2012-01-01

Control of a Robotic Prototype Multi-Axis Synergistic System

Julio A. Torres
University of Texas at El Paso, jatorres14@miners.utep.edu

CONTROL OF A PROTOTYPE ROBOTIC MULTI-AXIS SYNERGISTIC SYSTEM

JULIO ADRIAN TORRES FRAYRE

Department of Mechanical Engineering

APPROVED:

Thompson Sarkodie-Gyan, Ph.D., Chair

Noe Vargas, Ph.D., Co-Chair

Huiying Yu, Ph.D.

Miguel Pirela-Cruz, M.D.

Benjamin C. Flores, Ph.D.
Interim Dean of the Graduate School

Dedication

I dedicate this work to my parents Laurencia and Gabriel.

CONTROL OF A PROTOTYPE ROBOTIC MULTI-AXIS SYNERGISTIC SYSTEM

by

JULIO ADRIAN TORRES FRAYRE, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2012

Acknowledgements

I express gratitude to the people without whose support this research project would not have been possible. I wish to acknowledge my supervisor and great professor, Dr. Thompson Sarkodie-Gyan, for all his help, assistance, support and guidance, all over these two years. I take the opportunity to thank Dr. Noé Vargas-Hernández for the time he spent helping me organizing, correcting my ideas, and creating new ones.

I would like to acknowledge to Dr. Huiying Yu and Dr. Miguel Pirela-Cruz for their assistance in serving on my committee, and also for their pieces of advice during this work.

I convey special thanks to Gabriel Dávila for his significant contributions to this thesis, especially in the early stages. Similarly, I wish to recognize Pablo Rangel for his essential participation in this project.

My deepest gratitude is due to the members of the Laboratory for Industrial Metrology and Automation (LIMA), for giving me advice, with the purpose of having a better performance and for creating an excellent work environment: Luis Gerardo Sagarnaga López, Oscar Alonso Espino Flores, Jorge Garza Ulloa, Chethan Ramashandra Keladi, Melaku Bogale, Ryan Price, Murad Alaqtash and Beverly Rivera.

Abstract

Balance is an ability that every person has and uses in normal daily life, to walk, run, stand, dance, etc. However, there are people who suffer a kind of balance impairment. The Computerized Dynamic Posturography (CDP) is technique in balance rehabilitation, where the patient is asked to stand on a moving platform that is equipped with sensors that measure how well you maintain your balance. In this thesis, a new CDP mechatronic system, the Multi-Axis Synergistic System (MASS), for balance assessment is being developed.

As a first stage and as the objective of this research work the transferable control algorithm for position of a prototype MASS was developed. With the intention of achieving the mentioned objective; the inverse kinematics of a parallel mechanism in vector technique was utilized. The complete inverse kinematics analysis of the MASS was revised in this work. Correspondingly, the transferable programming of the algorithm was explained. Then, a series of experiments were performed in order to identify the electromechanical actuator of the MASS that resulted in the first order dynamic system model, in both analytical and experimental methods. After, several movements in the six degrees of freedom of the platform, through the programmed control algorithm were simulated in this work. The results are presented and found that the obtained first order model was the correct one for the actuator, however, some considerations relating to the error in transient state are discussed. Finally, the results of the simulations of the transferable algorithm and final conclusions are presented and the future work is defined.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The prototype robotic Multi-Axis Synergistic System (MASS) is a mechatronic device, based on the fundamentals of parallel mechanisms, which was produced for a very specific task: to develop the algorithm for the control of the MASS in Figure 1.1, a machine for the effective and efficient treatment and assessment of neuromuscular diseases relating to balance and posture, and that may also be used for training in different areas like sports, in the military and in space exploration. This prototype system consists of several development phases for different fields of engineering and other specialized areas, such as biomedical. Therefore, it exhibits high significance for multidisciplinary research.

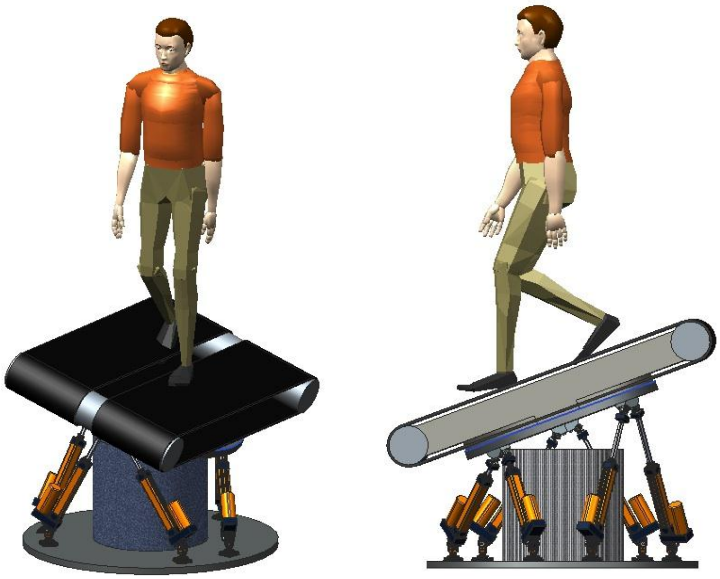

Figure 1.1: Multi-Axis Synergistic System (MASS) for assessment of balance and posture

It is noteworthy that despite the objectives of this thesis, there is also the development of the algorithm for the control of the MASS applying the control engineering theory. This was performed for the preliminary technical work to achieve the complete electromechanical functioning of the system, and

an exhaustive study of market was conducted for comparison with other existing commercial devices. Both preliminary works are also described along this writing in conjunction with other basic topics in which this development has its fundamentals, as inverse kinematics, that are found in Chapter 2.

1.1 Motivation

Balance is an ability that every person has and uses in normal daily life to perform many common activities such as walking, running, dancing, standing, sitting, etc. However, there is a significant number of people who suffer a balance impairment (National Institute of Health, 2009), i. e., they cannot perform even relatively easy tasks like other people.

Currently, there are different techniques to help people with this kind of disorders. We can find a wide range of physical examinations of all the systems involved in balance. Also, we can find several tests that medical doctors perform to balance impaired patients. Even, there exist medical devices to treat balance disorders [1]. However, the accuracy, repetition and reliability in the very first phase of rehabilitation, i. e., diagnosis of the patient, are very important characteristics in rehabilitation devices. In the most of the cases, those characteristics may be definitive for the plenty rehabilitation and recovery of the patient.

This situation requires research groups focused in the improvement of mentioned characteristics where the current medical devices can be enhanced. In the same way is very important the developing of new systems or medical devices for the treatment and assessment of balance impairments, which must consider the quantification; identification and differentiation at the moment of diagnose a patient.

Nowadays, a patient of Parkinson's disease is classified as just that, without considering that there could be significant differences between patients of the same disorder. Even when there are subcategories within the balance disorders, a patient who suffers of Multiple Sclerosis is affected in his or her balance [2], but must not be treated as the rest of Multiple Sclerosis patients. Here is where the

importance of this thesis lies. The motivation for this research is the desire to give back encouragement to all those patients affected by a balance disorder, and increase their quality of life.

1.2 Objectives

The problem that this thesis solves is the development of an algorithm of position control for the MASS device. However, this research work has very specific objectives. The control algorithm developed was created and tested on a prototype MASS. However, the followed developing approach needs to ensure, that the same algorithm will function in the real MASS, when it is created in the next years. Knowing the requirements, there are three specific objectives that define this thesis. The solutions to these three objectives are the main contributions that the author is defining at the end of this thesis. The three objectives are: a scaled prototype of the MASS, transferable control algorithm and closed-loop control approach.

1.2.1 Scaled prototype of the MASS

A scaled model prototype of the MASS was developed. There were different phases where different people were working on it. The mechanical design and also the assembly were achieved by a senior project team, from the Mechanical Engineering Department, under the supervision of Dr. Noe Vargas. After that phase, the electric and electrical parts connections and wiring procedures are part of this thesis, which are described in the following chapters.

1.2.2 Transferable Control Algorithm

The construction of the prototype is a short-term solution to the MASS, due to limitations in funds and time. For this reason, the control algorithm developed in this work is completely transferable, since the approach followed allows using the same program to control any system with the same architecture of the MASS, only with the definition of four parameters that are also discussed in Chapter 2.

1.2.3 Closed-loop control approach

The closed-loop control approach from Classic Control Theory is used in the algorithm. The actuators are systems of first order and have their own feedback sensors. An analysis of the feedback response of them is given. Characteristics as speed, settling time, time constant are shown. Also simulations of the behavior are analyzed.

Chapter 2: Literature Review

2.1 Posture, balance and its disorders

Posture is the characteristic that describes the position of any segment of the human body relative to the gravitational vector and is an angular measure from the vertical axis [3]. Balance is a term to describe the dynamics of posture to prevent falls and is related to inertial forces acting on the center of mass of the human body [3]. Balance is also defined as the ability that we have to keep our center of mass over its support base when walking, standing, running, dancing, etc. In Figure 2.1 the balance or unbalance condition is illustrated, during the activities mention before.

When people have proper functioning of balance, they can know their exact orientation with respect to gravitational vector and also determine in which direction and speed they are going to move. Consequently, they can make automatic adjustments in their posture to keep their center of mass within its support base and stability in different daily situations and activities [4].

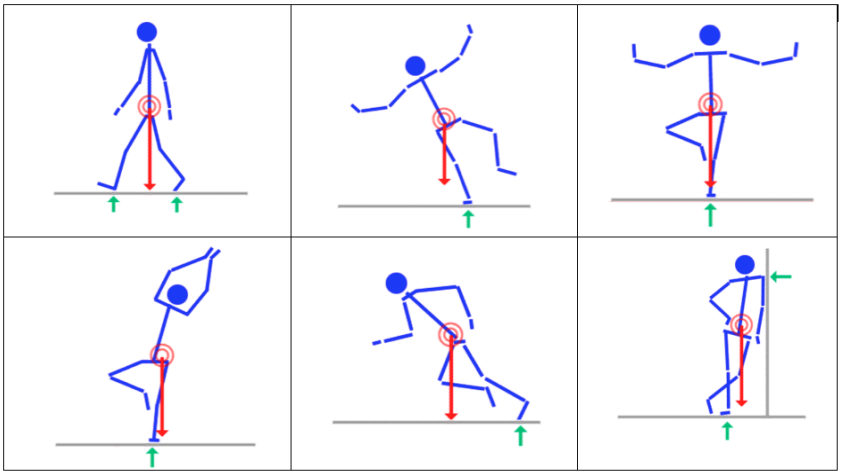

Figure 2.1: Balance or unbalance condition. Red circle: center of mass, green arrows: support bases.

The condition of proper balance is frequently taken for granted, however the complexity of the

balance system increases by itself the probability of having difficulties while balancing, which in most cases ends in an impairment. With impaired balance, activities even as stand up from a chair can be very fatiguing and also dangerous. Balance complexity lies on the fact that it is a coordinated response of the neuromuscular and musculoskeletal systems, as well as three main sensorial mechanisms that interact with each other: visual, vestibular and proprioceptive, as shown in Figure 2.2. Also in Figure 2.3, we can find the sensors used in balance and posture.

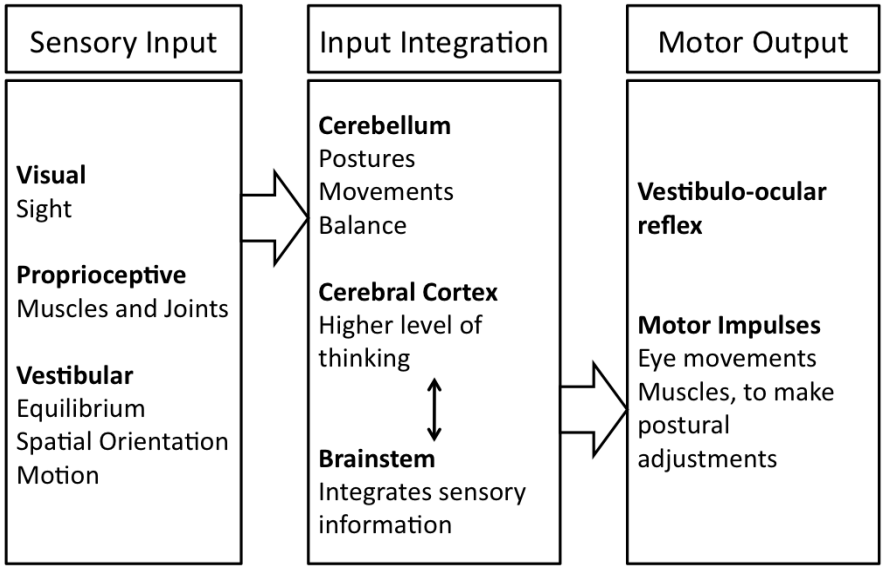

Figure 2.2 Systems involved in balance. Three main systems: visual, proprioceptive and vestibular.

Since, in balance we use reflexes to maintain stability in posture by extending muscles in the direction of an anticipated falls, failures in one of these systems can be compensated for by reliance on the other two systems. However, this defect reduces our ability to keep our center of mass over its support base, occasioning difficulties to realize daily activities that need balance, not allowing us to have a good quality of life. The mentioned defects are also known as a balance disorders that are disturbances that causes a person to feel unsteady, when standing, walking, dancing or running.

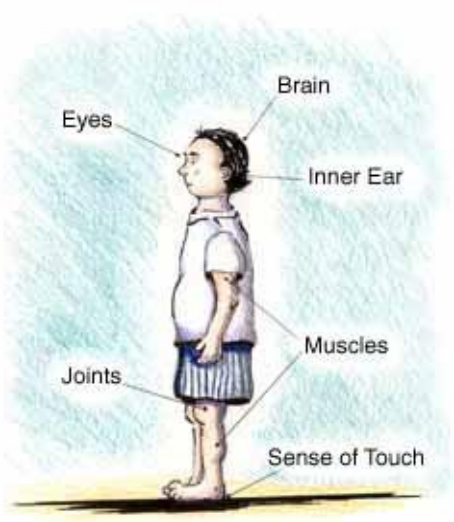

Figure 2.3 Balance sensors

There are different causes that affect a good balance. Some of them are related to loss of strength, generally starting in the feet. The cause of this imbalance is a muscle disease occasioned by general weakness in the thighs and shoulders first. Muscle and nerve disease could occur at any age. However, those that are aimed in the MASS development are the brain related causes such as degenerative, related to aging (it is commonly assumed that loss of balance is part of aging and patients do not look for medical attention); circulatory, with cases of cerebral ischemia and stroke; systemic, where we find multiple sclerosis (MP) or Parkinson's disease. In a lower range, there is another type of balance disorder: the Cerebral Palsy (CP) Ataxic Type, being the 10% of CP cases (Cerebral Palsy Source, 2008) [5].

A total of 8 million American adults have a chronic problem with balance. Also it is known that an additional 2.4 million American adults suffer of chronic problems with dizziness alone (National Institute of Health, 2009).

Although that alarming situation, it is possible the balance rehabilitation, since the learning of this ability is based on the practice and repetition methodology. When we learned to balance, impulses

were sent from the sensory systems to the brain and then are sent from the brain to the muscles creating paths of communication. Those paths can be recovered, if not in a 100% of functionality, but it is possible to give back a significant improvement. There are a variety of options for treating balance disorders [1].

