Polytechnic of Turin

Master's Degree in Mechatronic Engineering

# Docking for surface elements: design and simulation

**Supervisor**
Prof. Marcello Chiaberge

**Co-Supervisor**
Engr. Andrea Merlo

**Candidate**
Bongiovanni Lorenzo s288756

Academic year 2021–2022

## Abstract

This thesis aims at defining the most suitable docking solution for surface elements applications.The paper will go through the state of the art of the already existing technologies for soft  hard docking solutions between surface elements (i.e. manned rovers and surface habitats). These technologies are fundamental for safe exploration of Moon  Mars surface, when manned missions and outposts will be deployed. The focus of this thesis is the Moon, due to the already established Artemis missions.

Previous work had found which would have been the best mechanical structure to provide the soft docking, what has been established is the Stewart platform which will be later developed. The technical specifications of actuators, motors and sensors (which will not be deeply analyzed here) were examined and a suitable control architecture has been selected. In the end the final solution consists of a central control system, where an Arduino MEGA board is the master controller and the motors drivers (contained inside the linear actuators) are the slave elements.

This structure is part of the Artemis framework which already exists. This work goes through the electrical and mechanical implementation of the Stewart platform with a final Matlab ( Simulink ) simulation to prove the reliability and feasibility of the model for future physical implementation.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This thesis work starts from the need for an automatic control mechanism for an existing mechanical plant for the docking between a land lunar rover and any sort of entry port during the ARTEMIS missions on the moon, an example could be another rover or a pressurized lunar module for human leaving routine, pressurized laboratories etc . As already stated this electro-mechanical system is part of the ARTEMIS research program done under the collaboration of Thales Alenia Space.

The target of the thesis is to study the components and simulate them for an existing project of a Stewart platform. The main objective of the simulation is to test the inverse cinematic of the linear actuators considering already given both the translation and angles of rotation, they are simulated as already given by a sensor.

The result of the thesis will be subsequently implemented with hardware components and mounted on the Roxy rover, the Alenia Space's rover which is used to test every implementation in a simulated Mars field.

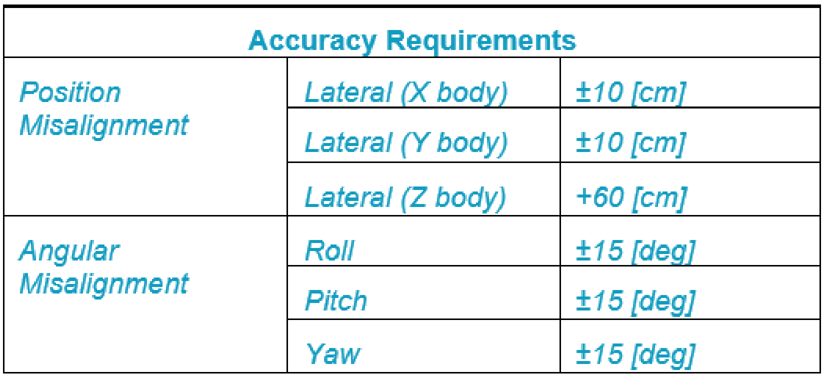
This project, as mentioned, is based on previous research and aims to simulate the system before a future hardware implementation. Looking at the system from a broad perspective, it is composed of a Stewart platform built from 6 linear actuators, each of which mounts a DC motor inside. Each linear actuator is connected to an H-bridge which in turn is connected to a board called "Arduino MEGA".

Here a brief description of the chapters and the structure of the thesis is exposed, without counting the introduction and conclusion, the remaining chapters are 4:

- in chapter 2 (State of the Art) the docking and already existing solutions are studied. While giving the definitions of docking it is also presented the evolution of the solutions adopted by the main space agencies.
- in chapter 3 (Mechanical Part) the Stewart platform, linear actuator and inverse cinematic concepts will be explained.
- in chapter 4 (Electronic part) the concept of DC motors and H-bridge.
- in chapter 5 (Simulation Part) the inverse kinematic will be tested through Matlab and Simulink, just to show the feasibility of that.

### 1.0.1 Complete circuit and brief explanation

A set of requirements on the relative displacements and rotations that the system has to compensate was given: The system has to be able to recover a translational misalignment of 10cm on the X and Y axes and a distance of 60cm on the Z axis. Also, it must be possible to orient the platform with an excursion of +/-15 deg for each coordinate axis.



| Accuracy Requirements | | |
|---|---|---|
| *Position Misalignment* | *Lateral (X body)* | *±10 [cm]* |
| | *Lateral (Y body)* | *±10 [cm]* |
| | *Lateral (Z body)* | *+60 [cm]* |
| *Angular Misalignment* | *Roll* | *±15 [deg]* |
| | *Pitch* | *±15 [deg]* |
| | *Yaw* | *±15 [deg]* |

Table 1.1: Requirements

# Chapter 2

# State of the Art

While until today the vision of docking was adopted only for the approach and docking of two separate free-flying space vehicles, today with the advent of space missions where the intent is to stabilize a small colony on the planet's surface, the docking technology become one of the main protagonists for docking between two non-flying object which stay on the planet surface. The first example can be a rover which has to approach and dock with a pressurized module.

In this particular case, the docking technology will be used exactly for that purpose, among possible rovers and pressurized modules on the Lunar surface, in particular for the ARTEMIS missions.

Here are some definitions of free-flying object docking.

Two main operational technology during missions involving more than one spacecraft are Rendezvous and docking (or berthing) RVD/B:

- RENDEZVOUS: it is a set of orbital manoeuvres between two spacecraft where the objective is to approach at a very close distance (e.g. within visual contact). To achieve the correct position the rendezvous needs a precise match of position vectors and orbital velocities of the two spacecraft.
- DOCKING: it is the act of approaching each other (generally one of the two spacecraft remains steady concerning the orbital velocity) and the other one modifies its velocity to reduce the remaining distance between the two spacecraft.
- BERTHING: It is the act of "putting" the passive spacecraft in mating position through the usage of a robotic arm

Looking at the history of space travel, the first situation in which a rendezvous and docking between two spacecraft took place was on 16 March 1966, when Neil Amstrong and Dave Scott manually performed rendezvous. The first automatic RVD took place on 30 October 1967, thanks to the Soviets. [2].

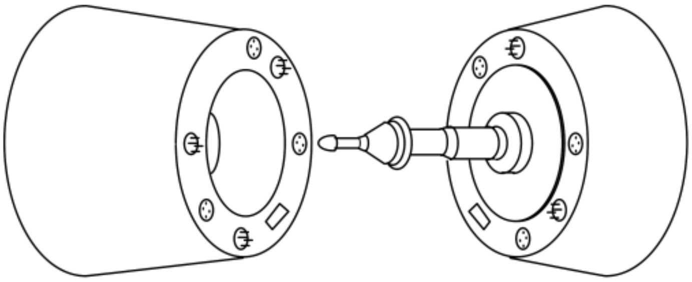


Figure 2.1: Classic Docking System

### 2.0.1 Flexible Electromagnetic Leash Docking system (FELDs)

Flexible Electromagnetic Leash Docking system (FELDs) is a technology demonstrator whose main objective is to test an electromagnetic soft docking system that guarantees a mechanical connection between two spacecraft through the use of a flexible cable.Docking is performed by launching a ferromagnetic probe towards the target, which attracts it with a static magnetic field.Since the connection between the probe and the launching spacecraft is flexible, the system is self-adjusting, with no need for precise positioning and attitude control; this represents a significant advantage over the existing mechanical docking systems, which require a high level of navigation accuracy.

