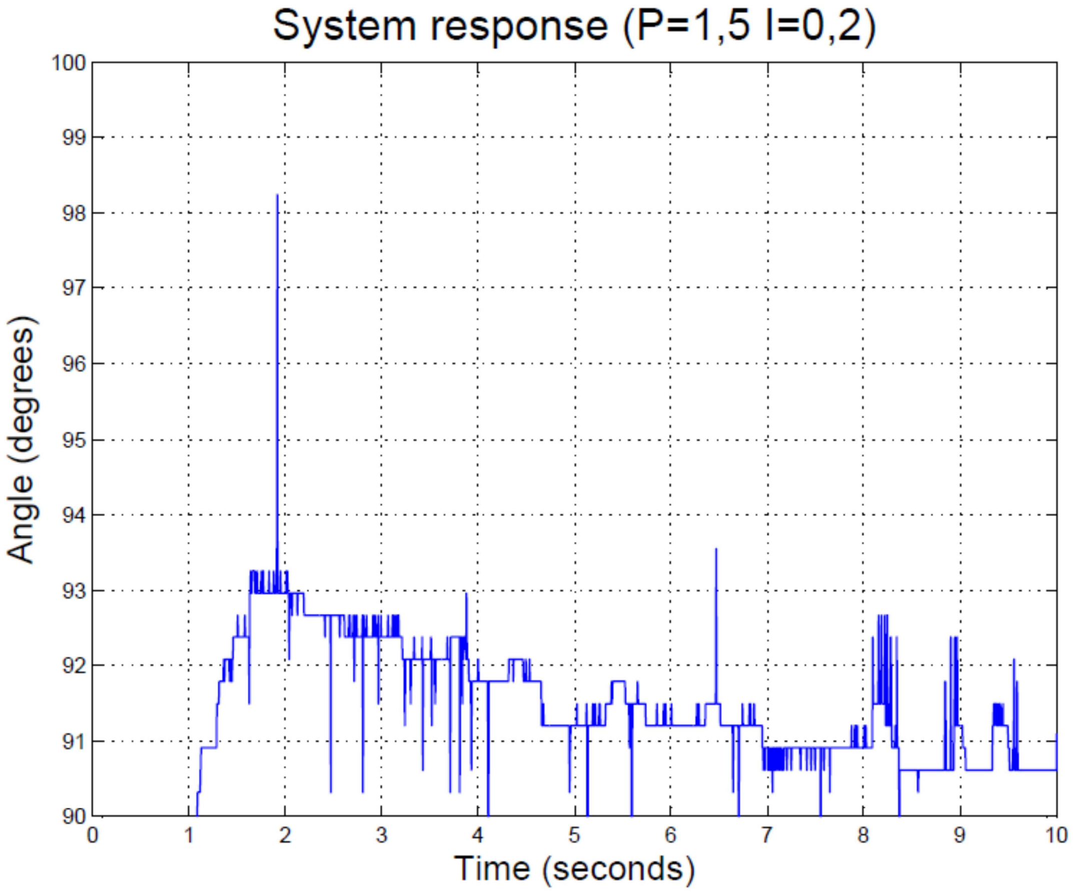


Figure 38 System response to a step input (detail)

Furthermore the system shows to become unstable for values of the proportional gain equal or greater than 6, as shown in Figure 39 below. This is the reason why the maximum proportional gain was set to 2.

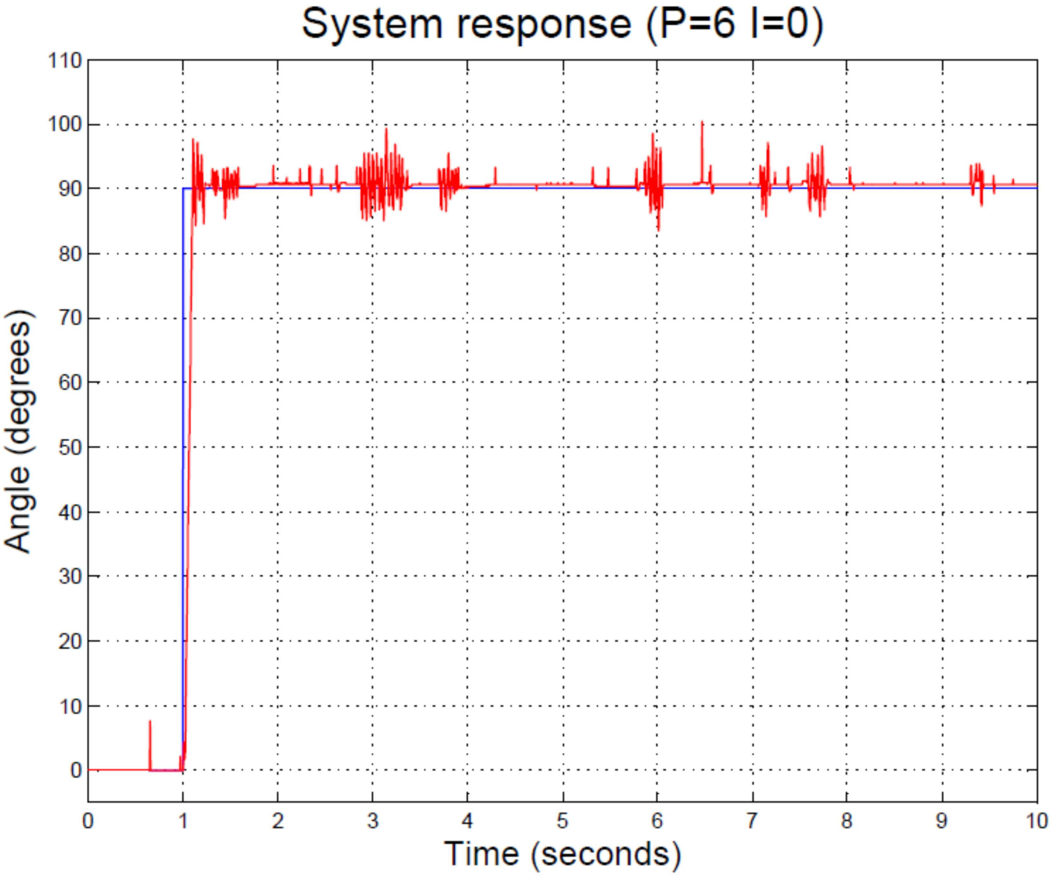


Figure 39 System unstable response to a step input (P=6 I=0)

7 Errors and mistakes quantification

7.1 Arduino resolution

The analogic input of the Arduino board has 1024 levels, from 0 to 1023, since it uses a 10 bit Anolog to Digital converter (ADC).

The resolution of the ADC depends on the applied voltage so with a voltage of 5V the resolution is equal to

$$\frac{5}{1024} = 0{,}00488\, V$$

It is possible to increase the resolution by applying a lower voltage, for example, 3,3 V or 1,5 V.

7.2 Potentiometer errors

Looking at Figure 40 it is possible to detect several spikes of different magnitude and randomly distributed.

These spikes are due to the potentiometer since this kind of electric equipment is well known to be noisy.

Fortunately, each spike lasts for just the sampling time of the system, that is equal to 0,05 s, so the spikes are not affecting the motor response too much because a change so limited, both in time and in amplitude, is not difficult to be countered by the controller.

This type of error is obviously a random error therefor it is possible to make an estimation of it by evaluating its mean and standard deviation.

Unpredictable fluctuations in a measure apparatus and reading misinterpretations are the most common source of random error. These ones are very difficult to eliminate and avoid the random too. So as a result random error is always present.

On the other hand, systematic errors can be either predicted or avoided. They are related to the true value, usually through a proportional constant.

Moreover, the cause of the systematic error can be eliminated if identified. These errors always affect the results in a predictable direction. The systematic error can be caused by instruments of measurements or method of observation, interference of the environment with the measurement process.

For this purpose, the signal coming out of the potentiometer when it is in a random position was recorded and analysed in Matlab. 10 trials were recorded for each position.

All the signals , lasting for 15 s, were recorded with a sampling time equal to 0,05 s, and one of this is shown in Figure 40 and Figure 41 below. The first one shows the whole signal while in the other one the time scale is changed in order to look deeper at the signal features.