Nevertheless, the main concern in this important possibility is at the diagnosis phase, the accuracy, reliability and repetition of medical tests is crucial. To determine the cause of a balance problem, first the subject is asked to describe the symptoms followed by a physical examination. Based on the subject's descriptions and examination results, the doctor develops a working diagnosis. Depending on the doctor's determination, some tests can be designed to confirm the diagnosis and also establish how key elements of subject's balance are functioning well or not. Table 2.1 shows the physical examinations and tests that are performed on subjects that suffer balance impairments [6].

Table 2.1: Physical examinations and tests performed on balance patients

Physical Examination	**Tests**
Assessment of Eye and Head Movement Functions Assessment of Cerebellar Function Assessment of Walking Function	Audiometric Nystagmography Positional Nystagmus Computerized Dynamic Posturography. Rotational Chair

From the different physical examination and tests, the Computerized Dynamic Posturography (CDP), in Figure 2.4, is the one of special interest and is a basis for this thesis. CDP is a series of tests that measure how well a subject is able to keep its balance under different conditions. The subject is

asked to stand on a moving platform, which has sensors that measure how well you maintain your balance. Some of the tests are designed to emulate different conditions you can have daily life. CDP is one of the most widely available of the technological options together with force plates, which measure subject's center of pressure (COP) [1],[7].

Figure 2.4: A Computerized Dynamic Posturography (Balance Quest™).

In Chapter 3, a detailed research of the state-of-the-art of the medical devices used for CDP is discussed.

2.2 Parallel robots and Stewart Platform

A parallel robot is made up of an end-effector with n degrees of freedom, and of a fixed base, linked together by at least two independent kinematic chains. Actuation takes place through n simple actuators [8]. The architecture of a parallel robot is illustrated in Figure 2.5.

Parallel robots are closed-loop mechanisms that have specials characteristics to present higher performances than serial robots in relation with capacity to support big loads, precision, rigidity, accuracy, inertia of moving parts [9]. The fact that the load is supported by more than one links allows us to have a high payload-to-weight ratio and high rigidity. The multiple links provide the high accuracy and not accumulating errors caused by serial joints. The disadvantages of the parallel robots or manipulators are limited workspace, low agility, complicated direct kinematics solution [10] (in this work is analyzed the inverse kinematics solution to solve control the MASS).

Figure 2.5: Parallel robot ABB FlexPicker.

Parallel robots or manipulators are used for many applications in a wide range of areas like: astronomy, flight simulators, manufacturing machine-tool industry and, for their high precision, medical surgeries. The implementation of parallel robots was first introduced for applications out of the medical field, but currently, the research of parallel robots for medical applications is increasing in a considerable manner. In Figure 2.6 a concept for a medical robot applicable to chest compression in the process of Cardiopulmonary Resuscitation (CPR) is shown [11].

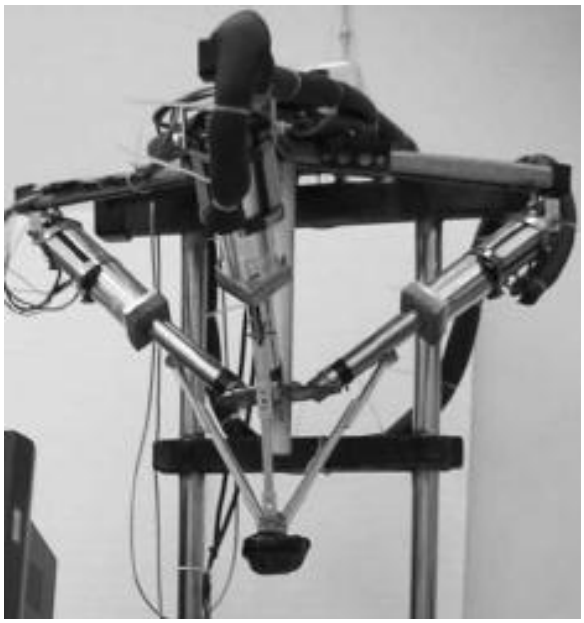

Figure 2.6: Medical robot concept for Cardiopulmonary

The Stewart Platform, in Figure 2.7, is one of the most common architecture of parallel robots with six degrees of freedom. Stewart presented it first in 1965, as a flight simulator [12]. This parallel mechanism was composed of a base and a triangular moving platform with three extensible links connecting the moving platform to the base. This device is frequently called a synergistic motion platform, because the motions are produced by a combination of movements of several actuators. Stewart platforms are applied in machine tool technology, crane technology and orthopedic surgery.

Their high accuracy is reached by using high-precision linear axes and high-precision joints. The upper platform of the Stewart Platform robot serves as a universal platform for different types of surgical instruments. These are developed specifically for different disciplines [13]. In Figure 2.8, a medical robot based on the Stewart platform is presented.

Figure 2.7: Example of a Stewart Platform (M-840 HexaLight™)

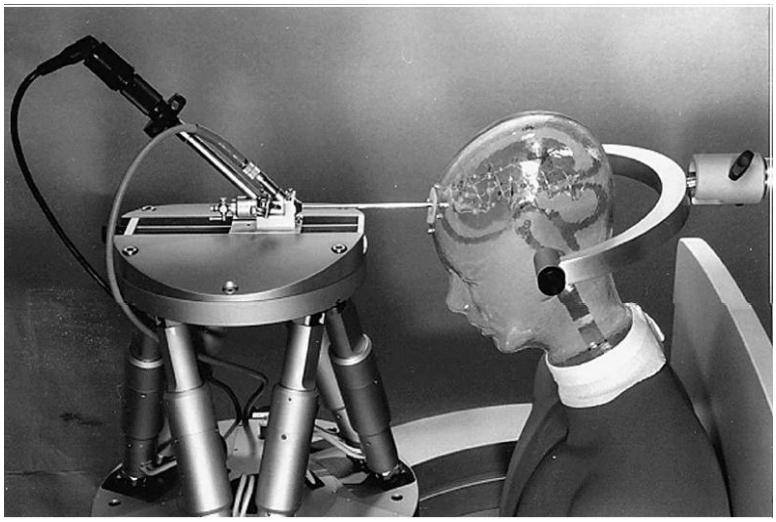

Figure 2.8: Medical robot based on the Stewart platform for precision surgery

2.3 Classic Control Theory

Classic control theory studies the behavior of dynamical systems and their control, and consists of several tools both mathematical and engineering. The main objective of the control theory is to compute solutions for the proper regulation of a variable (process variable) through the time, to make it stable and prevent it from oscillating around a reference. The reference is an external input to the dynamic system, which is the desired value in the process variable. The regulation of the process variable is performed by the controller, which manipulates the input to a system to obtain the desired effect on the process variable of that system. There are a number of key elements in the control scheme used in this project, as you will see described in this chapter. The control technique used is the closed-loop control, found in the classic control theory.

When dealing with dynamic systems, besides the elements of a closed-loop control technique, some concepts, such as transfer function, settling time, death time, system identification, the time constant, the response of a system and step input, need to be understood. These concepts are also explained in this chapter.

2.3.1 Closed-loop control

The closed-loop control is a method for systems that use feedback. The controller utilizes the feedback to make decisions, with the aim of change the control signal and manipulate the dynamic system. Contrary to closed-loop control, an open loop control does not have feedback.

A basic closed-loop control system is shown in Figure 2.9. This figure can describe a diversity of control systems, including different kind of processes like temperature, level, speed, to mention some [14].

Closed-loop control systems works by this principle: After reading each new reading from the sensor (feedback) measuring the process variable $y(t)$, the controller reacts to the system's new state by computing and correcting the control signal. The system is manipulated by the control signal and it responds to this change, another reading is measured, and the cycle starts again. The process is repeated until the system reaches the desired state reference $x(t)$, and the controller stops making decisions and manipulating the system. The cycle starts again when there is a disturbance in the system and it changes the real value of the process variable or when the reference was changed to another desired value by the user.

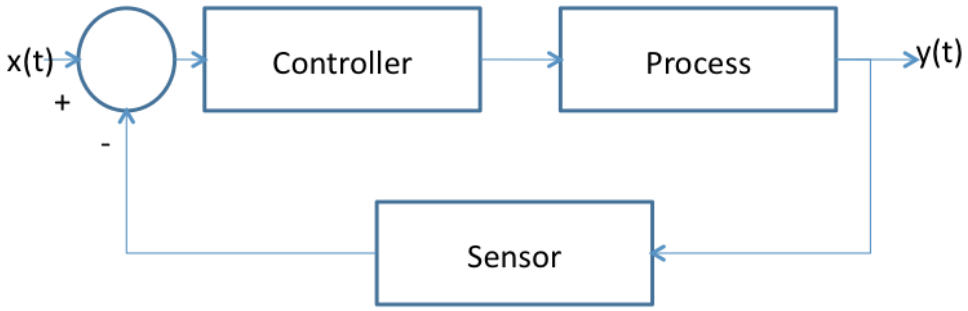

Figure 2.9: Closed-loop control diagram

2.3.2 Transfer function

A transfer function is a mathematical representation, in the frequency domain, of the relation that exists between the output and input of dynamic system and is represented as $H(s)$, with initial conditions equal to zero. Transfer functions are very used in the analysis of dynamic systems in control theory. It is important to mention that, with the intention of obtain the transfer function of a system; this system has to be linear and invariant with time. Although most real systems have non-linear behaviors,

many systems, when functioning in nominal parameters, have characteristics very close to linear, so transfer function is a suitable representation of those systems [15].

So as to obtain the transfer function of a dynamic system, it is necessary first to get the Laplace Transform of the differential equation(s) of the system. Laplace Transform $F(s)$ of a function $f(t)$ is defined as follows [14]:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t) dt$$

When the Laplace Transforms of the input and the output are obtained, under initial conditions equal to zero, the transfer function can be obtained by performing the division of the output over the input [14]:

$$H(s) = \frac{Y(s)}{X(s)}$$

Once the transfer function is obtained, it is possible to know the time history a system by performing the Inverse Laplace Transform of the transfer function multiplied by the input of the system [14]:

$$Y(s) = H(s)X(s)$$

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = \mathcal{L}^{-1}\{H(s)X(s)\}$$

The Inverse Laplace Transform is defined as follows:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi i} \lim_{\gamma \to \infty} \int_{\gamma-\pi}^{\gamma+\pi} e^{st} F(s) ds$$

However, in control theory, there are methods such as partial fractions, that make this process more direct, allowing even the use of tables of common transform pairs to convert from time domain to frequency domain or vice versa [14].

2.3.3 First order systems and time constant

A first order system is defined as the system whose general equation has one differential element on the left side and can be reduced to the next format:

$$\tau \frac{dy}{dt} + y(t) = x(t)$$

Where τ is the time constant of the system and depends of the physical characteristics of the system and its nature, that is to say if the system is mechanical, electrical, etc.

The transfer function of a first order system is then:

$$\tau[sY(s) - y(0)] + Y(s) = X(s)$$

$$\tau sY(s) + Y(s) = X(s)$$

$$(\tau s + 1)Y(s) = X(s)$$

$$Y(s) = \frac{X(s)}{\tau s + 1}$$

$$H(s) = \frac{Y(s)}{X(s)} = \frac{1}{\tau s + 1}$$

The time constant is a very significant concept in control theory. It is the main characteristic of a first order system. Physically, the time constant represents the time that the system takes to reach the 63.2 % of its final (asymptotic) value to the step response. A first order system reaches the 98.17 % of its final value (which is considered the final value under an error bandwidth of 2%) at a time equal to four times the time constant [14]. In Figure 2.10 the step response of a first order system is shown.

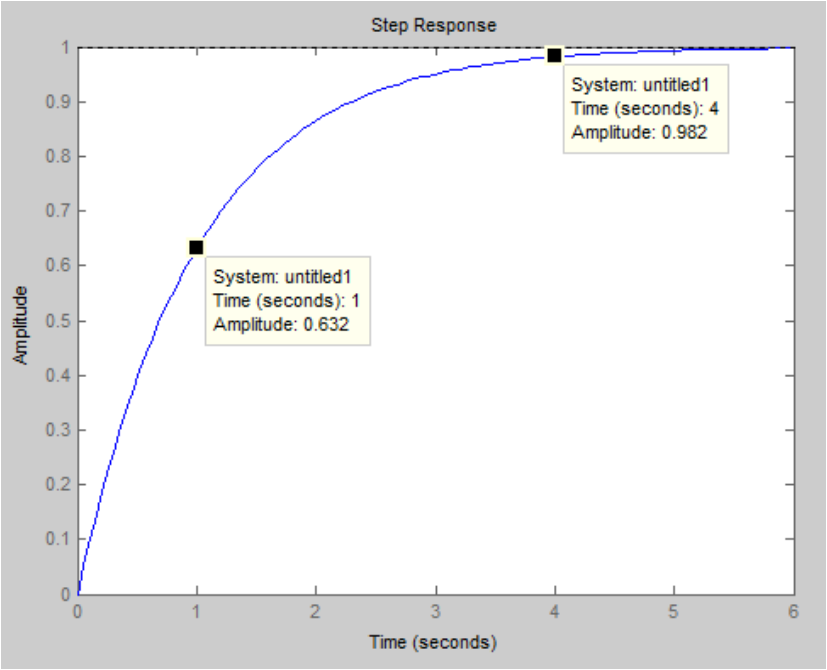

Figure 2.10: Step response of a first order system with time constant $\tau = 1$

The step response of a system is the output behavior in time, when its input is changed from zero to one in a very short time.

2.4 Inverse Kinematics

With the aim of define the inverse kinematics problem, understand *kinematics* as the study of the geometry of motion. *The word comes from Greek "kinema" that means movement. It takes place in accordance with the laws of dynamics. If a body such as a part of a machine is constrained to move in a certain manner or with a certain degrees of freedom, by virtue of its contact with other machine elements, its motion may be determined by kinematics.* [17]

The function of kinematics is to make sure the functionality of a mechanism, and the role of dynamics is to verify the acceptability of induced forces in parts. The functionality and induced forces are subject to various constraints depending on the mechanism design. An example of a cam operating a valve mechanism is shown in Figure 2.11 [18]

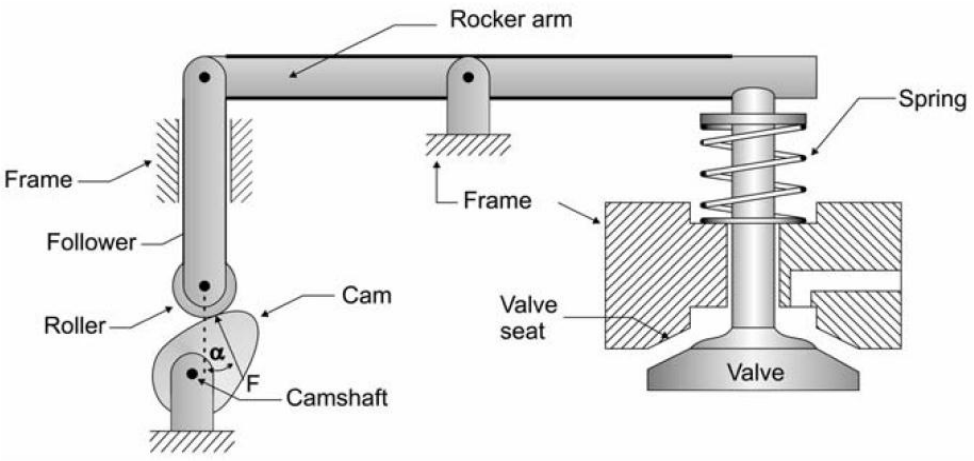

Figure 2.11: Cam operating valve mechanism

The problem of inverse kinematics is to find the joint angles, given the end effector position and orientation. In a mechanism, end effector is the part that interacts with the environment and its structure depends on the task the mechanism will be performing.

