Universidade de Aveiro Departamento de Engenharia Mecânica
2015

João Filipe Torres Valente

Servo-sistema para perturbação controlada do equilíbrio postural
Servo-system for controlled postural balance disruption

Universidade de Aveiro Departamento de Engenharia Mecânica
2015

João Filipe Torres Valente

Servo-sistema para perturbação controlada do equilíbrio postural
Servo-system for controlled postural balance disruption

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob orientação científica de Vítor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro e de António José Monteiro Amaro, Professor Coordenador s/ Agregação do Escola Superior de Saúde da Universidade de Aveiro da Universidade de Aveiro.

O júri / The jury

Presidente / President	**Prof. Doutor presidente** Professor Catedrático da Universidade de Aveiro
Vogais / Committee	**Prof. Doutor Vítor Manuel Ferreira dos Santos** Professor Associado da Universidade de Aveiro (orientador)
	Prof. Doutor arguente arguente da Universidade do xxxx(arguente)

Agradecimentos / Acknowledgements

Em primeiro lugar, agradeço ao professor Vítor Santos pela sua excelente orientação, incentivo e persistência.

Aos professores António Amaro e Filipe Silva pelo apoio prestado e motivação.

Aos professores José Azevedo e Bernardo Cunha pelo apoio técnico.

Aos meus colegas do LAR e arredores pela ajuda e boa disposição.

E finalmente, um especial agradecimento à minha família por sempre me apoiarem e suportarem ao longo do meu percurso académico.

Palavras-chave Estimulos; equilíbrio; pêndulo invertido; controlo.

Resumo

O estudo do equilíbrio postural é uma preocupação que se aplica igualmente a seres humanos e robôs com pernas. Em relação a Humanos este problema foca-se na deteção de patologias, avaliação da capacidade de manter o equilíbrio e recolher dados acerca da forma como o equilíbrio é mantido. Por outro lado, o estudo da postura de um robô tem como objetivo avaliar e melhorar o seu controlo. Dada a necessidade de testar o equilíbrio do Projeto Humanóide da Universidade de Aveiro - PHUA, surgiu a ideia de criar um sistema de distúrbio do equilíbrio. A ideia é o sistema perturbar o equilíbrio utilizando forças de tração que poderão ser aplicadas em todas as direções do plano horizontal. Como primeira abordagem, o conceito é testado num pêndulo assistido por elásticos, com o objetivo de perceber qual a melhor forma de controlar o sistema e encontrar possíveis problemas para o seu controlo. Numa primeira fase, é desenhado um controlador para criar um estado de balanço em que as forças de tração são iguais fazendo com que o pêndulo fique imóvel. Numa segunda fase, depois de atingir o estado descrito anteriormente, o pêndulo é brevemente tracionado num sentido e são medidas as forças e o ângulo do pêndulo para se analisarem os efeitos do estímulo. Assim, várias formas de aplicar estímulos são testadas assim como diferentes estímulos. Percebeu-se que: é possível aplicar estímulos repetiveis com este conceito; o facto de o sistema ser acoplado e não-linear causa dificuldades para o controlo, sendo necessário adoptar novas estratégias. Esta dissertação vai servir como suporte para o trabalho futuro, (que será construir o sistema à escala do PHUA) oferecendo um conjunto de informações e guias.

Keywords Stimuli; balance; inverted pendulum; control.

Abstract

Postural balance is a concern that applies to both humans and bipedal robots. For Humans, this issue is focused on detecting some pathologies, evaluating individual balance capability, and gather data on how equilibrium is maintained. On the other hand the robot's postural study aims to evaluate the machine's control and help tune it. Given the need to test the equilibrium of University of Aveiro Humanoid Project - PHUA, the idea of creating a balance disruption system came up. The idea is the system disrupt the balance using pull forces that can be applied in every direction of the horizontal plane. As a first approach, the concept is test on a pendulum assisted by elastics, with the objective of understanding which is the best way to control the system and find possible problems to its control. On an initial phase, a controller is designed to create a state of balance where the pull forces are equal making the pendulum stand still. On a second phase, after achieving the state of balance described previously, the pendulum is briefly pulled in a direction and tensile forces and the pendulum's angle are measured, with the purpose of analysing the effects of the stimulus. This way, different manners of applying stimuli and different types of stimuli are tested. It was realised that: it is possible to apply repeatable stimuli with this concept; the fact that the system is coupled and non-linear brings obstacles to its control, being necessary to adopt new control strategies. This dissertation will serve as a support for future work, (which is building the system at PHUA scale) offering a set of informations and guidelines.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Context and Motivation

Postural balance is a concern that applies both to humans and bipedal robots. For humans, this issue is focused on detecting some pathologies, evaluating individual balance capability, and gather data on how equilibrium is maintained. On the other hand, the robot's postural study aims to evaluate the machine's control and help tune it.

University of Aveiro Humanoid Project (PHUA) began in 2003/2004, as a result of a partnership between the Mechanical Engineering Department (DEM) and Electronics, Telecommunications and Informatics Department (DETI), of the University of Aveiro, with the purpose of creating a humanoid platform to study control, perception and behaviour. Its main goal is the development and application of hardware and software components in a functional low-budget platform, to carry out research in balance and locomotion tasks. The robot's current physical appearance is shown in Figure 1.1.

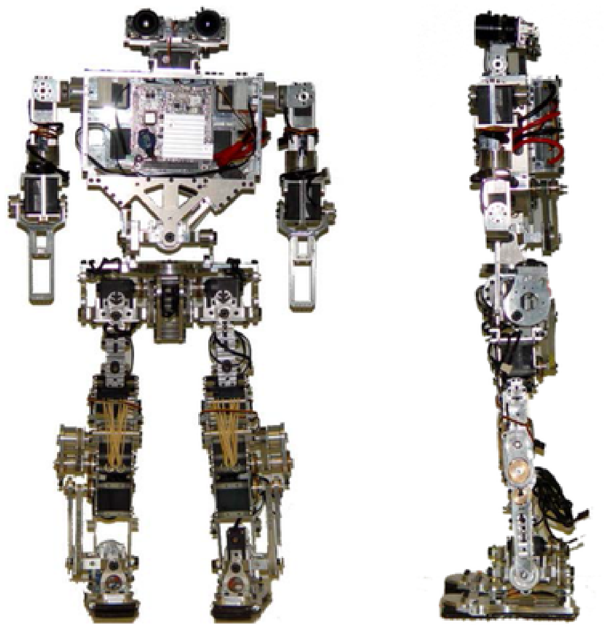


Figure 1.1: Front and side views of the PHUA platform's full body [2].

The high degree of complexity associated with PHUA places some restrictions on the definition of accurate models to the study of motion patterns. Some work has been done on trying to take advantage of the concept of robot learning from demonstration, where a human user teleoperates the robot in different motion and balance tasks using a haptic joystick. The data collected during the tests can be used in learning algorithms, so the robot will be able to move, balance and walk on its own [10].

Currently the humanoid project faces the problem of maintaining balance under perturbations. This means that the balance algorithms need to be studied better. To do this the induced perturbations have to be repeatable.

Also, there is the possibility of extending this technique to humans (to study pathologies, assess the recovery of a patient, etc.).

1.2 Problem Description and Solution Overview

For both robot and human, the chosen method to study postural equilibrium is done by applying a controlled and repeatable balance disrupting stimulus and measure responses, such as ground reaction forces, accelerations, body movements and energy expended. All this data will be analysed to obtain a model of the behaviour or set of reactions that counteracted the disturbance.

Measuring all these quantities can be achieved with a motion capturing system, a force plate, inertial systems and electromyography. For robots, this last technique can be partially replicated by measuring servo current consumption, for example.

For the success of this approach, the subject responses have to be precisely measured, and the stimuli must be well known, controlled and repeatable. Taking this in consideration, the first system to test the PHUA's balance will be a prototype of the human scaled one, which needs to be a servo-controlled system that can be applied in a biometric and motion capturing environment. Therefore, it must have a low visual influence, assure the subject safety, allow calibrations and adjustments as automatically as possible.

So, the system to test PHUA's robot has to be designed taking all this in consideration. Advancing to a possible solution, the type of stimulus to be used in the tests, can be traction forces (without excluding other possibilities) applied to the subject's body. Figure 1.2 visually describes this concept; the ropes would pull the subject in the posterior, anterior, right and left directions.

Servo motors would provide the tension needed, and backed up by force sensors, integrated on the ropes, to assure the stimulus data is stored and is within the desired range. On a final phase of the project, all other data will be gathered with the sensing infrastructure available in the motion capturing lab at ESSUA and by PHUA's own sensors.

As for the stimulus itself, one or two servos would pull, and the others would provide as much rope needed to maintain a certain tension, so that the subject doesn't become at risk of falling. Making the system like this, leaves room to use other types of stimuli that can be used alongside with this one, to interfere with the visual and vestibular systems.

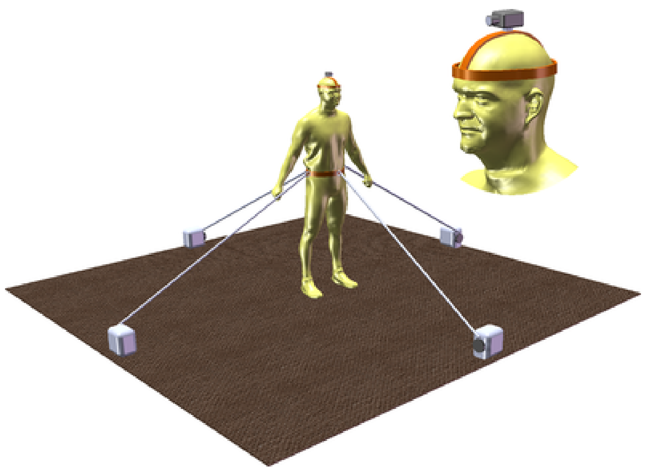


Figure 1.2: Possible solution for the balance disruption system

1.3 Objectives

The main objectives of this dissertation are the following:

- Conceive a mechanical set-up to test the concept.
- Conceive and test stimuli patterns to the mechanical system in order to assess the reliability of the concept.

1.4 Thesis Guide

The stages of this work are presented in the following chapters, organized as follows:

- Chapter 1 gave the context and motivation for the current work as well as a description of the problem, an overview of the solution and a presentation of the main objectives.
- Chapter 2 compiles relevant information for this project. It briefly presents human balance, balance testing systems and force control with servo motors.
- Chapter 3 details the hardware and software used in this work. More specifically it presents the characteristics of: PHUA; the servos and the interface to communicate with them; the sensors and their acquisition unit; the software platform.
- Chapter 4 documents experiments done with an inverted pendulum. It presents strategies to achieve a state of balance.
- Chapter 5 documents experiments done with the inverted pendulum. More specifically, some stimuli are applied to the pendulum and the results are presented and discussed.
- Chapter 6 discusses the main conclusions of this dissertation work, and presents some proposals for future works.

Chapter 2

Background Information and Related Work

This chapter of the dissertation discusses relevant background information for the project. It presents a brief introduction to human balance to give a better perspective on stimuli types. It also presents related works about balance assessment machines to learn what has been done and how it was done. And finally it presents a very brief introduction to force control with servos motors just to give a small idea on how to control servos based on force readings.

2.1 Brief Introduction to Human Balance

Human balance consists of maintaining the line of gravity (vertical line from centre of mass) of one's body within the base support and doing so with minimal postural sway [11]. Both Humans and humanoid robots only have two legs, this makes their upright posture mechanically unstable. The ability to maintain an upright posture is achieved with the assistance of three sensing systems. Perturbations applied to vestibular[12], visual[13] and proprioceptive[14] systems have provoked body sway, thus proving that these are the three input systems that contribute to postural control.

The vestibular system, located in the inner ear, is responsible for determining the orientation of the body with respect to gravity, self generated movements and external perturbations. It can produce many involuntary reactions to make compensatory movements or postural adjustments[15]. Some of these reactions control head and eye movement.

Vision also gives important information for stabilization. When we move, the images of objects in the environment move over our retina. Since the vestibular system measures accelerations it can't detect slow movements or movements at constant velocity. So our brain assumes that the visual environment is static. Consequently, when the visual world moves over our retina we have the perception that we move ourselves. This problem is solved by analysing all the sensory information; one sensor can be easily deceived, but to deceive all of them is much harder[13].

The proprioceptive system gives information about the relative position of the body parts and the effort being made to maintain posture. Some muscles, tendons and joint capsules are responsible for sensing all that data. This information is critical to employ the optimal strategy to maintain balance[16].

2.2 Balance Testing Systems

There are different systems to test the equilibrium of a human, and they can be split in two categories: static and dynamic. A static test measures some quantities, such as weight distribution and Center of Pressure (CoP) with a non moving subject. The dynamic analysis applies a balance disrupting stimulus, most commonly below the feet, such as a translation or rotation, or those two combined. Another type of dynamic assessment can be done using pull forces or a combination of these two dynamic tests. The measured quantities are CoP, weight distribution, accelerations and muscle electrical signals. The dynamic approach is more complete, giving more data to be analysed by experts in biomechanics and equilibrium.

2.2.1 Stimulus Applied Below the Feet

An example of a static and dynamic balance evaluating machine is IsiMove, seen in figure 2.1.

> [3] "It allows the evaluation of the static and dynamic balance of a human placed on a force plate. IsiMove is a robotic platform open kinematic with four degrees of freedom: anteroposterior tilt, mediolateral tilt, vertical rotation, and horizontal translation. It is capable of measuring the displacement of the center of pressure over time, with a resolution of 0.1 mm for each foot and support a human of about 120 kg. IsiMove can generate various types of balance perturbations based on parameters such as direction, amplitude, frequency and shape."

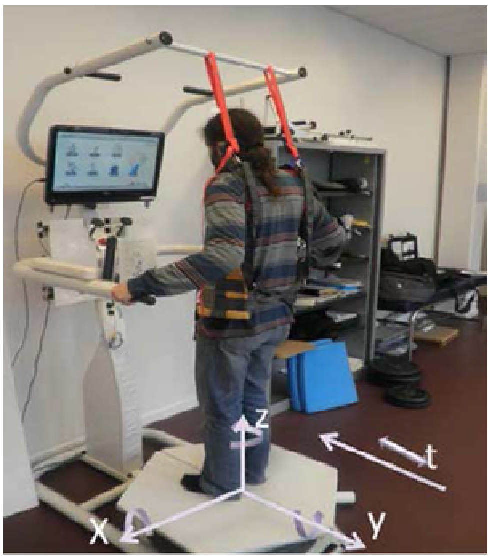


Figure 2.1: IsiMove platform and its axis of motion[3].

Another example of a similar system is IsiSkate, depicted in figure 2.2.

[4] "It is specially designed to reproduce the perturbation of a public transportation in a laboratory environment. It is equipped with force and inertial sensors intended for the evaluation of human posture for static and dynamic equilibrium analysis of a human subject standing on it. This platform is able to disturb the balance of a human weighing 120 kg with an acceleration of 3 m/s^2 , which is above the safe limit 1.9 m/s^2 in public transportation. The platform is capable of measuring in real time the displacement of the center of pressure independently for the left and right foot. With its integrated inertial sensor, the platform is also capable of estimating the displacement of center of mass of the human subject as well as the motion of the ankle, knee, and hip joint."

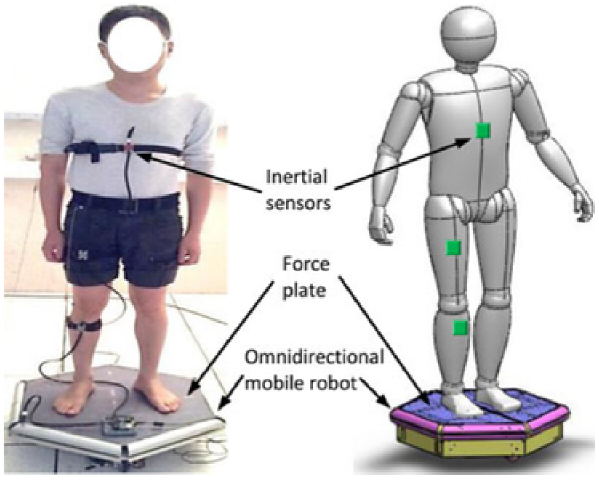


Figure 2.2: IsiSkate platform for balance analysis, user standing on omnidirectional mobile robot with three inertial sensors attached to his body[4].

2.2.2 Pull Force

Another type of balance recovery assessment, is the stress test, introduced by Wolfson et al.[17]. This test consists of postural perturbations that solicit balance recovery reactions measured during a normal upright position, using a simple pulley weight system that displaces the centre of gravity behind the base of support. More precisely, subjects have to withstand a series of posterior perturbations impulses that are applied at the level of the subject's waist using three different perturbation intensities (i.e., 1.5%, 3% and 4% of the body mass)[18]. To administer this test, subjects must be placed 2 m from the pulley structure, with their upper limbs parallel to the body and the feet 10 to 15 cm apart. There are two points where the perturbation can be applied, on the shoulder and on the waist (at the level of the iliac crests)[5], as shown in figure 2.3 and 2.4, respectively.

2.2.3 Pull Force and Stimulus Applied Below the Feet

The last method presented is the most similar to the one that is being designed [6]. This study uses a contact force as an external perturbation. The stimulus is created by applying a force controlled pull on the subject's body in the anteroposterior direction. It measures centre of mass (CoM) and centre of pressure (CoP) excursions to sinusoidal pull stimuli at different frequencies and with various force magnitudes. The setup allows the application of a pull on the subject's body in the sagittal plane and thereby exert a defined torque about their ankle joints. The stimuli is applied via a body harness by means of two force controlled rope winches (servo motors) under computer control, as shown schematically in figure 2.5. Sinusoidal stimulus profiles are applied at five different frequencies (0.05, 0.1, 0.2, 0.4 and 0.8 Hz) and three different amplitudes of torque about the ankle joint (± 1, ± 4 and ± 16 N m). During these stimuli, subjects are standing upright, heels 5 cm apart, on a motion platform. The platform rests on six 'legs', each incorporating a motor capable of producing a change in the length of the leg (hexapod with Stuart principle). The motors are controlled by a computer to generate platform tilts in the sagittal plane with the axis through subjects' ankle joint during some of the stimuli.

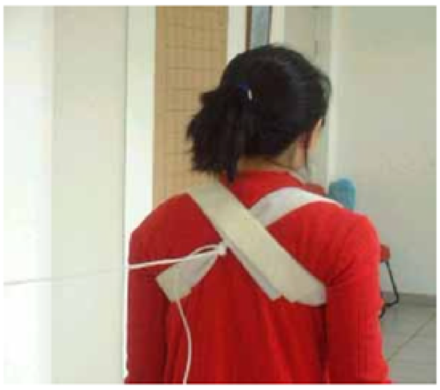


Figure 2.3: Balance disruption applied in the posterior direction at shoulder level[5].

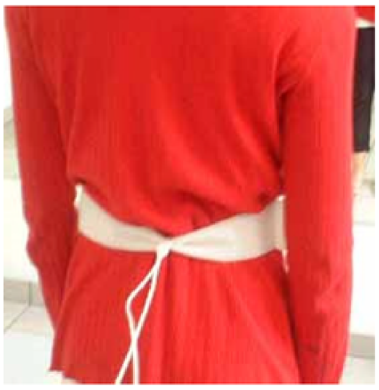


Figure 2.4: Balance disruption applied in the posterior direction at iliac crest level[5].

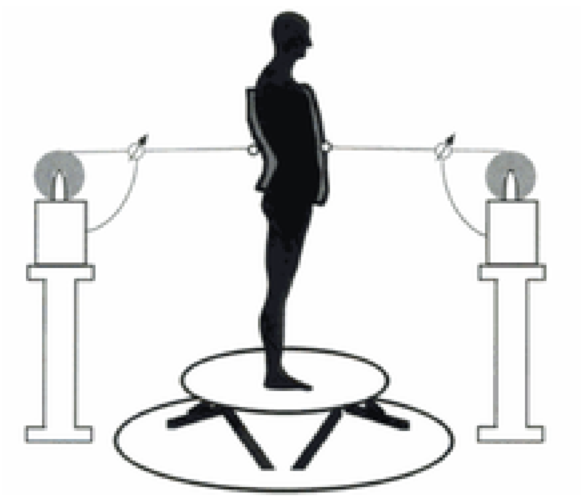


Figure 2.5: Schematic presentation of stimulus conditions. Sinusoidal pull stimuli were applied by means of two force-controlled cable winches via a padded body harness to exert defined torques about the subject's ankle joint in the a-p direction. The subject stood on a motion platform which was either stationary or was tilted in fixed register with spontaneous and stimulus-evoked body excursions[6].

2.3 Force control with servo motors

Force control is applied mainly in industrial manipulators to perform tasks that require force control along a nominal path or velocity.

> [7] “Robot force control involves integration of task goals, trajectory generation, force and position feedback, and modification of the trajectories. It requires understanding contact tasks so that effective strategies and trajectories can be planned and feedback data can be understood. It also requires control so that the robot’s responses will be stable. Finally, it requires filtering and estimation to remove unwanted signals, such as noise and robot motion errors, so that usable feedback information can be obtained. These issues - task analysis, strategy generation, control stabilization, and filtering - must be dealt with together if effective force control systems are to be created.”

A general example of robot force feedback architecture can be seen in figure 2.6.