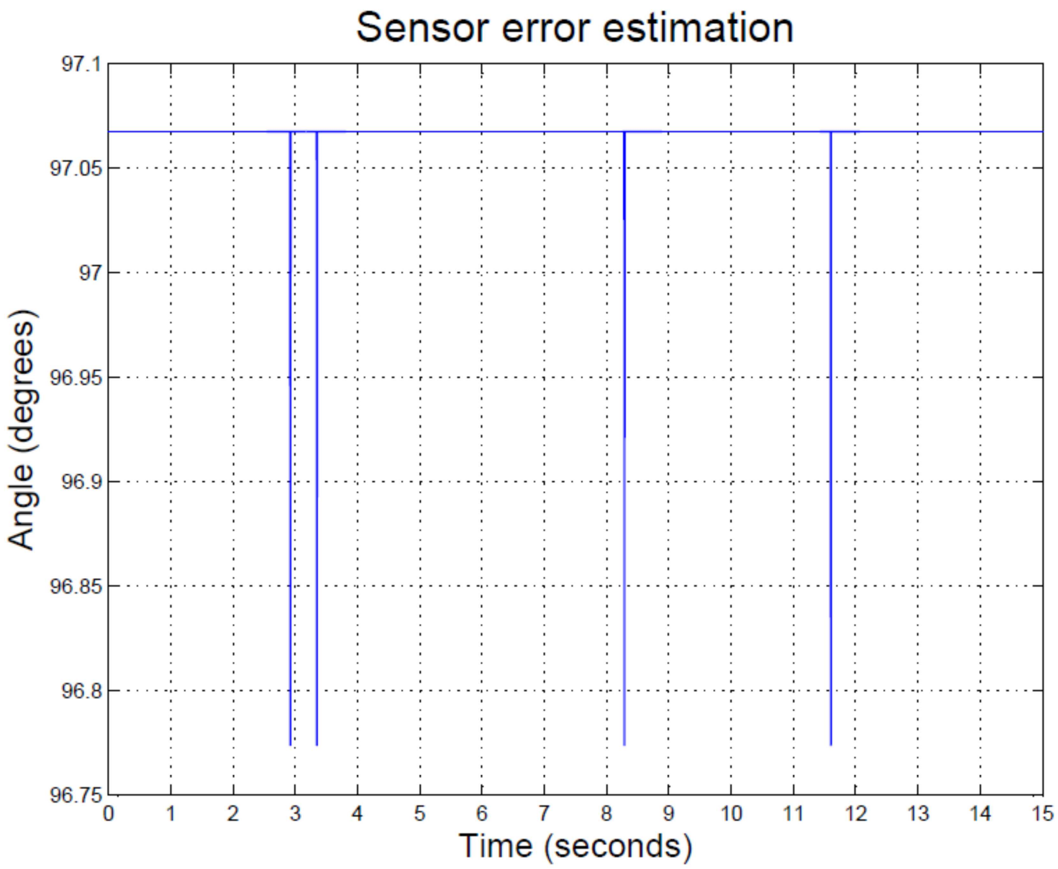


Figure 40 Sensor error estimation

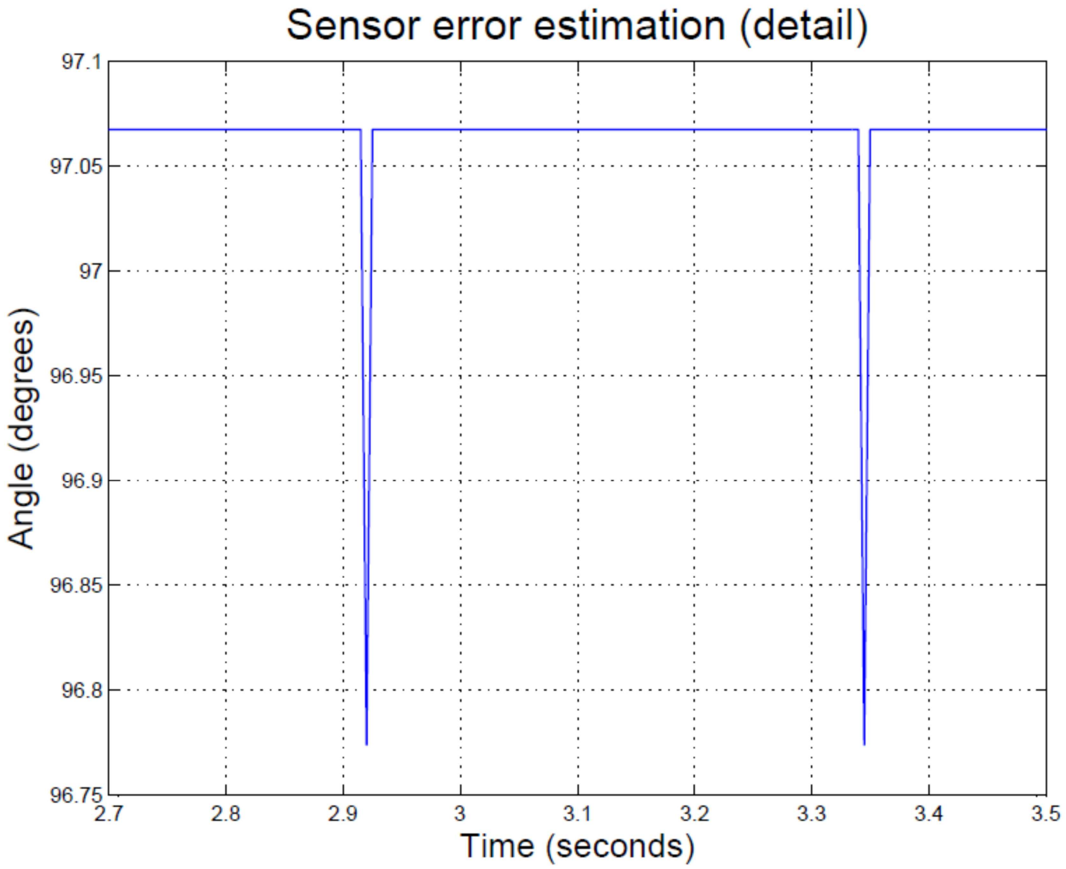


Figure 41 Sensor error estimation (detail)

The table below shows the report of the recorded signals :

Table 9 Sensor error values

Trial	Max (°)	Min (°)	Mean (°)	Standard Deviation (°)
1	97,0674	96,7741	97,067	0,0107
2	97,0674	96,7741	97,067	0,0107
3	97,0674	96,7741	97,0669	0,012
4	97,0674	97,0674	97,0674	2,13E-12
5	97,0674	96,7741	97,0671	0,0093
6	97,0674	96,7741	97,0673	0,0054
7	97,0674	96,7741	97,0668	0,0131
8	97,0674	96,7741	97,0664	0,0169
9	97,3606	96,7741	97,0669	0,0161
10	97,0674	96,7741	97,0671	0,0093
Mean	97,09672	96,80343	97,06699	0,01035
Max	--------	-------	--------	0,0169

Since the standard deviation can be used as a good estimation of the random error it can be concluded the random error due to the sensor is equal to 0,0169°. This value, as it can be noticed from the table above, corresponds to the maximum value and not to the mean one. This choice is obviously made in order to consider the worst case.

8 Result analysis

This section reports the output signals of the motor and their analysis using as input signals those discussed in sections 4.2.3 and 4.2.4.

8.1 Triangular input response

Figure 42, Figure 44 and Figure 46 show the response of the system to the triangular input, discussed in section 4.2.3, for different gains of the PI controller.

Each Figure shows two signals: the input one in blue and the output one in red.

All the spikes of the red signal shown in those figures are due to the noise of the potentiometer as described in chapter 7.2.

The absolute values of the difference between the input and output signals were calculated, filtered to reduce the noise and shown in Figure 43,Figure 45 and Figure 47.

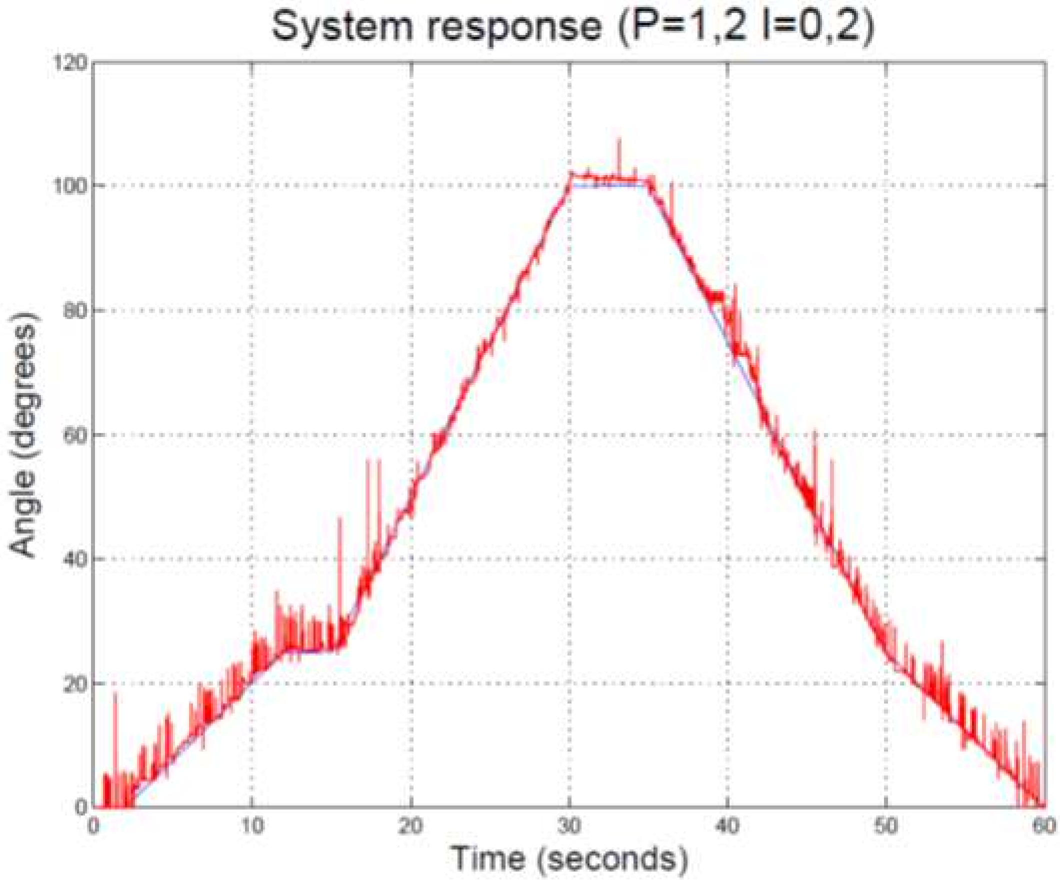


Figure 42 System response to triangular waveform (P=1,2 I=0,2)