In Figure 2.12 an inverse kinematics problem is presented.

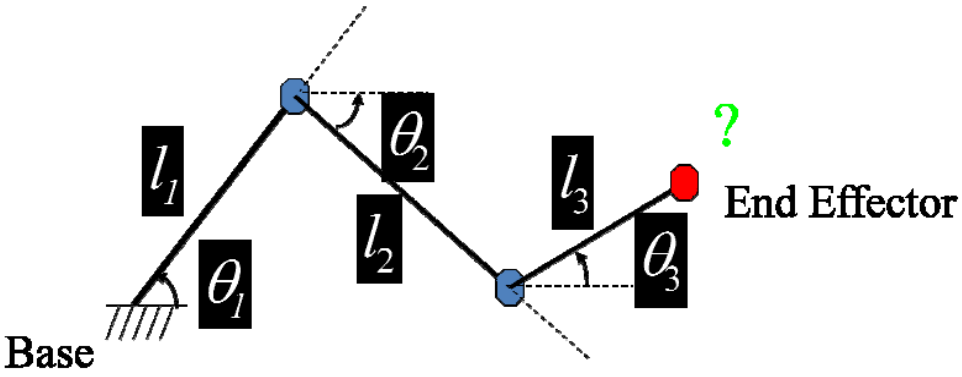

Figure 2.12: Inverse kinematic problem

If the position of the end effector (x, y) is given and knowing the three links length l_1, l_2 and l_3, find the angle joints θ_1, θ_2 and θ_3. The solution to this problem is hard, since we have only two equations for the system and three unknowns.

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3)$$

$$y = l_1 \sin(\theta_1) - l_2 \sin(\theta_2) + l_3 \sin(\theta_3)$$

In fact, for this problem there are multiple solutions. In general, inverse kinematics is more complex than forward kinematics [19], whose solution for the same problem (find the position of the end effector, given the joint angles) is straight forward, since we have two equations for two unknowns.

In fact, this is a very simple example in two dimensions. When analyzing three-dimensional mechanisms, the work becomes complex. Sometimes there is not analytical solution and an iterative process is necessary. In the case of redundant manipulators, there are many solutions from which we need to choose. Another consideration is that workspace limits may be over passed, where the point is outside the limits of the manipulator, or the joint limits are exceeded. For the MASS system, a workspace analysis is proposed as the next phase after this thesis.

Chapter 3: State-of-the-art

The current state-of-the-art in this field is vast and includes patents and commercial devices. A benchmark was done, with the purpose of know the commercial devices that already exist and know their advantages and disadvantages.

3.1 Patents

There are numerous patents related with the balance and posture rehabilitation field. However, they show devices like platforms with three degrees of freedom. Some of the are even only of two degrees of freedom, which limits the mobility of the device, not allowing having a good emulation of the reality, in daily events.

Assistive devices, called balancers, are human locomotion rehabilitation devices, which stimulate the lower extremities of the subject, changing the position of the base that supports the patient, modifying his or her balance and posture. Some examples of these devices are shown below.

The "Apparatus for the diagnosis and rehabilitation of balance disorders", in Figure 3.1, allows rotation off the base in the sagittal and frontal axes; this is two degrees of freedom system [20].

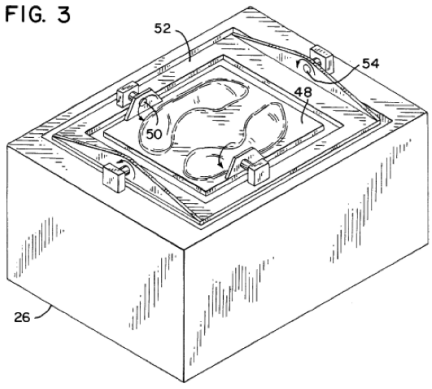

Figure 3.1: "Apparatus for the diagnosis and rehabilitation of balance disorders"

The "Adjustable Instability Apparatus for exercising, balancing, recreation and physical rehabilitation activities", in Figure 3.2, has around the x, y and z axes. In other words, it is three degrees of freedom system. However, these kinds of devices have their pivot points centered, which results in a limited range of motion [21].

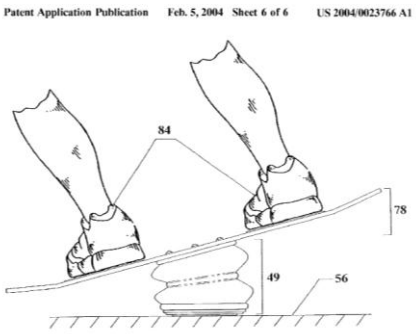

Figure 3.2: "Adjustable Instability Apparatus for exercising, balancing, recreation and physical rehabilitation activities"

Devices such as the "Balance Training Apparatus", in Figure 3.3, can produce an evoke potential on the subject by applying a force to the tilting mechanism which will change the base angle with respect to the ground [22].

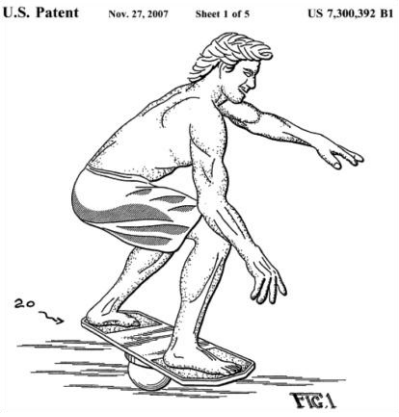

Figure 3.3: "Balance Training Apparatus"

There are other devices like the “System and method of balance training”, in Figure 3.4, that include open loop control system and involve the patient in the interaction with programed routines or games that react based on the balanced position of the patient [23].

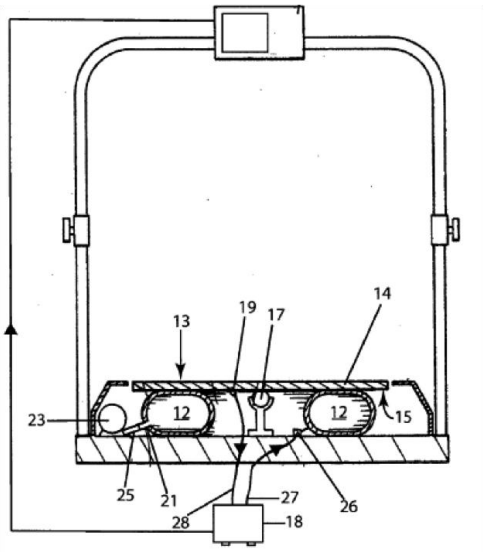

Figure 3.4: “System and method of balance training”

3.2 Commercial devices

In the commercial devices research there are five characteristics to identify the different products. These characteristics are very important in this project development, since they are the indicators of functionality in the current products for balance and posture rehabilitation. The first characteristics are the degrees of freedom, which gives us the quality in the emulation of more kind of movements. The sensors that the machines have or the variables that they measure are the second characteristic, and are important if we look for include closed-loop feedback from the patient to the machine. The type of control is other element that we classified in the commercial devices research. The

user interface is also important. Lastly, the applications which the machine is designed for are also of our interest.

Table 3.1 summarizes the commercial products found. They are classified in three levels of complexity: low, medium and high; depending of the characteristics mentioned previously. The number under the complexity levels is the number of devices found. In Figure 3.5, 3.6 and 3.7, devices of low complexity, medium complexity and high complexity are shown, respectively.

Table 3.1 Commercial devices for balance and posture rehabilitation

	Low complexity (7)	Medium complexity (3)	High complexity (1)
Degrees of freedom (DOF)	0	2-3	6
Measuring	Force, Center of pressure	Force, Center of pressure. Trunk movement in 6 DOF (1)	Force, Center of pressure. EMG. Cameras system.
Control	No control (4) Open Loop (2)	Open-Loop	Open-Loop
User Interface	Screen Results in real time	Results in real time	Virtual Reality
Applications	Medical, Sports	Medical, Sports	Medical, sports

Figure 3.5: Balance Test System. Changzhou Jianben Medical Rehabilitation Equipment Co., Ltd.

Figure 3.6: PROPIO 5000. Perry Dynamics

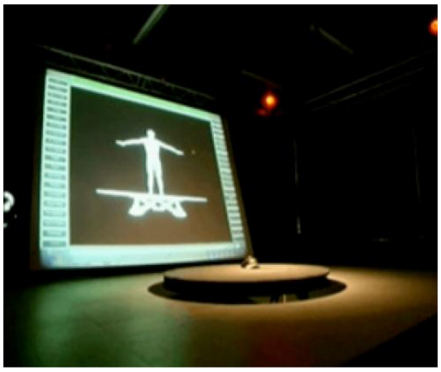

Figure 3.7: CAREN. MOTEK

Chapter 4: System architecture and development

In this chapter are described all the processes for the implementation for the MASS prototype (in Figure 4.1 a) and b) after the mechanical assembly, the hardware used and the general system architecture diagram. The mechanical design, manufacturing and assembly of this prototype are product of a research work, under the direction of Dr. Noe Vargas, from The University of Texas at El Paso. The electrical/electronic parts, the programming, inverse kinematics analysis, system identification and simulations are the core work of this thesis.

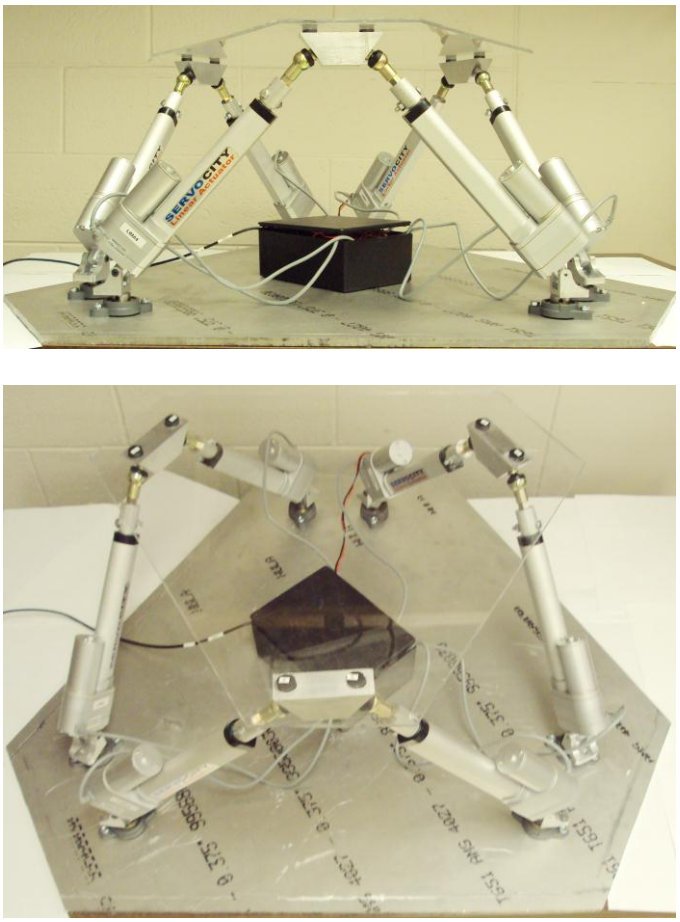

Figure 4.1: The prototype robotic Multi-Axis Synergistic System (MASS).

4.1 System architecture

The complete functioning of the prototype depends on several phases. Firstly, the global architecture is presented in Figure 4.2, the general flow diagram with the connections between components. After that, every single hardware element is described in Section 4.2.

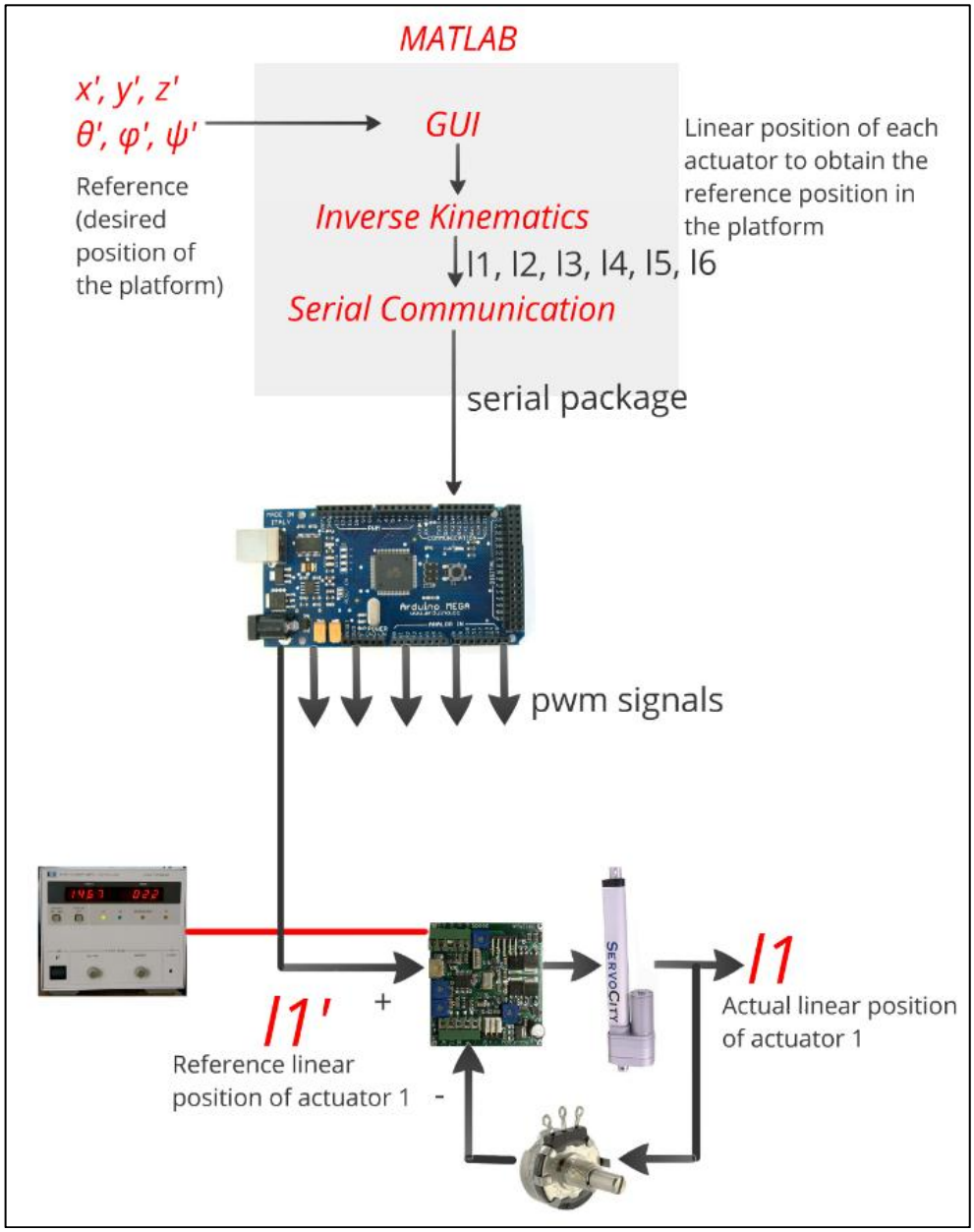

Figure 4.2: MASS general architecture

Everything begins with the reference position of the MASS, which is the desired position of the platform. The desired position of the platform is the position of the center of the moving (top) platform, in reference to the fixed (bottom) platform, as shown in Figure 4.3. The reference position consists of six parameters that are:

1) Linear displacement along x-axis,

2) Linear displacement along y-axis,

3) Linear displacement along z-axis,

4) Rotation around x-axis, which is known as roll θ,

5) Rotation around y-axis, also called pitch φ and

6) Rotation around z-axis, or yaw ψ

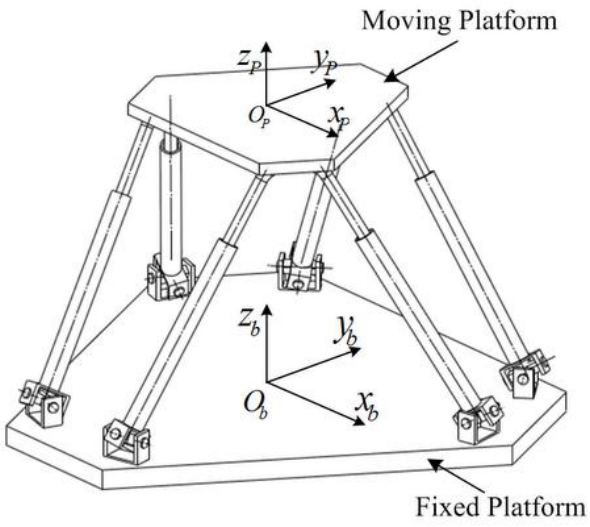

Figure 4.3: MASS moving and fixed platforms

The control of reference position has different levels of difficulty, depending if the user (manual control) or the controller (automatic control) is moving the platform:

1) Manual up and down movements
2) Manual left and right movements
3) Manual translation and rotation
4) Automatic translation and rotation

The reference position is given by a user, through a Graphical User Interface (GUI). The GUI and also the next programming phases, including the inverse kinematics process, as shown in Figure 4.2, are done in MATLAB. The developed GUI is shown in Figure 4.4.

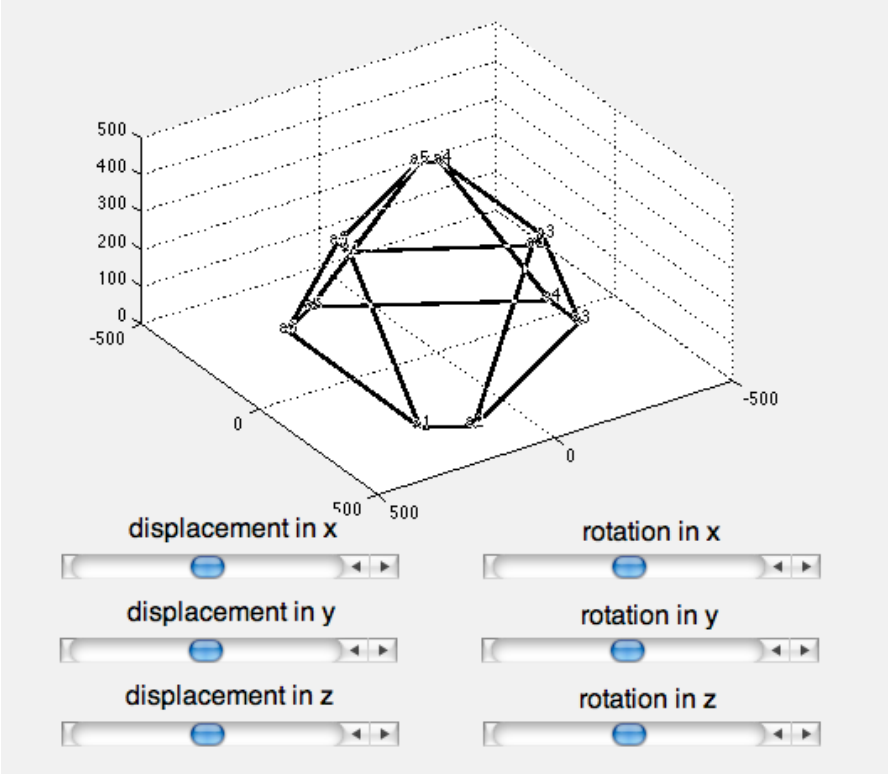

Figure 4.4: The MASS Graphical User Interface (GUI)

MATLAB is a programming environment for algorithm development, data analysis, visualization, and numerical computation [24]. It is used in a wide range of applications, including signal and image processing, communications, test and measurement, financial modeling and analysis and in this case, for control design, including the inverse kinematics solver for the MASS and the user interface. MATLAB allows connection with other devices in different standards, including serial communication, which makes the date transferring for the MASS, a matter of just configuration of some parameters such as the Baud Rate. These parameters are described more in detail below.

After receiving the six parameters of the reference position by the user, there is the "inverse kinematics solver", as mention before, programmed in MATLAB. The function of the "inverse kinematics solver" is to obtain, given the six reference position parameters, the linear position of each actuator, in order to reach the reference position of the MASS. As described in Chapter 2 Section 4, the inverse kinematics function is to obtain the position of the angles of the links necessary to place the end-effector in a certain position in space. Once the general architecture and hardware are explained, the inverse kinematics for the MASS is analyzed, in Chapter 5.

When the reference MASS position is translated to the single actuator linear positions, the linear actuator positions are sent to the Arduino microcontroller, through a serial communication interface that was configured in both the MATLAB program and the Arduino microcontroller and which details are found in Chapter 5, related to software. In this phase the six actuators positions are converted to six PWM signals using the Arduino libraries. The PWM frequency and other parameters are adjusted to have a proper quality in the Linear Actuator Control boards' performance.

Just before the actuators receive the control signals; these signals go through the Linear Actuator Control boards, configured to perform the closed-loop control of the position of the linear actuator. The LAC's receive the PWM signals from the Arduino microcontroller, i. e., the reference position of each of the six actuators. They also receive the real position of the actuators through the feedback provided by

the 10 KΩ potentiometers, that function as encoders and whose resistances are proportional to the actual position in a range from 0% to 100%. When having the reference position and the actual position, there is an error defined as:

$$error = position\ reference - actual\ position$$

This error is also called the delta position, Δp. The delta position is the signal sent to the controller, in this case the LAC, as can be seen on the closed-loop control diagram in Figure 2.9. Based on this delta position, the controller computes the control action that will be performed, with the purpose of reach the reference or desired position. The controller takes this action in several steps. This is a process of various iterations, as it was explained in Chapter 2, until the reference position is reached. During the travel to reach the reference position, the actuator behaves as a first order system, which also was explained in Chapter 2. The experimental identification of the actuator dynamic system was realized in this research work, to know the constant time of the system. The experimental identification was compared with the analytical identification, from which very good results were obtained. These results are shown in Chapter 6, in conjunction with the simulation results of the "inverse kinematics solver", in different cases.

Now, the used hardware is presented.

4.1 Hardware

The hardware elements for the construction of the MASS are described below. For more specifications, see the Appendix A.

4.1.1 Linear actuators

The linear actuators are the most important elements in the MASS prototype, since they make possible the mobility, with 6 degrees of freedom, to the upper platform. The system is integrated by 6

linear actuators. As mentioned in Chapter 3, each actuator provides one degree of freedom. Each actuator gives motion in a straight line, which differentiate it from circular motion of a conventional electric motor. Linear actuators are used in machine tools and industrial machinery and in many other places where linear motion is required. They are powered by different sources such as hydraulic or pneumatic or electromechanical. For safety reasons, for the MASS, we use the electromechanical. In the case of an outage, the system does not fall; it stays rigid in its current position. In the future, when the MASS is realized and a patient stands on the top, this characteristic will prevent the patient to fall down from the system.

For the prototype we use a Linear Actuator form Servocity, Figure 4.5. Its main specifications are summarized in Table 4.1.

Figure 4.5: Linear Actuator form Servocity.

Table 4.1: Servocity Linear Actuator specifications

Variable	Value
Control System	Pulse Width Modulation
Operating Voltage	6.0-12 Volts DC
Operating Speed (12V)	2.20” second
Dynamic Thrust (12V)	25 lbs. Thrust
Potentiometer	I0K

As shown in the table 4.1, this linear actuator has a 10K Ohms potentiometer (variable resistor), shown in Figure 4.6. This variable resistor does the function of an encoder, which is used to measure the angular position. The angular position, through the internal gears of the base of the actuator, is translated to the current linear position. In other words, the potentiometer gives us the linear position of the actuator. This is possible, since when the position of the actuator changes, the gears turn the potentiometer, changing its resistance proportionally to the linear change in the actuator. A voltage is applied to the potentiometer; with the intention of obtain an analog signal to know the position electronically. This process is done by the Linear Actuator Control Board that is explained in the next section.

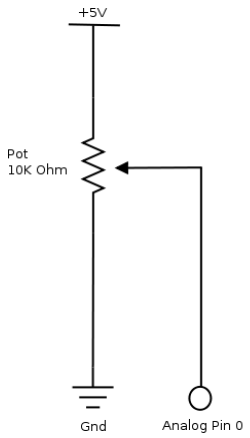

Figure 4.6: 10 K Ohms potentiometer used as encoder.

4.1.2 Linear Actuator Control Board

The Linear Actuator Control Board (LAC), in Figure 4.7, is control board specifically designed for linear actuators. The output of the LAC goes to the linear actuators; the input comes from the Arduino microcontroller. The LAC simplifies considerably the design, saving developing time in the processing associated with direct current motor control (motors used by the Servocity linear actuators).

The LAC allows digital and analog inputs and can be operated as both an interface board, or as a stand-alone controller with the addition of an external potentiometer and power supply.

In this prototype, the Pulse Width Modulation (PWM) input mode is used. This mode allows control of the actuator through a digital output from an external Arduino micro controller. The reference position of the actuator is obtained by means of the duty cycle of a 1kHz square wave of 3 Volt, where the percent duty cycle represents the actuator position to the same percent of full stroke six inches extension. The LAC uses an external power source, since the actuators are powered from the LAC and they require a high voltage and current to function.

Table 4.2 summarizes its principal characteristics.

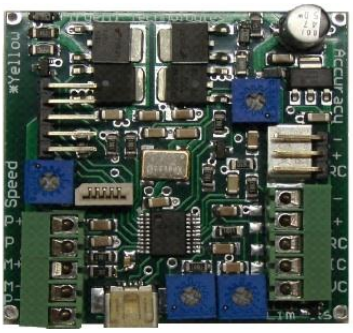

Figure 4.7: Linear Actuator Controller Board from Firgelli

Table 4.2: Firgelli Linear Actuator Control Board specifications

Variable	Value
Control input modes	Digital: USB, RC Servo, 1 kHz PWM Analog: 0–3.3 V, 4–20 mA
Compatible actuators	Larger Actuators with position feedback, 12 volts, 24 volts
Power	5–24 VDC, 4 Amps peak current

4.1.3 Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It can handle fifty-four digital inputs and output pins. The Arduino Mega can use fourteen pins as Pulse Width Modulation (PMW) outputs, from which six are required to send the control signals to the six LAC's. Then the output of the Arduino goes to the six linear actuator control boards and the input comes from a personal computer, by serial communication, through a program made in MATLAB. The physical connection between the microcontroller and the computer is made by an USB/Serial port, from which it also is powered. Arduino also has several very useful libraries, as the Servo.h that was used in the main program of the Arduino. In the next sections, the programming of the microcontroller is explained. Figure 4.8 shows the Arduinio Mega 2560 and Table 4.3, its main specifications.

Figure 4.8: Arduino Mega 2560

Table 4.3: Arduino Mega 2560 specifications

Variable	Value
Microcontroller	ATmega2560
Operating Voltage	5V
Digital I/O Pins	54 (14 PWM output)
Input Voltage (recommended)	7-12V

4.1.4 Power Supply

The power source utilized is the Hewlet-Packard model 6023A, 20 V, 30 (Figure 4.9). It is an auto ranging power supply that provides maximum power over an extensive range of voltage and current ratio without the necessity of selecting the proper output ratio manually. The power source provides the six linear actuators and the six linear actuators control boards. All the connections were made to rear-panel screw-on terminals. As a safety feature, since this is a prototype development and the actuators require very high current, this power source model has a thermostat that shuts down the supply if high temperature condition occurs and resets automatically. In table 4.4, the power source specifications are shown.

Figure 4.9: Hewlet-Packard 6023A Power Supply

Table 4.4: Hewlet-Packard 6023A Power Supply specifications

Variable	Value
DC volts output	0 to 20 V
DC amps output	0 to 30 A
Maximum power	200 to 242 W

The hardware was described. Now, in next chapter, the software design is introduced.

Chapter 5: Software Design

5.1 Inverse Kinematics Solver

The inverse kinematics solver was programed in MATLAB, a programming environment for algorithm development, as mention before. However, with the intention of make programmable the solver, it was done following some methods that allow use vectors, which simplify the computation. This characteristic also allows that this program is transferable and can be utilized in the MASS, not only in this prototype. Some fragments of the code are shown in the document, to have a better explanation of the used variables.

Firstly, with the purpose of make programmable the position of the joints, where the actuators are fixed, the technique proposed by Rene Graf in [25] was followed. Since the Stewart platform has symmetric geometry with two triangles and respectively two legs fixed at each angle. The legs are variable in length and are the linear actuators. The joints, that connect the linear actuators to the triangles, are completely free.