The prototype comprises two main subsystems, the launcher (GUN) and the receiving interface (SEC).The design of the SEC is based on empirical measurements and simulations of the magnetic field, as its effect on the probe is fundamental to the docking.Load cells and dampers are placed on the SEC mountings, both to reduce the effect of the impact and measure the system response [1].

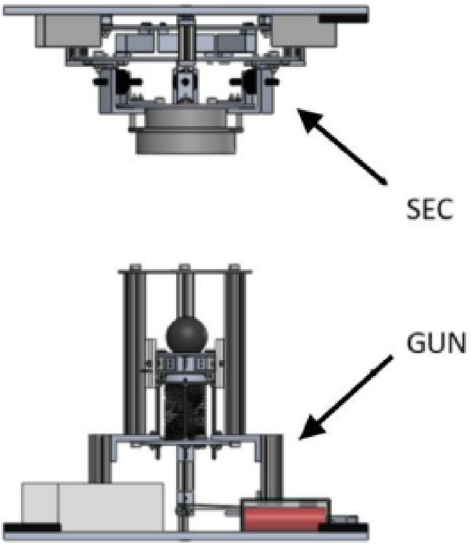


Figure 2.2: FELDs

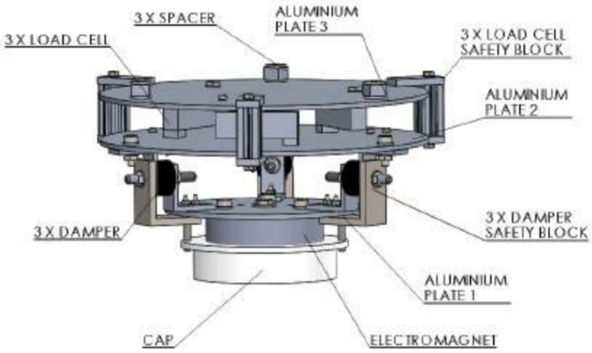


Figure 2.3: SEC

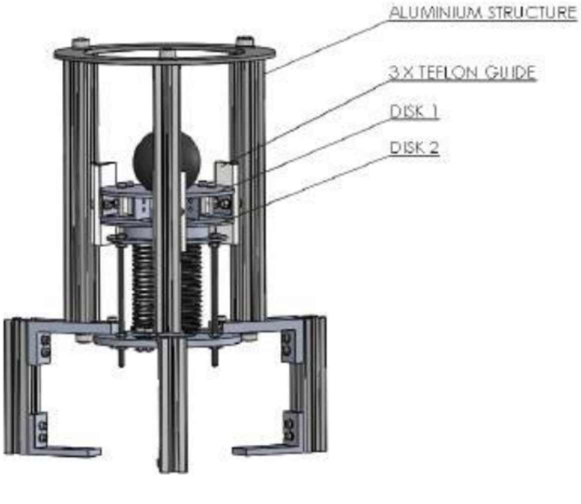


Figure 2.4: GUN

### 2.0.2 Gemini and Soyuz docking systems

The Gemini and Soyuz docking systems were the first two docking systems used in space exploration history. A central architecture was used in both cases. The mechanism in the Gemini mission consisted of a rigid male cone used as a probe and a cup interface used as a drogue, which was linked to the target spacecraft by seven shock absorbers to dampen the relative longitudinal and lateral velocities. For reusability, the shock absorbers were outfitted with an orifice damper and a parallel spring. The probe was used in conjunction with an alignment system that included a v-shaped counterpart as a guide in the female cone.

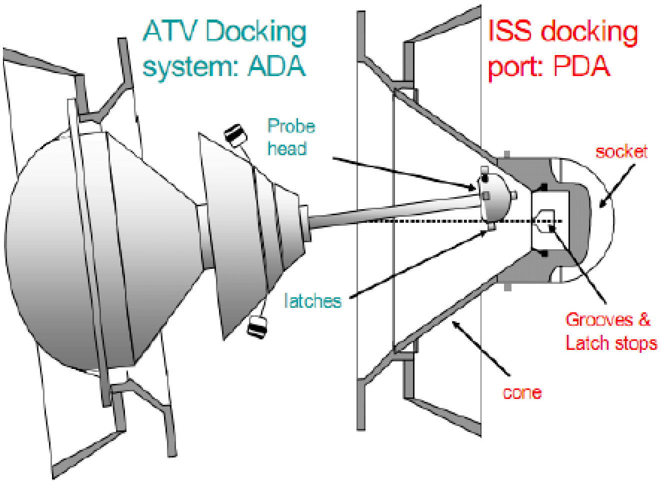


Figure 2.5: SOYUZ mechanism

When this element is retracted, it compresses a coil and a Belleville spring and causes an electromechanical brake to rotate. The capture was accomplished through the use of two latches on the probe head that reached the female socket. The end of the operation was confirmed by a transducer on the probe's head. The probe was retracted using the large ball screw, and the misalignment was corrected by the female guides. This solution was first used in 1967 and is still used today to create a transfer tunnel between the two satellites. The original design was modified by incorporating the probe and drogue mechanisms into the hatches and making the mechanism more compact. [3].