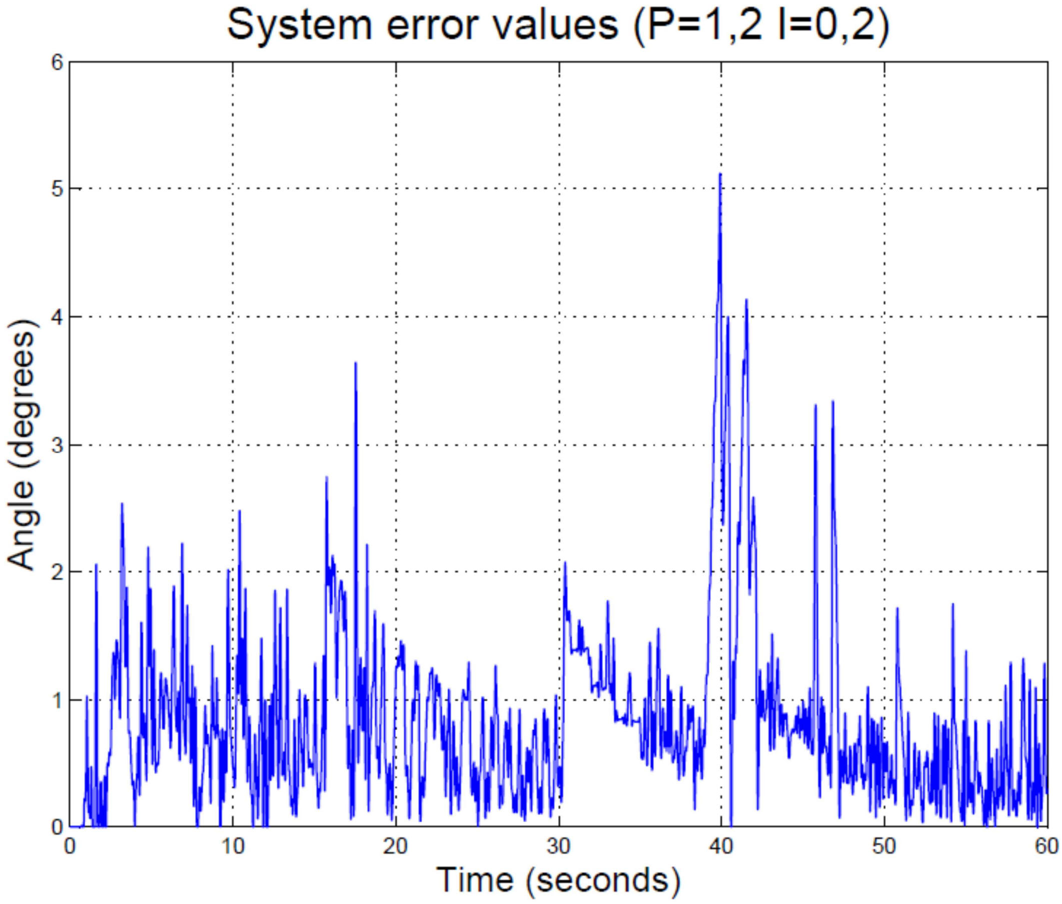


Figure 43 Evaluated error values between system response and triangular input waveform (P=1,2 I=0,2)

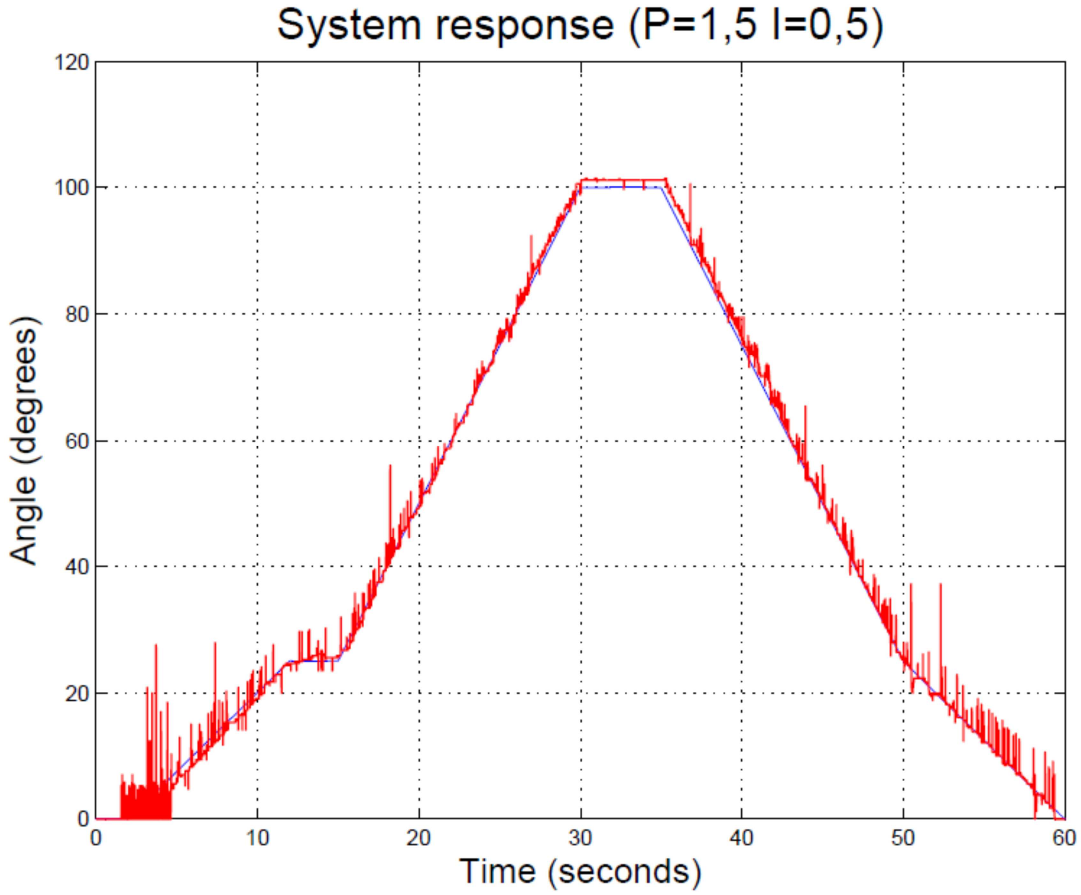


Figure 44 System response to triangular waveform (P=1,5 I=0,5)

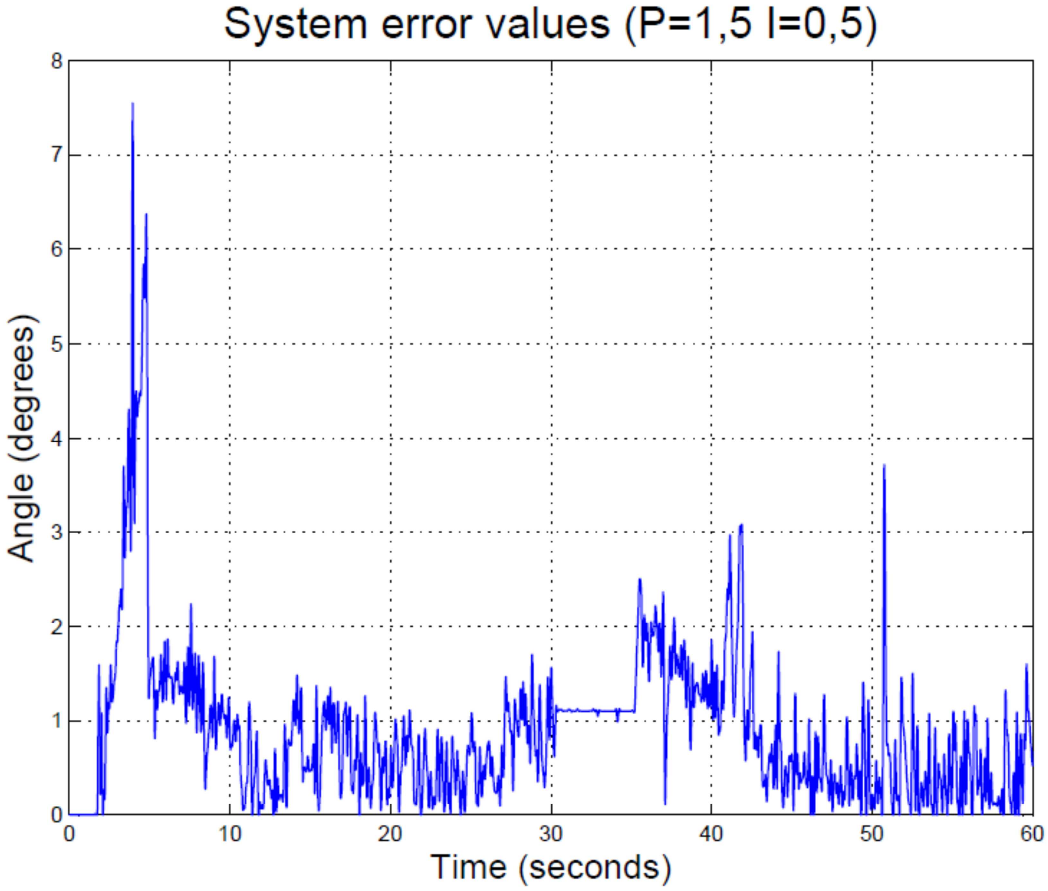


Figure 45 Evaluated error values between system response and triangular input waveform (P=1,5 I=0,5)

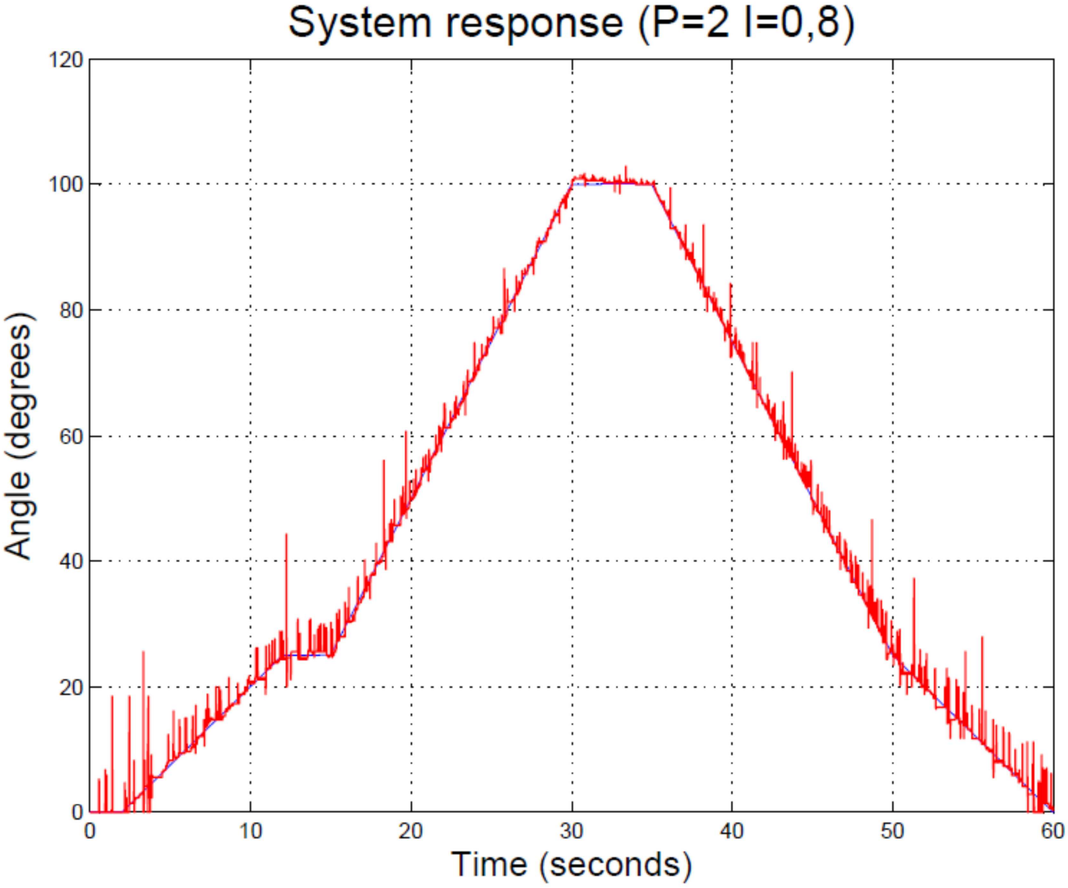


Figure 46 System response to a triangular waveform (P=2 I=0,8)