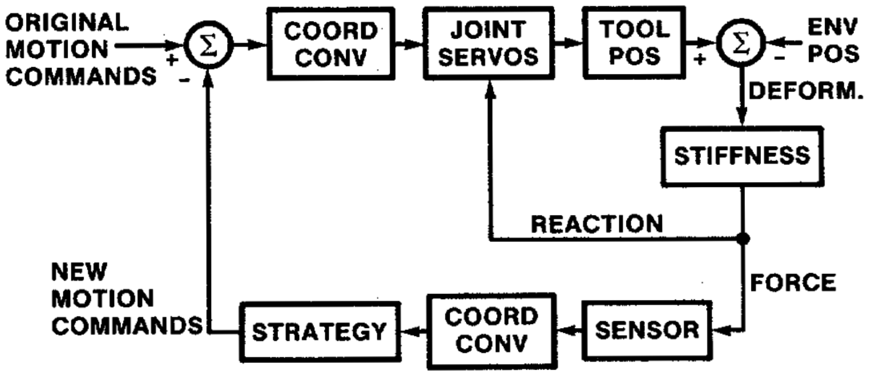


Figure 2.6: Architecture of robot force feedback[7].

Chapter 3

Experimental Infrastructure

In this chapter some specifications about PHUA are shown to complete its presentation started in chapter 1. As suggested before the stimuli are pull forces and they need to be measured so they can be controlled. To accomplish this several servo motors need to be connected in a bus with the possibility of force or torque control. To measure the stimuli tensile force sensors are needed with a measurement range close to the stimuli magnitude. Finally a software has to communicate with the motors and sensors interfaces. All these devices and software will be presented in this chapter.

3.1 University of Aveiro Humanoid Project

The platform has 27 DOF, visible in figure 3.1. HITEC® digital and analogue servomotors, combined with elastic elements, actuate 25 of the joints which are arranged as in figure 3.1 with the servomotors assembled as in table 3.1. The robot uses a belt and pulley adjustable transmission system on the legs and torso, providing torque to the joints with higher demand. The platform is also equipped with a set of 8 force sensors, 4 on each foot, from which weight distribution can be extracted. When assembled, it weights about 6 kg and has an approximated height of 65 cm [8].

Table 3.2 details the robot's joint anthropometric ranges and the total angular range, measured from the home position (in degrees), defined as the robot with all limbs at full stretch downwards, with the torso upright and a zero tilting in the head.

A simulation of PHUA on V-REP has shown that it's hip can move +/-50 mm in both sagittal and lateral direction and -50 mm vertically[10].

Table 3.1: PHUA platform's degrees-of-freedom and installed servomotors.

Joint	Number of DOF's	Servomotor
Toe	1(×2)	(none)
Ankle	2(×2)	HSR-5980SG
Knee	1(×2)	HSR-5980SG
Hip	3(×2)	HSR-5980SG

Trunk	3	HSR-5980SG(×2)
		HSR-8498SG(×1)

Shoulder	3(×2)	HSR-5980SG HSR-5498SG HS-82MG
Elbow	1(×2)	HSR-5498SG
Neck	2	HSR-5498SG HS-5056MG
Total	27	–

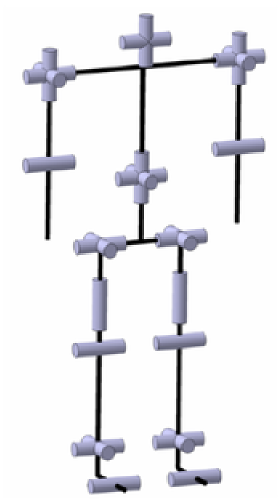


Figure 3.1: PHUA kinematic chains [8].

Table 3.2: PHUA robot degrees-of-freedom range limitation.

DOF	Minimum (°)	Maximum (°)	Total Range (°)
Head Tilt	0	10	10
Shoulder Flexion/Extension	-45	135	180
Shoulder Abduction/Adduction	0	180	180
Elbow Flexion/Extension	0	120	120
Torso Rotation	-90	90	180
Torso Flexion/Extension	-15	90	105
Torso Lateral Flexion/Extension	-90	90	180
Ankle Inversion/Eversion	-45	30	75
Ankle Extension/Flexion	-20	40	60
Knee Extension/Flexion	0	130	130
Hip Adduction/Abduction	-40	45	85
Hip Extention/Flexion	-30	120	150

3.2 Dynamixel RX-64

The servo motor used for the initial phase of the system's prototype is shown in figure 3.2. It was already present in the lab, ready to be used and also it has enough power to apply stimuli to the robot; some specifications can be seen in table 3.3. With a supplied voltage of 15 V the servos can produce a stall torque of approximately 632 N cm which is more than enough to lift PHUA's structure, if the radius of the pulling mechanism doesn't exceed 10 cm and minding PHUA's weight.

Figure 3.2: Dynamixel RX-64 servo

To communicate with the servos, Main Controller (3.3) has to send an Instruction Packet and receive a Status Packet, the structure of these packets can be seen in figure 3.3 and 3.4, respectively. Each box represents a byte; for both Packets, the four first bytes are the same. The first and second are the starting bytes, the third is the identification number of the servo, and the fourth is the length of the packet. For the Instruction Packet, the sixth until the last but one bytes, are the address of the parameters which the controller wants to read or write accordingly with the fifth byte which is the instruction. There are seven types of instructions and they can be seen in table 3.4. The last byte is used to check if the packet is damaged during communication. For the remaining bytes of Status Packet, the fifth byte displays if an error has occurred during the operation of the servo and which error it is. The sixth byte until the last but one, returns the value of the parameter requested by the Instruction. The last byte has the same usage as the checksum byte of the other packet.

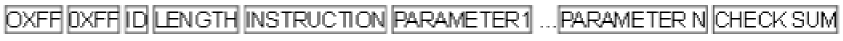


Figure 3.3: Structure of Instruction Packet sent to Dynamixel motors.

To view more detailed information about RX-64 and how to use it, please refer to Dynamixel RX-64 Manual[19].

Yet RX-64 manual doesn't cover some important technical features, which are pointed out by E. Tira-Thompson [20]. In this article it's mentioned that Present Speed and Present Load parameters are only updated every 130 ms, furthermore these parameters have some error.

From the manufacturer website it's possible to gather more important information about the motors. Load is only useful to discern which direction the torque is being applied. So, it will not be possible to use the Present Load parameter to measure the torque being outputted by the servos. Also it's advised that for stable motions, maximum

Table 3.3: Dynamixel RX-64 specifications.

Weight(g)	125
Dimension(mm)	40.2 x 61.1 x 41.0
Gear Reduction Ratio	1/200
Applied Voltage(V)	at 15 | at 18
Final Reduction Stopping Torque (kgf.cm)	64.4 | 77.2
Speed (s/60°)	0.188 | 0.157
Resolution (°)	0.29
Running Degree	300°, Endless Turn
Voltage (V)	12 ∼ 21 (Recommended: 18)
Max Current (mA)	1200
Running Temperature (°C)	-5 ∼ +85
Command Signal	Digital Packet
Protocol	RS485 Asynchronous Serial Communication (8bit,1stop, No Parity)
Link (Physical)	RS485 Multi Drop Bus
ID	254 ID (0 ∼ 253)
Communication Speed	7343bps ∼ 1 Mbps
Sensing & Measuring	Position, Temperature, Load, Input Voltage, etc.
Material Quality	Full Metal Gear, Engineering Plastic Body
Motor	Maxon RE-MAX
Standby Current	50

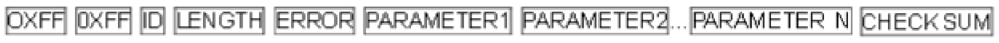


Figure 3.4: Structure of Status Packet sent by Dynamixel motors.

load can't be higher than 1/5 of stall torque.

An important aspect of these servos is their ability to be set to motion with either position or torque control; the latter assumes free spin mode is selected, where the engine is free to turn more than 360 degrees, whereas position control limits the motion to 300 degrees, as is depicted in figure 3.5.

Table 3.4: Instruction for Dynamixel.

Name	Function	Number of Parameters
PING	No execution. It is used when controller is ready to receive Status Packet	0
READ DATA	This command reads data from RX-64	2
WRITE DATA	This command writes data to RX-64	2 or more
REG WRITE	It is similar to WRITE DATA, but it remains in the standby state without being executed until the ACTION command arrives	2 or more
ACTION	This command initiates motions registered with REG WRITE	0
RESET	This command restores the state of RX-64 to the factory default setting	0
SYNC WRITE	This command is used to control several RX-64 simultaneously at a time	4 or more

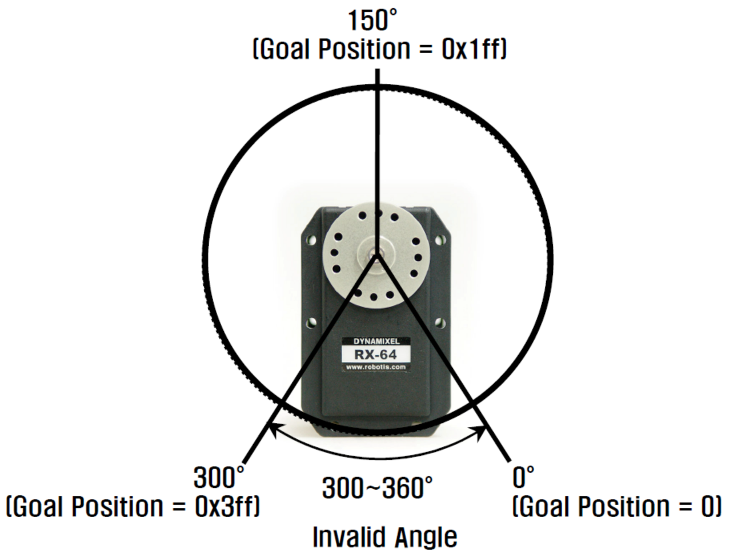


Figure 3.5: Position control angle restrictions of RX-64 servomotor.

3.3 DETI Dynamixel Interface

As encouraged by ROBOTIS [19] the main controller was custom made, instead of being bought, by the Department of Electronics, Telecommunications and Informatics (DETI), to use the servos without restrictions on Linux operating system. It supports RS485 Universal asynchronous receiver/transmitter (UART) because RX-64 uses a Multi-Drop Link method which connects several motors to a Node by using Half Duplex UART.

It has a FTDI FT232RQ USB to UART interface, responsible for the communication between the computer and PIC32MX795F. The speed of this communication is by default set at 230400 bps. The microprocessor is responsible for arranging the messages accordingly to the Dynamixel protocol (see figures 3.3 and 3.4), so that the servos can read the instruction packets and the computer can view the requested information sent by the Dynamixels in the return packet. A MAX3362EKA#T allows the RS485 communication between the microcontroller and the servos, with a speed of 1000000 bps. The board is supplied with 5 V, which is reduced to 3.3 V by a MCP1703 linear regulator.

This interface was built to control the Dynamixel servos series AX that consume less current then the RX series, so the interface was able to control and supply power to the AX motors. But, when building the interface, the designers made it capable of controlling other servo series of this brand just not powering them. To use only one power supply a 7805CT voltage regulator was connected to the interface, as seen in figure 3.6. The schematic for the voltage regulator board can be seen in figure 3.7. Figure 3.8 describes how to connect the power supply to the interface and the data cable that connects to the servo, the connection to the computer is done with an USB-A to USB-B cable shown in figure 3.6 (grey cable).

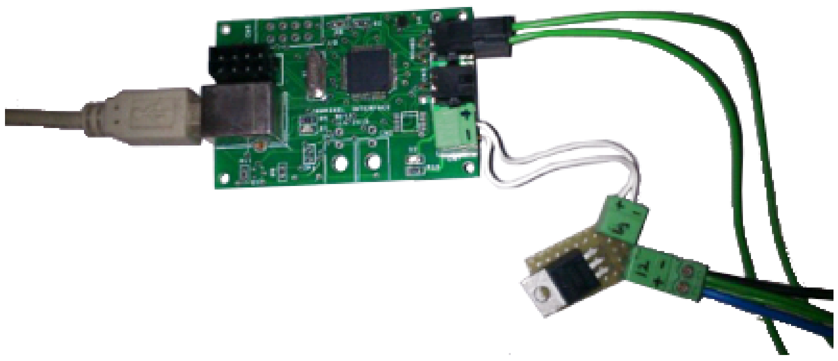


Figure 3.6: Deti Dynamixel Interface and voltage regulator.

To protect the DETI Dynamixel Interface a case was designed using the modelling tool Catia V5, shown in figure 3.9 and then a 3-D printing machine printed the case in thermoplastic (polylactic acid or PLA). The protection case is assembled with four screws that are inserted in the bottom part, figure 3.10 depicts the plastic case.

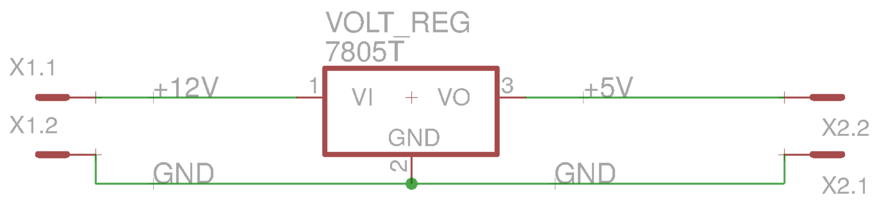


Figure 3.7: Voltage regulator board schematic.

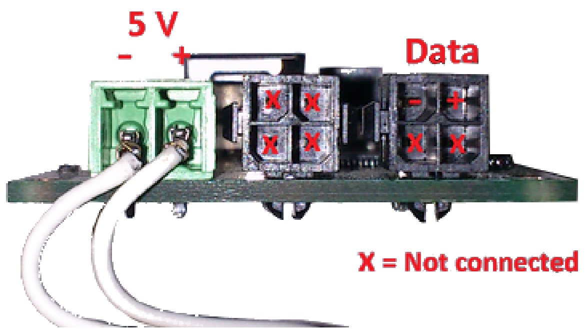


Figure 3.8: Deti Dynamixel Interface connections used.

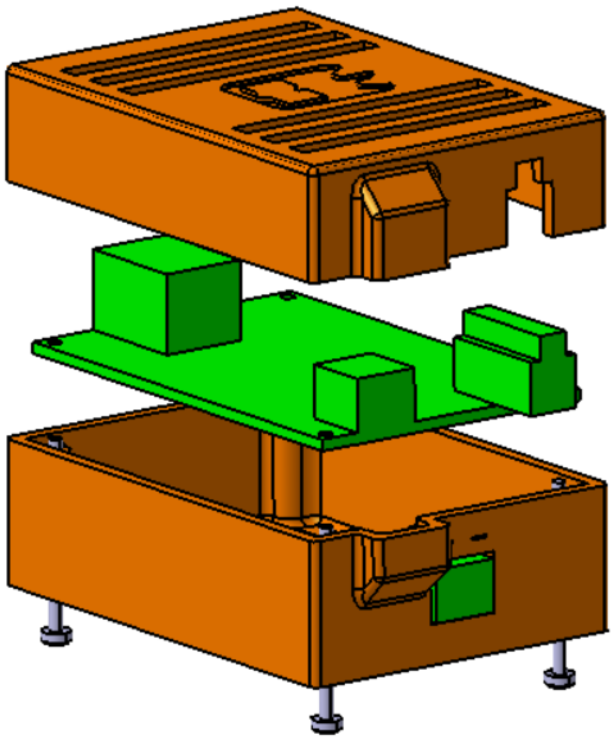


Figure 3.9: CAD assembly of the Deti Dynamixel Interface protection case.

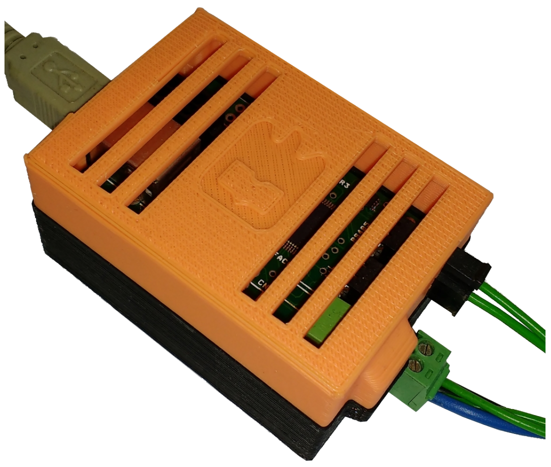
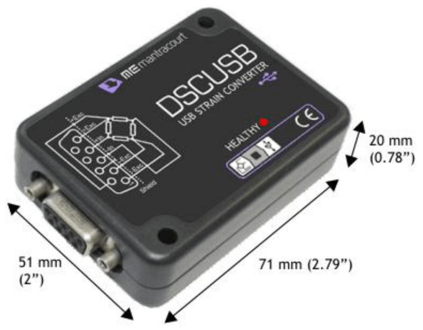


Figure 3.10: Deti Dynamixel Interface protection case.

3.4 Load Sensors

The contextualization for this section is that a cheap, easy and reliable tensile force measuring system was needed. The measurements need to be a digital signal readable by the software. Tension cells have very good construction and signal quality but their prices are too high and still require an analogue to digital signal converter, shown in figure 3.11. Strain gauges, shown in figure 3.12 are cheaper than tension cells but require the dimensioning of an appropriate Wheatstone bridge, analogue to digital conversion and a rigid material to measure strain, which is a very time costly solution. After some more thought on a possible solution it was realised that some spare load cells were available in the Laboratory of Automation an Robotics (LAR) at University of Aveiro, and with a simple mechanism (3.4.1) tensile force could be measured. So, this is why load cells are being presented to measure tensile force.

(a) APPLIED Measurements DDE series tension cell.



(b) APPLIED Measurements DSCUSB Digitiser module.

Figure 3.11: Tension cell.

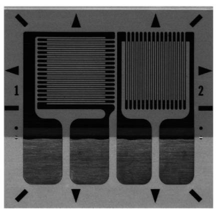


Figure 3.12: Micro Measurements Strain Gauge.

The load cells are Interface's LBS-5 and LBS-10, with a weight limit of 5 and 10 pounds of force respectively and their specifications can be seen in table 3.5.

To read the load cell's signal properly, a signal conditioning circuit and software that were designed by Estrelinha [21] were used. The signal conditioning board has four amplifiers (it can connect to four load cells) each with: i) a gain resistor of 453 Ω with 1% error; ii) a Texas Instruments®'s instrumentation amplifier INA 122; iii) a capacitor to ensure a stable power supply; iv) a first order low-pass filter to prevent aliasing that uses a 1k43 Ω resistor and a 220 nF capacitor. The low pass filter as a cut-off frequency of 506.15 Hz for an expected sample rate of 1k Hz. The circuit schematics can be seen in figure 3.13. This work was updated by Serra [9], and the circuit components were rearranged to fit in a smaller printed circuit board that can be connected to an Arduino™ nano, as seen in figure 3.15; in this image it can also be seen the instructions to connect the load sensors to the board.

The Arduino is programmed to read the analogue outputs and send the values to the serial port with the order seen in figure 3.15, at a speed of 115200 bps. The baudrate was increased to 230400 bps, after some search in forums one person stated the Arduinos's ATmega328 was capable of higher communication speed. Some tests confirmed this capability which isn't stated in the official specifications. The Arduino code sent the analogue readings as doubles this was changed to integers to improve communication speed. On the computer side, a buffer collects this data and stores it in a string shown in figure 3.14. The buffer keeps storing data until the byte EOF arrives. Note that the messages starts with sen1, ends with EOF and the spaces count as a byte, so a complete message has at least eight bytes, since the sensor value can have more than one byte. The software was modified to read values only from one Arduino with four load cells and the way that the sensor data was extracted from the string buffer was also modified, using the function `stringstream`, from the C++ library sstream, the four values are transferred into four doubles. With these modifications the sensor values are being read at approximately 900 Hz.

Table 3.5: LBS-5 and LBS-10's characteristics [1].