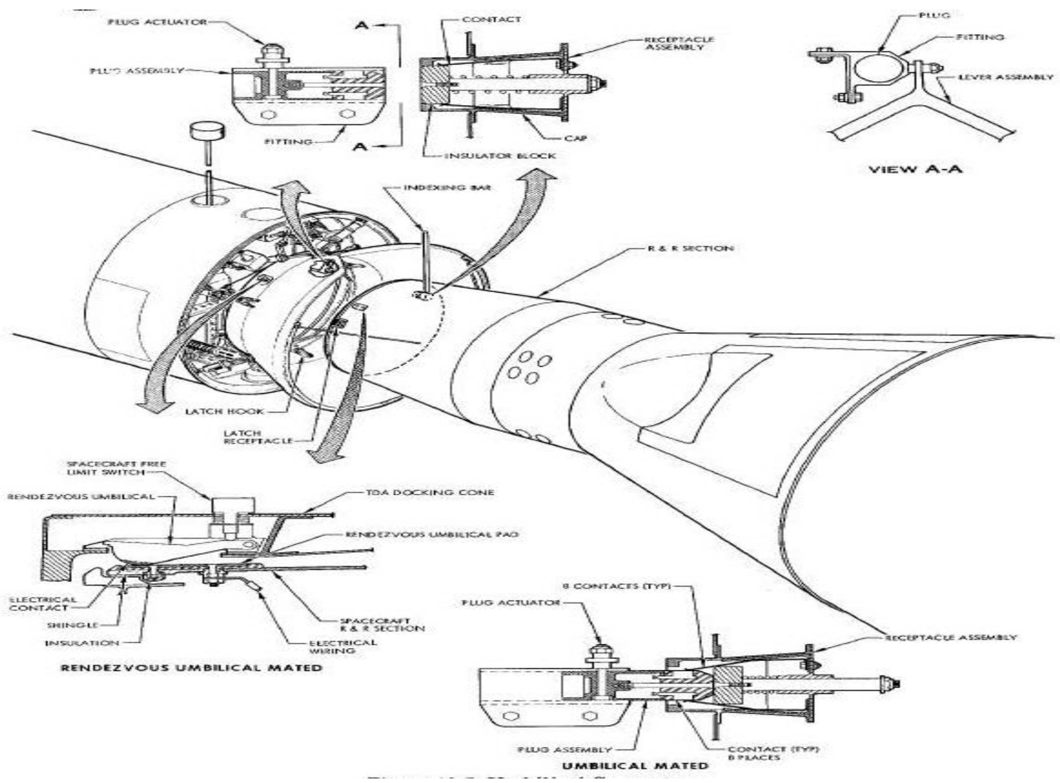


Figure 2.6: GEMINI

### 2.0.3 APAS

The ASTP mechanism inspired this docking mechanism. The first peripheral and androgynous docking mechanism was developed as a result of collaboration between American and Russian space agencies. The fact that it was androgynous implies that both sides are designed to work actively or passively, implying that if one of the two mechanisms fails, the other may be activated, increasing the likelihood of success. The design concept includes a ring with guides and capture latches mounted on movable rods that serve as attenuators and retracting actuators, as well as a docking ring with peripheral mating capture latches and a docking seal." In terms of attenuation technology, the two countries chose different approaches: the Americans used a hydraulic damper, while the Russians used their own electromechanical brake. This technology evolved in 1989 with the APAS-89 (Androgynous Peripheral Attachment System). The main changes were the use of EMB attenuation technology, mechanical latches loaded with springs for soft docking, and the redesign of the guide "petals," which went from an outwards to an inwards configuration. The APAS was initially developed for use on the Buran spaceship. Its subsequent versions were used in a variety of significant missions, including the Shuttle-ISS (APAS-95) and the Chinese one.

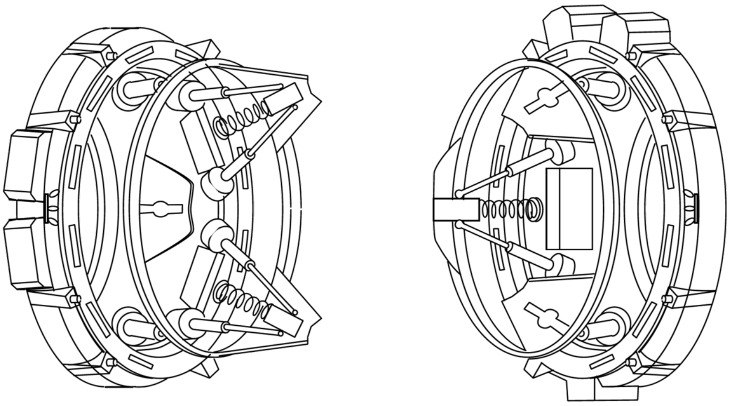


Figure 2.7: APAS System

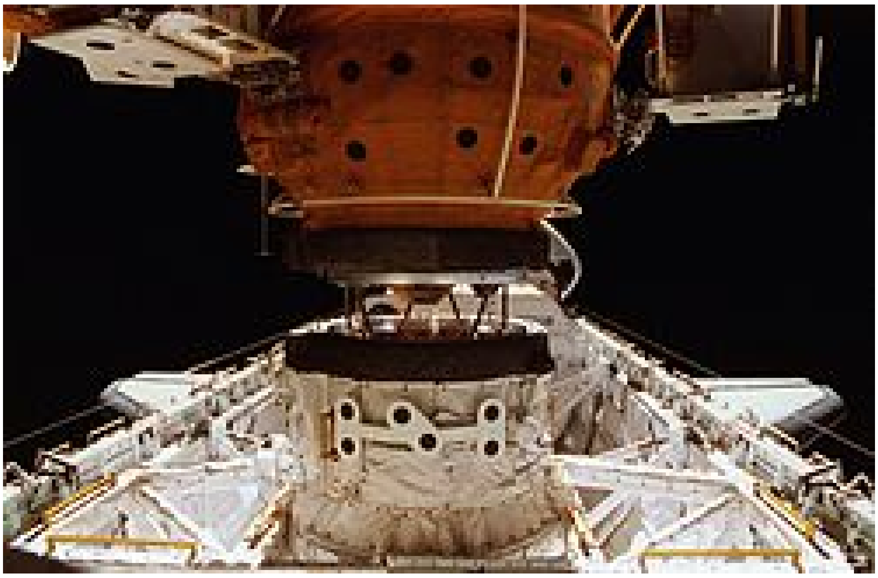


Figure 2.8: APAS System

### 2.0.4 IDSS

The international standard mechanism for spacecraft mating is the IDSS (International Docking System Standard). The first project was created in 2010 through a collaboration of the major space agencies. The goal is to provide distinct basic common design parameters that must be followed in order to create a mating mechanism for use with the International Space Agency. It is a peripheral androgynous system that allows cargo, crew, energy, and data to be transferred via dedicated connections. It is made up of two identical elements. Each component is made up of a ring that represents the mating surface, three guide petals, and a set of guide pins for proper alignment. The soft capture system is represented by a set of mechanical capture latches, whereas the hard capture system is represented by a set of hooks placed along the mating surface. The NDS is an example of an IDSS-compliant system (NASA Docking System). It is an evolution of the previous APAS system. The ring contains electromechanical actuators that form an active Stewart-Gough platform. ESA's IBDM (International Berthing and Docking Mechanism) is another example of an IDSS-compliant mechanism. This mechanism must still be tested in space. [5] .

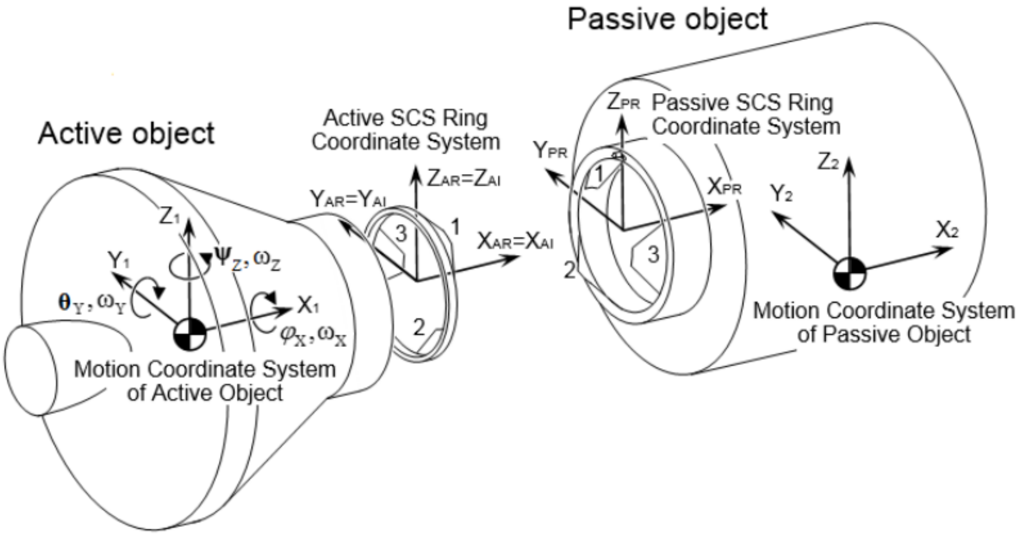


Figure 2.9: Coordinate system of docking objects (active and passive)