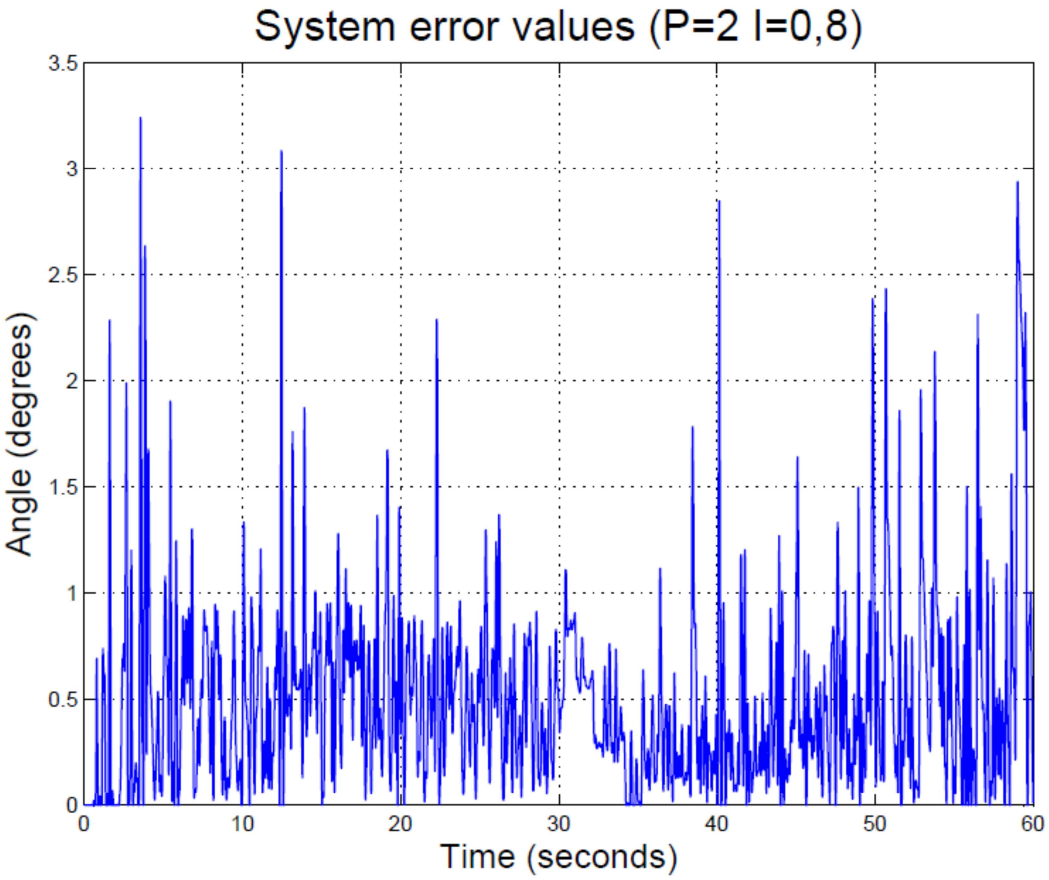


Figure 47 Evaluated error values between system response and triangular input waveform (P=2 I=0,8)

Table 10, shown below, reports information about the error, calculated as the difference of the values between input and output signals, according to the different gains of the PI controller. In particular, the informations shown are the average value, the maximum value and the standard deviation of the error values.

Table 10 System error values when the input is a triangular waveform

	Figure 43	Figure 45	Figure 47
P gain [k_p]	1,2	1,5	2
I gain [k_i]	0,2	0,5	0,8
Mean error value (°)	0,8424	0,8941	0,5075
Maximum error value (°)	5,1269	7,5449	3,2395
Standard Deviation of the error values (°)	0,6953	0,8271	0,4507

Looking at Table 10 it can be noticed that the best results are obtained using the higher gains, among all those evaluated, for both the proportional and integral gains of the PI controller. These values showed to produce one of the biggest overshoot during the trials and errors tuning of the PI controller, as reported in Table 8, but they also showed to guarantee a faster adaptability and a smaller error with a less ideal signal.

Moreover the small values of both the mean error and the standard deviation of the errors confirm that the maximum error value is not due to the system but to the noise produced by the potentiometer.

Furthermore, by looking at Figure 42, Figure 44 and Figure 46 it can be realized that there is no delay between the input signals and the system responses.

8.2 Electrooptical signal

Figure 48, Figure 50 and Figure 52 shown the response of the system to the real signal, discussed in section 4.2.4, for different gains of the PI controller.

Each Figure shows two signals: the input one in blue and the output one in red.

All the spikes of the red signal shown in those figures are due to the noise of the potentiometer as described in chapter 7.2.

The absolute values of the difference between the input and output signals were calculated, filtered to reduce the noise and showed in Figure 49, Figure 51 and Figure 53.

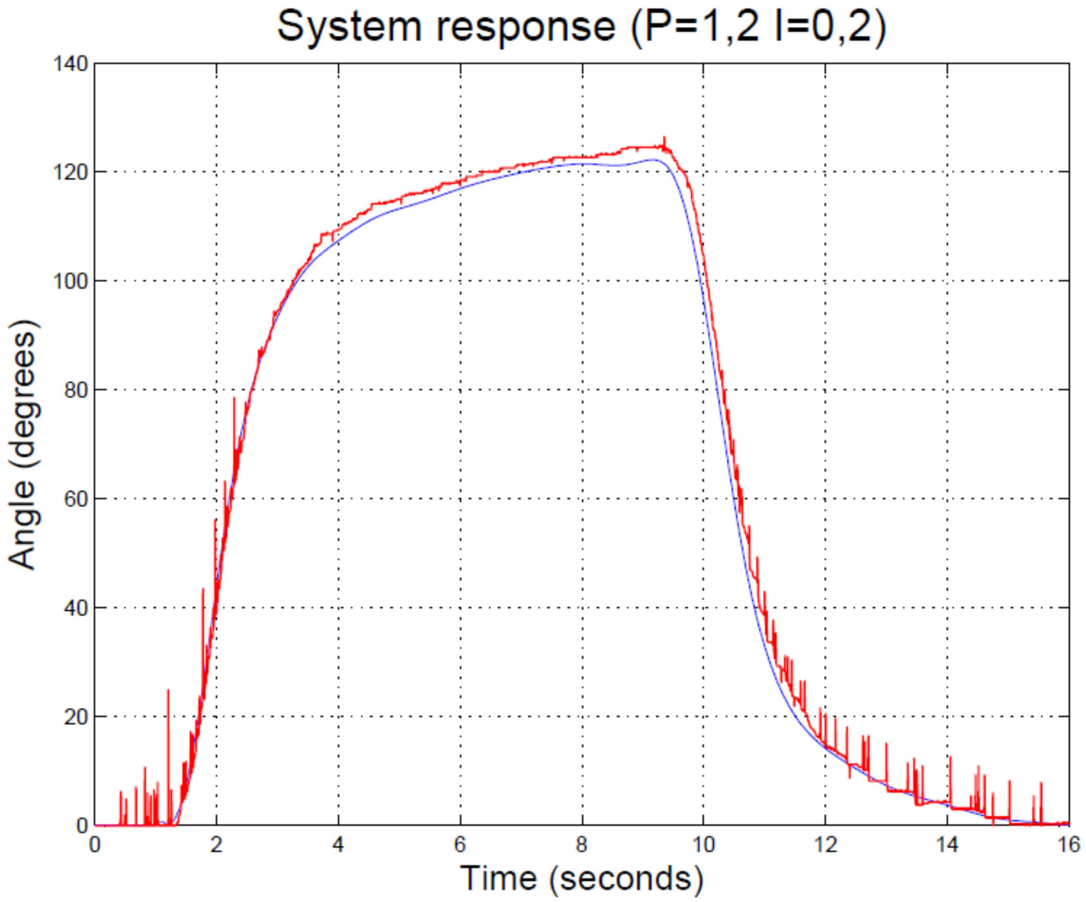


Figure 48 System response when triggered by a muscle signal (P=1,2 I=0,2)