Characteristics	LBS-5	LBS-10
Nominal Load (lbs)	5	10
(N)	22.2	44.4
Excitation, nom (V)	5	5
max (V)	7	7
Safe Overload	150%	150%
Non-linearity-% FS	0.25	0.5
Compensated Temp. Range (°F)	60 - 160	60 - 160
Bridge Resistance (Ω)	350	350
Rated Output (mV/V)	2.0	2.0

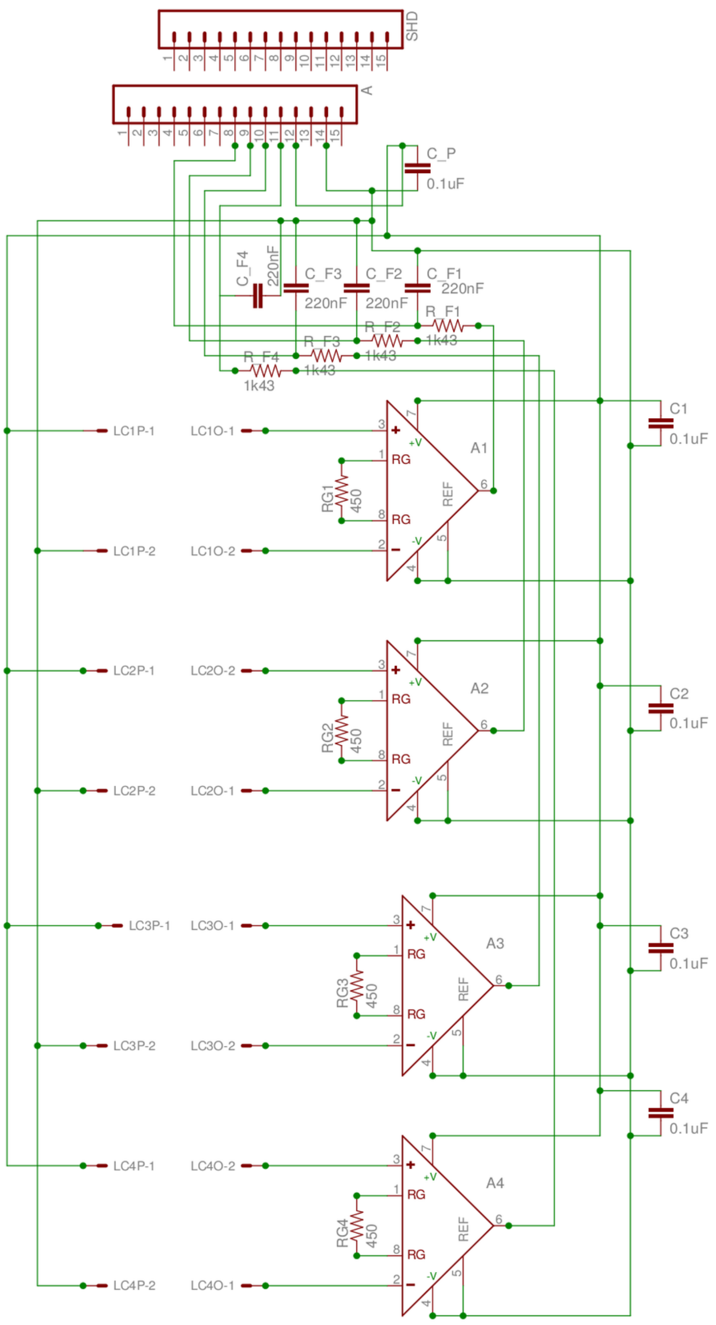
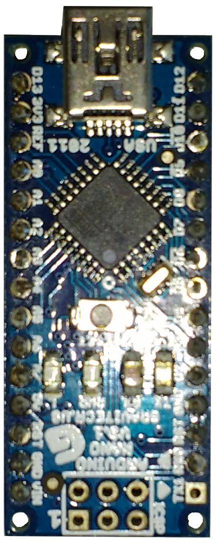
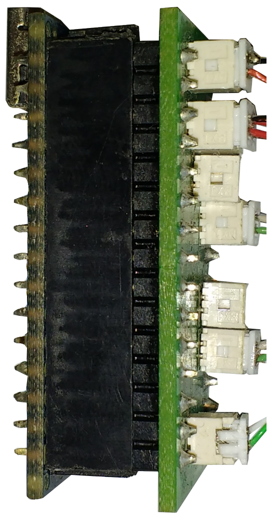


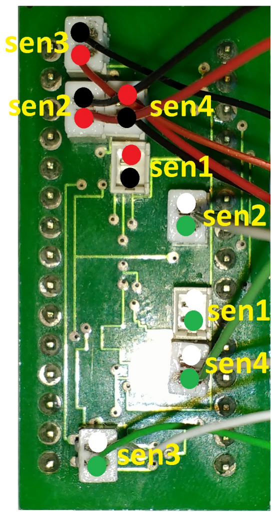
Figure 3.13: Signal conditioning circuit updated by Serra [9].

Sen1		Sen2		Sen3		Sen4		EOF

Figure 3.14: Message sent by the Arduino with the sensors readings.


(a) Arduino Nano.


(b) Signal conditioning board connected to Arduino Nano.

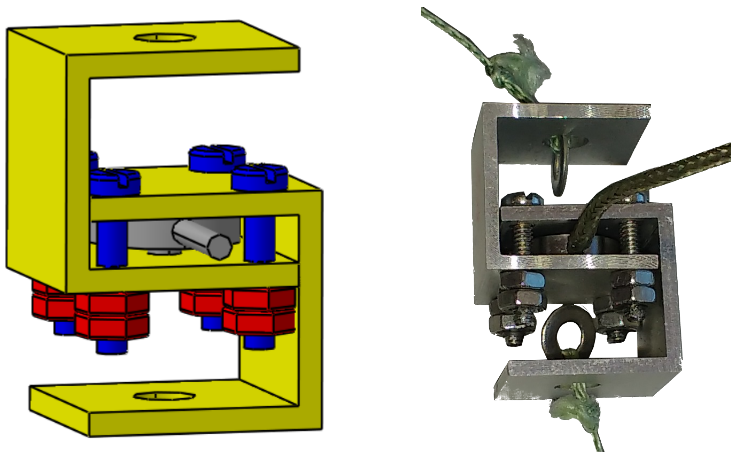


(c) Load cells connections on the Signal conditioning board.

Figure 3.15: Arduino, signal conditioning board and load cells connections.

3.4.1 Tension Compression Mechanism

Figure 3.16 shows the simple mechanism that allows the measurement of tensile force using load cells. It's made with two pieces of 14 mm U shape 15×15×1.5 aluminium bar (see appendix A), eight M2 stainless steel nuts and four M2×10 stainless steel bolts. Two nuts are used to prevent them from unscrewing and not compress the cell.



(a) CAD of the Tension Compression Mechanism.

(b) Tension Compression Mechanism Assembled.

Figure 3.16: Tension Compression Mechanism.

3.4.2 Cell Calibration

The main objective of using load cells is to know the force exerted in the cable. As proven by Estrelinha [21], each cell needs its own calibration curve. The same universal testing machine was used, model AGS-X 10kN by Shimadzu Corporation shown in figure 3.17. Using the machine to tension the load cell mechanism, force measurements were manually stored as well as the analogue values sent by the Arduino. The linear coefficients and their error of determination were extracted from the collected data and the results are shown in table 3.6. The calibration process consist on multiplying the analogue value x by a and adding b giving the tensile force magnitude in Newtons.

Table 3.6: Load cell linear coefficients ($y = a \times x + b$) and their error of determination.

Cell number	Cell Reference	a	b	r^2
2	X281413	0.06	-3.17	0.99
3	X271169	0.03	-1.80	0.99
4	X281448	0.07	0.25	0.99

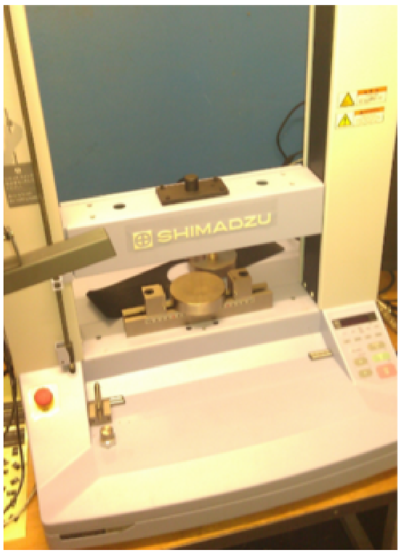


Figure 3.17: AGS-X 10kN by Shimadzu Corporation.

3.5 Software Platform

The software used is Robot Operating System (ROS) with C++ language on a catkin workspace. This platform was chosen because it is open source, uses the same programming language as Dynamixel and it allows the use of previous work done in LAR.

The package `dyna_comm` was created to carry out the main operations which are communicating with the servo motors and acquire data from the load sensors.

The node `dyna_comm` is responsible for communicating with the motors and calculate the required action to be sent to the Dynamixel. It connects with the servos using DETI libraries which establish the communication protocol with the main controller and have the functions needed to send the necessary commands. This node can publish servo information, commands sent and other values related with the servo control on the topic `/servo_info`; this is used for debugging and to adjust the control parameters. It subscribes to a topic called `/servo_command` to receive user commands and send them to the Dynamixels, this is mainly used to start a motor motion. If this node sends only one instruction to the servos (per ROS loop) its frequency is ∼333 Hz, which is its current frequency; if more messages are sent, the frequency will drop. To use the highest frequency possible with this setup, the only message sent per ROS loop has to be of the type sync write, seen in table 3.4.

Figure 3.18 shows the nodes interactions and figure 3.19 shows, in a simple way, the node's architecture. The ROS tool `rqt_plot` is an option to see in real time the sensor values and some servo information. The other ROS tool used is `rostopic` and its function `pub` to publish commands that the node `dyna_comm` will interpret.

The node `dyna_pressure_cells` is responsible for gathering a complete message sent by the force acquisition unit and then calibrate those values as described in 3.4.2. After these calculations are done, the node publishes the tensile force measured by the sensors in the topic `/values` with a frequency of 900 Hz. This node doesn't subscribe to a topic, only publishes on one.

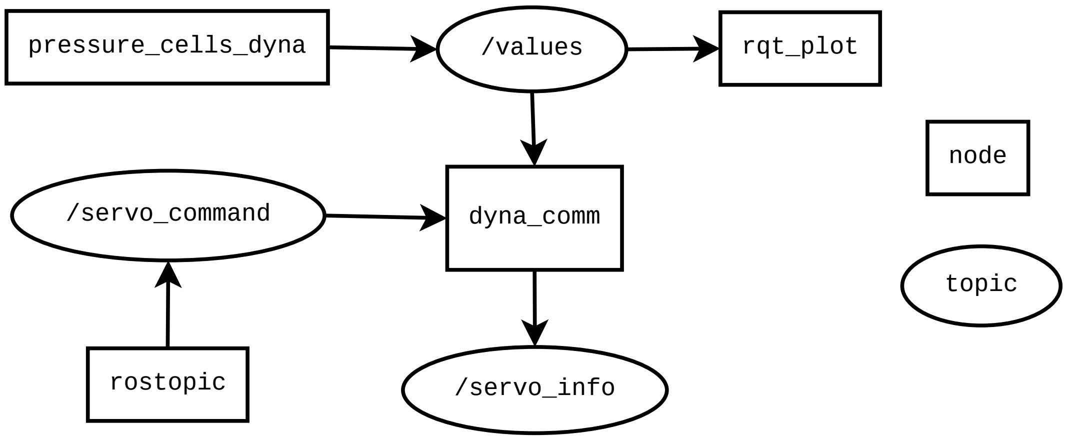


Figure 3.18: ROS Nodes interactions and Subscribed/Published topics.

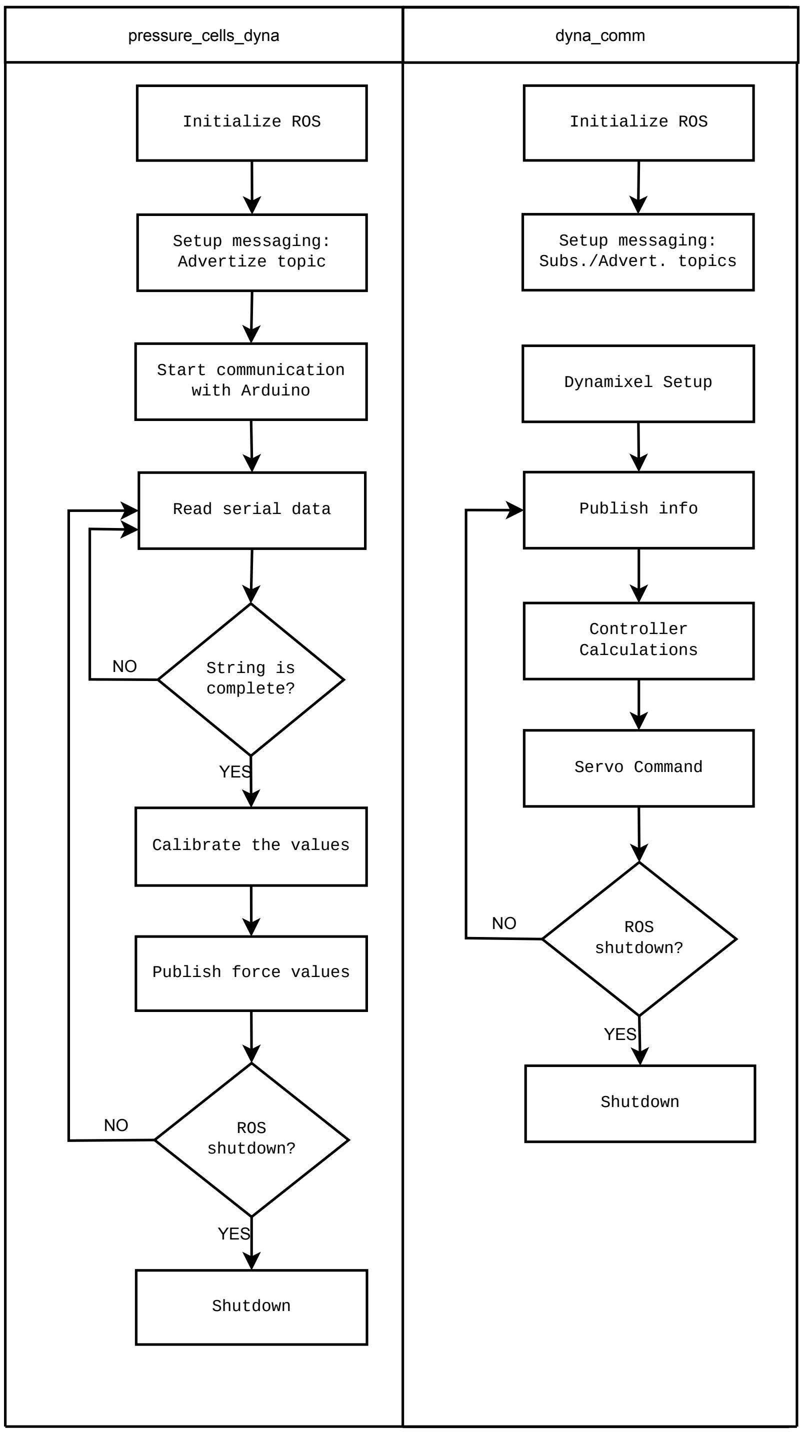


Figure 3.19: Simplified scheme of the `pressure_cells_dyna` and `dyna_comm` ROS nodes.

Chapter 4

Experiments with an Inverted Pendulum

In this chapter the concept first test will be presented. Using the hardware and software described in chapter 3 along with a mechanical structure, an inverted pendulum. An objective of this step is to achieve the initial conditions necessary for the application of stimuli. These initial conditions need to be achieved automatically by the system; repeatable and customizable. Another objective is to find controlling problems inherent to this concept and to the physical structure.

4.1 Inverted Pendulum

An inverted pendulum, figure 4.1, was chosen because it's simple system that only has one degree of freedom, this will help building the control necessary to keep the ropes similarly tensioned with a customizable magnitude. The ropes have to be tensioned so that the applied stimulus can be replicated more than once. The controller will only mind the sensors information, yet it is known that if the pendulum isn't in the upright position the ropes won't be similarly tensioned, the downward force will be mostly absorbed by the rigid pole and the remaining forces will pull one rope and relieve the other. This effect is aggravated as the pendulum gets closer to the horizontal. To counteract this effect elastics are used to keep the upright position and also to emulate the reaction of attempting to remain upright. The weigh is used to emulate inertia.

To tension the ropes a sheave was designed, see appendix B, taking in consideration that the RX-64 servos only allow 300° rotation(in position control mode) and that PHUA's maximum range in the four planar directions is 50 mm the sheave's inner radius is 25 mm which provides a linear translation of about 131 mm for 300°, keeping in mind that only half of that translation can be used since the servos have to rotate clockwise (CW) and counter clockwise (CCW). This radius can't be much higher because it can hinder on it's base or other objects that in the future can be near the servo.

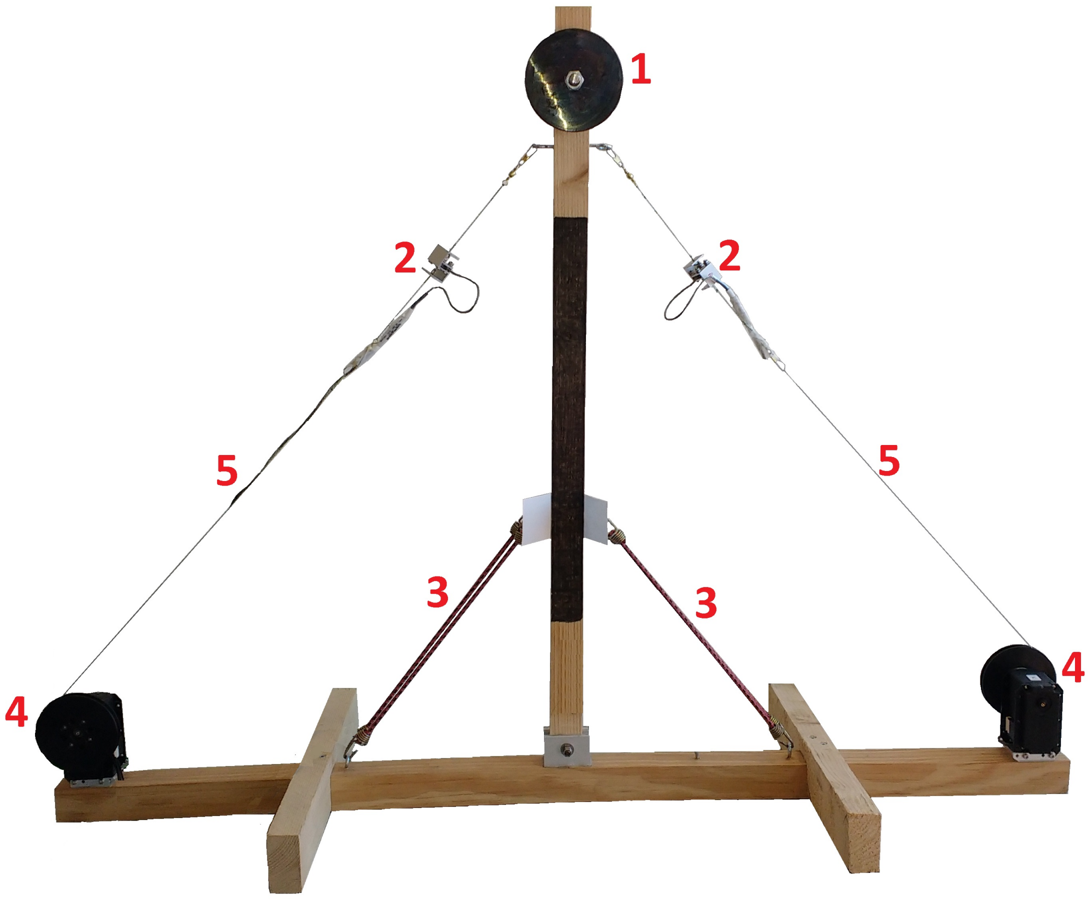


Figure 4.1: Inverted pendulum. 1 - Weight; 2 - Tensile force sensors; 3 - Elastic units; 4 - Servo motor and sheave; 5 - Traction ropes.

4.1.1 Control with Dynamixel RX-64

With these servos there are two types of control that can be designed, position control or torque control. Even though the sheave was designed to produce 131 mm of linear translation this is not enough because of extreme situations that will require more rope than that. An example is depicted in figure 4.2 this situation needs a translation of 300 mm.

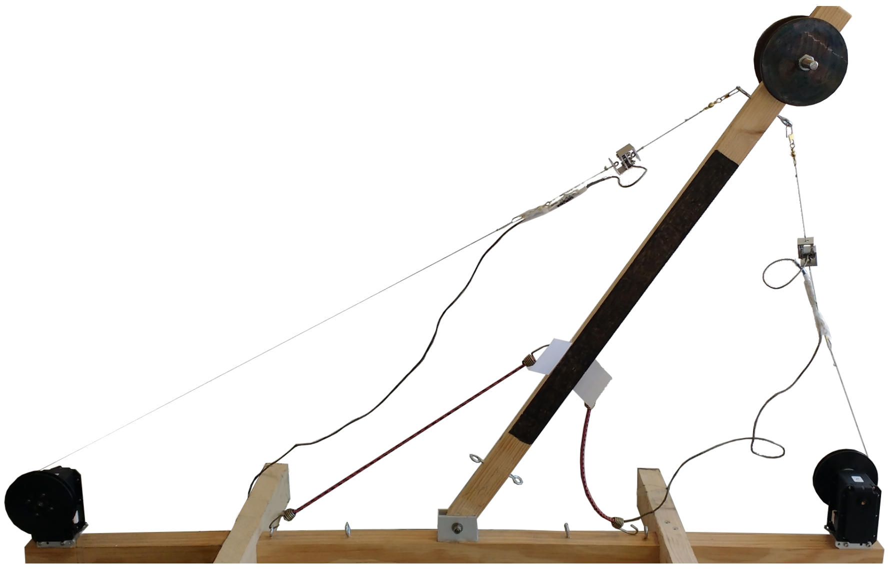


Figure 4.2: Pendulum at an extreme position.

Torque control was explored: for this, the servos have to be programmed to what Dynamixel calls “wheel mode”. To set the servos to this mode all it needs to be done is to define as 0 the counter clock-wise and clock-wise angle limits. To make the servo produce torque a value different from 0 or 1024, and between 1 and 2047 needs to be written in the goal speed parameter (this is misleading because speed is not being controlled, only toque is controlled in the “wheel mode”). Values between 1 and 1023 make the servos rotate CCW and values between 1025 and 2047 make the servos rotate CW.

To get familiarised with this mode, a simple experiment was done: it consisted of raising a mass with different torque values. The procedure for this experiment was to fasten a servo and have it raise a cup with some weights while recording tensile force, commanded torque and load parameter. All the results were the same for every different weight lifted so only one graphic is shown in figure 4.3.

From sample 0 to ≈15000 the servo didn’t raise the cup, it started to be able to raise the cup if the commanded torque was equal or higher than 105. The higher the torque the less time the trial takes and also some errors are observed in the load parameter readings as expected. This load parameter was recorded just to ensure that it can’t be used for torque measurement.