# Chapter 3

# Hardware Chapter

## 3.1 Mechanical Parts

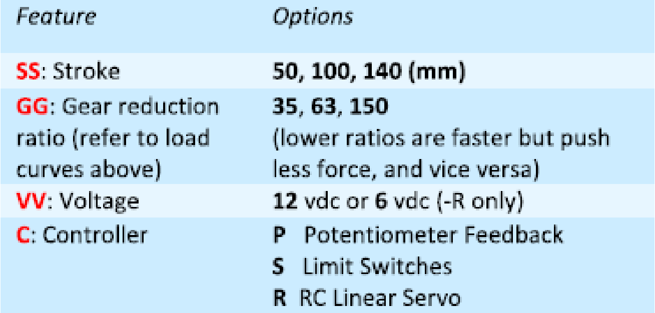
### 3.1.1 Linear Actuator

#### What Is An Electric Linear Actuator and How does it work?l

An electric linear actuator is a device that converts an AC or DC electric motor's rotational motion into linear motion. The linear motion is created by rotating the actuator's screw via the motor. The screw turns either clockwise or counter-clockwise, and this causes the shaft (which is a nut on the screw) to move in a line, up and down, creating the push/pull effect for the load. A Direct Current (DC) motor, is a rotary machine that converts electric energy into mechanical energy. This functionality is based on the principle of induction – an electromagnetic force created by input current, which in turn creates a rotational movement.

#### Chosen Linear Actuator

The Linear Actuator we've chosen is the Actuonix Linear Actuator of the Miniature Linear Motion Series - L16, in Figure 3.3 the specifications are shown.



| *Feature* | *Options* |
|---|---|
| SS: Stroke | **50, 100, 140 (mm)** |
| GG: Gear reduction ratio (refer to load curves above) | **35, 63, 150**<br>(lower ratios are faster but push less force, and vice versa) |
| VV: Voltage | **12** vdc or **6** vdc (-R only) |
| C: Controller | **P** Potentiometer Feedback<br>**S** Limit Switches<br>**R** RC Linear Servo |

Figure 3.1: Options

For the scale realization we've chosen to take the 150:1 version since we decided that the stroke of 140mm was what I needed.

This type of Actuator is designed to pull or push a load at its full stroke length. As written in the datasheet the Actuator speed can be reduced or augmented by modifying the drive voltage.

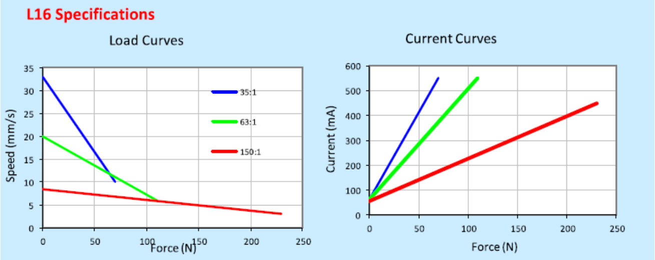
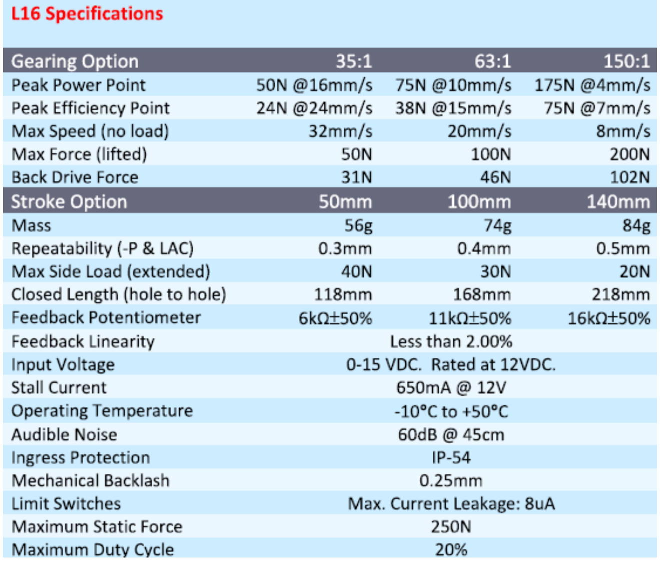


Figure 3.2: Load curves

In Figure 3.1 the options from which I could choose are shown. Because I needed a bus to feed the feedback signal, the Actuonix actuator already had it so I've chosen the Option P - Potentiometer position Feedback which gives me that opportunity.



L16 Specifications

| Gearing Option | 35:1 | 63:1 | 150:1 |
|---|---|---|---|
| Peak Power Point | 50N @16mm/s | 75N @10mm/s | 175N @4mm/s |
| Peak Efficiency Point | 24N @24mm/s | 38N @15mm/s | 75N @7mm/s |
| Max Speed (no load) | 32mm/s | 20mm/s | 8mm/s |
| Max Force (lifted) | 50N | 100N | 200N |
| Back Drive Force | 31N | 46N | 102N |
| **Stroke Option** | **50mm** | **100mm** | **140mm** |
| Mass | 56g | 74g | 84g |
| Repeatability (-P & LAC) | 0.3mm | 0.4mm | 0.5mm |
| Max Side Load (extended) | 40N | 30N | 20N |
| Closed Length (hole to hole) | 118mm | 168mm | 218mm |
| Feedback Potentiometer | 6kΩ±50% | 11kΩ±50% | 16kΩ±50% |
| Feedback Linearity | Less than 2.00% | | |
| Input Voltage | 0-15 VDC. Rated at 12VDC. | | |
| Stall Current | 650mA @ 12V | | |
| Operating Temperature | -10°C to +50°C | | |
| Audible Noise | 60dB @ 45cm | | |
| Ingress Protection | IP-54 | | |
| Mechanical Backlash | 0.25mm | | |
| Limit Switches | Max. Current Leakage: 8uA | | |
| Maximum Static Force | 250N | | |
| Maximum Duty Cycle | 20% | | |