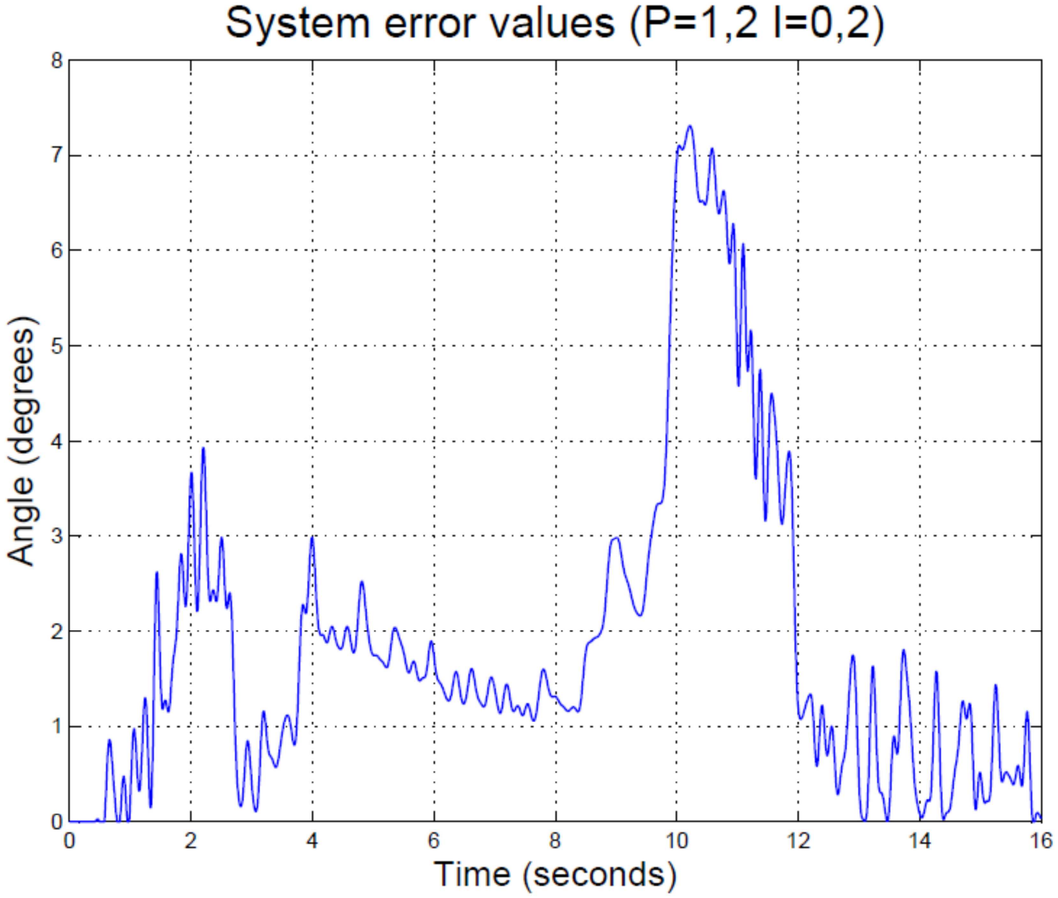


Figure 49 Evaluated error values between system response and angle signal related to the isotonic contraction signal (P=1,2 I=0,2)

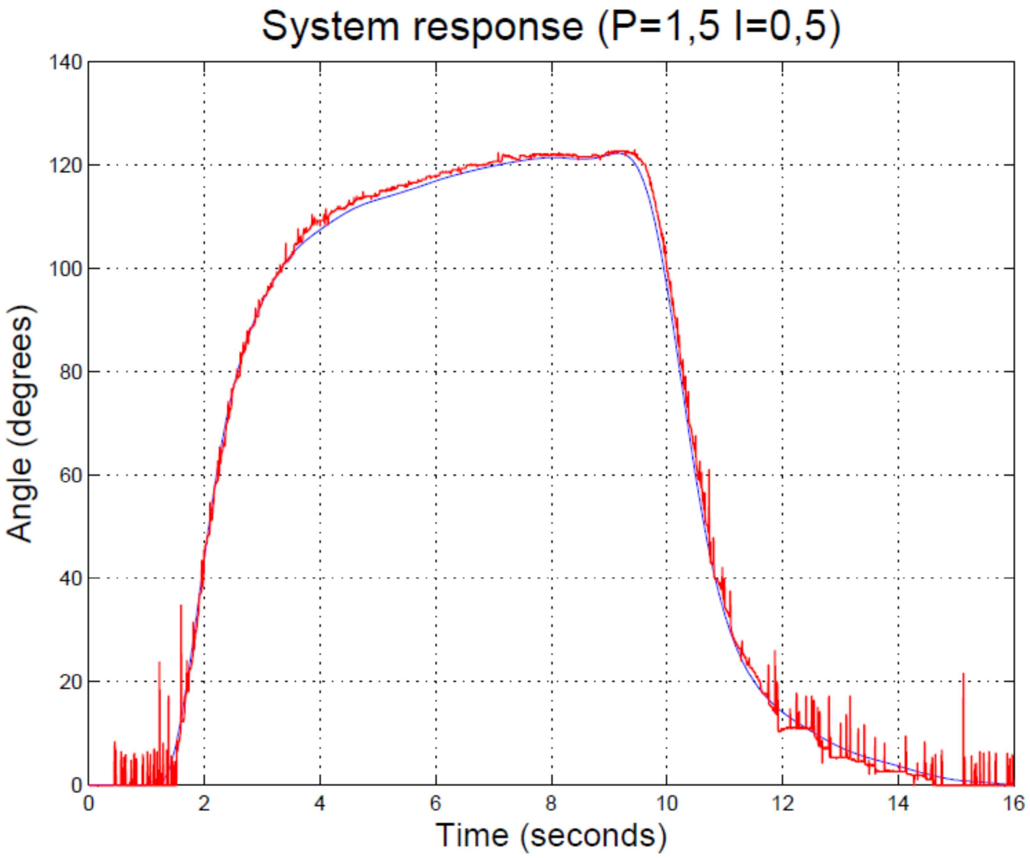


Figure 50 System response when triggered by a muscle signal (P=1,5 I=0,5)

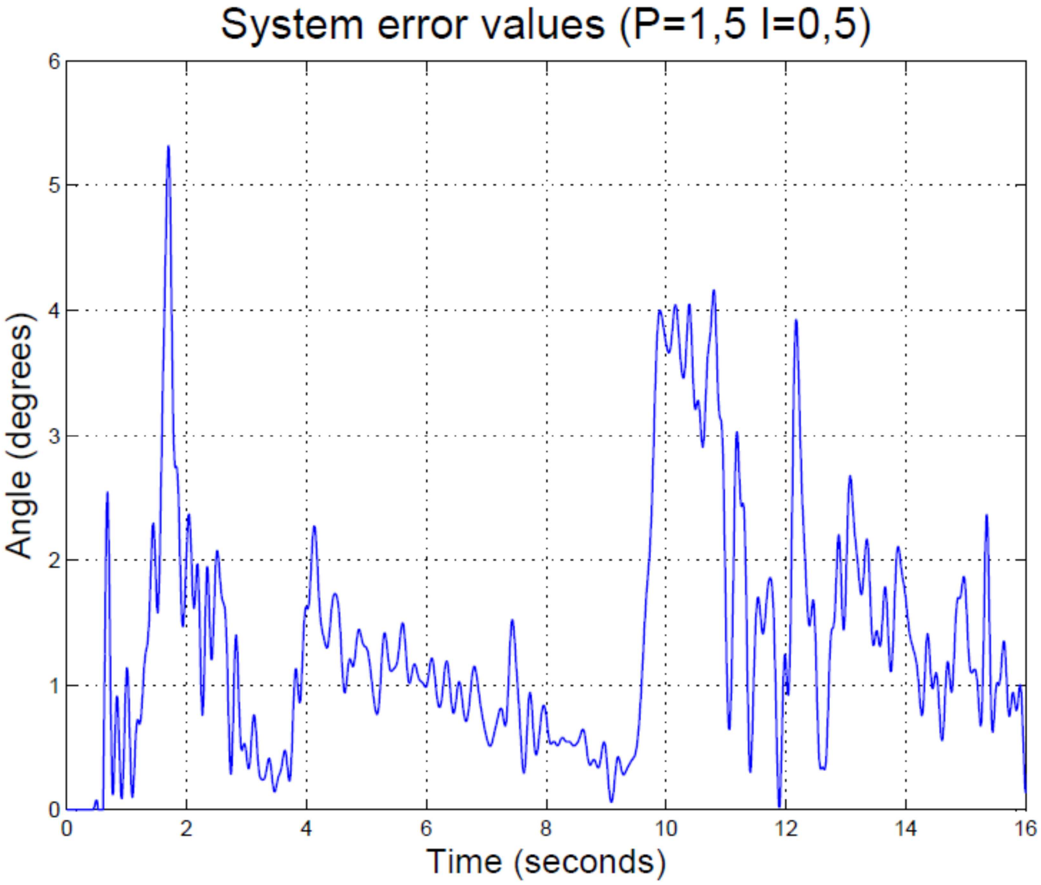


Figure 51 Evaluated error values between system response and angle signal related to the isotonic contraction signal (P=1,5 I=0,5)

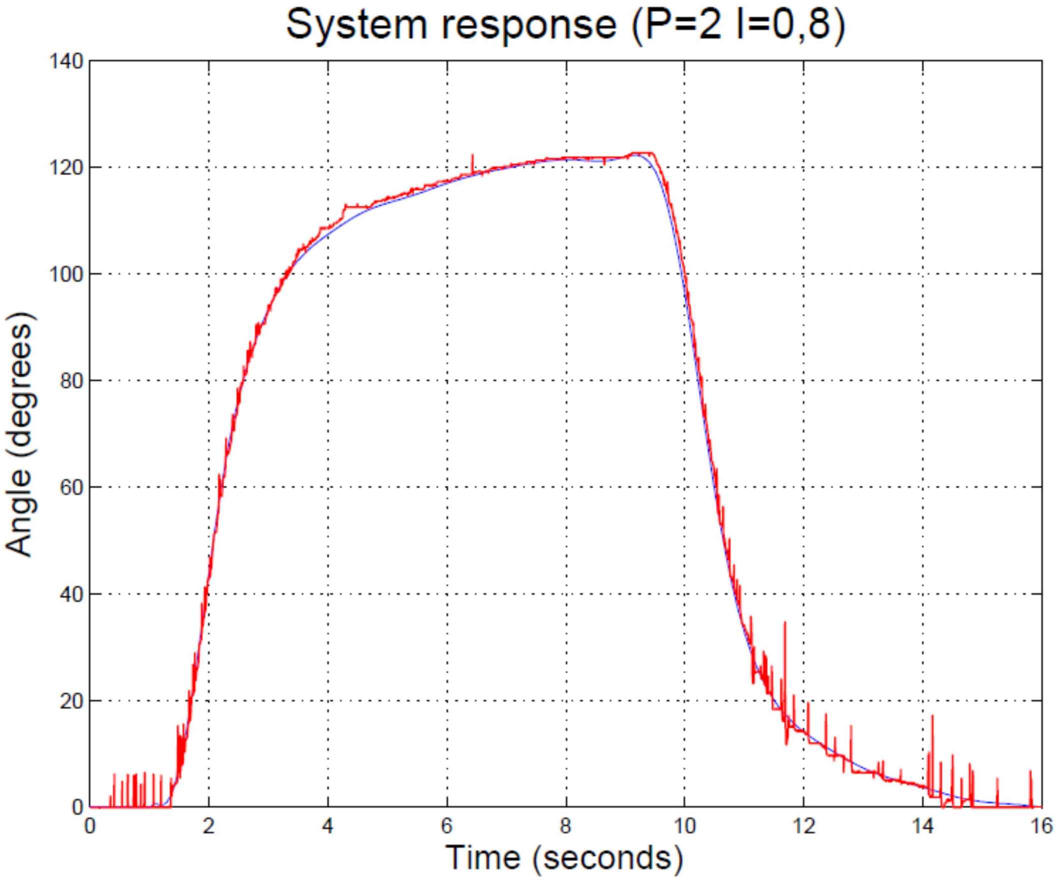


Figure 52 System response when triggered by a muscle signal (P=2 I=0,8)