With these trials it’s predictable that the controller will need an integral term to increase or decrease the servo torque if the rope’s tensile force is below or above the desired value since the servos can get stalled.

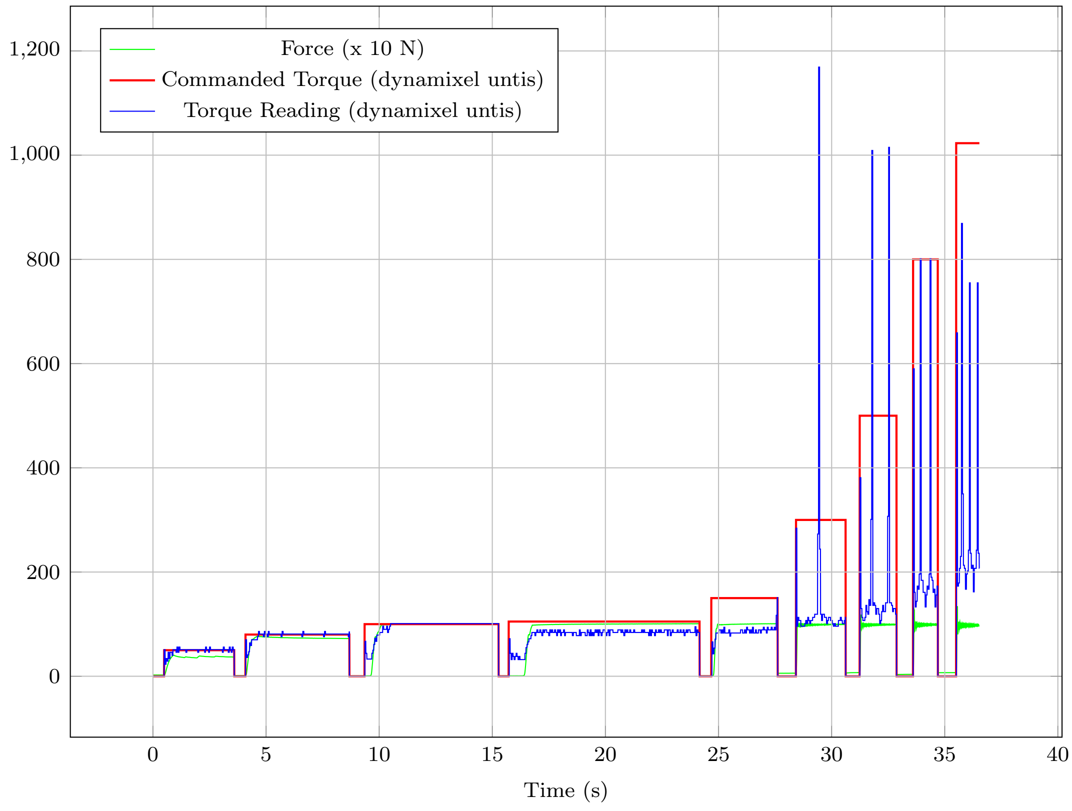


Figure 4.3: Results from the mass raising experiments. Nine different torque values were sent to the servo motor.

4.1.2 Proportional and Integral Controller

A proportional and integral controller is a control loop feedback algorithm that will be used to achieve the pendulum's state of equilibrium. The proportional term will produce an output value that is proportional to the current error between the present sensor reading and the desired tensile force value. The integrative term output will be proportional to the magnitude and duration of the error, meaning that it will be the sum of the errors between the sensor reading and the desired tensile force value.

These terms need to be multiplied by a constant to adjust their effect and produce values compatible with RX-64 units which are integers between 0 and 2047.

The first control algorithm tested can be seen in equation 4.1. Each servo is controlled with it's own equation, the sensor input comes from the sensor that's attached to the servo rope. Like this each servo is controlled independently from the other, but still get influenced by the other servo actions because the ropes connect them physically. $Torque(n)$ is the torque value calculated, at time n, that is going to be sent to the servo; $Tension_{limit}$ is the customizable value for the rope tension; $Sensor(n)$ is the measured tensile force on the servo's rope, at time n; K_P is a constant to convert the proportional term to dynamixel values and to adjust the term's influence on the output ($Torque(n)$); K_I is a constant to convert the integral term to dynamixel values and to adjust the term's influence on the output ($Torque(n)$).

$$Torque(n) = K_P.(Tension_{limit} - Sensor(n)) + K_I.\sum_{i=0}^{n}(Tension_{limit} - Sensor(n_i)) \quad (4.1)$$

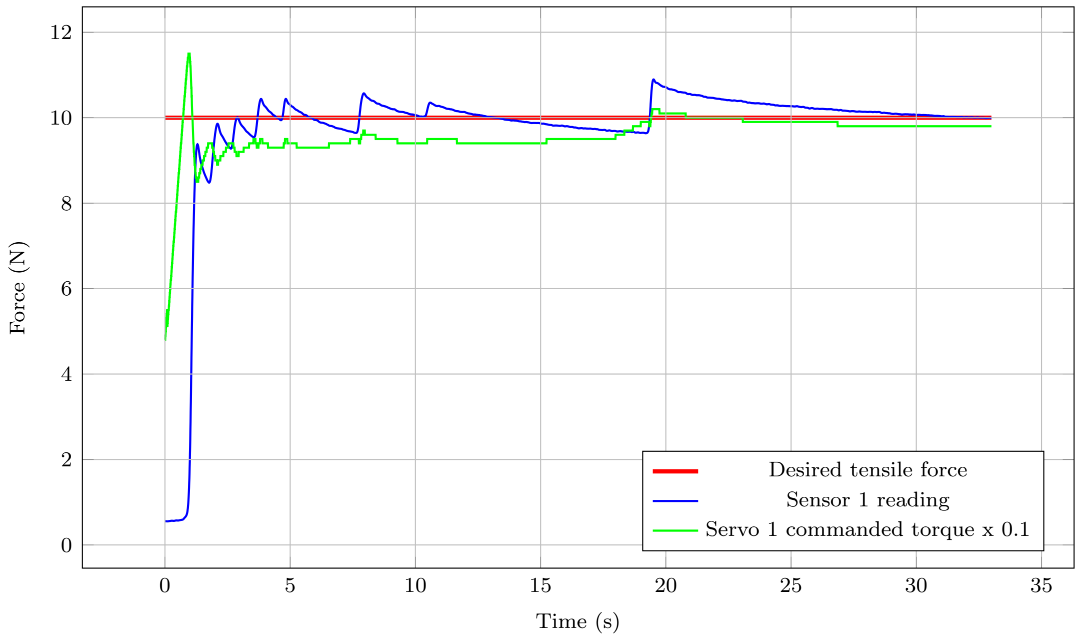
This control caused positive feedback because accumulated too much error and produced positive feedback, making the controller unstable. The rotation direction is based on the output being greater or lesser than zero; this works well when the ropes are both below or above the tension limit, but if one rope is tensioned above the limit and the other is below the control will aggravate this state of unbalance. To understand this effect better, the control equation can be broken down to a simple set of conditions, if the sensor value is below the limit, the servo has to wind the rope to raise the tension and if the sensor value is above the limit the servo has to unwind rope to lower the tension. This means that the control for each servo needs one more term that considers the state of the other servo's sensor.

A new term was designed to oppose the positive feedback that takes in consideration the sensor of the opposite servo and acts like a proportional term. This term is the difference between the servo's rope tensile force ($Sensor(n)$) and the opposite servo's rope tensile force ($Sensor_{opposite}(n)$), at time n; K_P is a constant to convert the term to dynamixel values and to adjust the term's influence on the output ($Torque(n)$). Equation 4.2 shows the controller with the new term.

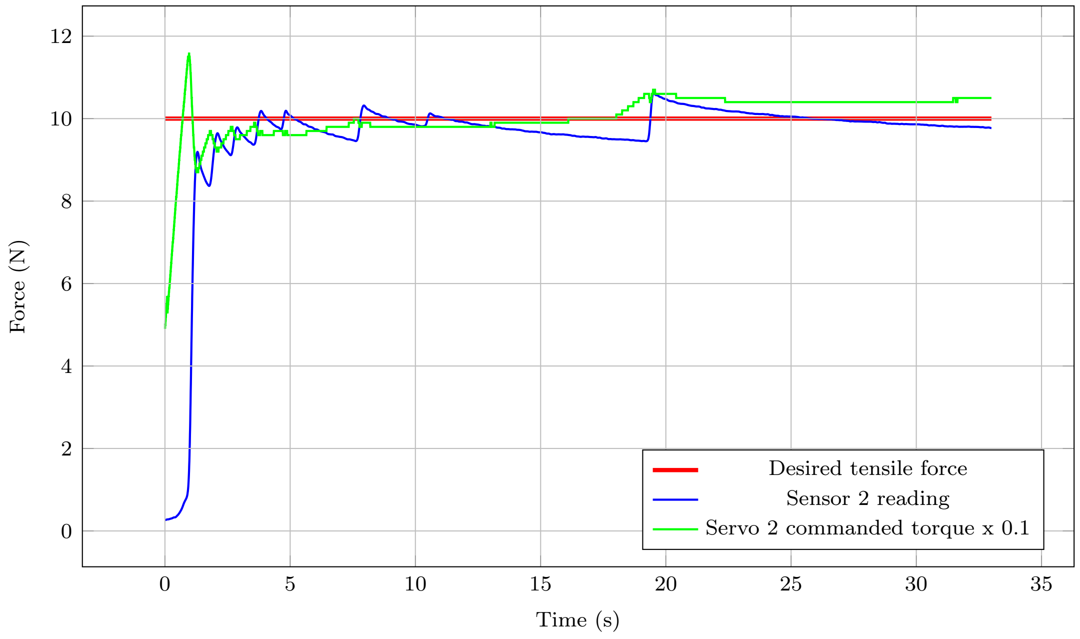
$$\begin{aligned} Torque(n) = {} & K_{P1}.(Tension_{limit} - Sensor(n)) \\ & + K_{P2}.(Sensor(n) - Sensor_{opposite}(n)) \\ & + K_I.\sum_{i=0}^{n_i}(Tension_{limit} - Sensor(n_i)) \end{aligned} \quad (4.2)$$

The result of this controller can be seen on figure 4.4. It's not a smooth control because of the non-linearities of the physical system and because of the resolution and internal friction of the servos. It can be seen, between some samples, that even though the servo torque was being raised by the controller the tension kept decreasing until the torque could overcome internal friction and caused the oscillation in the sensor values. Another problem inherent to this dynamixel control mode is that when the applied torque has to change it's direction the tension on the rope pulls the servo causing instability, this can be seen in figure 4.5.

Because of this problem these servos aren't ideal to control this system. So the servos were switched with the Dynamixel MX-106R because they were available, have more functions, better resolution and can be fitted in the physical system, their specifications can be seen in section 4.2.

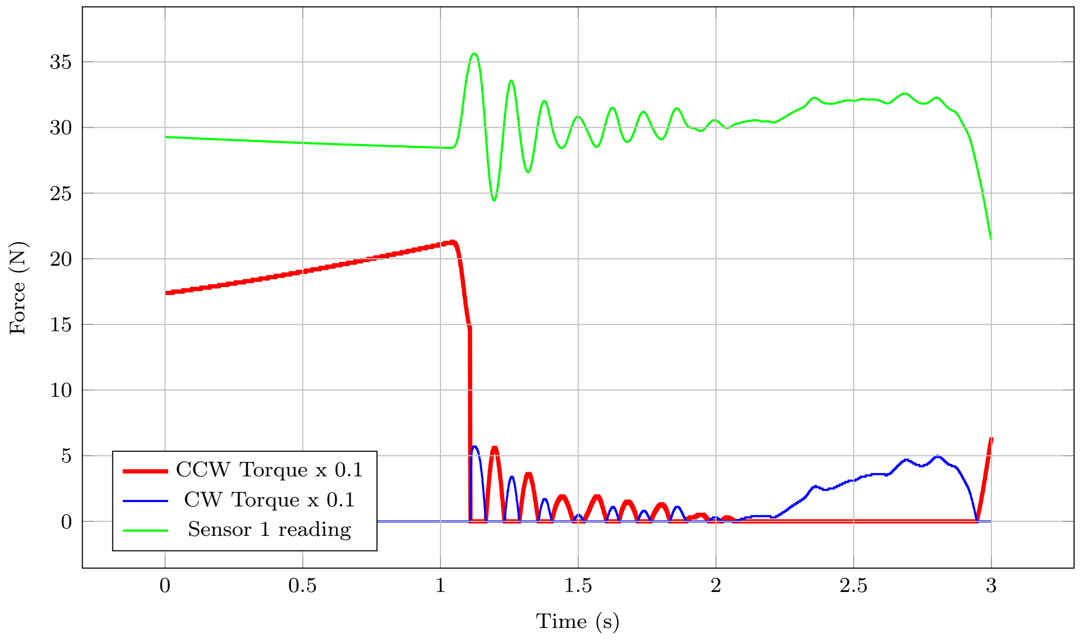


(a) Servo 1 side.

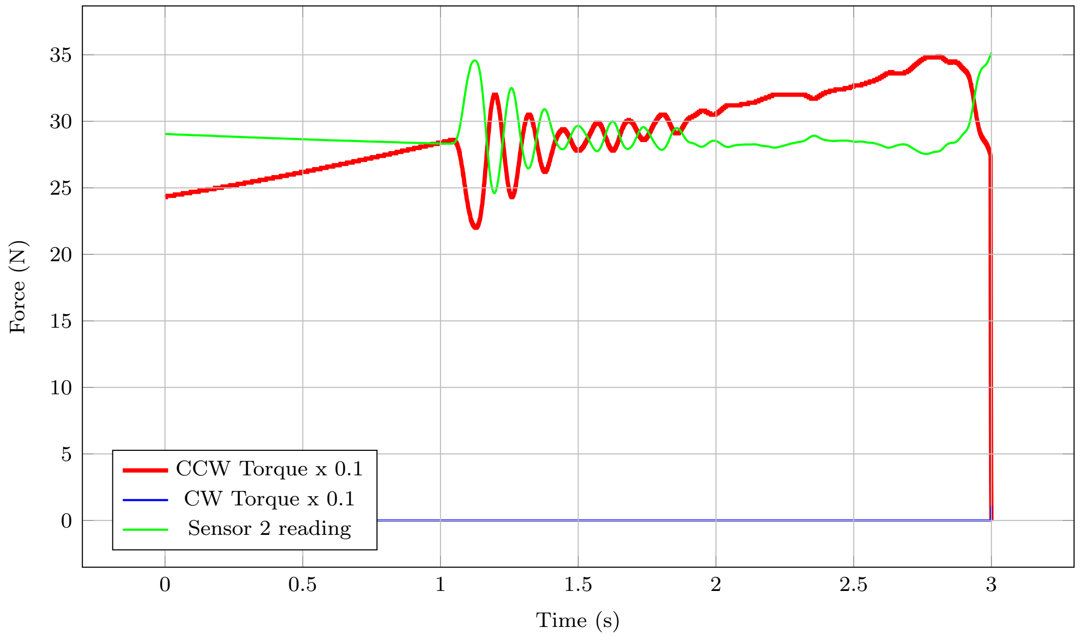


(b) Servo 2 side.

Figure 4.4: Controller test.



(a) Servo 1 side.



(b) Servo 2 side.

Figure 4.5: Effects of the torque direction change.

4.2 Dynamixel MX-106R

In this section, the dynamixel MX-106R will be briefly presented, emphasizing it's advantages over the RX-64 that are relevant to this control. Figure 4.6 shows the new servos. Table 4.1 shows some of the servo specifications; compared to the previous servos these have more detailed information. One of the best improvement is the contactless absolute encoder that allows a full turn; this means that position control is now an option. The communication with these servos is done the same way as the others.

The MX series allows three types of control: wheel mode, joint mode and multi-turn mode.

While on wheel mode it is possible to control torque and the servo will behave just like RX-64's wheel mode, or it is possible to control its speed and the servo will rotate in the commanded direction maintaining the desired speed.

Joint mode is not very interesting, the servos move to position commands restricted to 360°.

The more interesting mode that the MX series offers is the multi-turn which allows the servo to rotate more than one turn while controlling its position, more specifically: seven turns in each direction if the resolution divider parameter is set to one; fourteen turns in each direction if the resolution divider parameter is set to two; twenty one turns in each direction if the resolution divider parameter is set to three and twenty eight turns in each direction if the resolution divider parameter is set to four.

Figure 4.6: Dynamixel MX-106R servo.

Table 4.1: Dynamixel MX-106R specifications.

Weight(g)	153g		
Dimension(mm)	$40.2 \times 65.1 \times 46$		
Gear Reduction Ratio	225 : 1		
Applied Voltage (V)	11.1	12	14.8
Consumed Current (A)	4.8	5.2	6.3
Stall Torque (N.m)	8.0	8.4	10.0
No Load Speed (rpm)	41	45	55
Resolution (°)	0.088		
Speed (rpm/unit)	$\sim$ 0.114		
Running Degree (°)	0 $\sim$ 360, Endless Turn		
Voltage (V)	10 $\sim$ 14.8 (Recommended 12)		
Running Temperature (°C)	-5 $\sim$ +80		
Standby Current (mA)	100		
Position Sensor	Contactless absolute encoder (12BIT,360°)		

4.2.1 Learning Dynamixel MX-106R

Before trying to use these new servos to control the system some small experiments were carried out to learn how to use some of the MX's modes. This revealed that the desired Multi-turn mode wasn't functioning properly. After a quick search through some online forums, one person's comment stated that this mode could only be used with the latest dynamixel firmware. Robotis was contacted and confirmed that the firmware version required for this purpose was 36, these servos had the version 31. Coincidently in LAR there are two robotic arms controlled with dynamixel servos that possess two usb2dynamixel, seen in figure 4.7, which are needed to upgrade the firmware. Using Roboplus, the upgrade was quickly done without any problem. Roboplus is a software given by Robotis that communicates and controls their products, but it needs their physical interface (usb2dynamixel) to communicate with the servos.

Figure 4.7: Usb2dynamixel.

After this, another simple experiment was done to see if the servos were capable of performing all functions which concluded that they are able to perform the functions stated on their online manual. This experiment consisted of using the DETI dynamixel libraries to control the servos in the modes presented in section 4.2.

4.2.2 Control with Dynamixel MX-106R

To control MX-106's with the previous algorithm, equation 4.2, only the constants (K_{P1}, K_{P2} and K_I) were adjusted so the output would be position increments ($Position_{increment}(n)$); basically, they were all lowered. These increments can be positive or negative, a positive increments means that the servo should rotate CCW and a negative increment means the opposite. The new equation can be seen in 4.3.

$$\begin{aligned} Position_{increment}(n) &= K_{P1}.(Tension_{limit} - Sensor(n)) \\ &+ K_{P2}.(Sensor(n) - Sensor_{opposite}(n)) \\ &+ K_I.\sum_{i=0}^{n}(Tension_{limit} - Sensor(n_i)) \end{aligned} \tag{4.3}$$

To adjust this controller, several trials were done, until a good balance between the gains was found. K_{P1} has an important role to the systems response speed, K_I makes the system find the desired force and finally K_{P2} gives the system the right direction to find balance. On an initial phase if the pendulum is tilted, producing a high force difference between the sensors, K_{P2} will make the servo with the higher tension wind the rope bringing it to the upright position. While the other two gains only consider their servo tension and produce the necessary output to bring the tension to the desired value.

For these trials, the desired tension was 5 N because if an overshoot occurs it's less likely to damage the load cells.

The first trials adjusting the gains produced the outcome shown in figure 4.8. It never achieved the desired 5 N tension because it would overshoot.

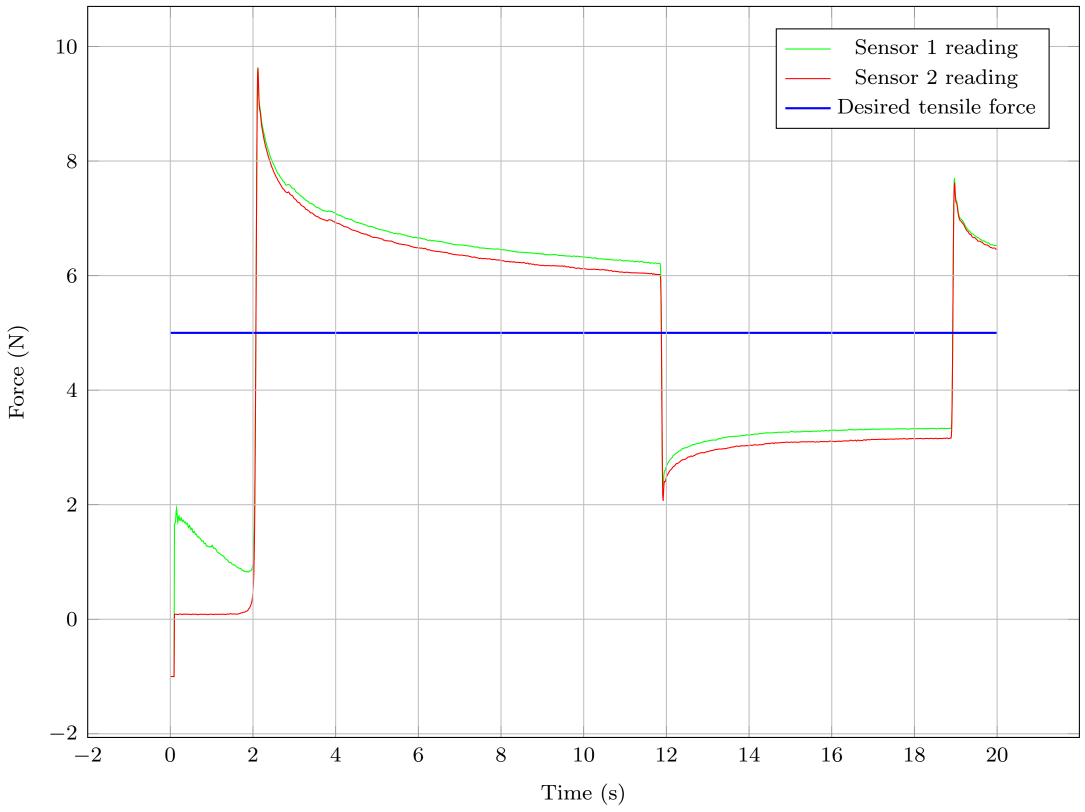


Figure 4.8: Example of the system's behaviour with the first trials.