Figure 3.3: L16 Specifications

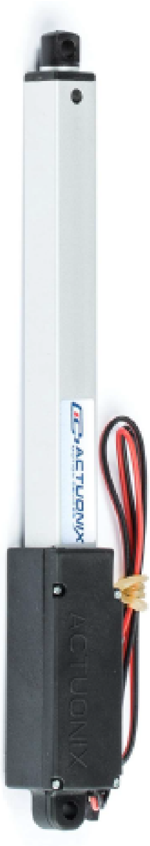
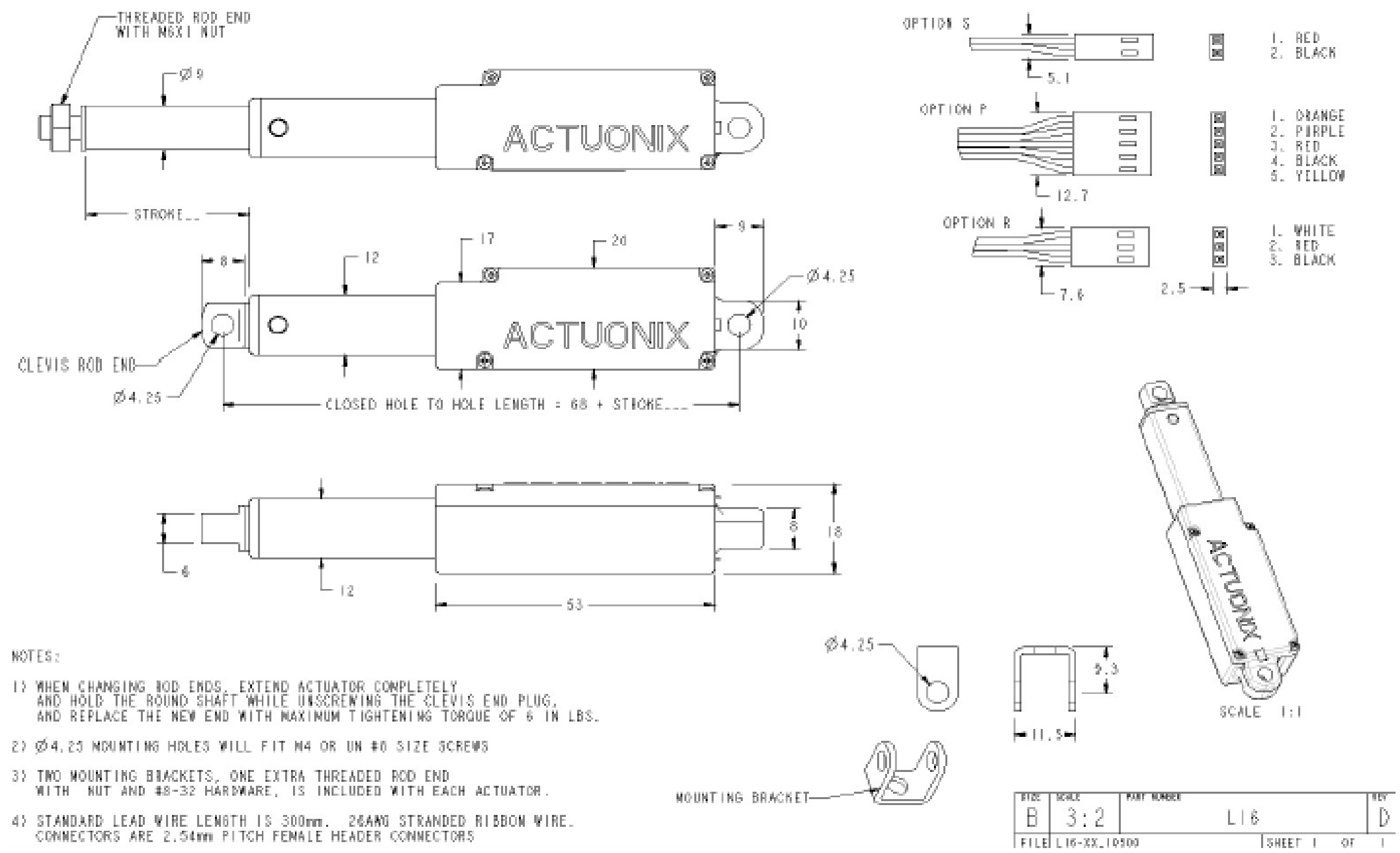


Figure 3.4: linear actuator

Regarding the wiring these options gave me: a feedback Potentiometer negative reference rail, a Feedback Potentiometer wiper, A Motor V+ (6V or 12V) connector, a Motor V- (Ground) connector and a Feedback potentiometer positive reference rail.

This configuration permits me to automatically control positioning system, which is relevant in our application (due to inverse kinematics implementation), even though the controller is not already inside the actuator and future work on the project should implement custom. While the voltage is applied to motor power pins the actuator extends, while instead if you invert the pin the actuator retracts. This inversion can be implemented differently but what I've chosen is the usage of H-bridge components.Pros of H-bridge will be discussed in the Electronic Chapter. The stroke position can be monitored by the internal potentiometer.



NOTES:

1) WHEN CHANGING ROD ENDS, EXTEND ACTUATOR COMPLETELY AND HOLD THE ROUND SHAFT WHILE UNSCREWING THE CLEVIS END PLUG, AND REPLACE THE NEW END WITH MAXIMUM TIGHTENING TORQUE OF 6 IN LBS.

2) Ø4.25 MOUNTING HOLES WILL FIT M4 OR UN #8 SIZE SCREWS

3) TWO MOUNTING BRACKETS, ONE EXTRA THREADED ROD END WITH NUT AND #8-32 HARDWARE, IS INCLUDED WITH EACH ACTUATOR.

4) STANDARD LEAD WIRE LENGTH IS 300mm. 26AWG STRANDED RIBBON WIRE. CONNECTORS ARE 2.54mm PITCH FEMALE HEADER CONNECTORS

Figure 3.5: Data sheet

### 3.1.2 Cardan and revolut design joint

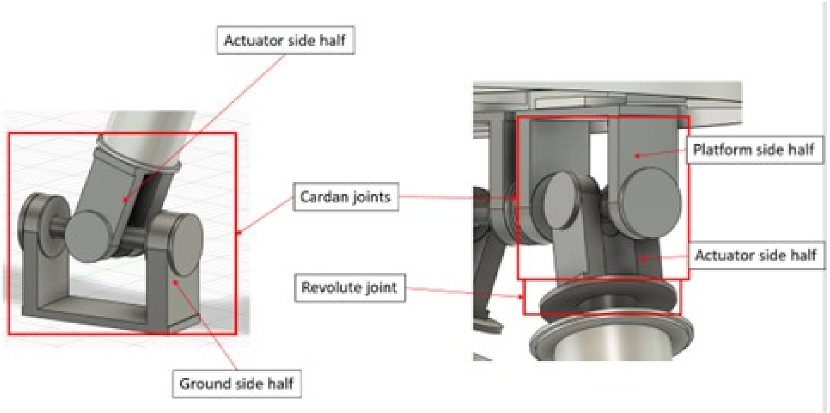


Figure 3.6: Cardan and revolut joint

This joint also called a universal joint is a joint or coupling connecting rigid shafts whose axes are inclined to each other. It is commonly used in shafts that transmit rotary motion. It consists of a pair of hinges located close together, oriented at 90° to each other, and connected by a cross shaft. The universal joint is not a constant-velocity joint. Cardan joints are typically used with independent and beam axles, and can accommodate high angles which are suitable for our application and for the Stewart platform itself.

## 3.2 Inverse Kinematics

To reach the desired position we need to calculate from the final position (in coordinates) the precise Rotation matrix and translation matrix needed to move correctly the actuators. To implement it is used the Inverse Kinematics.

In our case we will fake to already have calculated the two matrices and we will start operating with them.