To compute the kinematic of a Stewart platform, first, the position of the joints has to be determined. Figure 5.1 shows the position of the joint of the base platform ($^{A}J_1$, $^{A}J_2$, $^{A}J_3$, $^{A}J_4$, $^{A}J_5$ and $^{A}J_6$) and the moving platform ($^{B}J_1$, $^{B}J_2$, $^{B}J_3$, $^{B}J_4$, $^{B}J_5$ and $^{B}J_6$) triangle in the corresponding coordinate system. The joints of platforms, base and moving are marked on orange in the figure. The vectors of the joints of the base platform are defined as follows [25]:

$$\begin{bmatrix} {}^{A}J_1 \\ {}^{A}J_2 \\ {}^{A}J_3 \\ {}^{A}J_4 \\ {}^{A}J_5 \\ {}^{A}J_6 \end{bmatrix} = R_A \begin{bmatrix} \cos(\delta_A) & \sin(\delta_A) & 0 \\ \cos(120-\delta_A) & \sin(120-\delta_A) & 0 \\ \cos(120+\delta_A) & \sin(120+\delta_A) & 0 \\ \cos(240-\delta_A) & \sin(240-\delta_A) & 0 \\ \cos(240+\delta_A) & \sin(240+\delta_A) & 0 \\ \cos(-\delta_A) & \sin(-\delta_A) & 0 \end{bmatrix}$$

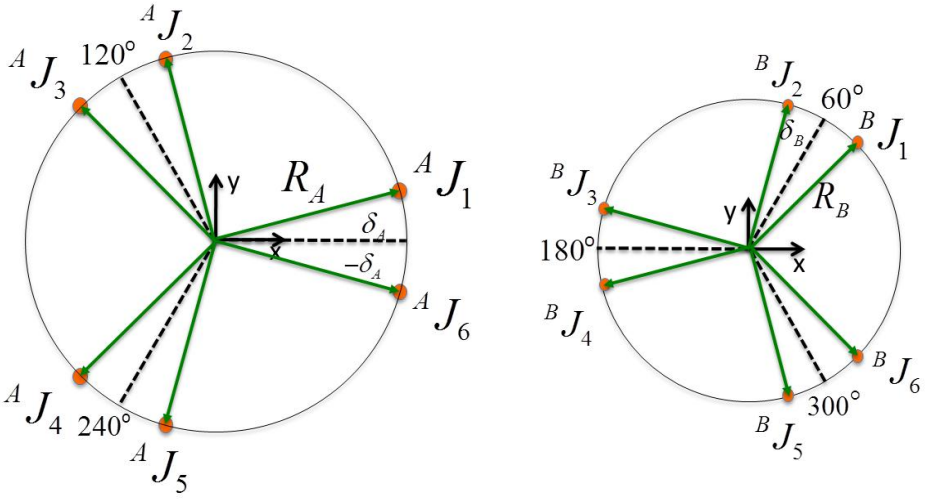

Figure 5.1: Position of the joints in the base platform and the moving platform.

In the same way, the vectors of the joints in the moving platform are defined as:

$$\begin{bmatrix} {}^{B}J_1 \\ {}^{B}J_2 \\ {}^{B}J_3 \\ {}^{B}J_4 \\ {}^{B}J_5 \\ {}^{B}J_6 \end{bmatrix} = R_B \begin{bmatrix} \cos(60-\delta_B) & \sin(60-\delta_B) & 0 \\ \cos(60+\delta_B) & \sin(60+\delta_B) & 0 \\ \cos(180-\delta_B) & \sin(180-\delta_B) & 0 \\ \cos(180+\delta_B) & \sin(180+\delta_B) & 0 \\ \cos(300-\delta_B) & \sin(300-\delta_B) & 0 \\ \cos(300+\delta_B) & \sin(300+\delta_B) & 0 \end{bmatrix}$$

The vector method utilized allows making the algorithm completely programmable and transferable with the definition of only four parameters:

1) The radius of the base platform R_A,
2) The angle of the joints of the base platform δ_A,
3) The radius of the moving platform R_B and
4) The angle of the joints of the moving platform δ_B.

The prototype MASS dimensions were measured. The radius and the delta's of the prototype are the shown in the next MATLAB code. The base platform is defined as BotBase (bottom base) and the moving platform as TopBase (top base). The dimensions are in millimeters.

```
radBotBase=365;       %// bottom base radius
deltaBotBase=60-10;   %// delta of bottom base actuators
radTopBase=265;       %// top base radius
deltaTopBase=60-5.75;%// delta of top base actuators
home=[0,0,365]';      %// home position
```

It is noteworthy that there is another parameters, besides the four mentioned, that also has to be predefined in the code. This is the home position that is the height where the moving platform is positioned when it is in the zero position, in reference to the zero of the base platform. The home is a default position in the z-axis. For the case of the prototype MASS, the home is located at 365 millimeters form the universal zero at (x = 0, y = 0 and z = 0) and depends on the geometry and the nature of the mechanism.

After understanding how the joint are defined. The next step is the solution of the inverse kinematics in a vector approach [26]. Consider the Figure 5.2. We have the two platforms, base and moving. We also have two coordinated systems, one for the base platform, which, in this case, is the reference coordinated system or the universal coordinated system and is defined with x_A, y_A and z_A. The other is the moving coordinated system. The position of this coordinate system in relation to the base coordinated system is the actual position of the moving platform. In other words, the position of the moving coordinated system gives us the position of the center of the moving platform that is where its position is specified from. This coordinated system is defined with x_B, y_B and z_B.

To simplify a little bit the analysis, only one joint is shown in the figure. The vector of the joint in the base platform is defined as ${}^{A}J$ and the vector joint in the moving platform, as ${}^{B}J$. Again, the orange

circles are the joints. The vector ${}^{A}B_{ORG}$, in red, is the translation that exists from the base platform to the moving platform. This translation is compound of two terms:

$$AB_{ORG} = home\ position + displacement\ in\ x, y\ and\ z$$

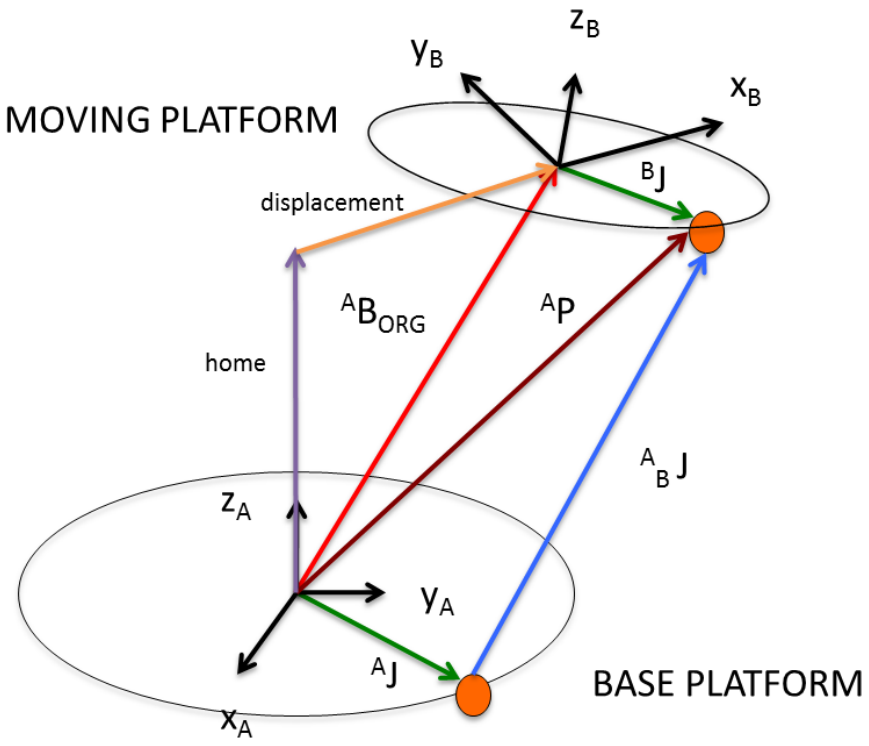

Figure 5.2: Vectors definition in the inverse kinematic analysis of the MASS

The position of the moving platform does not only depends on its displacement in x, y and z from the home position. There are six parameters related with its position. The other three parameters are the rotations around the x, y, and z axis. The rotations give us the orientation of the moving platform coordinated system in relation to the universal coordinated system. With the aim of find this orientation is necessary to perform the spatial transformation [26]. Figure 5.3 shows the translation and rotation of coordinated system.

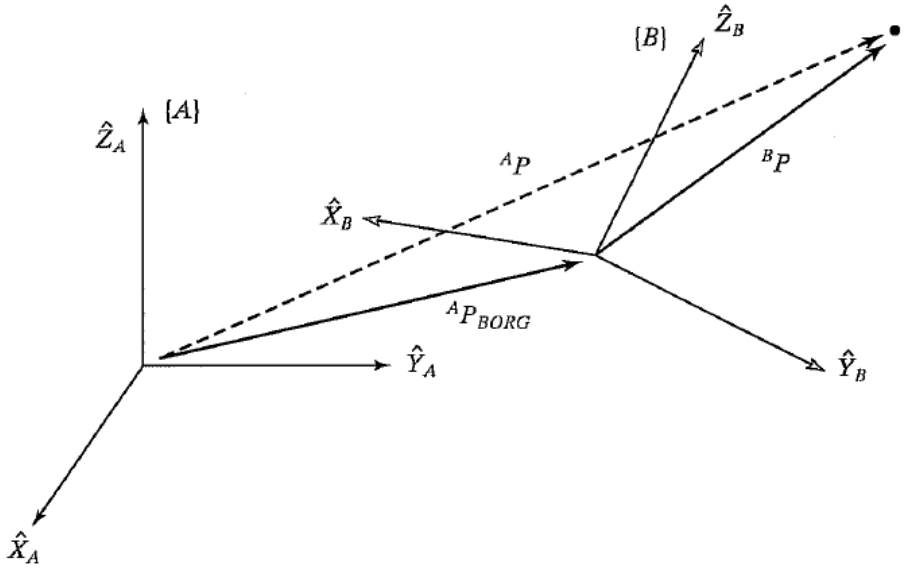

Figure 5.3: Translation and rotation of the B coordinated system. John Craig.

When we look for place the moving platform in a specific position, through the vector ${}^AB_{ORG}$ and its rotation, what we are really interested in is the linear position that we need in the actuators, so we can obtain the desired position. If we define the actuators vector as ${}^A_BJ$, in blue in Figure 5.2. The procedure to get their magnitude is the following.

1) Obtain the vectors of the joints moving platform from the base coordinated system frame ${}^AP$, which is given by:

$$ {}^AP = {}^A_BR\,{}^BJ + {}^AB_{ORG} $$

Where ${}^A_BR\,{}^BJ$ are the vectors of the joints of the moving platform in the moving coordinated system multiplied by the rotation matrix ${}^A_BR$. The corresponding fragment of the MATLAB code is written to explain this point:

```
% DO NOT CHANGE THIS:
alpha=[deltaBotBase 120-deltaBotBase 120+deltaBotBase 240-deltaBotBase 240+deltaBotBase -deltaBotBase]*pi/180;
beta=[60-deltaTopBase 60+deltaTopBase 180-deltaTopBase 180+deltaTopBase 300-deltaTopBase 300+deltaTopBase]*pi/180;
```

```
    aPborg=[x,y,z]' + home;                                                    %/
    gamma=gamma*pi/180; theta=theta*pi/180; phi=phi*pi/180;                    %/
    abRotMatrix = [cos(phi)*cos(theta), cos(phi)*sin(theta)*sin(gamma) - sin(phi)*cos(gamma)
    %abRotMatrix=[1,0,0;0,1,0;0,0,1];                                          %/
for i=1:6
    topBaseAct(:,i)=abRotMatrix*(radTopBase*[cos(beta(i)) sin(beta(i)) 0]')+aPborg;   %#
    text(topBaseAct(1,i)+50,topBaseAct(2,i)+50,topBaseAct(3,i)+50,strcat('a',num2str(i)));%/
end
```

In this code, in line 6 we define the vector of the joints of the moving platform in the base coordinated system as topBaseAct, using the rotation matrix and multiplying by the vectors of the joints of the moving platform defined in the moving coordinated system and previously defined as:

$$\begin{bmatrix} {}^{B}J_1 \\ {}^{B}J_2 \\ {}^{B}J_3 \\ {}^{B}J_4 \\ {}^{B}J_5 \\ {}^{B}J_6 \end{bmatrix} = R_B \begin{bmatrix} \cos(60-\delta_B) & \sin(60-\delta_B) & 0 \\ \cos(60+\delta_B) & \sin(60+\delta_B) & 0 \\ \cos(180-\delta_B) & \sin(180-\delta_B) & 0 \\ \cos(180+\delta_B) & \sin(180+\delta_B) & 0 \\ \cos(300-\delta_B) & \sin(300-\delta_B) & 0 \\ \cos(300+\delta_B) & \sin(300+\delta_B) & 0 \end{bmatrix}$$

2) Obtain the actuators vectors ${}^{A}_{B}J$ performing the operation:

$${}^{A}_{B}J = {}^{A}P - {}^{A}J$$

In the code, we find this part here, in line three:

```
% ACTUATORS ------------------------------------------------------------------

for i=1:6
    line('color',color,'LineStyle','-','erasemode','xor','LineWidth',1.5, 'xdata
    act(:,i)=topBaseAct(:,i)-botBaseAct(:,i);                                  %
    lengthAct(i) = sqrt(act(1,i)*act(1,i)+act(2,i)*act(2,i)+act(3,i)*act(3,i));%
end
```

Remember that topBaseAct are the vectors of the joints of the moving platform, seen from the base coordinated system (the universal coordinated system), i. e., ${}^{A}P$, and the vectors of the joints of the base platform, seen from the base coordinated system, are ${}^{A}J$. The subtraction of these two is the vector of the actuator.

3) Once we have the vectors of the actuators, the linear positions of them are going to be given by their scalar magnitudes after making a linear conversion, to adjust to the specific dimension of the used actuator. The MATLAB code to obtain the magnitude of the vectors is:

```
lengthAct(i) = sqrt(act(1,i)*act(1,i)+act(2,i)*act(2,i)+act(3,i)*act(3,i));
```

4) Finally, an adjustment to the scale is made. To get the exact stroke, we subtract 370 millimeters that is the longitude of the actuator when it is fully retracted, as in Figure 5.4. The factor 100/150 is added, because the stroke of the actuator is 150 millimeters but the controller receives this signal in a range from 0% to 100%. This is the code:

```
out=(lengthAct'-370).*(100/150);
```

The MATLAB file is called invKinePlat.m. The inputs are the six parameters that define the moving platform position and the output are six percentages from 0 to 100, of the necessary stroke of each actuator to obtain the reference position, as explained in Figure 5.5.

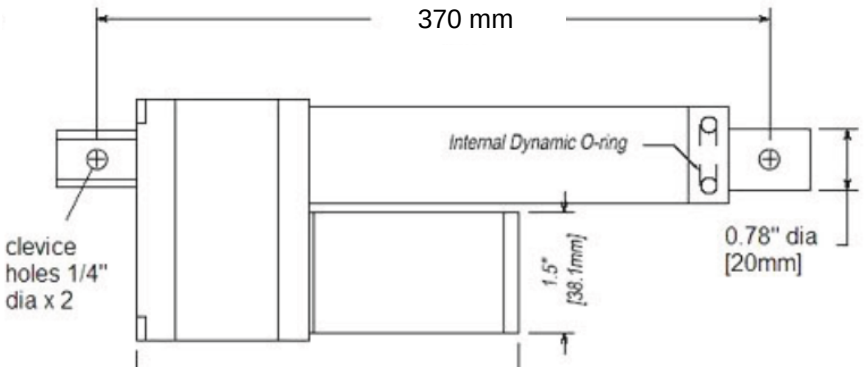

Figure 5.4: Dimensions of the Servocity linear actuator.

The way of use this code is as follows:

In the MATLAB command window call the function like this: invKinePlat(x, y, z, θ, φ, ψ), where x is the desired linear displacement along x-axis; y, the linear displacement along y-axis; z, the linear displacement along z-axis; θ, the rotation around x-axis; φ, the rotation around y-axis and ψ, the rotation around z-axis.