After some more trials, the final result that was achieved can be seen in figure 4.9 using the gain values seen in table 4.2. All the trials started with the pendulum at the position represented in figure 4.2 that's why the tension is above the 5 N limit. It still has problems because the tension in the ropes wasn't matched and the start wasn't smooth has it should be.

Table 4.2: Controller gains obtained and adjusted by trial and error.

Gain	Value
K_{P1}	0.1
K_{P2}	0.7
K_I	0.00001

In the end, it is clear that this controller isn't the optimal, yet it proved useful to identify problems that the concept brings to its control. The fact that the system is coupled makes it very difficult to achieve equal tensile force on the ropes. As it can

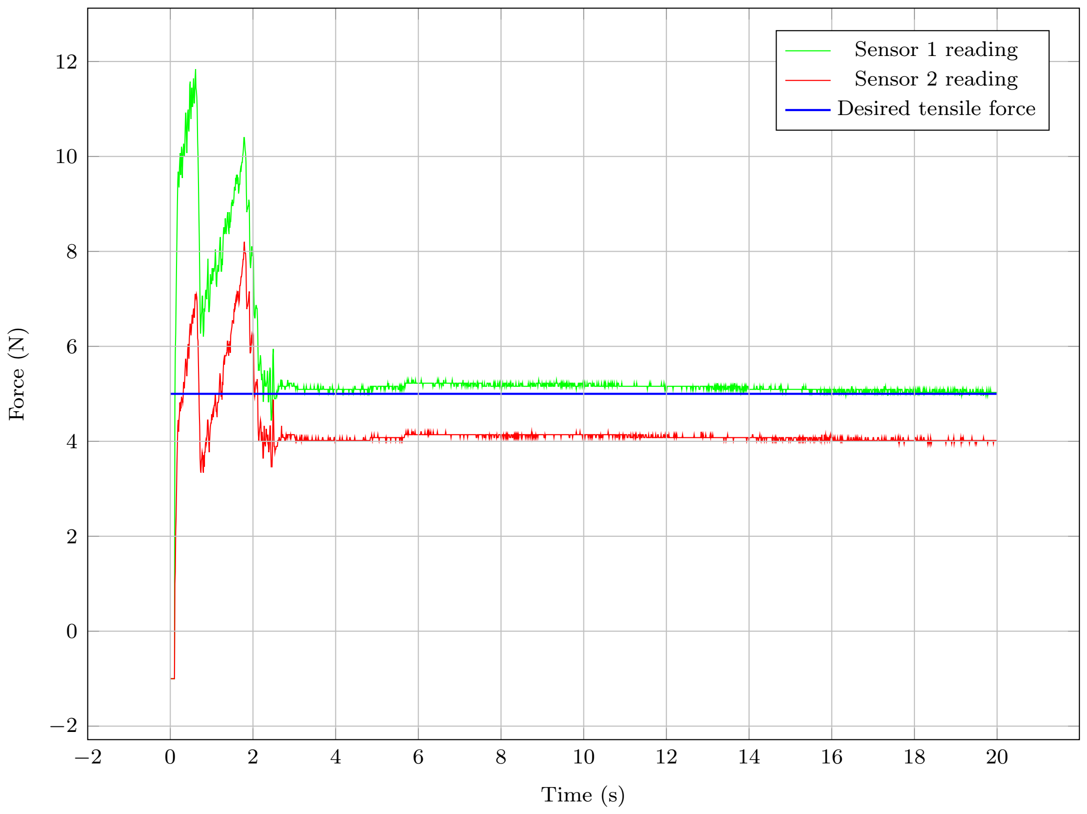


Figure 4.9: Best result achieved with trial and error method.

be seen in figure 4.9 and from the experience tuning the controller if the ropes start to be tensioned with different forces the controller isn't capable of matching the forces. Another problem is the speed of the controller that sometimes times produces a spike in the tensile force as it can be seen in the first two seconds of figure 4.9. So, now that the problems of the concept's control have been identified and the controller can achieve a state of balance where the ropes have stable and similar tensile force the next step is to study the application of stimuli.

Chapter 5

Results from Stimuli Experiments

The main objective of this chapter is to: evaluate the reliability of the application of stimuli using the developed concept; verify the importance of the initial conditions for the repeatability of the stimuli; evaluate different types of stimuli; and also different types of Dynamixel commands will be used to control the servos. First, some stimuli will be applied to the pendulum controlling the servos position, torque and velocity to assess the differences of the stimuli produced by these commands. The second step is to narrow the control methods for the servos and test some stimuli with shorter durations.

5.1 Procedure planning

For all the trials data will be collected over 10 seconds and stored in a csv (comma separated values) file for later analysis. A script plots the information contained in the csv files to provide visual feedback at the end of each trial. This information consists of: sensor readings, applied command to the servos, and the pendulum's angle. Regarding the latter, vision is used to get the pole's angle, more detailed information can be seen in section 5.2. The pendulum's angle is measured just to have more information to compare and thus confirm the reliability and consistency of the results. When the trials were finished a script calculated Pearson's product moment correlation coefficient [22] between the repeated tests, the method used can be seen in equation 5.1. Note that the correlation is done between every repeated test so , as an example, if the a trial is repeated five times this method will produce ten correlations, table 5.1 shows this method.

$$r = \frac{\sum_{i=1}^{n} (x_i - \overline{x}).(y_i - \overline{y})}{\sqrt{[\sum_{i=1}^{n} (x_i - \overline{x})^2].[\sum_{i=1}^{n} (y_i - \overline{y})^2]}}, \tag{5.1}$$

where x_1, x_2, ..., x_n and y_1, y_2, ..., y_n are the measured values. $\overline{x}$ (equation 5.2) and $\overline{y}$ (equation 5.3) are the variables means.

$$\overline{x} = \frac{1}{n}.\sum_{i=1}^{n} x_i \tag{5.2}$$

$$\overline{y} = \frac{1}{n}.\sum_{i=1}^{n} y_i \tag{5.3}$$

The first trials will test different actuations more specifically ramp, ramp step, sawtooth and triangle and different control types such as position, torque and velocity. Only one servo is used, the other one is decoupled. This means that the rope is always loose so the servo actuation effect will be delayed and it is the reason why the pulse step wasn't used on these trials. Each trial was repeated five times to get a reliable correlation. This procedure is depicted in table 5.2

The second trials had in mind the previous results, so only torque and velocity control were used. The only step used here was the pulse with three durations 0.1, 0.25 and 0.5 seconds and three different torque and speed values. This was decided based on the experience with the system and the experiments seen with other robot's balance test. For these set of trials the two servos were coupled, so the data collected was the force from the two sensors and the pendulum's angle. These trials had two distinct phases, one where the servos would adjust both rope's tension to about 4 N using the controller designed and some manual adjustments. The second phase was the stimulus application, where only servo 1 would pull the pole and servo 2 would just release the rope. When the stimulus time had finished servo 1 would unwind the rope at maximum speed to allow the pendulum to freely return to balance. Each trial was repeated only three times because they consumed some time and the previous ones proved to be reliable. The procedure for the second trials is depicted in table 5.3

Table 5.1: Correlations between repeated trials.

	x_1	x_2	x_3	x_4	x_5
x_1		r_{1-2}	r_{1-3}	r_{1-4}	r_{1-5}
x_2			r_{2-3}	r_{2-4}	r_{2-5}
x_3				r_{3-4}	r_{3-5}
x_4					r_{4-5}
x_5					

Table 5.2: Procedure plan for the first trials.

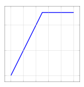
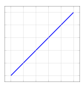
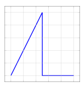
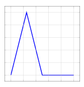
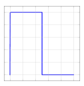
	Position commands	Velocity commands	Torque commands
 Ramp step	×5	×5	×5
 Ramp	×5	×5	×5
 Sawtooth	×5	×5	×5
 Triangle	×5	×5	×5

Table 5.3: Procedure plan for the second trials.

	Torque commands			Velocity commands		
	800	900	1023	800	900	1023
0.1	×3	×3	×3	×3	×3	×3
0.25	×3	×3	×3	×3	×3	×3
0.5	×3	×3	×3	×3	×3	×3

5.2 Measuring the Pole Angle with Vision

To get the pole's angle a simple 30 Hz web cam was used. Another node, `dyna_comm_camera`, was added to the `dyna_comm` package to deal with the image processing and publish the pole's angle, this way the node `dyna_comm` writes the csv file with synchronized data. OpenCV libraries were used to connect with the usb camera and do all the necessary steps to obtain the pole's angle. These steps are the following by order: transform the coloured image to grey scale; transform it to black and white and invert it because the pendulum is black; get all objects contours which are ordered from biggest to smallest, the pendulum being the biggest; get the orientation of the biggest object.

To assure that the pendulum is the biggest object the background is white and the pole was painted black, this way the lighting doesn't influence the image analysis process. These features and the pole's detection can be seen in figure 5.1. To assure that the correct angle is published the camera is level with the horizontal.

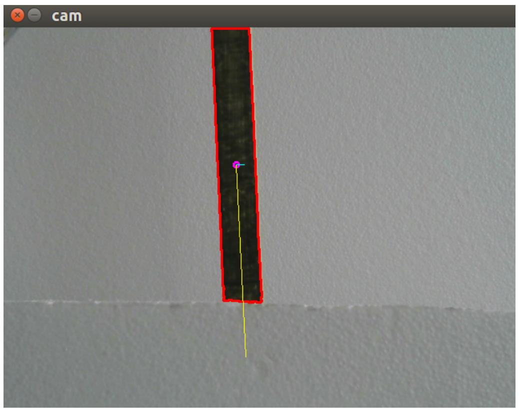


Figure 5.1: Pole painted black and white background to favour its detection.

5.3 First Trials Results Presentation and Discussion

Annex D shows one set of graphics per trial, these sets were repeated five times. Each figure represents one trial, the left graphic is the sensor readings and the right graphic is the pendulum's angle. Figure 5.2 shows an example of these graphics. Tables 5.4, 5.5 and 5.6 show the correlation mean and its standard deviation, using equation 5.4 and equation 5.5, respectively.

$$\overline{r} = \frac{1}{n}.\sum_{i=1}^{n} r_i \tag{5.4}$$

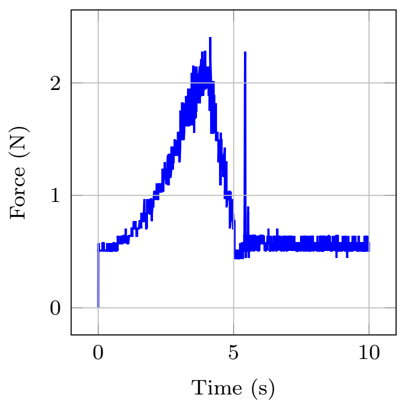
$$r_\sigma = \sqrt{\frac{1}{n}.\sum_{i=1}^{n}(r_i - \overline{r})^2} \tag{5.5}$$

On some of the ramp step, sawtooth and triangle graphics it is noticeable a spike in the tensile force; this is due to the pendulum's inertia. The sawtooth trials have the worst correlation because the servo couldn't unwind rope fast enough to allow the pendulum to freely return to the balance position. As for the decision to choose a control type for the servos it is still difficult to do; for now, only position will be discarded since it requires the definition of other parameters, namely velocity and torque.

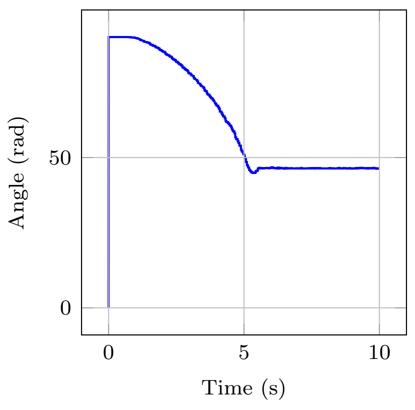
As for the repeatability of the tests it can almost be assumed that the system is reliable because the initial conditions weren't exactly the same for every trial and still the correlations for the ramp and ramp step are good. As written before the initial conditions aren't exactly the same because the rope was loose. This somewhat acceptable correlation and the fact that these trials consume some time led to the decision of only repeating the second trials three times.

Table 5.4: Correlation of the results produced by different stimuli, controlling servo position.

	$\overline{r}$		r_σ	
	Force	Angle	Force	Angle
Ramp step	0.9421	0.9333	0.0127	0.0457
Ramp	0.9623	0.9424	0.0055	0.0741
Sawtooth	0.8598	0.9744	0.0798	0.0221
Triangle	0.8547	0.9675	0.0235	0.0393



(a) Tensile Force.



(b) Angle of the pendulum.

Figure 5.2: Ramp step with torque control.

Table 5.5: Correlation of the results produced by different stimuli, controlling servo velocity.

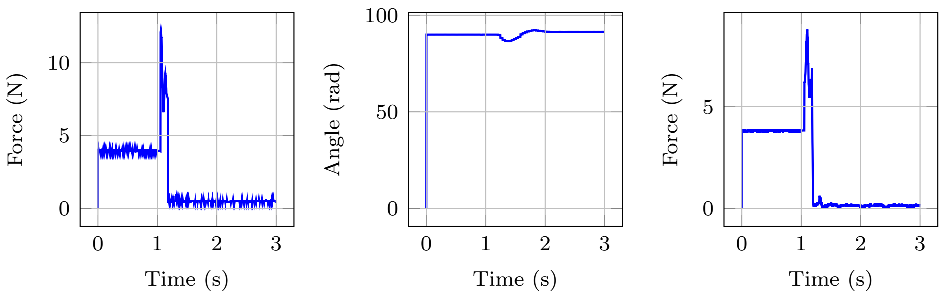
	$\overline{r}$		r_σ	
	Force	Angle	Force	Angle
Ramp step	0.9410	0.9269	0.0102	0.0522
Ramp	0.9141	0.9115	0.0396	0.0718
Sawtooth	0.4043	0.9280	0.1601	0.0558
Triangle	0.9192	0.6951	0.0215	0.2878

Table 5.6: Correlation of the results produced by different stimuli, controlling servo torque.

	$\overline{r}$		r_σ	
	Force	Angle	Force	Angle
Ramp step	0.9435	0.9538	0.0214	0.0396
Ramp	0.9411	0.9737	0.0303	0.0298
Sawtooth	0.3289	0.9201	0.0673	0.0845
Triangle	0.8238	0.6961	0.0634	0.3843

5.4 Second Trials Results Presentation and Discussion

Annex E shows one set of graphics per trial, these sets were repeated three times. From these graphics it can be concluded that the stimuli should be applied at maximum torque or velocity and for short durations to avoid the spikes seen in some of the actuating force graphics. The servos can't unwind rope at the same speed that the pendulum's elastic units react. Figure 5.3 shows a set of these graphics as an example.



(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

Figure 5.3: Pulse with 800 torque and 0.1 s duration.

As for the quality of these results, it can be seen in tables 5.7 and 5.9, for torque and velocity respectively. The initial conditions and its variations can be seen in tables 5.8 and 5.10, for torque and velocity respectively.

Crossing the correlation tables with the respective initial values table makes it clear that the trial can be easily repeated because even though there were some high deviations of the initial tensile force this did not affect the test correlation. All the correlations coefficients are sufficiently high to consider enough the number of trial repetitions, which was three. It was considered that for statistical purposes three trials might not be enough but since the correlations are mostly above 0.99 these results are considered valid.

And so, the evaluation of the application of stimuli proved that this concept satisfies the need for repeatability. When it comes to different types of stimuli there is a need to test if it can be repeatable; the sawtooth and triangle stimuli proved this fact that not all stimuli can be replicated. As for the type of Dynamixel commands it is still not clear which one is the best for this kind of application; torque and velocity produce very similar results and are easier to apply since they only require the definition of one command as opposed to position commands which require more detailed definitions. Regarding the initial conditions, considering that the repeated trials had variations between ≈ 0.02 and ≈ 0.4 and still managed to get high correlation values gives the impression that they can support a considerable error ($\approx 10\%$ in the case of torque stimuli with 0.5 s duration) and produce reliable results.

Table 5.7: Correlation of the results produced by pulse stimuli with different durations, controlling servo torque.

		$\overline{r}$			r_σ		
Torque	Time (s)	Force 1	Angle	Force 2	Force 1	Angle	Force 2
800	0.1	0.9984	0.9947	0.9986	0.0008	0.0021	0.0002
	0.25	0.9926	0.9642	0.9930	0.0046	0.0438	0.0055
	0.5	0.9956	0.9975	0.9972	0.0007	0.0008	0.0004
900	0.1	0.9947	0.9965	0.9954	0.0017	0.0015	0.0030
	0.25	0.9842	0.9957	0.9903	0.0115	0.0005	0.0072
	0.5	0.9825	0.9943	0.9968	0.0092	0.0032	0.0017
1023	0.1	0.9924	0.9974	0.9943	0.0035	0.0012	0.0040
	0.25	0.9932	0.9958	0.9940	0.0045	0.0012	0.0045
	0.5	0.9929	0.9970	0.9940	0.0042	0.0006	0.0044

Table 5.8: Initial conditions of trials with the pulse stimuli with different durations, controlling servo torque.

		Initial Values Mean			Initial Values Standard Deviation		
Torque	Time (s)	Force 1 (N)	Angle (°)	Force 2 (N)	Force 1 (N)	Angle(°)	Force 2 (N)
800	0.1	4.0125	89.9958	3.8597	0.0871	0.0740	0.0445
	0.25	3.7747	89.9719	4.1636	0.1771	0.0683	0.0807
	0.5	3.8791	90.0486	4.1636	0.4013	0.0372	0.3078
900	0.1	3.7717	89.9520	4.1923	0.2287	0.0360	0.1487
	0.25	3.9325	89.9476	4.0179	0.3211	0.0441	0.1124
	0.5	3.7028	90.0218	4.0295	0.1426	0.0545	0.1395
1023	0.1	3.7827	90.0120	4.0037	0.0266	0.0146	0.1851
	0.25	3.8746	90.0398	3.9139	0.1095	0.0751	0.2224
	0.5	3.5811	90.0284	3.8578	0.0165	0.0482	0.2550

Table 5.9: Correlation of the results produced by pulse stimuli with different durations, controlling servo velocity.

		$\overline{r}$			r_σ		
Velocity	Time (s)	Force 1	Angle	Force 2	Force 1	Angle	Force 2
800	0.1	0.9965	0.9704	0.9970	0.0005	0.0233	0.0012
	0.25	0.9986	0.9836	0.9982	0.0005	0.0061	0.0008
	0.5	0.9403	0.9866	0.9946	0.0242	0.0038	0.0021
900	0.1	0.9974	0.9477	0.9977	0.0011	0.0428	0.0011
	0.25	0.9939	0.9828	0.9971	0.0035	0.0121	0.0015
	0.5	0.9804	0.9934	0.9915	0.0131	0.0039	0.0059
1023	0.1	0.9967	0.9810	0.9960	0.0021	0.0133	0.0029
	0.25	0.9923	0.9969	0.9975	0.0042	0.0006	0.0014
	0.5	0.9467	0.9884	0.9901	0.0325	0.0085	0.0070

Table 5.10: Initial conditions of trials with the pulse stimuli with different durations, controlling servo velocity.

		Initial Values Mean			Initial Values Standard Deviation		
Velocity	Time (s)	Force 1 (N)	Angle (°)	Force 2 (N)	Force 1 (N)	Angle(°)	Force 2 (N)
800	0.1	3.7166	89.1194	4.0151	0.1186	0.7606	0.1045
	0.25	3.6941	89.2527	3.9003	0.0828	0.8207	0.2572
	0.5	3.6489	89.6939	3.9247	0.1229	0.5049	0.2141
900	0.1	3.6304	89.1316	3.8276	0.1884	0.9508	0.1806
	0.25	3.7447	89.7734	4.0183	0.2516	0.8168	0.2853
	0.5	3.6319	89.1251	3.9527	0.1795	0.5337	0.2627
1023	0.1	4.0992	90.0413	4.1990	0.1611	0.1963	0.4638
	0.25	3.9386	90.0330	3.8830	0.1650	0.0994	0.1817
	0.5	3.7914	90.0630	3.9111	0.2261	0.1279	0.3125

Chapter 6

Conclusion and Future Work

The final conclusions of this dissertation are detailed on this chapter, evaluating the success of the initially proposed objectives and paths that this work took. For future works, some proposals are made, suggesting some approaches considering the present work results.

6.1 Conclusion and Considerations

The main objective of this dissertation work was to implement and test a balance disrupting system. This objective was accomplished, an balanced inverted pendulum was built and stimuli were applied to it and studied giving valuable information about the concept of disturbing balance using pull forces. On a first step some background information was gathered that influenced some decisions throughout this project. Then the hardware and software needed was acquired, designed and built. And finally the concept was test, evaluated and discussed.