[There are three simple rotation matrices, one for every axis x,y, z] [A complete rotation matrix is a succession of three rotations for which there are different solutions]

To obtain the rotational matrix for the moving platform I used the Euler angles, so that the rotation angles concerning z,x and y axes are carried out.

$$R_P^B = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{3.1}$$

$$R_P^B = \begin{vmatrix} c(\beta)c(\gamma) & s(\alpha)s(\beta)\mathrm{c}(\gamma) - c(\alpha)s(\gamma) & c(\alpha)s(\beta)c(\gamma) + s(\alpha)s(\gamma) \\ c(\beta)s(\gamma) & s(\alpha)s(\beta)s(\gamma) + c(\alpha)c(\gamma) & c(\alpha)s(\beta)s(\gamma) + s(\alpha)c(\gamma) \\ -s(\beta) & s(\alpha)c(\beta) & c(\alpha)c(\beta) \end{vmatrix} \tag{3.2}$$

Then we can generalize coordinate position of the moving platform with respect to the fixed one as follows.

$$q = [t_x t_y t_z \alpha\beta\gamma]^T \tag{3.3}$$

The inverse kinematic theory stated that the length of the legs, in our cases of the linear actuators, is found according to planned trajectories of the moving platform position. As already stated before we will end up with six coordinates of which three of them are rotational and three translational.

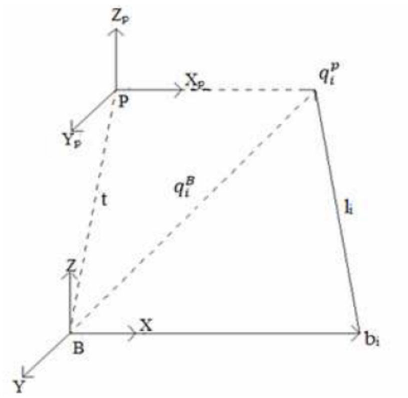


Figure 3.7: Closed-Loop of one Leg

To study the length of the linear actuator of the Stewart platform, the closed-loop of one leg is considered as shown in Figure 3.8 Two reference frames are needed and they are the mass centre of the moving platform P $(X_p, Y_p, Z_p)$, and the second one is base platform B $(X, Y, Z)$.

$$L_i = q_i^B - b_i \tag{3.4}$$

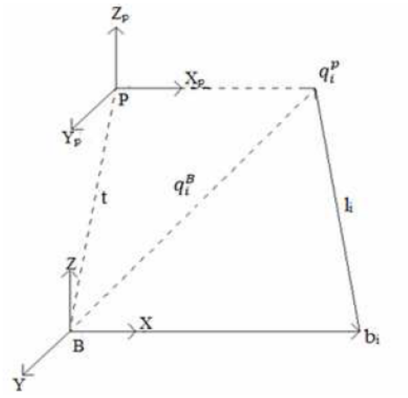


Figure 3.8: Closed-Loop of one Leg

Here the equation of the $i$th upper junction point with respect to B:

$$q_i^B = t + R_P^B q_{Pi} \tag{3.5}$$

After having substituted eq.() in equation () it results in :

$$L_i = T_P^B q_i^P - b_i = t + R_P^B q_i^P - b_i \tag{3.6}$$

$$L_i = L_{xi} + L_{yi} + L_{zi} \tag{3.7}$$

Now that we have the vectorial expression of the $i$th length we just need to apply the Euclidian Norm to it resulting in the length module:

$$l_i = \sqrt{(L_{xi})^2 + (L_{yi})^2 + (L_{zi})^2(3.8)}$$

Once we have the length of our linear actuator we will know how and how much to power up the DC motors to reach that length.

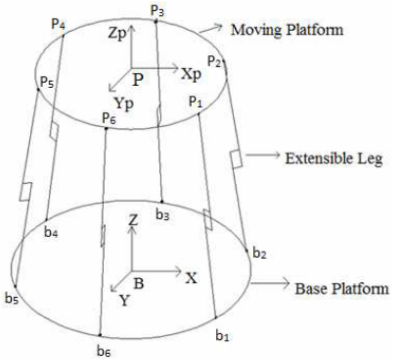


Figure 3.9: Schematic representation of the Stewart Platform

# Chapter 4

# Electronic Chapter

## 4.1 H-Bridge

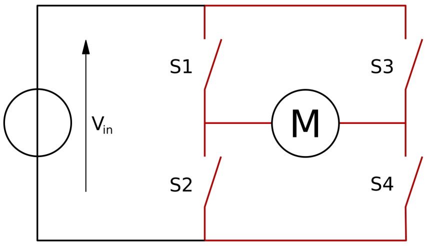
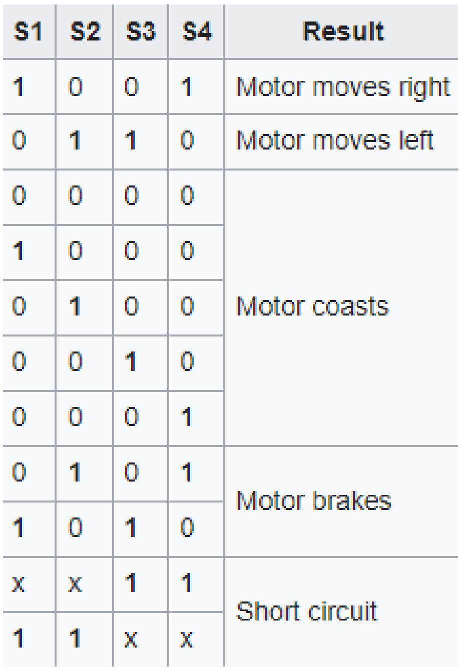


Figure 4.1: H-bridge circuit

The H-bridge term comes from its typical structure as we can see in Figure 4.1. An H-bridge is built with four switches (solid-state or mechanical). When switches S1 and S4 (according to the first figure) are closed (and S2 and S3 are open) a positive voltage is applied across the motor. By opening the S1 and S4 switches and closing the S2 and S3 switches, this voltage is reversed, allowing the reverse operation of the motor.

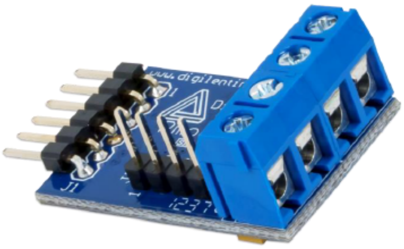
Using the nomenclature above, switches S1 and S2 should never be closed at the same time, as this would cause a short circuit on the input voltage source. The same applies to switches S3 and S4. This condition is known as shoot-through. An H-bridge is used to supply power to a two-terminal device. By proper arrangement of the switches, the polarity of the power to the device can be changed. Two examples are discussed below, DC motor Driver and transformer of a switching regulator. Note that, not all of the case of switching condition is safe. The "short"(see below in the "DC motor driver" section) cases are dangerous to the power source and the switches.