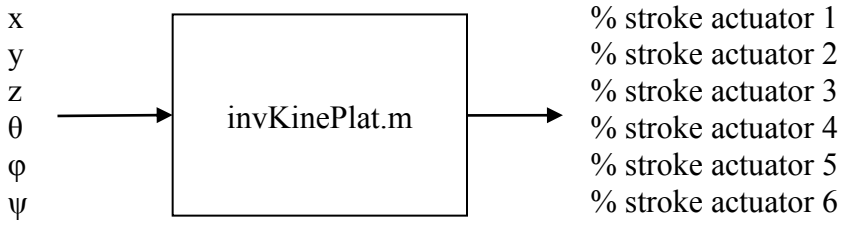

Figure 5.5: Inputs and outputs of the invKinePlat program

When we have the percentage of stroke for each actuator, the commands to send the instructions to the Arduino microcontroller are (write these commands on the MATLAB command window):

```
s = serial('SerialPortName', 'TimeOut', 1, 'BaudRate', 9600);
fopen(s);
fwrite([% act 1, % act 2, % act 3, % act 4, % act 5, % act 6]);
```

The parameters in red are the ones that changes. The serial port name must consider with the selected port in the Arduino software. The baud rate was selected 9600 as default, but it may be changed, just be sure that have the same baud rate in both the MATLAB command and in the Arduino program.

5.2 Serial Communication

In the microcontroller code, the library Servo.h was use. That is the reason why the actuators appear as servo objects. As in the case of the MATLAB code, some fragments of the Arduino code are shown to explain its functioning.

First part:

```
#include <Servo.h>

Servo act1, act2, act3, act4, act5, act6 ;  // create servo objects to control the linear actuators

int a[6];                                  // Vector with the current positions of actuators
int incomingBuffer[6];                       // Vector that will keep the new positions

// Method that fill the buffer with a 6 size vector with the new actuator positions from MATLAB
void fillBuffer() {
  int inbyte;
  int bufPosition=0;
  while(Serial.available()>0) {
    // Read only one character per call
    inbyte = Serial.read();
    incomingBuffer[bufPosition]=(int)inbyte;
    bufPosition++;

    if(bufPosition-1==6) {
      Serial.println("ERROR Command Overflow");
      bufPosition=0;
    }
  }
}
```

In the first part, the actuator objects are created in conjunction with the variable that store the actuator strokes and the buffer that receives these strokes from the MATLAB code. After this, the method to fill the mentioned buffer, to receive six numbers from 0 to 100, is programmed.

Second part:

```
// Setup method
void setup()
{
  Serial.begin(9600);  //Serial starts at 9600 bps
  act1.attach(6);  // attaches the servo on pin 9 to the servo object
  act2.attach(7);
  act3.attach(8);
  act4.attach(9);
  act5.attach(10);
  act6.attach(11);
}

void loop()
{
  // When there is a new position
  if(Serial.available()>0)
  {   fillBuffer();
      Serial.write('OK');
  }

  act1.write(map(incomingBuffer[0], 375, 525, 48, 127));  // sets the servo position
  act2.write(map(incomingBuffer[1], 375, 525, 48, 127));
  act3.write(map(incomingBuffer[2], 375, 525, 48, 127));
  act4.write(map(incomingBuffer[3], 375, 525, 48, 127));
  act5.write(map(incomingBuffer[4], 375, 525, 48, 127));
  act6.write(map(incomingBuffer[5], 375, 525, 48, 127));
  delay(15);  // waits for the servo to get there
}
```

In the second part the actuator objects are attached to a PWM output pin. The baud rate is set at 9600, but it may be changed, just keeping the concordance with the same baud rate in MATLAB.

The main loop of the code (void loop) is waiting until there is a new value (a change) for the actuator strokes. When a change is detected, the buffer starts to fill and the values are sent to the actuators through the instruction act.write. The map function converts a value from 0 to 100, to a value from 48 to 127, which is the range where the Servo.h library works.

Chapter 6: Simulation analysis

6.1 Actuator mathematical model

With the purpose of identify the actuator system experimentally; a data set was collected, in real time. The response to the step is presented in Figure 6.1.

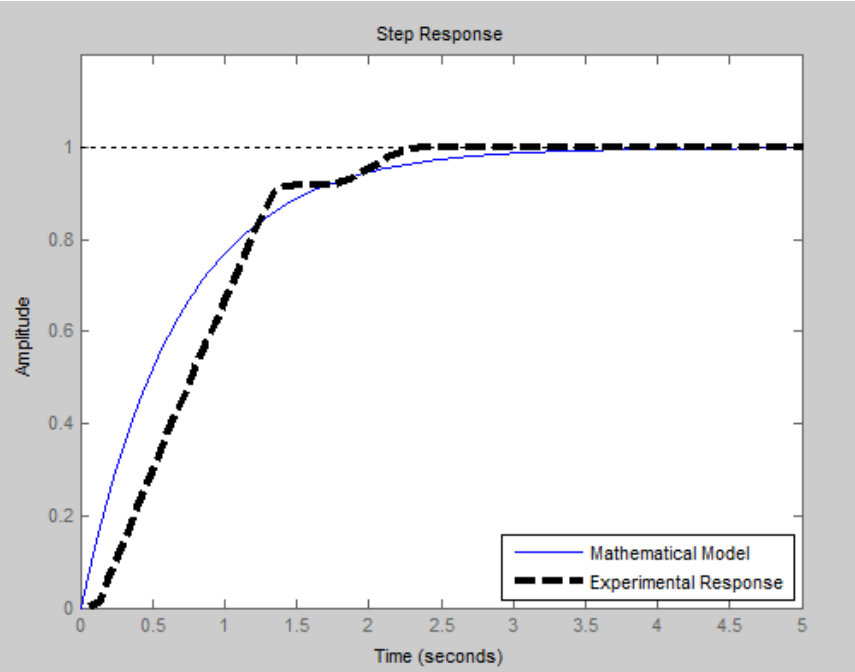

Figure 6.1: Step response of the actuator.

The mathematical model was obtained from the datasheet specifications. The time constant was calculated from the time that the actuator lasts to reach its final value, i. e., the settling time of the actuator:

$$\tau = 0.68\ s$$

Then the differential equation of the system is [14]:

$$\tau \frac{dy}{dt} + y(t) = x(t)$$

And in the Laplace domain:

$$\tau s Y(s) + Y(s) = X(s)$$

With the intention of obtain the analytical model of the system; we proceed to get the transfer function of the system, the relation of the output over the input in the Laplace domain, under initial conditions equal to zero.

$$Y(s)(\tau s + 1) = X(s)$$

From where:

$$\frac{Y(s)}{X(s)} = \frac{1}{(\tau s + 1)} = \frac{1/\tau}{(s + 1/\tau)}$$

Substituting the value of the time constant obtained analytically we get:

$$\frac{Y(s)}{X(s)} = \frac{1/0.68}{(s + 1/0.68)}$$

This is the system represented, in blue, in figure 6.1. The discontinuous black line represents the experimental values obtained during the test. Several tries were done in the experiment, all the collected data can be found in the Appendix B.

Analyzing the two curves, we can see that the experimental behavior of the actuator over the time is as the behavior of a first order system.

There are some differences: at the beginning, we can see that the actuator has a death time (it takes more time to react and start moving). This is due to the high current needed to overcome the static friction and the inertia of the electromechanical actuator. When the current at this point was measured, we can detect the high peak consumed by the motor.

The second difference is in the trajectory of the actuator to its final value. The error in the analytical value and the real value is duo to the friction in the gears inside the actuators. We found the actuators have difference and their speeds caused by friction; nevertheless, this difference is not a greater to be considered in the model.

At the steady state, we found that the actuator does not have steady state error.

6.2 Inverse Kinematics

In the simulation of the "inverse kinematics solver", also several tests were done. Here some cases are presented:

1) First case: x=0, y=0, z=0, θ=0, φ=0, ψ=0

   This is to show the MASS system in home position. Three views are presented in Figures 6.2, 6.3 and 6.4.

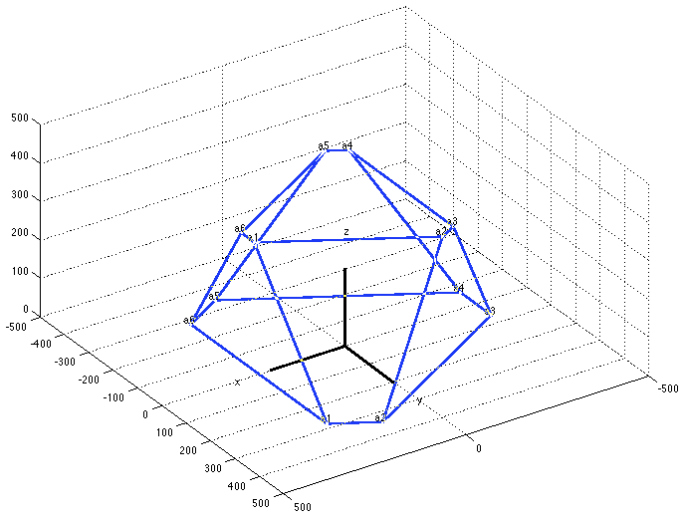

Figure 6.2: Isometric view of the MASS in home position.

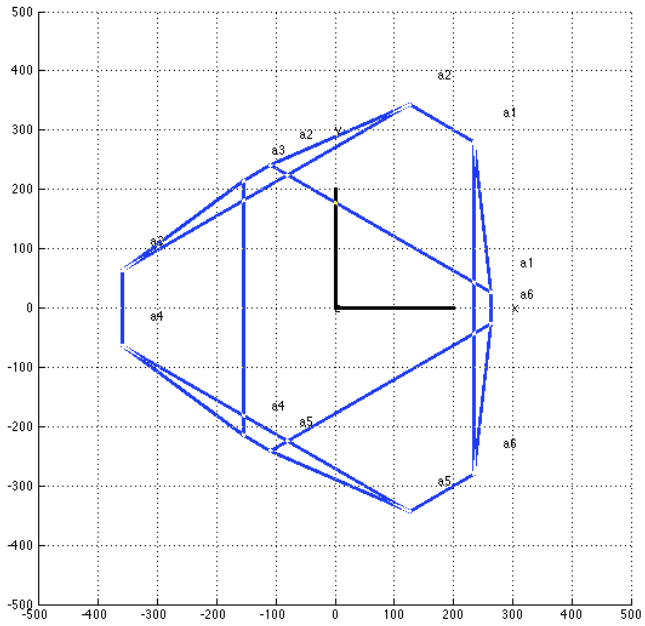

Figure 6.3: Top view of the MASS in home position.

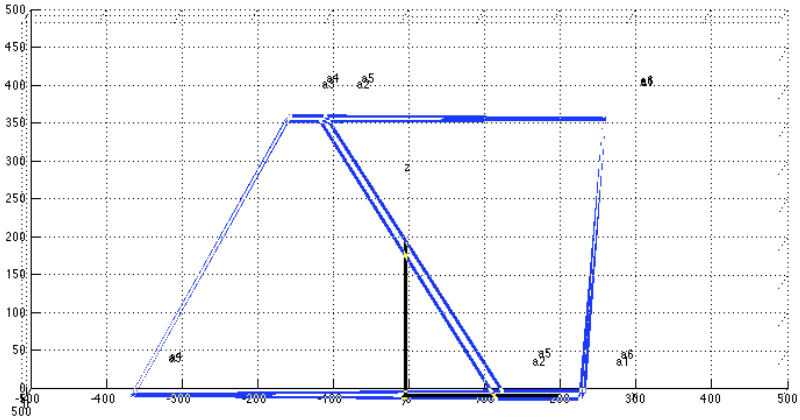

Figure 6.4: Lateral view of the MASS in home position.

2) Second case: x=50, y=-30, z=20, θ=0, φ=0, ψ=0

This is to show the MASS system with linear displacement, but no rotation. Three views are presented in Figures 6.5, 6.6 and 6.7

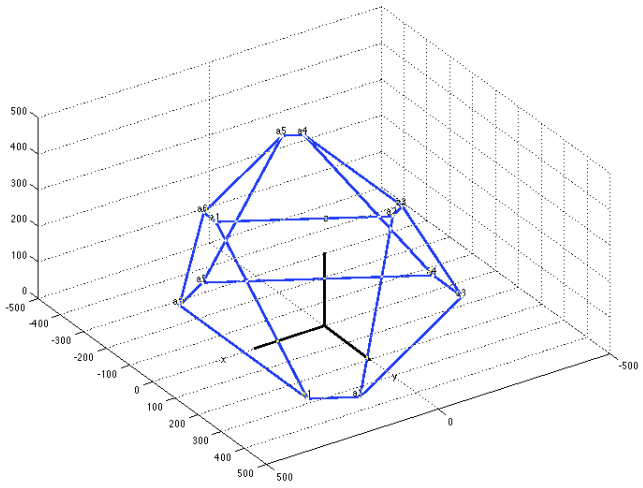

Figure 6.5: Isometric view of the MASS displaced.

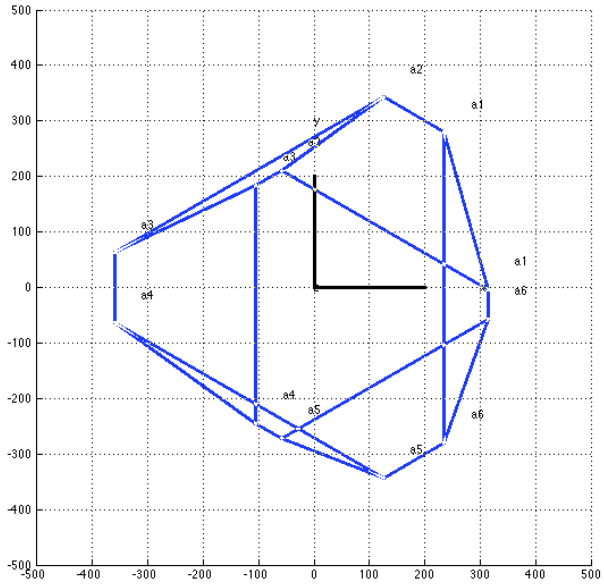

Figure 6.6: Top view of of the MASS displaced.

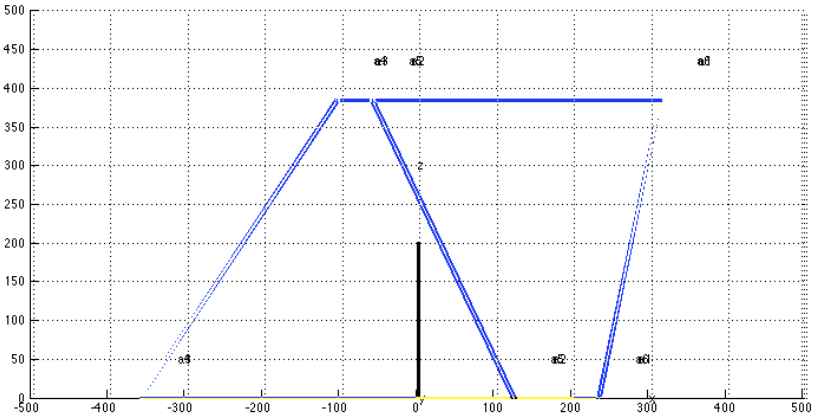

Figure 6.7: Lateral view of of the MASS displaced.