As for the hardware needed: a protective case for the DETI Dynamixel Interface was designed, printed and implemented; a small circuit board was designed and built to supply the DETI Dynamixel Interface with 5 V; a mechanism to use load cells as tensile force sensors was successfully designed and implement. Regarding the software: a ROS infrastructure was created to control the servos, read the load cells and process images all in real time and while storing important information. A Solidworks assembly of the pendulum and SimMechanics tutorial was created.

The PID controller obtained by trial and error allowed the identification of problems, such as: its speed which can be good for a fast responding system but can also be bad because it can easily produce instability; the rope's non linearity makes the stabilization harder, when the intended tensile force is reached the ropes begin to settle and thus making the tension drop, to avoid this problem a rope with a higher spring stiffness should be used; and finally the fact that the system is coupled makes it very hard to lower the difference between the ropes tensile force after they are tensioned.

The dynamixel servos are very easy to communicate with and control, yet the fact that the servo don't control torque, as it was thought, is a considerable disadvantage. By torque control it was thought that the servo would maintain an intended torque value in the commanded direction, as an example: if the servo is commanded to produce 1 N m in the CW direction and it is pulled in the CCW direction with a greater torque it locks;

and what was expected was that it should adapt it's rotation to maintain 1 N m in the CW direction.

Finally, some stimuli were tested which allowed the conclusion that not all types of stimuli are repeatable, meaning that it is necessary to test every new type of stimulus to assess it's repeatability. As for the initial conditions that are required for the reliability of the stimuli application, they still need to be studied better because the trials that were done seem to indicate that an error up to $\approx 10\%$ doesnât influence the trials repeatability. By error it is meant the difference between the initial tensile force measured on the ropes for the repeated tests.

6.2 Future Work and Recommendations

For future work it is recommended an extensive search on different types of controllers and controllers for rope coupled system's. Also it is recommend to read all the articles about the Cable-driven Auto-levelling Parallel Robot ([23], [24] ,[25] ,[26] and [27]) and to search more about that project. The Cable-driven Auto-levelling Parallel Robot project managed to built a control for a system with four cables with tensile force sensors and faced the problem of it being a coupled system. They used a fuzzy controller and also did the stabilization of their system in two steps, their robot would stabilize a pair of ropes and then the other pair. Yet, their system had more than the tensile force inputs, so all their strategies might not help this project.

The future steps should be as follows:

- The influence of the initial conditions need to better understood.
- Try to do a simulation of the system using the tutorial or something else, and try to obtain the system's equations of motion.
- Test more stimuli, possibly with longer duration, such as a sinusoidal wave.
- Find the best way to control the dynamixel servos or find another type of servos that produce the intended torque control.
- Consider the concept requirements, maybe make the initial tensioning of the ropes slower.
- Create a graphical user interface that allows the user to create and apply stimuli as intended.
- Acquire more or other tensile force sensors.
- A structure needs to be designed and built to accommodate four servos and have PHUA in the middle to apply stimuli to the humanoid robot;

Bibliography

[1] Interface. *Model LBS Miniature Compression Load Button Datasheet*, 2009.

[2] J. Barros, F. Serra, V. Santos, and F. Silva. Tele-kinesthetic teaching of motion skills to humanoid robots through haptic feedback. In *IEEE-RAS International Conference on Humanoid Robots*, 2014.

[3] H .Kharboutly, J . Ma, A. Benali, P. Thoumie, V. Pasqui, and M. Bouzit. Design of multiple axis robotic platform for postural stability analysis. *Neural Systems and Rehabilitation Engineering*, 23(1):93–103, 2015.

[4] J. Ma, H. Kharboutly, A. Benali, F. B. Amar, and M. Bouzit. Design of omnidirectional mobile platform for balance analysis. *IEEE/ASME Transactions on Mechatronics*, 19(6):1872–1881, 2014.

[5] T. Capato. Eficácia de um programa de treinamento motor para melhora do equilíbrio associado a pistas rítmicas e suas repercussões na marcha e aspectos não motores de pacientes portadores de doença de parkinson. Master's thesis, Universidade de São Paulo, 2007.

[6] T. Mergner, C. Maurer, and R. J. Peterka. A multisensory posture control model of human upright stance. *Progress in brain research*, 142:189–201, 2003.

[7] D. E. Whitney. Historical-perspective and state-of-the-art in robot force control. *The International Journal of Robotics Research*, 6(1):3–14, 1987.

[8] P. Cruz. Haptic interface data acquisition system. Master's thesis, Universidade de Aveiro, 2012.

[9] F. Serra. Haptic Interface for the Tele-operation of a Humanoid Robot. Master's thesis, Universidade de Aveiro, 2014.

[10] J. Barros. Cooperative haptics for humanoid robot teleoperation. Master's thesis, Universidade de Aveiro, 2014.

[11] A. Shumway-Cook, D. Anson, and S. Haller. Postural sway biofeedback: its effect on reestablishing stance stability in hemiplegic patients. *Archives of physical medicine and rehabilitation*, 69(6):395–400, 1988.

[12] B. L. Day, A. S. Cauquil, L. Bartolomei, M. A. Pastor, and I. N. Lyon. Human body-segment tilts induced by galvanic stimulation: a vestibularly driven balance protection mechanism. *Journal of Physiology*, pages 661–672, 1997.

[13] T. Dijkstra. *Visual control of posture and visual perception of shape*. PhD thesis, Katholieke Universiteit Nijmegen, Nijmegen, Netherlands, 1994.

[14] A. Kavounoudias, J. C. Gilhodes, R. Roll, and J. P. Roll. From balance regulation to body orientation: Two goals for muscle proprioceptive information processing? *Experimental Brain Research*, 124(1):80–88, 1999.

[15] L. Gray. Chapter 10: Vestibular System: Structure and Function.

[16] B. L. Riemann and S. M. Lephart. The sensorimotor system, part ii: The role of proprioception in motor control and functional joint stability. *Journal of Athletic Training*, 37(1):80–84, 2002.

[17] L. I. Wolfson, R. Whipple, P. Amerman, and A. Kleinberg. Stressing the postural response. a quantitative method for testing balance. *Journal of the American Geriatrics Society*, 34(12):845–850, 1986.

[18] U. Granacher, T. Muehlbauer, and M. Gruber. A qualitative review of balance and strength performance in healthy older adults: Impact for testing and training. *Journal of Aging Research*, 2012, 2012.

[19] ROBOTIS. *Dynamixel RX-64 User's Manual*, 2006.

[20] E. Tira-Thompson. Digital servo calibration and modeling. Technical Report CMU-RI-TR-09-41, Robotics Institute, Pittsburgh, 2009.

[21] E. Estrelinha. Tele-operation of a humanoid robot using haptics and load sensors. Master's thesis, Universidade de Aveiro, 2013.

[22] M. Mukaka. A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal: The Journal of Medical Association of Malawi*, 24(3):69–71, 2012.

[23] J. Zhang, D. Zhao, J. Yi, and Y. Yu. Modelling of a rope-driven self-levelling crane. In *The 3rd International Conference on Innovative Computing Information and Control*, 2008.

[24] Y. Yu, J. Yi, C. Li, and D. Zhao. Fuzzy Logic Based Adjustment Control of a Cable-driven Auto-leveling Parallel Robot. In *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[25] J. Zhang, J. Yi, and Y. Yu. Intelligent control of a four-rope-driven level-adjustment device with constrained outputs. In *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*, 2009.

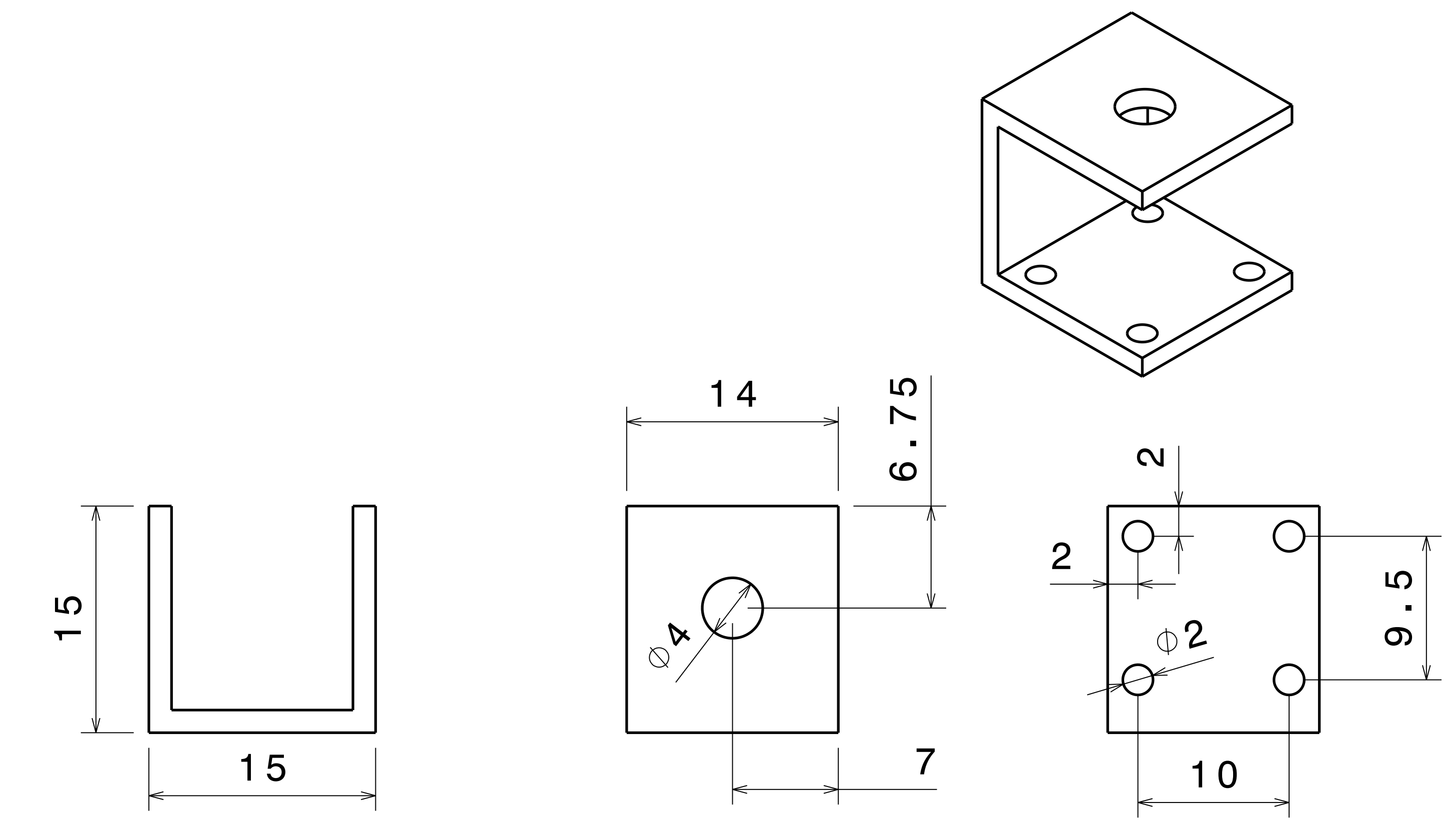
[26] L. Cheng-Dong, Y. Jian-Qiang, Y. Yi, and Z. Dong-Bin. Inverse control of cable-driven parallel mechanism using type-2 fuzzy neural network. *Acta Automatica Sinica*, 36(3):459–464, 2010.

[27] J. Zhang, J. Yi, X. Tan, and Y. Yu. Fuzzy control of a four-rope-driven level-adjustment robot considering all constrained situations. In *American Control Conference on O'Farrell Street*, 2011.

Appendices

Appendix A

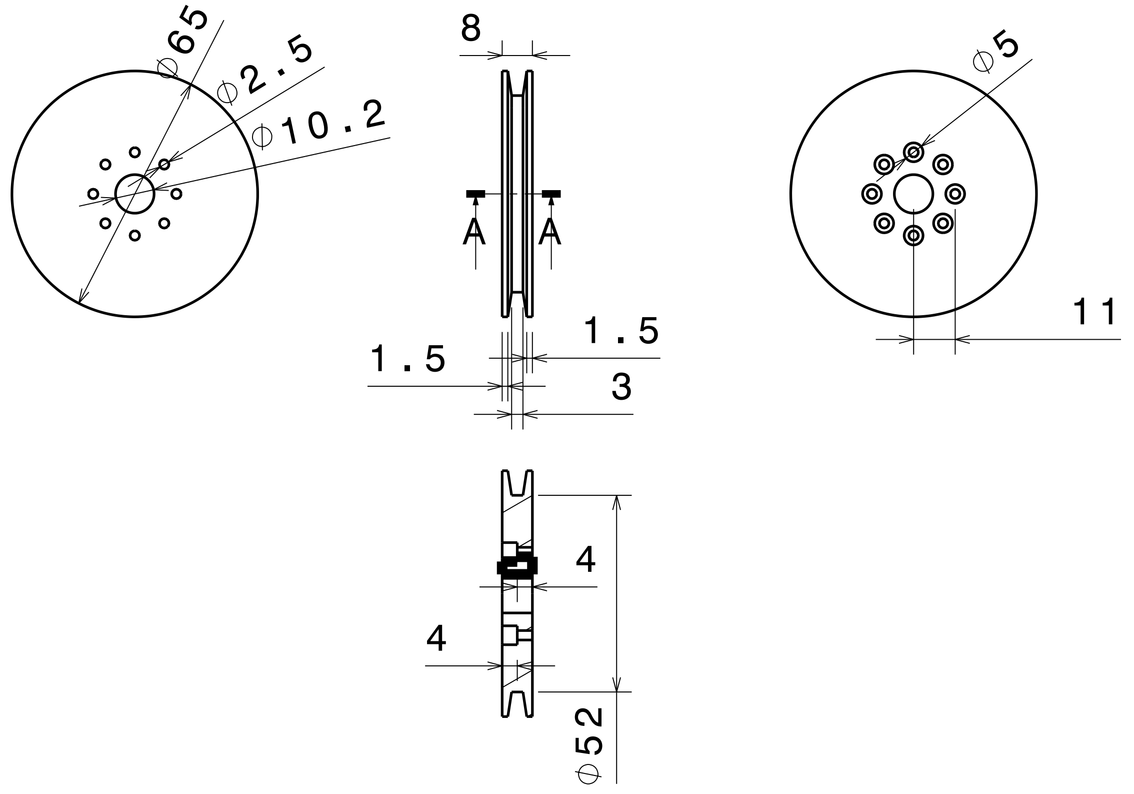
Load Cell Mechanism



DESIGNED BY: JoãoFilipe DATE: 30/06/2015	Name	Load Cell Mechanism	I	_
	Material	Aluminium	H	_
CHECKED BY: XXX DATE: XXXX	Quantity	20	G	_
	Thickness	XXXX	F	_
SIZE A4	SequenceID	XXXX	E	_
	GroupID	XXXX	D	_
SCALE 2:1 / WEIGHT (kg) 0,00			C	_
			B	_
			A	_

Appendix B

Sheave



DESIGNED BY: JoãoFilipe DATE: 10/07/2015		Name	Sheave	I	_
		Material	Plastic	H	_
CHECKED BY: XXX DATE: XXX		Quantity	4	G	_
		Thickness	XXXX	F	_
SIZE A4		SequenceID	XXXX	E	_
		GroupID	XXXX	D	_
SCALE 1:2	WEIGHT (kg) 0,00			C	_
				B	_
				A	_

This drawing is our property; it can't be reproduced or communicated without our written agreement.

Appendix C

Simulation

C.1 Introduction to SimMechanics™

SimMechanics provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. The multibody system is modelled using blocks representing bodies, joints, constraints, and force elements, and then SimMechanics formulates and solves the equations of motion for the complete mechanical system. Models from CAD systems, including mass, inertia, joint, constraint, and 3D geometry, can be imported into SimMechanics. An automatically generated 3D animation allows the visualization of the system dynamics.

The models can be parametrized using Matlab variables and expressions, and design control systems for a multibody system in Simulink®. Electrical, hydraulic, pneumatic, and other components can be added to a mechanical model using Simscape™ and test them all in a single simulation environment.

A new multibody model can be started by typing `smnew` in the Matlab command line. A new model window opens with commonly used blocks. The SimMechanics block library also opens. Figure C.1 shows the new model window. To begin modelling a new multibody system, blocks from the library are dragged into the model window. These blocks belong to the SimMechanics Second Generation library which has more tools and functionalities compared to First Generation.

Mechanical links are common building blocks in linkages, mechanisms, and machines. A link needs end frames to connect to joints. Rigid Transform Blocks provide the end frames, while a Solid block provides geometry, inertia and colour. Joint blocks, as in real life, connect two solids. The Solver Configuration block specifies the solver parameters that the model needs before the simulation can begin. The World Frame block represents the global reference frame in a model. This frame is inertial and at absolute rest. Rigidly connecting a frame to the World frame makes that frame inertial. Frame axes are orthogonal and arranged according to the right-hand rule. In a frame network, the World frame is the ultimate reference frame. Directly or indirectly, all other frames are defined with respect to the World frame. The Mechanism Configuration block provides mechanical and simulation parameters to a mechanism, i.e., a self-contained group of interconnected SimMechanics blocks. Parameters include gravity and a linearisation delta for computing numerical partial derivatives during linearisation. These parameters apply only to the target mechanism, i.e., the mechanism that the block connects to. These are the main blocks to construct a multibody model and they can be seen in figure C.2.

When the model is finished the joint blocks can be actuated with a Signal block to perceive the systems behaviour or with a PID block to find an optimal controller. Also to be noted that the Matlab version should be higher than R2014 to access the new SimMechanics blocks that have more sensing options. To find out more detailed information and help for SimMechanics please refer to the the official MathWorks® website.

Building a multibody model using the SimMechanics and Simscape libraries can be time consuming and difficult since objects with complex geometries are impossible to accurately model using only the Solid block, so fortunately there is a faster way to build a model which is using the tool SimMechanics Link.

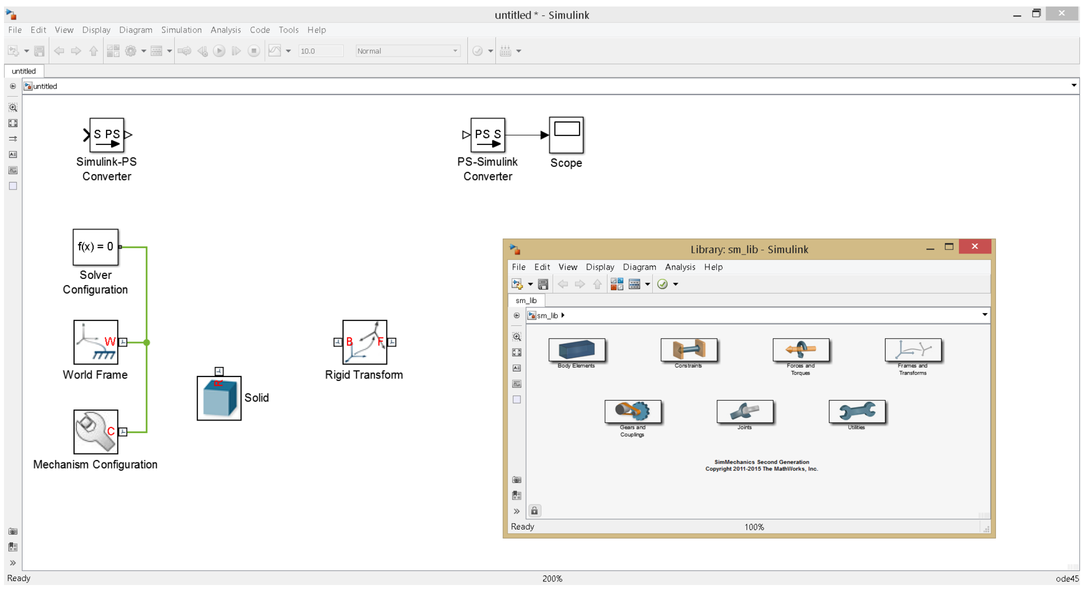


Figure C.1: New model window and SimMechanics Second Generation library.

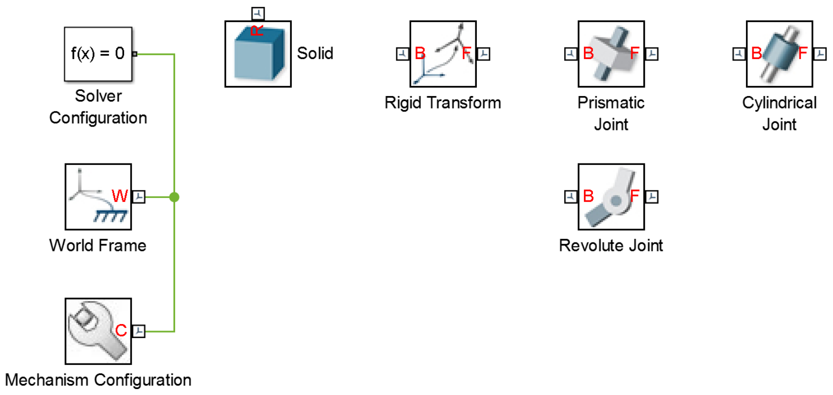


Figure C.2: Main blocks necessary to build a multibody system.