Changing the polarity of the power supply to the DC motor is used to change the direction of rotation. Apart from changing the rotation direction, the H-bridge can provide additional operation modes, "brake" and "free run until frictional stop". The H-bridge arrangement is generally used to reverse the polarity/direction of the motor, but can also be used to 'brake' the motor, where the motor comes to a sudden stop, as the motor's terminals are shorted. In shorted case, the kinetic energy of a rotating motor is consumed rapidly in form of electrical current in the shorted circuit. The other case, is to let the motor 'free run' to a stop, as the motor is effectively disconnected from the circuit. The following table summarizes the operation, with S1-S4 corresponding to the diagram above. In the table below, "1" is used to represent the "on" state of the switch, "0" to represent the "off" state [4].



| S1 | S2 | S3 | S4 | Result |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | Motor moves right |
| 0 | 1 | 1 | 0 | Motor moves left |
| 0 | 0 | 0 | 0 | Motor coasts |
| 1 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 0 | |
| 0 | 0 | 0 | 1 | |
| 0 | 1 | 0 | 1 | Motor brakes |
| 1 | 0 | 1 | 0 | |
| x | x | 1 | 1 | Short circuit |
| 1 | 1 | x | x | |

Figure 4.2: Options



*The PmodHB3.*

Figure 4.3: H-bridge

## 4.2 DC-motor

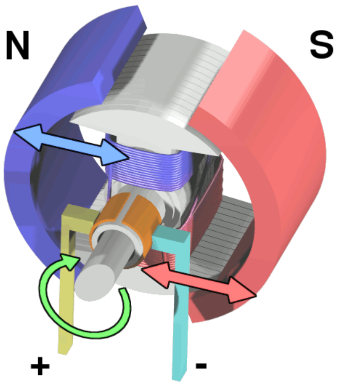


Figure 4.4: DC-motor

A DC motor is any of a class of rotary electrical motors that converts direct current (DC) electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in a part of the motor.

DC motors were the first form of motor widely used, as they could be powered by existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of the current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in the propulsion of electric vehicles, elevators and hoists, and drives for steel rolling mills. The advent of power electronics has made the replacement of DC motors with AC motors possible in many applications.

# Chapter 5

# Simulation

## 5.1 Purpose and Implementation

The purpose of the simulation is to study the feasibility of the inverse kinematics and the correctness of the formulae. The simulation will be performed in a Matlab environment using also Simulink ambient. Firstly in Matlab I address the coordinate of the points and performed the length of the elongated legs of the Stewart platform. Once the length are calculated I was ready to simulate the kinematic of the solution in Simulink, the Simulation was performed with a closed loop feedback controller using an already existing PID model of Simulink (a brief explanation of a PID controller will be given later). A suitable representation of DC motors and H-bridge was chosen to simulate approximately but accurately what could have been the final output.

## 5.2 How a PID controller works and what is it

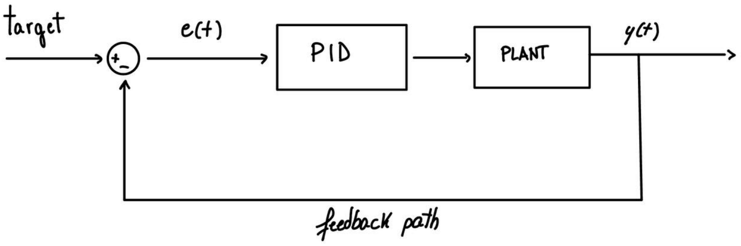


Figure 5.1: PID

To use in a simple way all these tools are usually to simplify in blocks specific parts of the system. As we can see from Figure 5.1 we have PID and PLANT blocks, the first one is our controller while the latter is the physical system transformed in a mathematical (or a physical to Simulink as we will see) way for better performance in a simulation, it also is the system we want to control.

The signal entering the plant is called an "actuating signal", it is generated by the PID controller which is fed by the error commonly called *e(t)* which is the result of a subtraction among target value and feedback value as shown in the figure. The simple way to see how it works is: " We give a target value to the PID, which sees which is the actual value and based on the error provides a signal to the lant to reach the provided target value.

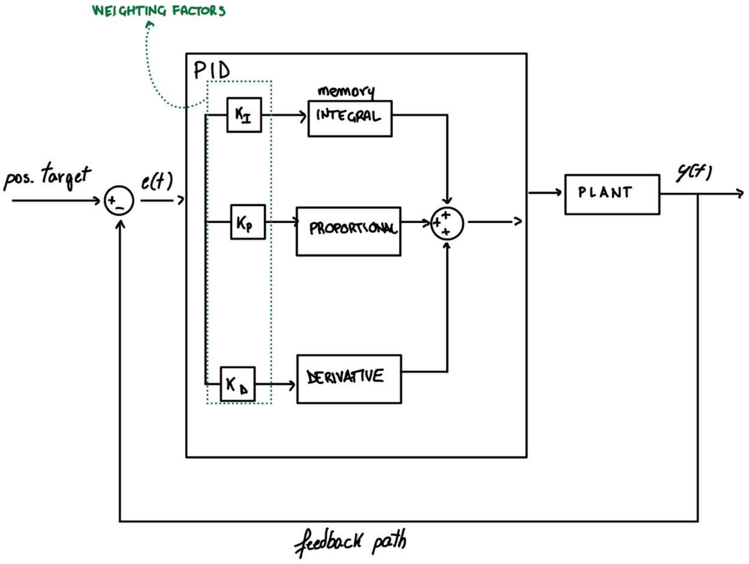


Figure 5.2: PID

Here in Figure 5.2 we can see what a complete PID looks like, it is also worth it to mark that P stand for Proportional, i for Integrative and D for Derivative, they are the fundamental actors of our controller and a brief explanation will be given.

Looking at the previously mentioned figure we can see that for every block (Proportional Integrative and Derivative) there are also some weighting factors, that allow proper tuning to give more relevance to one block or the others. The proportional action takes account of the so-called *present error*, relying on how "much effort" is needed to reach the target value. the problem with (only) that it is that we can occur on a steady state error. To mitigate it we operate also with the Integral block, which takes account of the *past error* which in a certain way calculate how many time the same error is counted, giving more effort in the case the error is present for a long time. There are cases where significant changes occur and the Integral keep adding error, to adjust that behaviour a Derivative block is added, it takes into account the *future error* looking at how fast the output is approaching the target value and slowing it down once the gap is closing.

Figure 5.3 shows an example of a case where our linear actuator for some motivation cannot stretch and keeps, for a certain amount of time, the error steady (pink line). In this case we can see as the integral command keeps growing because it sees that the error doesn't go down, consequentially the rpm command continues growing until it reaches its limit (green line). The fact the rpm reaches its saturation limit doesn't stop the integral which keeps growing its error creating an error which can become relevant if the error grows without control (thanks to the derivative block we will be able to control it). Let's imagine the case that our actuator suddenly starts lengthening (knee of pink line) and the error will start going down since the rpm command is high. As the figure show there will be a period where the integral command will ask for an rpm higher than the saturation point resulting in a non-decreasing rpm while approaching the target position. It will end in an exaggerated lengthening of the actuator over the target value.