3) Third case: x=0, y=0, z=0, θ=30, φ=-12, ψ=15

This is to show the MASS system rotated, but not linear displacement. Three views are presented in Figures 6.8, 6.9 and 6.10

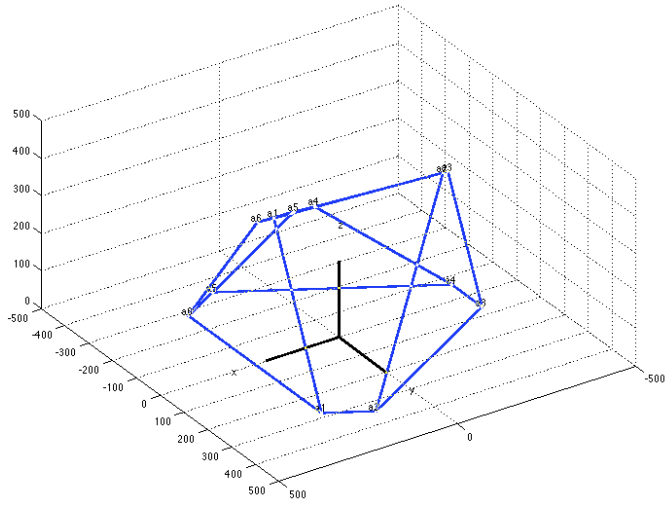

Figure 6.8: Isometric view of the MASS rotated.

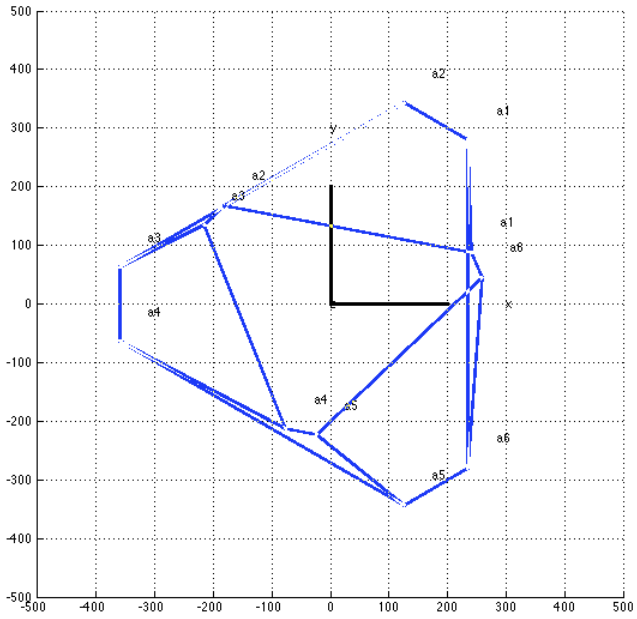

Figure 6.9: Top view of the MASS rotated.

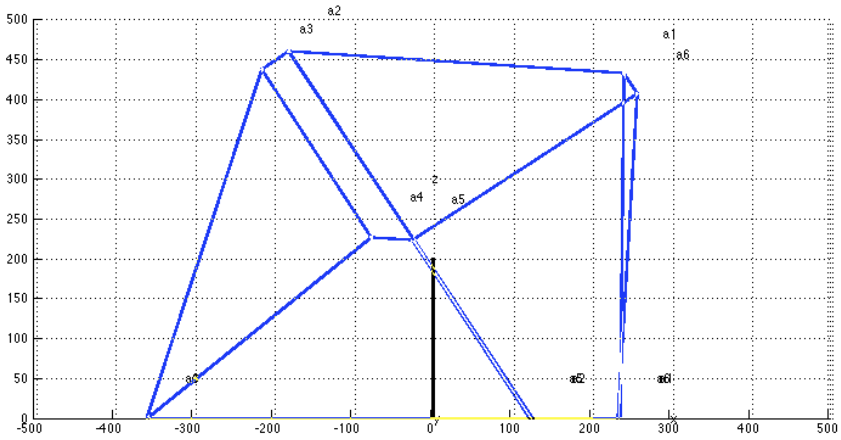

Figure 6.10: Lateral view of the MASS rotated.

4) Fourth case: x=10, y=-10, z=10, θ=10, φ=-12, ψ=-15

This is to show the MASS system both displaced and rotated. Three views are presented in Figures 6.11, 6.12 and 6.13

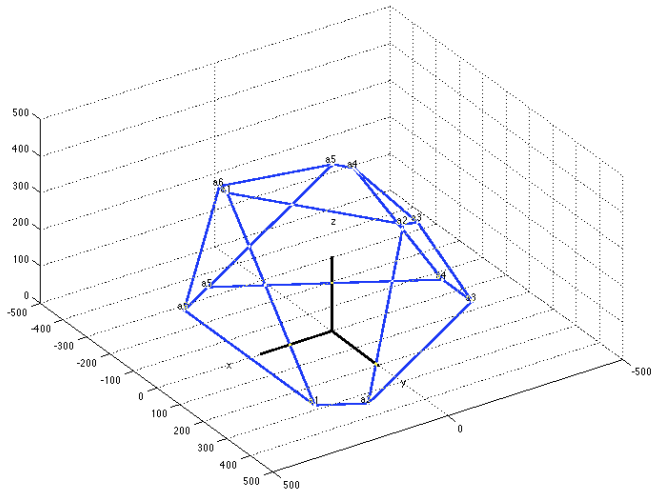

Figure 6.11: Isometric view of the MASS discplaced and rotated.

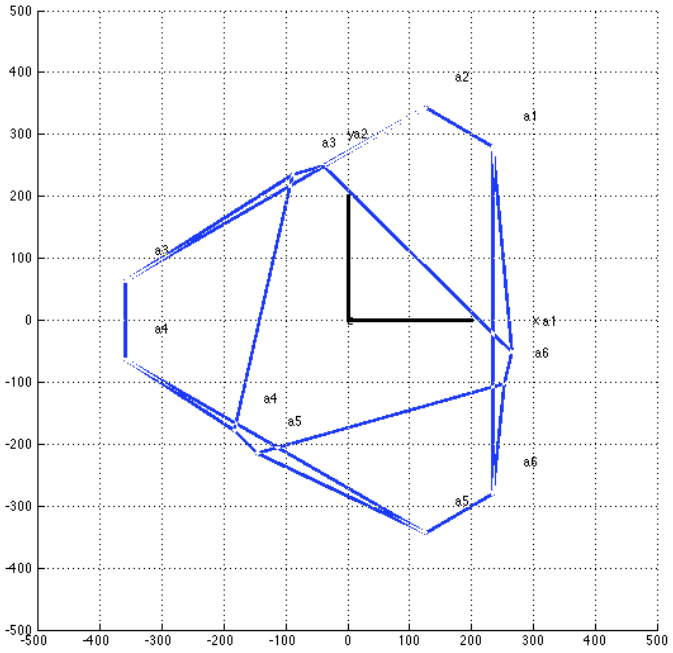

Figure 6.12: Top view of the MASS discplaced and rotated.

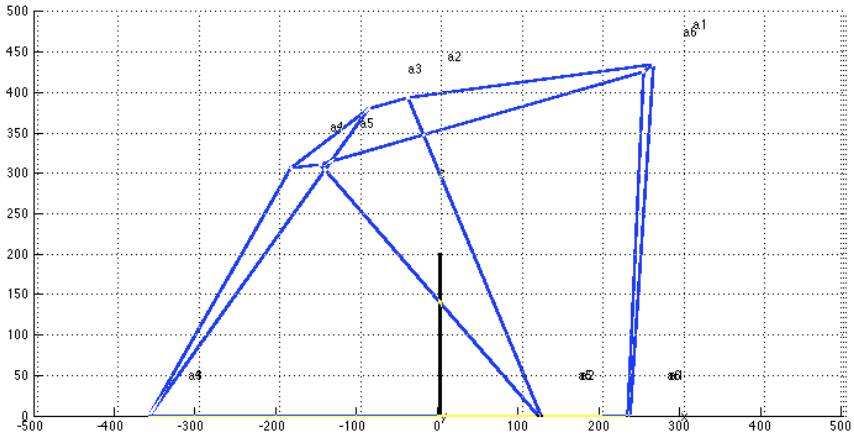

Figure 6.13: Lateral view of the MASS discplaced and rotated.

In general, as seen in the last pictures, the “inverse kinematics solver” results are the proper. For the four presented cases, the computation of the stroke percentage is correct and for the prototype MASS dimensions works in a fine way.

Chapter 7: Conclusions and contributions

First, from the state-of-the-art research we can emphasize the necessity of developing a system that allows six degrees of freedom, which allows the same ranges of motion that the human posture and balance require. The thesis provides a system with a closed-loop control, such that the machine has an intelligent control and the patient's performance over the system is the one who commands the machine. The prototype MASS is the first phase in this project.

The control algorithm of the prototype MASS meets the necessary requirements for being transferable to the MASS system. In general, the objectives proposed in this work were accomplished. The scaled prototype was constructed in different phases: mechanical design, mechanical manufacturing, mechanical assembly, electronic design, electronic assembly, inverse kinematics analysis, mechanism simulation and control algorithm design. The closed-loop control was utilized in the control algorithm, where the actuator system was identified both mathematically and experimentally. The method used in the inverse kinematics allows having a program that is transferable to the MASS.

Some observations and comments during the realization of this project are: 1) At the first stages, when programming the serial communication between MATLAB and the LAC's, before using the Arduino controller, we tried to make the communication through LABVIEW, because the LAC's have their own software to make that communication. However, when connecting each LAC via USB to the computer, every time the USB addresses changed, so have a control of the ports was not easy. In fact, the addresses change randomly every time the computer starts. We found that not always the software included in a product is the best. Finally, we find a pretty fine control and communication in the Arduino microcontroller. 2) When identifying the actuator system, analyzing the two curves the analytical model and the experimental behavior, we can see that the experimental behavior of the actuator over the time is as very close and can be approximated by a first order system as explained in [14]. The linear actuator is

a first order response. Time constant = 0.68. There is a transient state error in the analytical value and the real value produced by the friction in the gears inside the actuators, since friction causes that the systems does not have a complete linear behavior. We also found that the actuator has a death time, due to the high current needed to overcome the static friction and the inertia of the electromechanical actuator. When the current at this point was measured, we can detect the high peak consumed by the motor. It was also found that the actuators have difference and their speeds caused by friction; nevertheless, this difference is not a greater to be considered in the model, so it is valid.

3) In the inverse kinematics analysis, in general, as seen in simulation pictures, the "inverse kinematics solver" results are the proper. For the four presented cases, the computation of the stroke percentage is correct and for the prototype MASS dimensions works in a fine way and the MATLAB code to solve the inverse kinematics of the MASS can be used to solve any platform, by setting five parameters. The prototype MASS was used to test the positioning of the platform. The result of the tests was satisfactory. The serial communication, the GUI interface, and all the hardware is working properly.

Finally, the contributions of this thesis are:

1. A position solution for the MASS, based on inverse kinematics.
2. A transferable control algorithm for the MASS.
3. A first order system experimental identification, i. e. system model.
4. A graphical user interface.

Chapter 8: Future Work

The work that follows this research is vast. They are classified in short-term and long-term. 1) A workspace analysis is required. During the realization of the inverse kinematics analysis, we found that there are some positions that the prototype MASS cannot reach, because of its dimensions limitations. This limitation depends on the mechanism geometry, which in turn depends on the model or type of mechanism utilized. 2) Speed and acceleration. In this research the control algorithm for the position was developed. The next step is to make a kinetic analysis, involving forces, in order to develop an algorithm for controlling the acceleration and speed in the prototype MASS. 3) Ball on platform experiment. The intention to include an external feedback is another task to consider. There is a classic experiment called “ball and beam”, where the position of the ball on the beam is controlled. This experiment can be extrapolated to the platform, in 3D. For long-term there are four tasks, which are aimed to the realization of the MASS, however the prototype MASS is the bridge to its development. 1) Wearable sensors for real-time acquisition. 2) Intelligent control system (MIQ), algorithm for compensation of the adaptive locomotion and postural control. 3) Sensor fusion and sensor validation scheme.

References

[1] Jacobson, G. P. Handbook of balance function testing. 1st Edition. Cengage Learning. 1997.

[2] Cook, S. D. Handbook of multiple sclerosis. 4th Edition. Informa Healthcare. 2006.

[3] Winter, D. A. Human balance and posture control during standing and walking. Gait & Posture. Volume 3, Issue 4, December 1995, Pages 193-214.

[4] Haven, L. The human balance system – A complex coordination of central and peripheral systems. The Vestibular Disorders Association. 2008

[5] http://www.cerebralpalsysource.com/Types_of_CP/ataxic_cp/index.html

[6] http://balanceandmobility.com/patient_info/testing.aspx

[7] Whitney, S. L. A comparison of accelerometry and center of pressure measures during computerized dynamic posturography: A measure of balance. Gait & Balance. Volume 33, Issue 4, April 2011, Pages 594-599.

[8] Merlet, J.-P. Parallel Robots. 2nd Edition. Springer. 2006.

[9] Chebbi, A.-H. Kinestostatic and singularity analyses of the 3 UPU translational parallel robot. Computational Kinematics. 2009, Pages 61-68.

[10] Gogu, Gregori.Structural synthesis of parallel robots: Methodology. 1st Edition. Springer. 2008.

[11] Li, Yangmin. Design and development of a medical parallel robot for cardiopulmonary resuscitation. Transactions on Mechatronics. Volume 12, Issue 3, June 2007, Pages 265-273.

[12] Stewart, D. A platform with six degrees of freedom. Proceedings of the Institution of Mechanical Engineers. Volume 180, Issue 1, June 1965, Pages 371-386.

[13] Wapler, M. A Stewart platform for precision surgery. Transactions of the Institute of Measurement and Control. Volume 25, Issue 4, 2003. Pages 329-334.

[14] Burns, R. S. Advanced Control Engineering. 1st Edition. Butterworth-Heinemann. 2001.

[15] Kuo, B. C. Automatic control systems, Volume 1. 8th Edition. Wiley , John & Sons, Incorporated. 2002.

[16] Ogata, K. Modern control engineering. 5th Edition. Prentice Hall. 2009.

[17] Beggs, J. S. Kinematics. 1st Edition. Taylor & Francis, Inc. 1983.

[18] Vinogradov, O. G. Fundamentals of Kinematics and Dynamics of Machines and Mechanisms, Volume 1

[19] Ceccarelli, M. Fundamentals of mechanics of robotic manipulation. 1st Edition. Kluwer Academic Publishers. 2004.