C.2 Introduction to SimMechanics Link

SimMechanics Link is a Matlab function that can be downloaded in the official MathWorks website. This function creates a plug-in for a Computer Aided Design (CAD) software such as Solidworks™, Creo-Pro/E™ and Inventor™. The selected CAD software was Solidworks because it was the only one dealt with before.

When the function is installed and the plug-in is activated on the CAD software a 3D assembly of the system can be built with all needed constraints to make the model behave just as desired. These constraints are important because when the assembly is exported to SimMechanics they will be transformed into joints. There are two options to export the assembly which are SimMechanics First and Second Generation. Second Generation is preferred because it provides more features.

C.3 Pendulum Model

C.3.1 Construction of the Pendulum Assembly on Solidworks and SimMechanics Import

Modelling the pendulum brought up a problem which is to represent the winding of rope, this physical process can be visually produced but that is not enough because rope can be tensioned but not compressed and the final assembly wouldn't be able to reproduce that property. So, knowing that constraints on an assembly produce joints when exported to a SimMechanics model a simpler solution was approached.

The part seen in figure C.3 will duplicate the winding. Its hole will allow rotation, mimicking the rope's rotation around the pulley's radial plane and two edges are coincident with the opposing part (highlighted in figure C.3) will mimic the length variation of unwinded rope. On other words one servo with a pulley and rope is simplified with two solids and three joints, more specifically two rotational joints and one prismatic. The hole constraint will result in a revolute joint and the two coincident edges will result in a prismatic joint.

The final assembly is composed of six parts seen in figure C.4. To simplify even more the virtual pendulum it is symmetric, contrary to reality, were the pulleys aren't on the same plane producing a slight rotation on the pole. This was done to increase Matalab's computation time.

After the 3D model was completed it was exported to Matlab. On the Matlab side the assembly was imported to a SimMechanics multibody, the blocks can be seen in figure C.5. To note that this process produced cylindrical joints instead of prismatic, this is not a problem because it is a joint with one prismatic and one revolute primitives possessing parallel motion axes, meaning that the importing process considered that the constraint between the two parts (figure C.3) has some torsion.

C.3.2 SimMechanics Model Changes

Joint spring stiffness and friction parameters can't be specified on Solidworks, so it was added friction and a spring coefficient to the pole's rotational joint. These values were adjusted until the simulation would behave similarly to the real system. Note that this model is self supported by the spring in the pole's joint, this mimics the real pendulum when it is supported with the elastics.

It was not an objective to construct the virtual model as identical as possible to the real system because the main goal of this approach is to assess if a controller can be designed and see the effects of stimuli on the system.

Figure C.6 shows the implementation of a hard stop on the pole's bottom joint to prevent it from going through the base.

The solver chosen to run the system simulation was ode15s, recommended by MathWorks.

After all these changes the model is ready to start being measured and controlled. On a first approach the reaction forces from the prismatic joint were analysed while a force was being applied to the pole. This did not produce the expected output which was a force along the Z axis of the linear joint with values close to the force applied. Many settings were tweaked but none produced values close to a real situation.

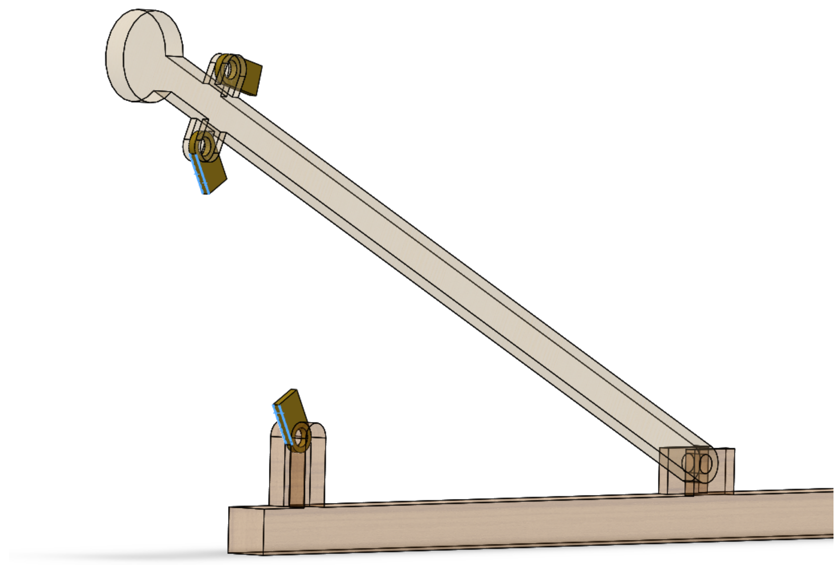


Figure C.3: Part designed to simplify the winding process.

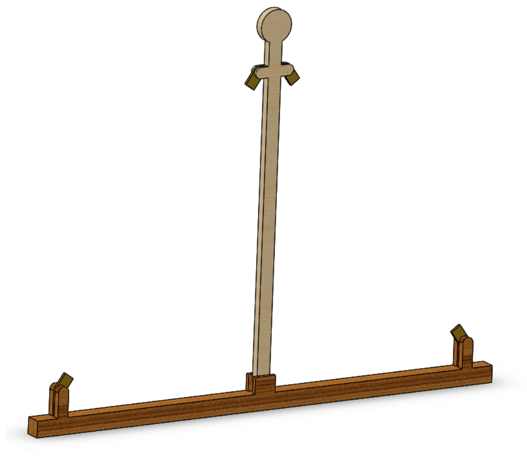


Figure C.4: Pendulum assembly.

From the experience of other users a solution to measure tensile force between two parts connected by a linear joint could be adding a spring and damper force in series. To use this block to aid this situation it is advised to set its spring stiffness to a high level, so it becomes almost rigid, and then measure the output force. Unfortunately this solution did not solve the problem, the values of the outputted force continued to be distinct from the expected. The experience with the real pendulum supports the failure to read the tensile force on the virtual model.

None of these force readings were similar to what was expected because their value was always between 1 and -1. When a force is applied on the pendulum the rope is pulled and its tensile force will have a magnitude close to the applied force, this is what happens with the real pendulum.

Even more, after some online research and conversations with other SimMechanis users it was realised that this tool doesn't allow the extration of the model's mathematical equations, it only solves them to produce the simulation.

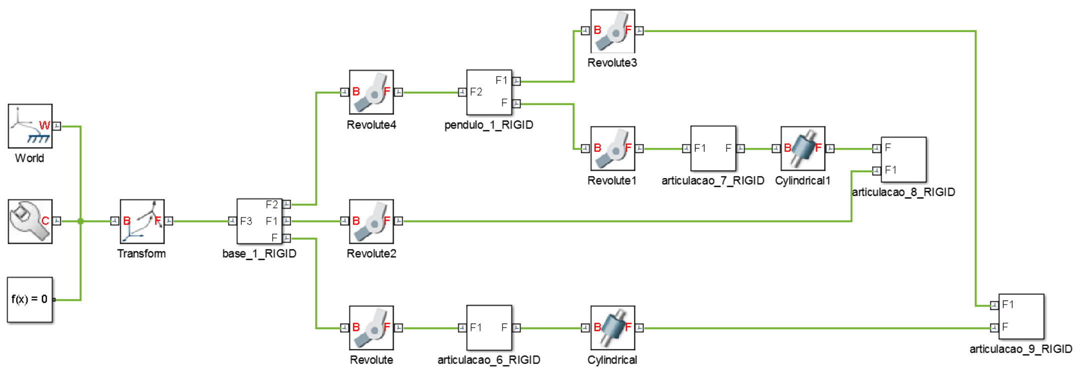


Figure C.5: Pendulum assembly after it was imported to SimMechanics.

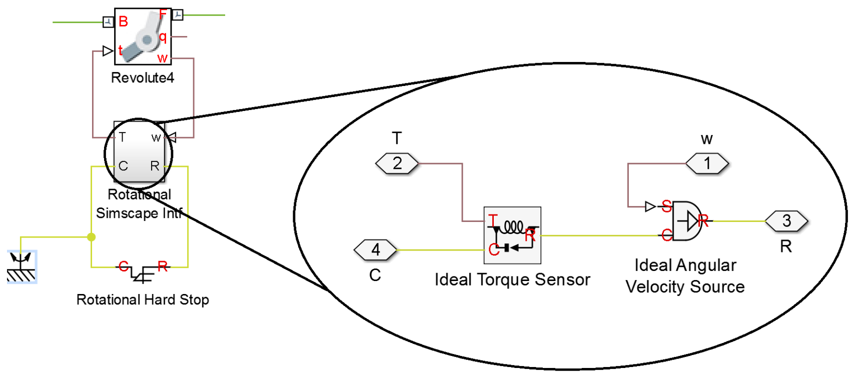
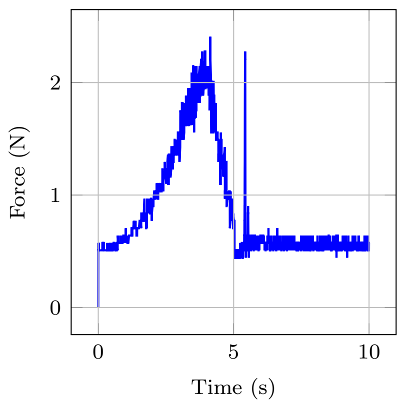


Figure C.6: Rotational Hard Stop to limit the pendulum's revolute joint movement range.

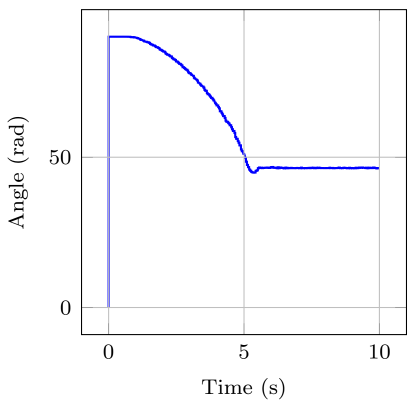
Appendix D

First Stimuli Trials

D.1 Torque Commands

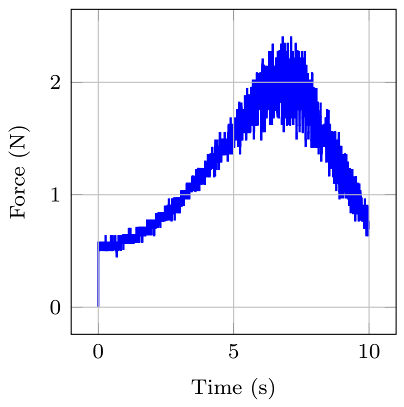


(a) Tensile Force.

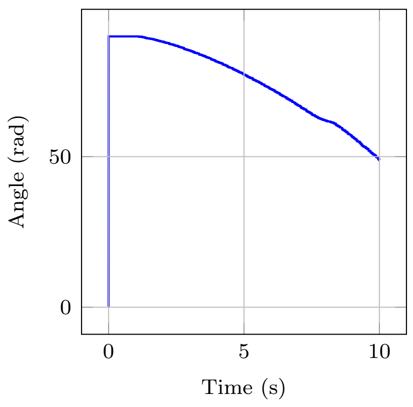


(b) Angle of the pendulum.

Figure D.1: Ramp step with torque control.

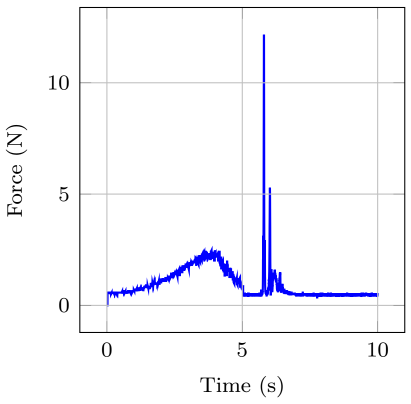


(a) Tensile Force.

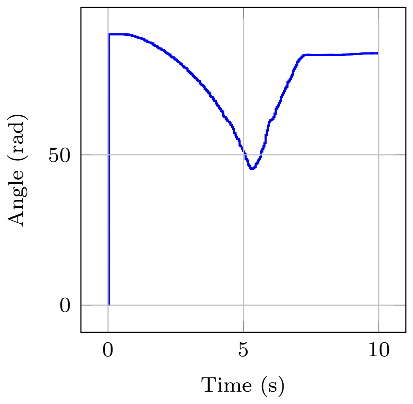


(b) Angle of the pendulum.

Figure D.2: Ramp with torque control.

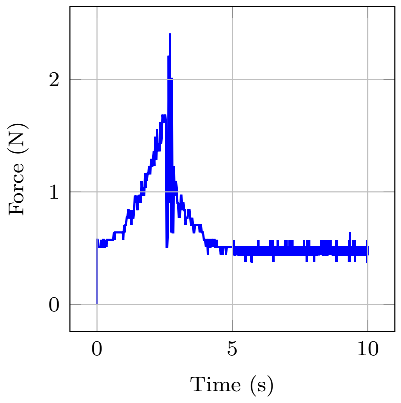


(a) Tensile Force.

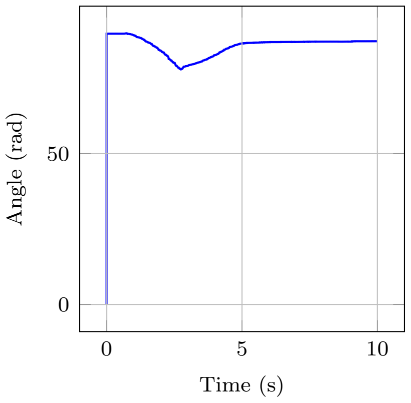


(b) Angle of the pendulum.

Figure D.3: Sawtooth with torque control.



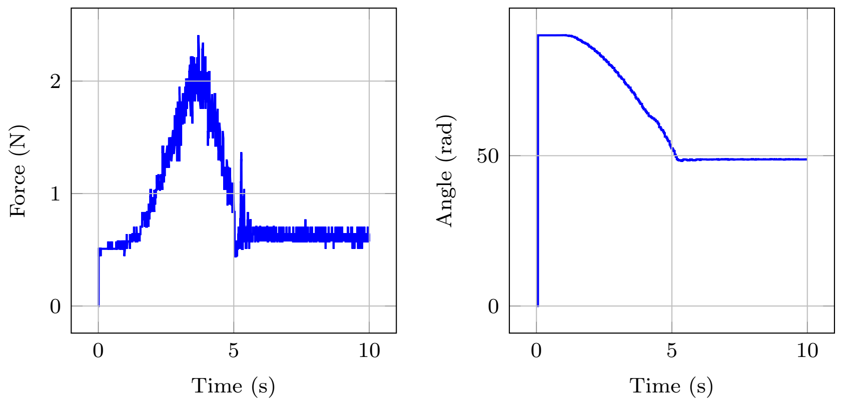
(a) Tensile Force.



(b) Angle of the pendulum.

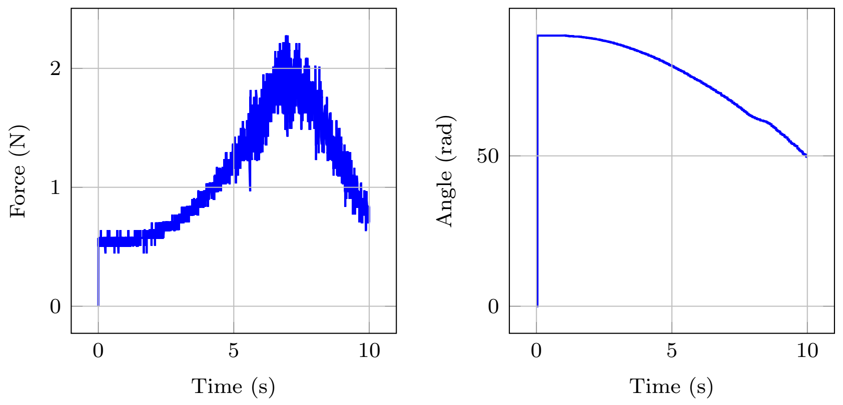
Figure D.4: Triangle with torque control.

D.2 Velocity Commands



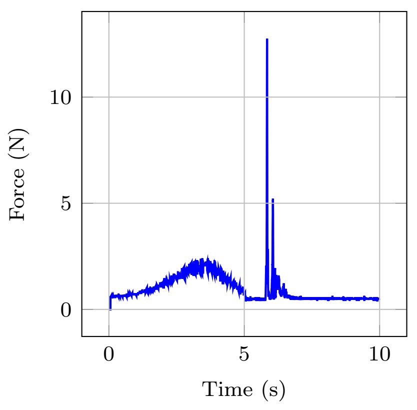
(a) Tensile Force. (b) Angle of the pendulum.

Figure D.5: Ramp step with velocity control.

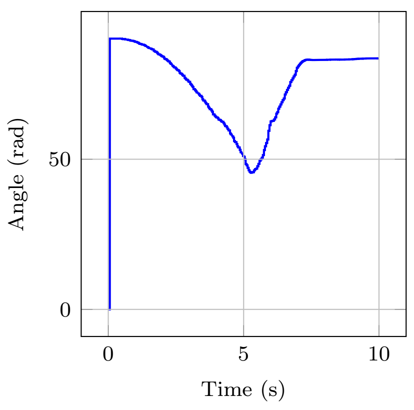


(a) Tensile Force. (b) Angle of the pendulum.

Figure D.6: Ramp with velocity control.

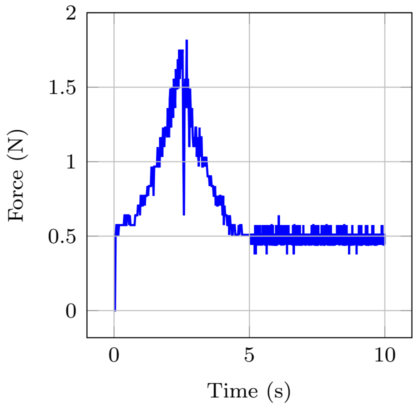


(a) Tensile Force.

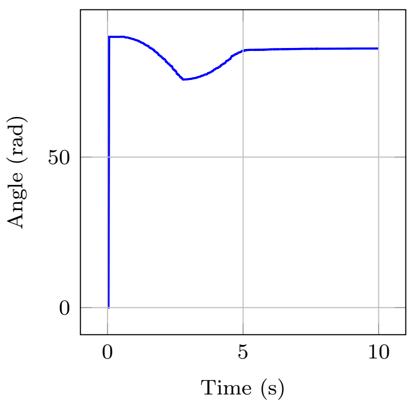


(b) Angle of the pendulum.

Figure D.7: Sawtooth with velocity control.



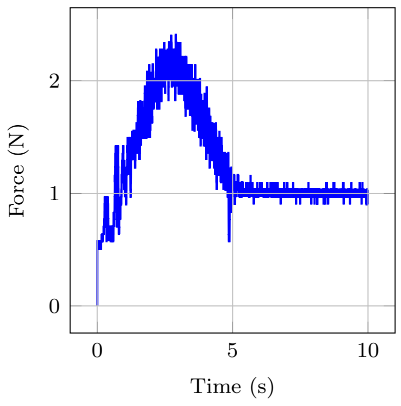
(a) Tensile Force.



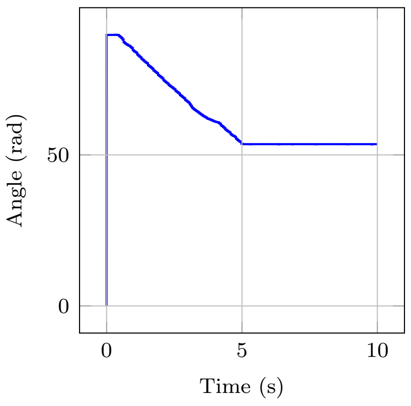
(b) Angle of the pendulum.

Figure D.8: Triangle with velocity control.

D.3 Position Commands

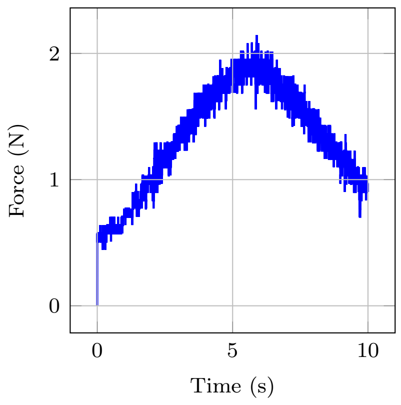


(a) Tensile Force.

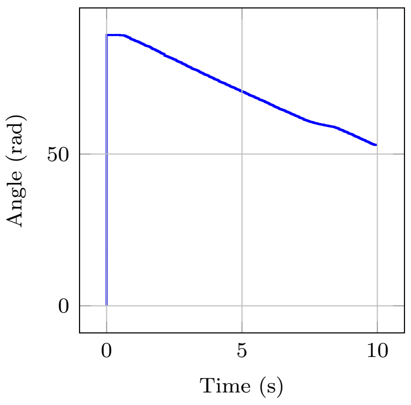


(b) Angle of the pendulum.

Figure D.9: Ramp step with position control.

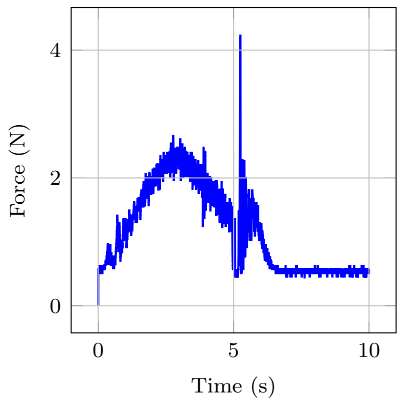


(a) Tensile Force.