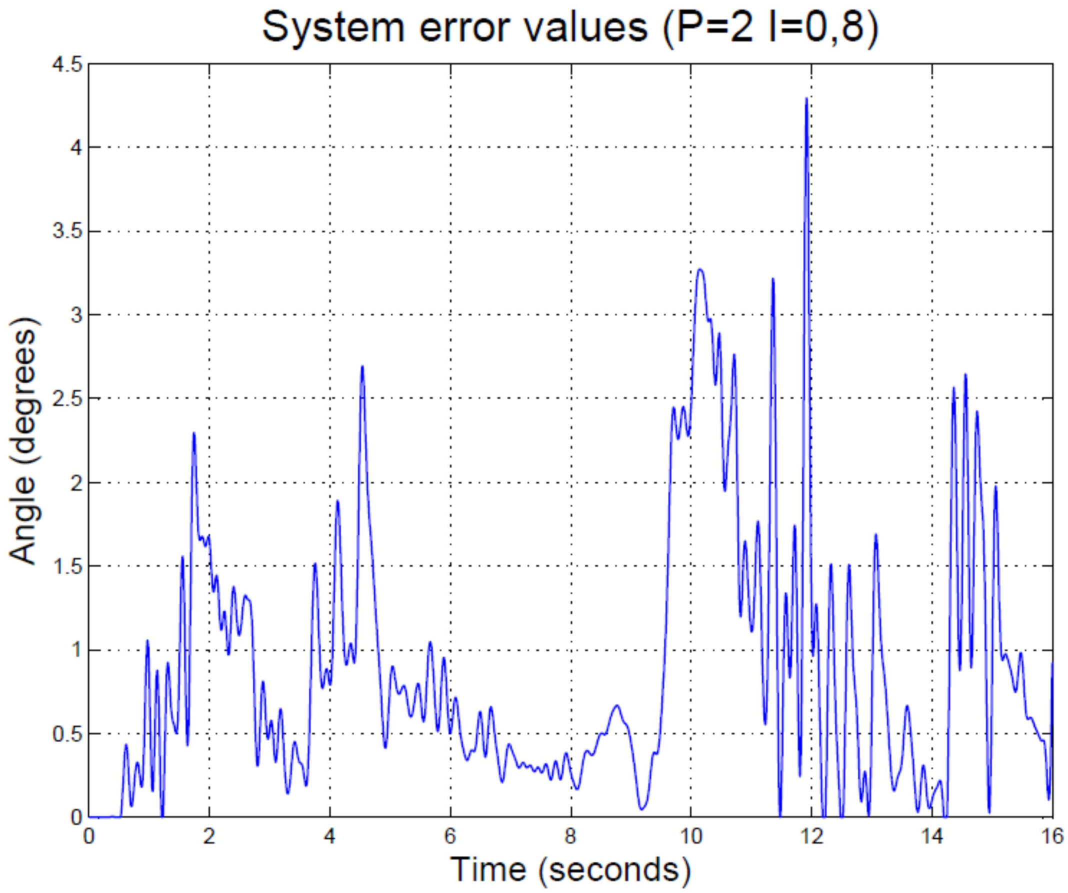


Figure 53 Evaluated error values between system response and angle signal related to the isotonic contraction signal (P=2 I=0,8)

Table 11, shown below, reports information about the error, calculated as the difference of the values between input and output signals, according to the different gains of the PI controller. In particular, the informations shown are the average value, the maximum value and the standard deviation of the error values.

Table 11 System error values when the input is related to the muscle activity

	Figure 49	Figure 51	Figure 53
P gain [k_p]	1,2	1,5	2
I gain [k_i]	0,2	0,5	0,8
Mean error value (°)	1,8569	1,3265	0,8941
Maximum error value (°)	7,3057	5,3210	4,2929
Standard Deviation of the error values (°)	1,6934	0,9813	0,7745

Once again using the greatest values for the PI gains ensured the best results as it can be noticed from Table 11. In fact, with a proportional gain equal to 2 and the integral gain equal to 0,8 the average error of the system is equal to 1° with the same value for the standard deviation.

Since the maximum error values are very far from the average and considering the relative small values of the standard deviation it can be concluded, as in the case of the triangular waveform, that these maximum values are not due to the system but to the noisy angular sensor.

Moreover, no time delay was introduced between the input signal and the system response as it can be noticed from Figure 48, Figure 50 and Figure 52.

9 Transfer function

The Transfer function of the whole system was calculated both in open loop and closed loop configuration.

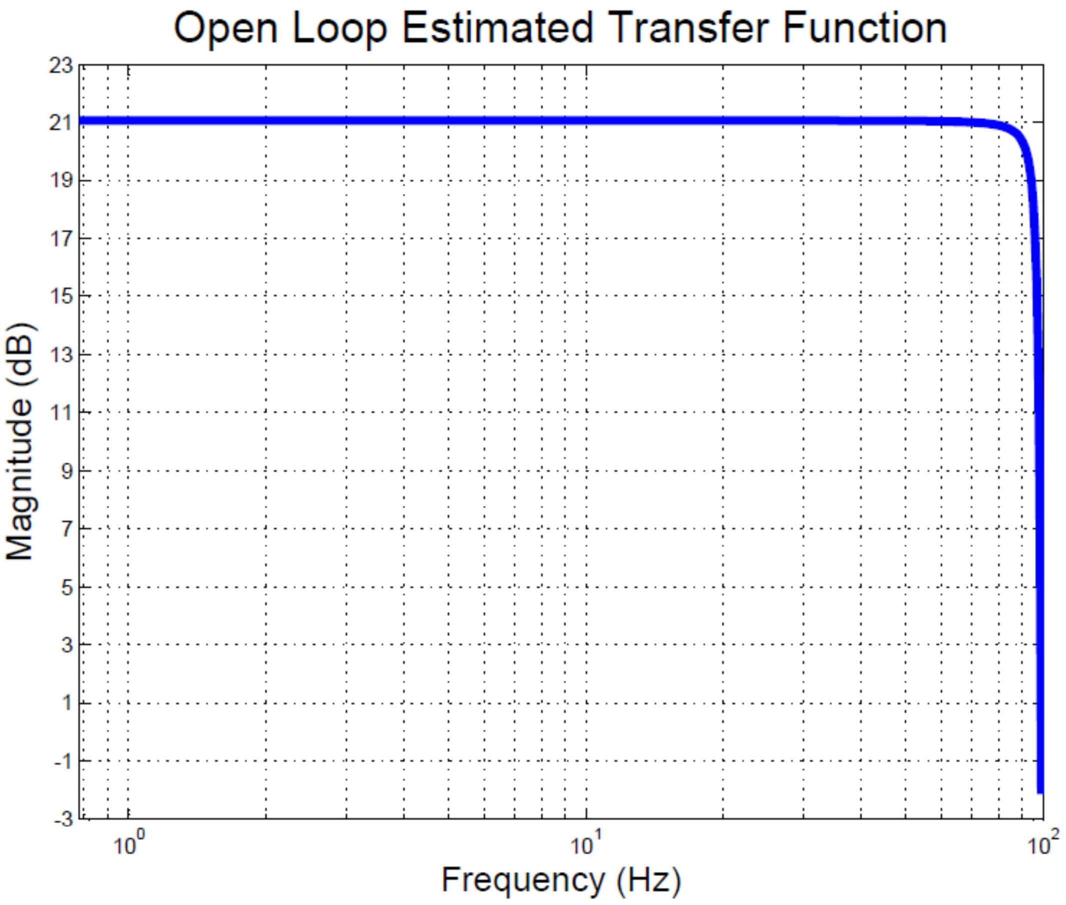


Figure 54 Open loop estimated transfer function

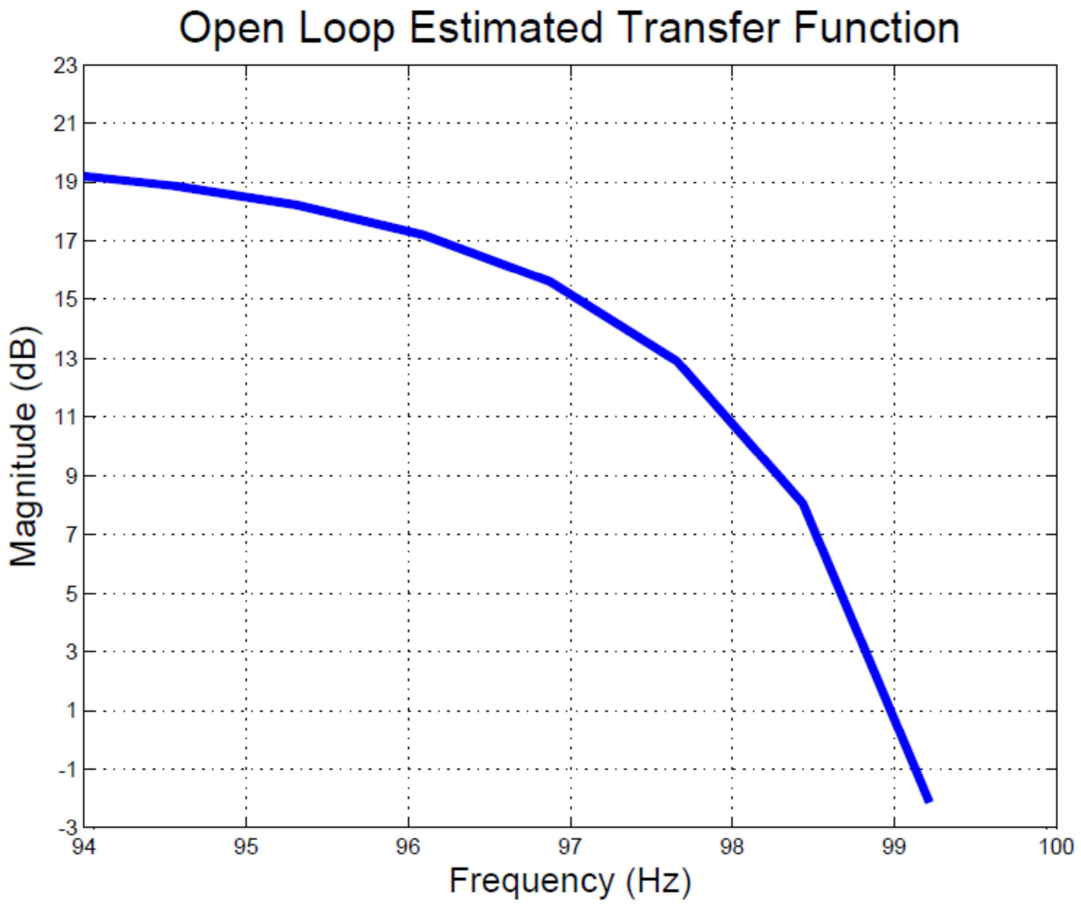


Figure 55 Open loop estimated transfer function (detail)