[20] Apparatus for the diagnosis and rehabilitation of balance disorders. US Patent: 6,063,046

[21] Adjustable Instability Apparatus for exercising, balancing, recreation and physical rehabilitation activities. US Patent Application: 2004/0023766 A1

[22] Balance Training Apparatus. US Patent Application: 2007/0207900 A1

[23] System and method of balance training. US Patent Application: 2010/007508 A1

[24] http://www.mathworks.com/products/matlab/

[25] Graf, R. A flexible controller for a Stewart platform. Second International Conference on Knowledge-Based Intelligent Electronic Systems, 21-23 April 1998, Adelaide, Australia. Editors, L.C. Jain and R.K.Jain

[26] Craig, J. J. Introduction to robotics - Mechanics and control. 3rd Edition. Prentice Hall. 2005

Appendix

Appendix A. Hardware

Servocity Linear Actuator

Variable	Value
Control System	+Pulse Width Control 1500usec Neutral
Required Pulse	3-5 Volt Peak to Peak Square Wave
Operating Voltage	6.0-12 Volts DC
Operating Temperature Range	-20 to +60 Degree C
Operating Speed (12V)	2.20” second at No load
Operating Speed (12V)	1.37” second at Max load
Dynamic Thrust (12V)	25 lbs. Thrust
Static Load	500 lbs.
Fully Retracted	2000usec Pulse Width
Half Extended	1500usec Pulse Width
Fully Extended	1000usec Pulse Width
Current Drain (12V)	11.9mah Idle
Current Drain (12V)	800mah operating No load
Current Drain (12V)	3200mah operating Max load
Dead band Width	8usec
Motor Type	3 Pole Ferrite
Potentiometer	I0K
Gear Type	4 Metal Gears
Connector Wire Length	24"

Recommended Fuse	5amp Inline Blade Fuse
IP Grade	2IP 63-total dust protection, water resistant
Duty Cycle	25% (25% on, 75% off)
Load force	224.8 lbs
Static load capacity	517 pounds
Housing Material	Zinc Alloy

Firgelli Linear Actuator Control Board

Variable	Value
Control input modes	Digital: USB, RC Servo, 1 kHz PWM Analog: 0–3.3 V, 4–20 mA
Controller	10-bit Dual Sample Rate Quasi PD
Compatible actuators	PQ12 Actuators with position feedback, 6 or 12 volts L12–P Actuators with position feedback, 6 or 12 volts L16-P Actuators with position feedback, 6 or 12 volts Larger Actuators with position feedback, 12 volts, 24 volts
Dimensions	50 mm x 50 mm (excluding battery

	holder)
Power	5–24 VDC, 4 Amps peak current at 10% duty cycle
Operating environment	–10 to +70°C at 10–80% relative humidity

Arduino Mega 2560 specifications

Variable	Value
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Hewlet-Packard 6023A Power Supply specifications

Variable	Value
DC volts output	0 to 20 V
DC amps output	0 to 30 A
Maximum power	200 to 242 W
Load regulation	Voltage: 0.01% + 2 mV Current: 0.01% + 9 mA
Line regulation	Voltage: 0.01% + 1mV Current: 0.01% + 6 mA
Ripple and noise	Voltage: 3 mVrms/30 mVp-p Current: 30 mArms
Load effect transient recovery	Time (10%/50%): 1 ms/2 ms Level (10%/50%): 50 mV/150 mV
Front panel voltmeter range	20 V, 200 V
Front panel ammeter range	200 A
Display OVP	200 V
Maximum AC input current (at 120 Vac)	6.5 A
DC floating voltage	±240 Vdc
Typical efficiency	80% on maximum output boundary

Appendix B. Actuators experimental identification data

Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
24.6334	1.85728	79.3744	19.6481	79.4721
24.6334	1.85728	79.3744	19.7458	79.4721
24.6334	1.85728	79.3744	19.7458	79.4721
24.6334	1.85728	79.3744	19.7458	79.4721
24.6334	1.85728	79.3744	19.7458	79.4721
24.6334	1.85728	79.3744	19.7458	79.4721
24.6334	1.85728	79.3744	19.7458	79.4721
24.7312	1.85728	79.3744	19.7458	79.4721
24.7312	2.24829	79.3744	19.6481	79.4721
24.6334	4.69208	79.3744	19.7458	79.4721
24.6334	7.52688	79.3744	19.7458	79.4721
24.6334	10.2639	79.3744	19.6481	79.4721
24.6334	13.1965	79.3744	19.6481	79.4721
24.6334	16.0313	79.3744	19.6481	79.4721
24.6334	18.8661	79.3744	19.6481	79.4721
24.7312	21.7009	79.3744	19.6481	78.1036
25.4154	24.5357	79.3744	19.7458	75.8553
28.2502	27.3705	79.3744	19.6481	73.7048
31.085	30.2053	79.3744	19.7458	71.3587
33.9198	33.0401	79.3744	19.6481	69.1105
36.7546	35.8749	79.3744	19.7458	66.8622
39.5894	38.7097	79.3744	19.6481	64.5161
42.4242	41.5445	79.3744	19.6481	62.1701
45.3568	44.477	79.3744	20.0391	59.9218
48.1916	47.3118	79.3744	21.5054	57.5758
50.7331	49.9511	79.3744	23.3627	55.3275
53.4702	52.7859	79.3744	25.0244	52.8837
56.1095	55.6207	79.3744	26.6862	50.5376
58.7488	58.4555	79.3744	28.2502	48.2893
61.3881	61.2903	79.3744	29.8143	45.9433
64.0274	64.1251	79.3744	31.2805	43.695
66.7644	66.9599	79.3744	32.8446	41.2512
69.4037	69.7947	79.3744	34.4086	39.0029
72.043	72.5318	79.3744	35.8749	36.6569
74.0958	75.2688	79.3744	37.4389	34.3109
74.8778	78.1036	79.3744	38.9052	31.9648
74.9756	80.9384	79.3744	40.3715	29.521
74.9756	83.7732	79.3744	41.9355	27.2727
74.9756	86.5103	79.3744	43.4995	25.3177
74.9756	88.0743	79.3744	44.9658	23.9492

75.0733	88.7586	79.3744	46.5298	23.0694
75.3666	88.7586	79.3744	47.9961	22.5806
75.7576	88.7586	79.3744	49.4624	22.3851
76.4418	88.7586	78.4946	50.8309	22.3851
77.1261	88.7586	76.9306	52.2972	22.3851
77.7126	88.7586	75.1711	53.7634	22.0919
78.3969	88.7586	73.5093	55.2297	21.6031
78.9834	88.7586	71.652	56.5982	21.0166
79.3744	88.7586	69.8925	57.9668	20.3324
79.4721	88.7586	68.1329	59.433	19.9413
79.4721	88.7586	66.2757	60.7038	19.8436
79.4721	88.7586	64.5161	62.0723	19.8436
79.4721	88.7586	62.6588	63.4409	19.8436
79.4721	88.7586	60.8016	64.8094	19.8436
79.4721	88.7586	58.9443	66.1779	19.8436
79.4721	88.7586	57.087	67.5464	19.8436
79.4721	88.7586	55.2297	68.915	19.8436
79.4721	88.7586	53.3724	70.2835	19.8436
79.4721	88.7586	51.4174	71.652	19.8436
79.4721	88.7586	49.6579	72.825	19.8436
79.4721	88.7586	47.8006	73.1183	19.8436
79.4721	88.7586	45.8456	73.1183	19.8436
79.4721	88.7586	43.9883	73.1183	19.8436
79.4721	88.7586	42.0332	73.1183	19.8436
79.4721	88.7586	40.176	73.3138	19.8436
79.4721	88.7586	38.2209	73.607	19.8436
79.4721	88.7586	36.3636	74.0958	19.8436
79.4721	88.7586	34.4086	74.6823	19.8436
79.4721	88.7586	32.4536	75.0733	19.8436
79.4721	88.7586	30.5963	75.5621	19.8436
79.4721	88.7586	28.739	76.0508	19.8436
79.4721	88.7586	26.8817	76.5396	19.8436
79.4721	88.7586	25.4154	77.0283	19.8436
79.4721	88.7586	24.6334	77.4194	19.8436
79.4721	88.7586	24.0469	77.8104	19.8436
79.4721	88.7586	23.8514	78.2014	19.8436
79.4721	88.7586	23.8514	78.4946	19.8436
79.4721	88.7586	23.8514	78.8856	19.8436
79.4721	88.7586	23.7537	79.1789	19.8436
79.4721	88.7586	23.3627	79.4721	19.8436
79.4721	88.7586	22.6784	79.4721	19.8436
79.4721	88.7586	21.8964	79.4721	19.8436
79.4721	88.7586	21.1144	79.4721	19.8436
79.4721	88.7586	20.3324	79.4721	19.8436

79.4721	88.7586	19.9413	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436
79.4721	88.7586	19.6481	79.4721	19.8436
79.4721	88.7586	19.7458	79.4721	19.8436

Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Trial 11
19.8436	79.6676	20.2346	20.3324	20.2346	75.0733
19.8436	79.6676	20.2346	20.3324	22.1896	75.8553
19.8436	79.6676	20.2346	20.3324	24.8289	75.9531
19.8436	79.6676	20.2346	20.3324	26.393	75.9531
19.8436	79.6676	20.2346	20.3324	27.2727	75.9531
19.8436	79.6676	20.2346	20.3324	27.4682	76.2463
19.8436	79.6676	20.2346	22.1896	27.566	76.6373
19.8436	79.6676	20.2346	25.0244	27.7615	77.1261
19.8436	79.6676	20.2346	27.7615	28.0547	77.5171
19.8436	79.6676	20.2346	30.5963	28.6413	78.1036
19.8436	79.6676	20.2346	33.3333	29.4233	78.4946
19.8436	79.6676	20.2346	36.1681	30.0098	78.8856
19.8436	79.6676	20.2346	39.0029	30.0098	79.2766
19.8436	79.6676	20.2346	41.74	30.0098	79.4721
19.8436	78.9834	20.2346	44.5748	30.0098	79.4721
19.8436	76.2463	20.2346	47.3118	30.0098	79.4721
19.8436	73.4115	20.2346	49.9511	30.0098	79.4721
19.8436	70.479	20.2346	52.6882	30.0098	79.4721
19.8436	67.6442	20.2346	55.4252	30.0098	79.4721
19.8436	64.7116	20.2346	58.1623	30.0098	79.4721
19.8436	61.7791	20.2346	60.8993	30.0098	79.4721
20.3324	58.9443	20.2346	63.5386	30.0098	79.4721
22.5806	56.0117	20.2346	66.2757	30.0098	79.4721
24.9267	52.9814	20.2346	69.0127	30.0098	79.4721
27.3705	50.1466	20.2346	71.7498	30.0098	79.4721

29.6188	47.3118	20.2346	73.998	30.0098	79.4721
31.6716	44.3793	20.2346	75.1711	30.0098	79.4721
33.8221	41.4467	20.2346	75.4643	30.0098	79.4721
35.9726	38.5142	20.2346	75.4643	30.0098	79.4721
38.0254	35.5816	20.2346	75.4643	30.0098	79.4721
40.0782	32.6491	20.2346	75.5621	30.0098	79.4721
42.131	29.7165	20.2346	75.5621	30.0098	79.4721
44.1838	26.784	20.2346	75.8553	30.0098	79.4721
46.3343	24.4379	22.0919	76.3441	30.0098	79.4721
48.2893	22.6784	24.9267	77.1261	30.0098	79.4721
50.2444	21.5054	27.6637	77.8104	30.0098	79.4721
52.2972	20.8211	30.4008	78.4946	30.0098	79.4721
54.2522	20.5279	33.1378	79.2766	30.0098	79.4721
56.305	20.4301	35.9726	79.6676	30.0098	79.4721
58.26	20.4301	38.7097	79.6676	32.0626	77.6149
60.2151	20.4301	41.4467	79.6676	34.6041	74.7801
62.0723	20.2346	44.2815	79.6676	36.1681	71.9453
64.1251	20.2346	47.0186	79.6676	36.9501	69.0127
66.0802	20.2346	49.7556	79.6676	37.1457	66.1779
68.0352	20.2346	52.3949	79.6676	37.2434	63.9296
69.9902	20.2346	55.132	79.6676	37.3412	62.2678
71.9453	20.2346	57.869	79.6676	37.7322	61.2903
73.5093	20.2346	60.6061	79.6676	38.3187	60.7038
73.998	20.2346	63.3431	79.6676	39.1007	60.4106
73.998	20.2346	65.9824	79.6676	39.6872	60.2151
73.998	20.2346	68.7195	79.6676	39.8827	60.2151
73.998	20.2346	71.3587	79.6676	40.176	60.1173
74.1935	20.2346	73.9003	79.6676	40.176	60.1173
74.6823	20.2346	75.0733	79.6676	40.176	60.1173
75.1711	20.2346	75.4643	79.6676	40.176	60.1173
75.8553	20.2346	75.3666	79.6676	40.176	60.1173
76.6373	20.2346	75.3666	79.6676	40.176	60.1173
77.3216	20.2346	75.4643	79.6676	40.176	60.1173
78.0059	20.2346	75.5621	79.6676	40.176	60.1173
78.6901	20.2346	75.8553	79.6676	40.176	60.1173
79.2766	20.2346	76.4418	79.6676	40.176	60.1173
79.6676	20.2346	77.1261	79.6676	40.176	60.1173
79.6676	20.2346	77.8104	79.6676	42.131	60.1173
79.6676	20.2346	78.4946	79.6676	45.0635	60.1173
79.6676	20.2346	79.1789	79.6676	47.9961	60.1173
79.6676	20.2346	79.6676	79.6676	50.7331	60.1173
79.6676	20.2346	79.6676	79.6676	53.5679	60.1173
79.6676	20.2346	79.6676	79.6676	56.5005	60.1173
79.6676	20.2346	79.6676	79.6676	59.2375	60.1173

79.6676	20.2346	79.6676	79.6676	62.0723	60.1173
79.6676	20.2346	79.6676	79.6676	65.0049	60.1173
79.6676	20.2346	79.6676	79.6676	67.7419	60.1173
79.6676	20.2346	79.6676	79.6676	70.5767	60.1173
79.6676	20.2346	79.6676	79.6676	73.4115	60.1173
79.6676	20.2346	79.6676	79.6676	75.0733	60.1173
79.6676	20.2346	79.6676	79.6676	75.9531	60.1173
79.6676	20.2346	79.6676	79.6676	76.0508	60.1173
79.6676	20.2346	79.6676	79.6676	76.0508	60.1173
79.6676	20.2346	79.6676	79.6676	76.0508	60.1173
79.6676	20.2346	79.6676	79.6676	76.2463	60.1173
79.6676	20.2346	79.6676	79.6676	76.6373	60.1173
79.6676	20.2346	79.6676	79.6676	77.1261	60.1173
79.6676	20.2346	79.6676	79.6676	77.6149	60.1173
79.6676	20.2346	79.6676	79.6676	78.1036	60.1173
79.6676	20.2346	79.6676	79.6676	78.4946	60.1173
79.6676	20.2346	79.6676	79.6676	78.8856	60.1173
79.6676	20.2346	79.6676	79.6676	79.3744	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173
79.6676	20.2346	79.6676	79.6676	79.4721	60.1173

Vita

Julio Adrian Torres Frayre was born on April 18th, 1985 in Juan Aldama, Zacatecas, Mexico. He finished his bachelor's degree in Mechatronics Engineering and a specialization in Industrial Automation on December 2008, from the Instituto Tecnologico y de Estudios Superiores de Monterrey, in Monterrey, Mexico. He entered to the Mechanical Engineering Department at The University of Texas at El Paso in August 2010, to pursue his Master's Degree, under the supervision of Dr. Thompson Sarkodie-Gyan, from the Electrical and Computer Engineering department.

He worked in the Laboratory for Industrial Metrology and Automation, where he has fulfilled the position of Research Assistant. Also, he obtained a position of Teaching Assistant in the Electrical and Computer Engineering.

His areas of specialization are: Control Engineering, Linear Systems Analysis, Digital Control, Industrial Logic Control, Fuzzy Logic, Identification and Modern Control, Instrumentation for Measuring and Control, Industrial Networks, Manufacturing Systems Automation, Manufacturing Technologies, Parallel Mechanisms.

This thesis was typed by Julio Adrián Torres Frayre