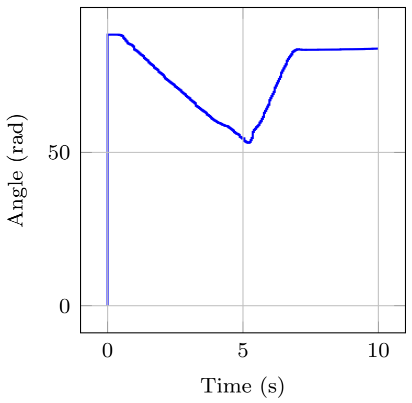


(b) Angle of the pendulum.

Figure D.10: Ramp with position control.

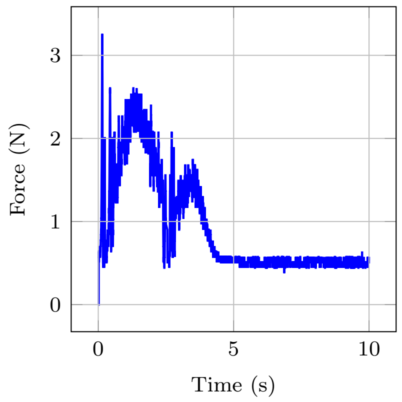


(a) Tensile Force.

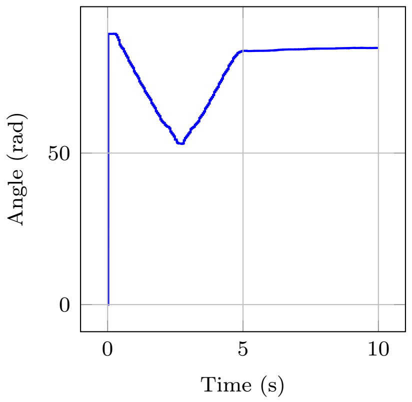


(b) Angle of the pendulum.

Figure D.11: Sawtooth with position control.



(a) Tensile Force.



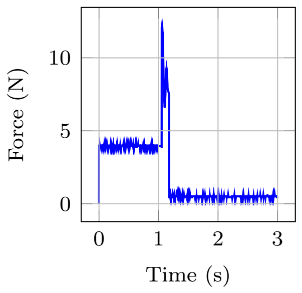
(b) Angle of the pendulum.

Figure D.12: Triangle with position control.

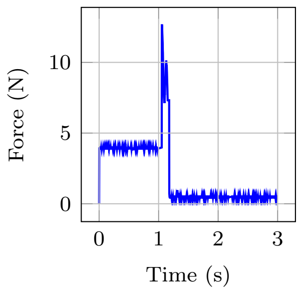
Appendix E

Second Stimuli Trials

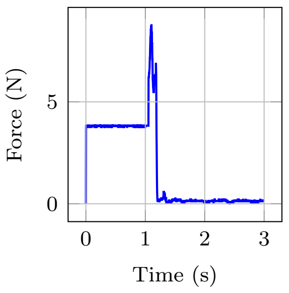
E.1 Torque Commands



(a) Tensile Force 1, actuating cable.

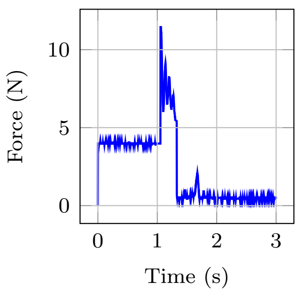


(b) Angle of the pendulum.

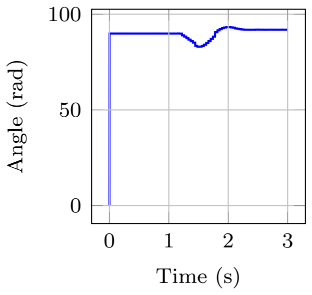


(c) Tensile Force 2, passive cable.

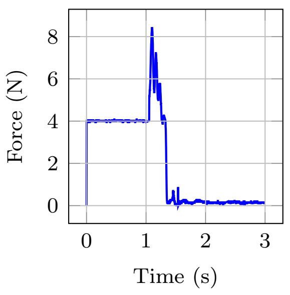
Figure E.1: Pulse with 800 torque and 0.1 s duration.



(a) Tensile Force 1, actuating cable.

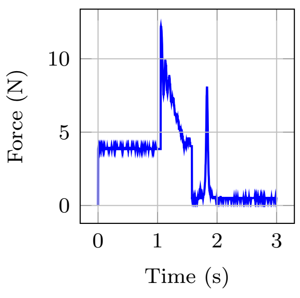


(b) Angle of the pendulum.

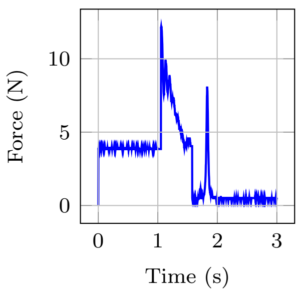


(c) Tensile Force 2, passive cable.

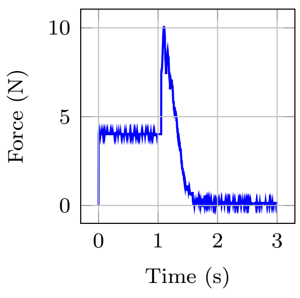
Figure E.2: Pulse with 800 torque and 0.25 s duration.



(a) Tensile Force 1, actuating cable.

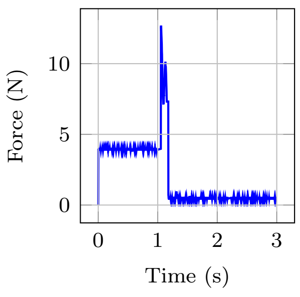


(b) Angle of the pendulum.

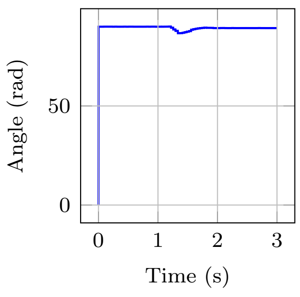


(c) Tensile Force 2, passive cable.

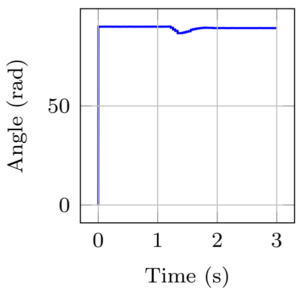
Figure E.3: Pulse with 800 torque and 0.5 s duration.



(a) Tensile Force 1, actuating cable.

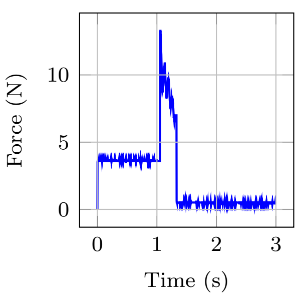


(b) Angle of the pendulum.

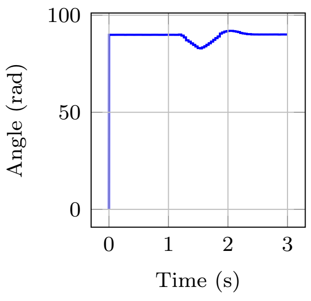


(c) Tensile Force 2, passive cable.

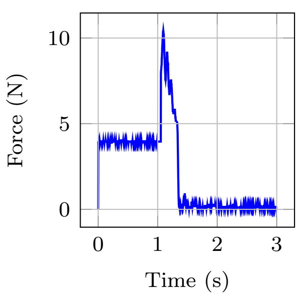
Figure E.4: Pulse with 900 torque and 0.1 s duration.



(a) Tensile Force 1, actuating cable.

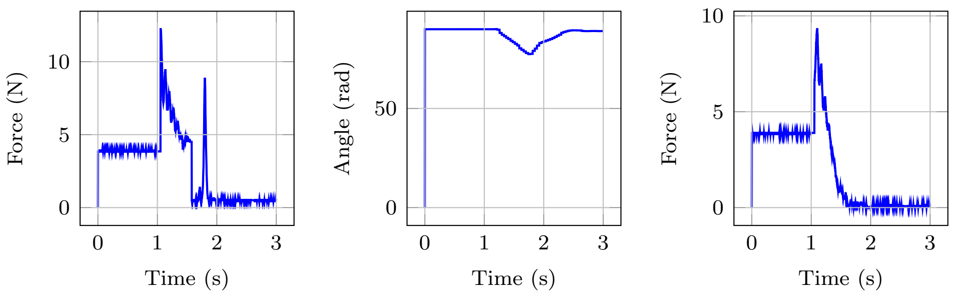


(b) Angle of the pendulum.



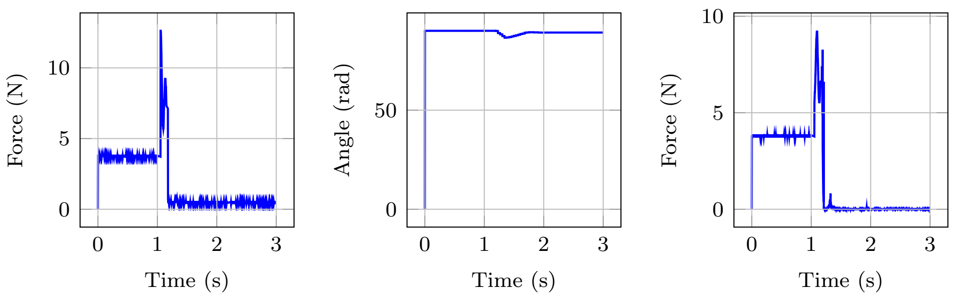
(c) Tensile Force 2, passive cable.

Figure E.5: Pulse with 900 torque and 0.25 s duration.



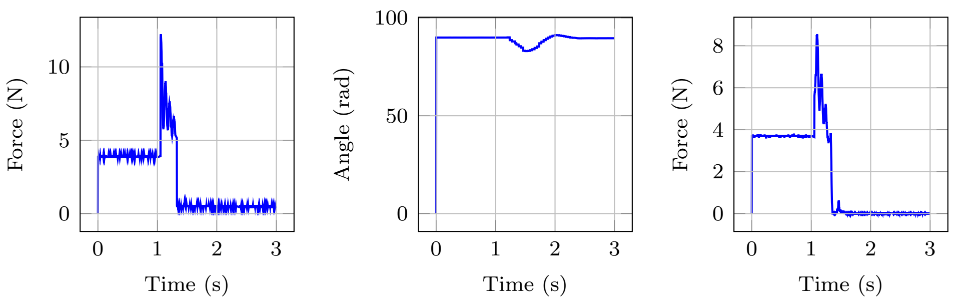
(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

Figure E.6: Pulse with 900 torque and 0.5 s duration.



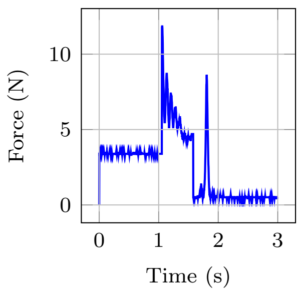
(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

Figure E.7: Pulse with 1022 torque and 0.1 s duration.

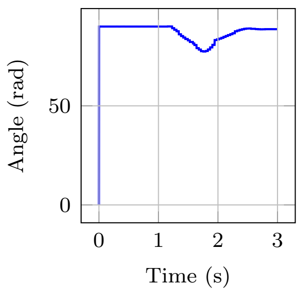


(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

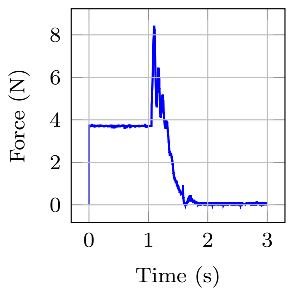
Figure E.8: Pulse with 1022 torque and 0.25 s duration.



(a) Tensile Force 1, actuating cable.



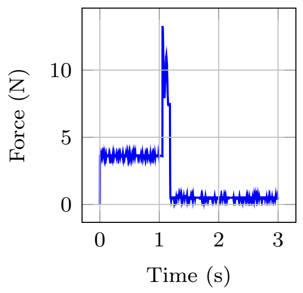
(b) Angle of the pendulum.



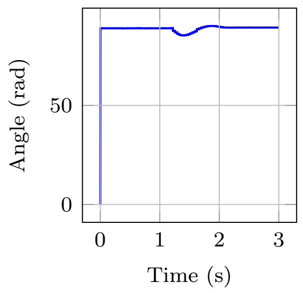
(c) Tensile Force 2, passive cable.

Figure E.9: Pulse with 1022 torque and 0.5 s duration.

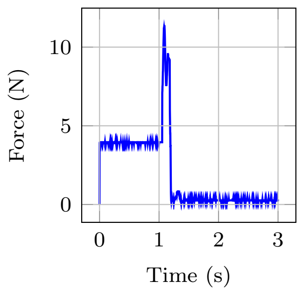
E.2 Velocity Commands



(a) Tensile Force 1, actuating cable.

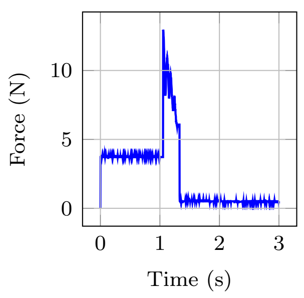


(b) Angle of the pendulum.

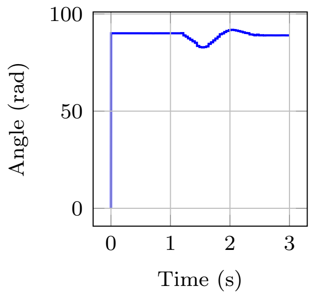


(c) Tensile Force 2, passive cable.

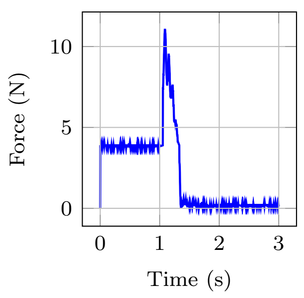
Figure E.10: Pulse with 800 velocity and 0.1 s duration.



(a) Tensile Force 1, actuating cable.

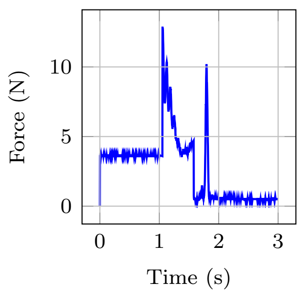


(b) Angle of the pendulum.

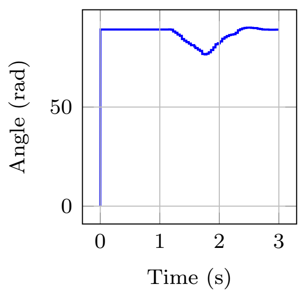


(c) Tensile Force 2, passive cable.

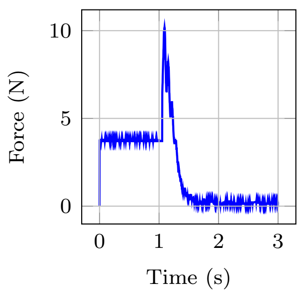
Figure E.11: Pulse with 800 velocity and 0.25 s duration.



(a) Tensile Force 1, actuating cable.

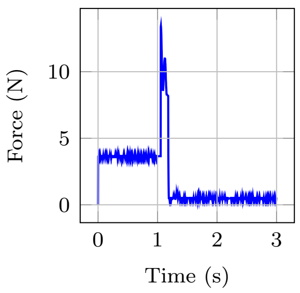


(b) Angle of the pendulum.

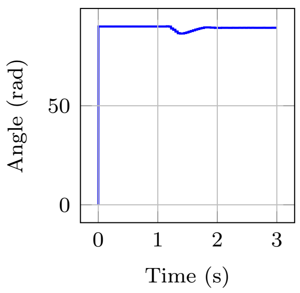


(c) Tensile Force 2, passive cable.

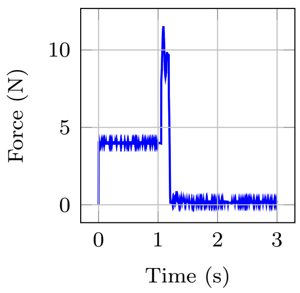
Figure E.12: Pulse with 800 velocity and 0.5 s duration.



(a) Tensile Force 1, actuating cable.

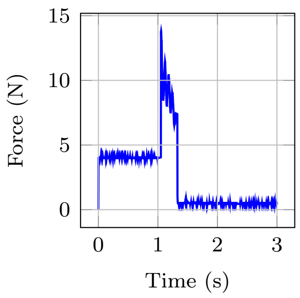


(b) Angle of the pendulum.

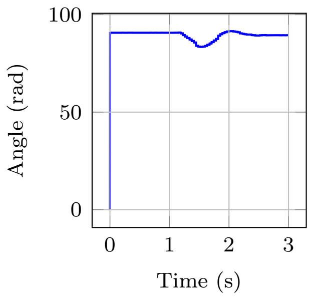


(c) Tensile Force 2, passive cable.

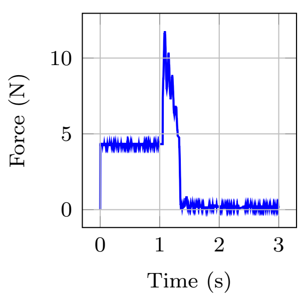
Figure E.13: Pulse with 900 velocity and 0.1 s duration.



(a) Tensile Force 1, actuating cable.

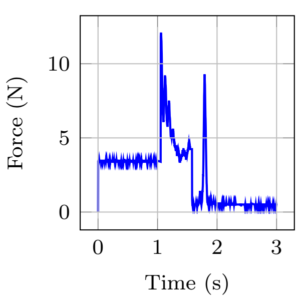


(b) Angle of the pendulum.

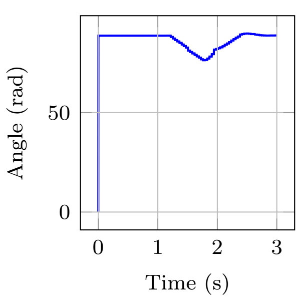


(c) Tensile Force 2, passive cable.

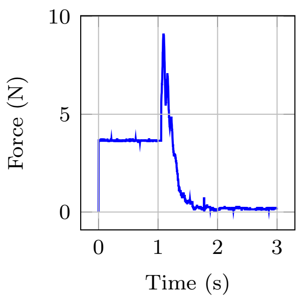
Figure E.14: Pulse with 900 velocity and 0.25 s duration.



(a) Tensile Force 1, actuating cable.

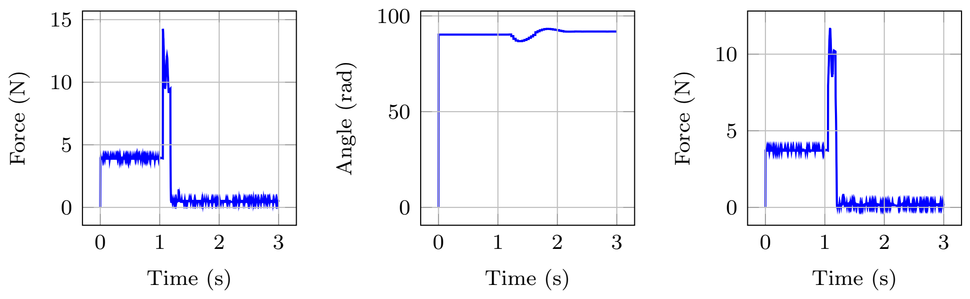


(b) Angle of the pendulum.



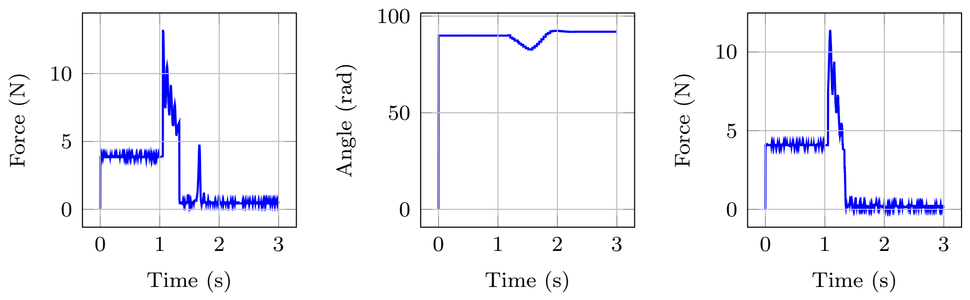
(c) Tensile Force 2, passive cable.

Figure E.15: Pulse with 900 velocity and 0.5 s duration.



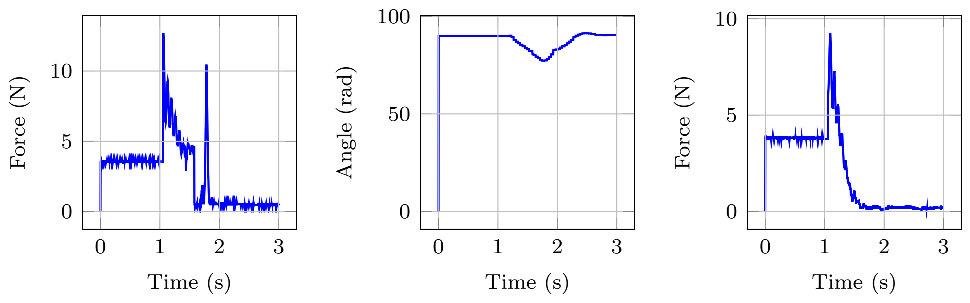
(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

Figure E.16: Pulse with 1022 velocity and 0.1 s duration.



(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

Figure E.17: Pulse with 1022 velocity and 0.25 s duration.



(a) Tensile Force 1, actuating cable. (b) Angle of the pendulum. (c) Tensile Force 2, passive cable.

Figure E.18: Pulse with 1022 velocity and 0.5 s duration.