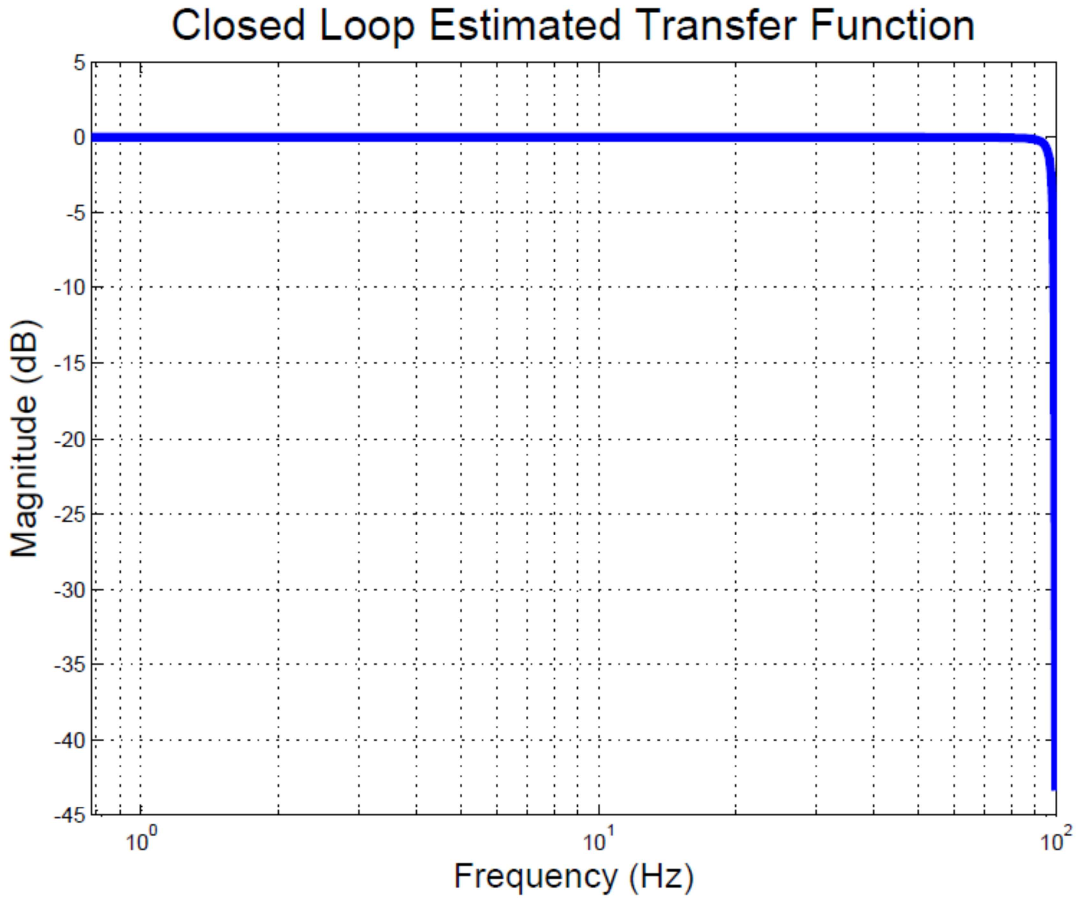


Figure 56 Closed loop estimated transfer function

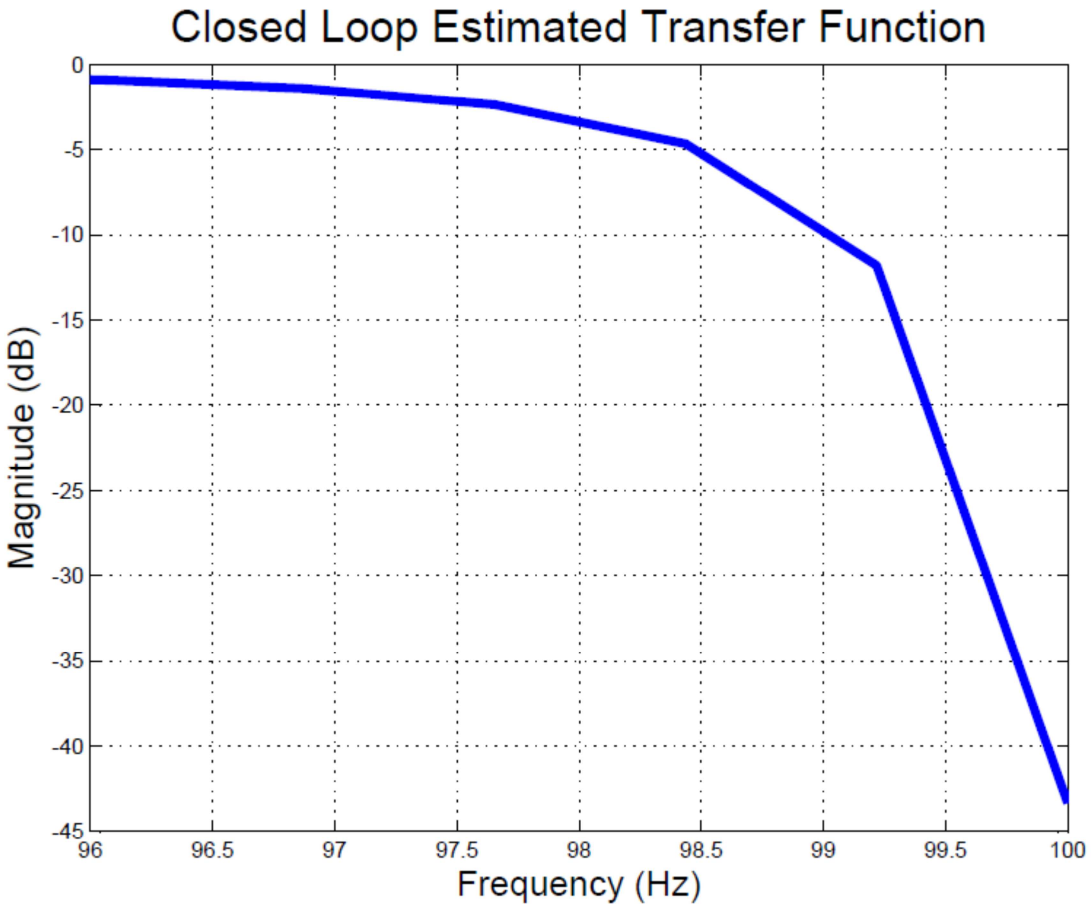


Figure 57 Closed loop estimated transfer function (detail)

The closed loop system showed to have a gain equal to 0 dB and a -3 dB frequency of 97,87 Hz. This value is mainly due to the sampling frequency that is equal to 200 Hz.

The gain of 0 dB ensures the stability of the system, but a notch filter could be used to eliminate some disturbance like the 50 Hz.

On the other hand, the gain of the open loop system is equal to 21 dB with a -3 dB frequency of 95,45 Hz.

Conclusion

The aim of this thesis work was to design a controller for a rehabilitation exoskeleton with only one DOF. The Simulink software running on a normal computer is used to control the exoskeleton actuator. The signals used were either related to a person’s muscle activity or ideal signals generated in Simulink, according to the different purposes of the rehabilitation.

All the tests conducted on the designed controller showed a good response of the system under the profile of both the angle amplitude and the timing. The system was tested both in robot in charge and patient in charge tasks with great success.

The overshoot values were successfully limited to less than 5% and the steady state error was equal to 0,03%, so a good controllability of the exoskeleton is guaranteed during the rehabilitation sessions.

The use of a normal computer to control the exoskeleton allows a great customisation of the parameters according to the needs. In fact, the physiotherapist can decide time by time the parameters of a robot in charge task rehabilitation session via the designed Simulink mask.

The system showed a good stability and a -3 dB frequency of 97,87 Hz that allows for several uses with different signals.

Bibliography

1. Díaz I, Gil J, and Sánchez, E. Lower-limb robotic rehabilitation: literature review and challenges. Journal of Robotics, 2011.

2. Vitiello N, Lenzi T, Roccella S, De Rossi S, Cattin E, Giovacchini F, Vecchi F and Carrozza M. (2013). NEUROExos: A powered elbow exoskeleton for physical rehabilitation. Robotics, IEEE Transactions on, 2013, 29(1), pp.220-235.

3. Aisen ML, Krebs HI, Hogan N, McDowell F, Volpe BT. The effect of robot assisted therapy and rehabilitative training on motor recovery following stroke. Archives of Neurology 1997, 54:443-446.

4. Volpe BT, Krebs HI, Hogan N, Edelsteinn L, Diels CM, Aisen ML. Robot training enhanced motor outcome in patients with stroke maintained over 3 years. Neurology 1999, 53(8):1874-1876.

5. Lum PS, Burgar CG, Shor PC, Majmundar M, Van der Loos M. Robot assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. Archives of Physical Medicine and Rehabilitation 2002, 83(7):952-959.

6. Fasoli SE, Krebs HI, Stein J, Frontera WR, Hughes R, Hogan N. Robotic therapy for chronic motor impairments after stroke: Follow-up results. Archives of Physical Medicine and Rehabilitation 2004, 85(7):1106-1111.

7. Kahn LE, Rymer WZ, Reinkensmeyer DJ. Adaptive assistance for guided force training in chronic stroke. Conf Proc IEEE Eng Med Biol Soc 2004,4:2722-2725.

8. Lum PS, Burgar CG, Van der Loos M, Shor PC, Majmundar M, Yap R. MIME robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study. Journal of Rehabilitation Research & Development 2006, 43(5):631-642.

9. Dovat L, Lambercy O, Gassert R, Maeder T, Milner T, Leong TC, Burdet E. HandCARE: a cable-actuated rehabilitation system to train hand function after stroke. IEEE Transactions on Neural Systems and Rehabilitation Engineering 2008, 16(6):582-91.

10. Dovat L, Lambercy O, Salman B, Johnson V, Milner T, Gassert R, Burdet E, Leong TC. A technique to train finger coordination and independence after stroke. Disabil Rehabil Assist Technol 2010, 5(4):279-87.

11. Bouzit M, Burdea G, Popescu G, Boian R. The Rutgers Master II-New Design force-feedback glove. IEEE/ASME Transactions on Mechatronics 2002, 12(4):399-407.

12. Boian R, Sherman A, Han C, Merians A, Burdea G, Adamovich S, Recce M,Tremaine M, Poizner H. Virtual-reality-based post-stroke hand rehabilitation. Studies in Health Technology and Informatics 2002, 85:64-70.

13. Merians AS, Jack D, Boian R, Tremaine M, Burdea GC, Adamovich SV,Recce M, Poizner H. Virtual reality-augmented rehabilitation for patients following stroke. Physical Therapy 2002, 82(9):898-915.

14. Lambercy O, Dovat L, Gassert R, Burdet E, Teo CL, Milner T. A haptic knob for rehabilitation of hand function. IEEE Transactions Neural Systems Rehabilitation Engineering 2007, 15(3):356-66.

15. Masia L, Krebs HI, Cappa P, Hogan N. Design and Characterization of Hand Module for Whole-Arm Rehabilitation Following Stroke. IEEE ASME Transactions on Mechatronics 2007, 12(4):399-407.

16. Adamovich SV, Fluet GG, Mathai A, Qiu Q, Lewis J, Merians AS. Design of a complex virtual reality simulation to train finger motion for persons with hemiparesis: a proof of concept study. Journal of NeuroEngineering and Rehabilitation 2009, 17:6-28.

17. Wege A, Zimmerman A: Electromyography sensor based control for a hand exoskeleton. Proceedings of the 2007 IEEE ICRB 1470-1475.

18. Fischer HC, Stubblefield K, Kline T, Luo X, Kenyon RV, Kamper DG. Hand rehabilitation following stroke: a pilot study of assisted finger extension training in a virtual environment. Topics in Stroke Rehabilitation 2007, 14(1):1-12.

19. Takahashi C, Der-Yeghiaian L, Le V, Cramer SC. A robotic device for hand motor therapy after stroke. Proceedings of IEEE 9th International Conference on Rehabilitation Robotics: Frontiers of the Human-Machine Interface Chicago, Illinois 2005, 17-20.

20. Schabowsky et al. Development and pilot testing of HEXORR: Hand EXOskeleton Rehabilitation Robot. Journal of NeuroEngineering and Rehabilitation 2010 7:36.

21. Colombo G, Joerg M, Schreier R, and Dietz V. Treadmill training of paraplegic patients using a robotic orthosis, Journal of Rehabilitation Research and Development, vol. 37, no. 6, pp. 693–700, 2000.

22. Freivogel S, Mehrholz J, Husak-Sotomayor T, and Schmalohr D. Gait training with the newly developed “LokoHelp”-system is feasible for non-ambulatory patients after stroke, spinal cord and brain injury. A feasibility study, Brain Injury, vol. 22, no. 7-8, pp. 625–632, 2008.

23. Reinkensmeyer D, Wynne J, and Harkema S. A robotic tool for studying locomotor adaptation and rehabilitation, in Proceedings of the 2nd Joint Meeting of the IEEE

Engineering in Medicine and Biology Society and the Biomedical Engineering Society, vol. 3, pp. 2013–2353, October 2002.

24. Reinkensmeyer DJ, Aoyagi D, Emken JL, et al. Tools for understanding and optimizing robotic gait training, Journal of Rehabilitation Research and Development, vol. 43, no. 5, pp. 657–670, 2006.

25. Banala SK, Agrawal SK, and Scholz JP. Active Leg Exoskeleton (ALEX) for gait rehabilitation of motor-impaired patients, in Proceedings of the 10th IEEE International Conference on Rehabilitation Robotics, (ICORR '07), pp. 401–407, Noordwijk, The Netherlands, June 2007

26. Beyl P, van Damme M, van Ham R, Versluys R, Vanderborght B, and Lefeber D. An exoskeleton for gait rehabilitation: prototype design and control principle, in Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '08), pp. 2037–2042, Pasadena, Calif, USA, May 2008.

27. Pietrusinski M, Cajigas I, Mizikacioglu Y, Goldsmith M, Bonato P, and Mavroidis C. Gait rehabilitation therapy using robot generated force fields applied at the pelvis, in Proceedings of the IEEE Haptics Symposium, (HAPTICS '10), pp. 401–407,Waltham, Mass, USA, March 2010.

28. Yoon J, Novandy B, Yoon CH, and Park KJ. A 6-DOF gait rehabilitation robot with upper and lower limb connections that allows walking velocity updates on various terrains, IEEE/ASME Transactions on Mechatronics, vol. 15, no. 2, Article ID 5424007, pp. 201–215, 2010.

29. Peshkin M, Brown DA, Santos-Munne JJ et al. KineAssist: a robotic overground gait and balance training device, in Proceedings of the 9th IEEE International

Conference on Rehabilitation Robotics, (ICORR ’05), pp. 241–246, Evanston,Ill, USA, July 2005.

30. Bouri M, Stauffer Y, Schmitt C et al. The walktrainer: a robotic system for walking rehabilitation, in Proceedings of the IEEE International Conference on Robotics and Biomimetics, (ROBIO ’06), pp. 1616–1621, Kunming, China, December 2006.

31. Kawamoto H and Sankai Y. Power assist system hal-3 for gait disorder person, in Proceedings of the 8th International Conference on Computers Helping People with Special Needs, pp. 196–203, London, UK, 2002.

32. Seo KH and Lee JJ. The development of two mobile gait rehabilitation systems, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 17, no. 2, Article ID 4785182, pp. 156–166, 2009.

33. Schmitt C, M´etrailler P, Al-Khodairy A, et al. The motion maker: a rehabilitation system combining an orthosis with closed-loop electrical muscle stimulation, in Proceedings of the 8th Vienna International Workshop on Functional Electrical Stimulation, pp. 117–120, Vienna, Austria, September 2004.

34. Saglia JA, Tsagarakis NG, Dai JS, and Caldwell DG. A high-performance redundantly actuated parallel mechanism for ankle rehabilitation, International Journal of Robotics Research, vol. 28, no. 9, pp. 1216–1227, 2009.

35. Krebs HI, Dipietro L, Levy-Tzedek S, et al. A paradigm shift for rehabilitation robotics, IEEE Engineering in Medicine and Biology Magazine, vol. 27, no. 4, pp. 61–70, 2008.

36. Khanna I, Roy A, Rodgers MM, Krebs HI, MacKo RM, and Forrester LW. Effects of unilateral robotic limb loading on gait characteristics in subjects with chronic stroke, Journalof NeuroEngineering and Rehabilitation, vol. 7, no. 1, article 23, 2010.

37. Sulter G, Steen C, and De Keyser J. Use of the barthel index and modified rankin scale in acute stroke trials, Stroke, vol. 30, no. 8, pp. 1538–1541, 1999.

38. Bubnicki Z. Modern control theory. 1st ed. Berlin: Springer, 2005.

39. Barabas M. DC motor speed and position control system connects directly to microcomputer chips. Scientific Bulletin of the Petru Maior University of Tarlu Mures 2011, pp. 1-7.

40. Altheris.com. High Precision Potentiometers and Rotary angle sensors, 2014. Available at: http://www.altheris.com/potentiometers-overview.htm [Accessed 4 Aug. 2014].

41. http://eecs.oregonstate.edu/.ECE 111 Lab Manual, 2014. Available at: http://eecs.oregonstate.edu/education/docs/ece111/CompleteManual.pdf.

42. Barr M. Introduction to Pulse Width Modulation. Embedded, 2001. Available at: http://www.embedded.com/electronics-blogs/beginner-s-corner/4023833/Introduction-to-Pulse-Width-Modulation [Accessed 9 Aug. 2014].

43. Ni.com. PID Theory Explained - National Instruments, 2014. Available at: http://www.ni.com/white-paper/3782/en/ [Accessed 4 Aug. 2014].

44. Classical PID Control by Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado Control System Design, Prentice Hall PTR

45. PID Control of Continuous Processes by John W. Webb Ronald A. Reis Programmable Logic Controllers, Fourth Edition, Prentice Hall PTR

46. Johnson M, Moradi M, and Crowe J. PID control. 1st ed. New York: Springer, 2005.

47. Arduino.cc, (2014). Arduino - Introduction. Available at: http://arduino.cc/en/Guide/Introduction [Accessed 4 Aug. 2014].

48. Arduino.cc, (2014). Arduino Mega 2560. Available at: http://arduino.cc/en/Main/arduinoBoardMega2560 [Accessed 4 Aug. 2014].

49. Chianura A, Giardini ME. An electrooptical muscle contraction sensor. Medical & biological engineering & computing, 2010, 48 (7), pp. 731--734.

50. Servodatabase.com. TowerPro SG-5010 Servo Specifications and Reviews, 2014. Available at: http://www.servodatabase.com/servo/towerpro/sg-5010 [Accessed 4 Aug. 2014].

51. Ang K, Chong G, and Li Y. PID control system analysis, design, and technology. Control Systems Technology, IEEE Transactions on, 2005, 13(4), pp.559-576. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1453566 [Accessed 18 Jul. 2014].