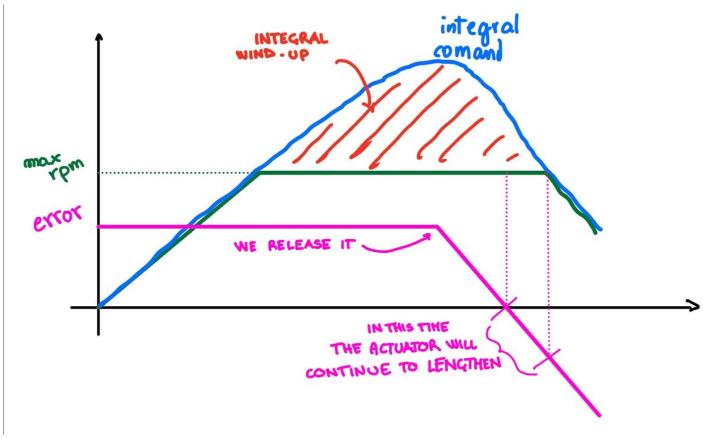


Figure 5.3: PID behaviour

## 5.3 Matlab code

### 5.3.1 Introduction

The Matlab platform is used as a base platform to carry out the algebraic accounts. First I have defined the starting position characteristics for both the fixed and the moving platform, the vector which relates the initial rest position of the moving platform with respect to the fixed one. Then as the reader will be able to see below, the different working values are depicted. These different situations have been chosen to be able to depict every possible situation that the Stewart platform could be called to actuate.

To be clear and complete there could be another situation where the centre of rotation is not the centre of the moving platform even though generally it will not be the case because of the fundamental request of simplicity and because of the fact the translation a rotation itself can make up for that case (it is worth mentioning that even the rover will be able to move and slide, correcting the bigger translation gap to the objective final position.

### 5.3.2 Code Explanation

As shown in chapter 3 the Stewart platform is described with vectors of which the description is done in the chapter mentioned before. Below we can see the starting vector values representing the rest position for our platform.

For clarity beside the $\mathrm{q_{pi}}$ has been added the position of the same joint concerning the fixed base frame. Going forward the expression of $t$ and $q$ can be found, which refer to the reciprocal position of the moving platform concerning the fixed one.

```
clear all
close all
clc

% STARTING POSITION
b_1 = [5; -33; 0];
b_2 = [26; 22; 0];
b_3 = [32; 11; 0];
b_4 = [-26; 22; 0];
b_5 = [-32; 11; 0];
b_6 = [-5; -33; 0];

q_p_1 = [25; -20; 0];
q_p_2 = [30; -11; 0];
q_p_3 = [4; 32; 0];
q_p_4 = [-4; 32; 0];
q_p_5 = [-30; -11; 0];
q_p_6 = [-25; -20; 0];

t_x = 0;
t_y = 0;
t_z = 16;
```

```
t = [t_x; t_y; t_z];

ah_f = 0;
bh_f = 0;
gh_f = 0;

q=[t_x, t_y, t_z, ah_f, bh_f, gh_f]';

```

In the following code as the title already explain the working values and working positions are completed. Three cases are illustrated:

- JUST TRANSLATION
- JUST ROTATION
- ROTATION AND TRANSLATION

In every case both ????Euler???? angles and translation values are shown, just for sake of clarity.

During the simulation one can ”commentate” (and so make the code unreadable for the coder) 2 of the 3 cases to apply only the remaining values to simulate. Doing so the final rotation matrix $R_{bp}$ will be filled with the wanted values.

```
 %% WORKING VALUES

% JUST TRANSLATION
ah = 0;
bh = 0;
gh = 0;

t_s = [0,10,7];

% JUST ROTATION
ah = pi;
bh = pi/4;
gh = pi/8;

t_s = [0,0,0];

%ROTATION AND TRANSLATION
ah = pi;
bh = pi/4;
gh = pi/8;

t_s = [0,10,7];

% R_bp = R_z(gh)*R_y(bh)*R_x(ah)

R_bp = [cos(bh)*cos(gh), sin(ah)*sin(bh)*cos(gh)-cos(ah)*sin(gh),
    cos(ah)*sin(bh)*cos(gh)+sin(ah)*sin(gh);
```

```
        cos(bh)*sin(gh), sin(ah)*sin(bh)*sin(gh)+cos(ah)*cos(gh), cos(ah)*sin(bh)*sin(gh)+sin(ah)*cos(gh);
        -sin(bh), sin(ah)*cos(bh), cos(ah)*cos(bh)];

```

Equations of moving platform joints with respect to the fixed one

```
%% q_i_b values

q_1_b = t + R_bp*q_p_1
q_2_b = t + R_bp*q_p_2
q_3_b = t + R_bp*q_p_3
q_4_b = t + R_bp*q_p_4
q_5_b = t + R_bp*q_p_5
q_6_b = t + R_bp*q_p_6

```

As shown in chapter 3.2 here are coded the length equation for every Stewart leg.

```
%% Length
L_1=t + R_bp*q_p_1 - b_1;
L_2=t + R_bp*q_p_2 - b_2;
L_3=t + R_bp*q_p_3 - b_3;
L_4=t + R_bp*q_p_4 - b_4;
L_5=t + R_bp*q_p_5 - b_5;
L_6=t + R_bp*q_p_6 - b_6;

l_1 = norm(L_1,2);
l_2 = norm(L_2,2);
l_3 = norm(L_3,2);
l_4 = norm(L_4,2);
l_5 = norm(L_5,2);
l_6 = norm(L_6,2);

```

# 5.4 Simulink structure

## 5.4.1 Schematics

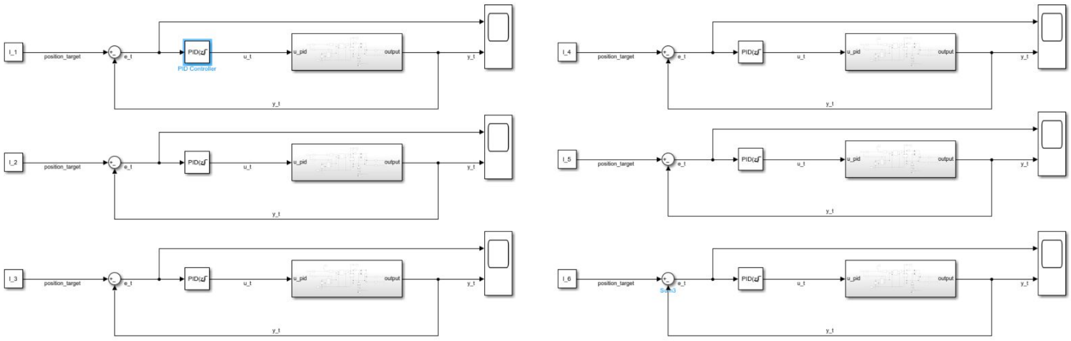


Figure 5.4: Overall view

As already said before Simulink has been used to simulate the system, Simulink is a MATLAB-based graphical programming environment for modelling, simulating and analyzing multidomain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries [6]

Figure 5.4 shows the overall view of our system, six are the subsystem and six are Stewart's legs. The reason why different subsystems were made is only due to the software. Since auto-tuning was used to have a better performance of the PID controller there was a need to have a simple plant to make it easy to linearize. So first a single system has been developed, tuned and configured and then the same system was copied and pasted for the other five linear actuators, the only thing that has changed for every linear actuator is the target value.

To make the final simulation useful six scopes have been added one for every linear actuator. They will be used to see the simulation results. Every value was taken from the MATLAB workspace with the code exploited before in chapter 5 section 5.4.

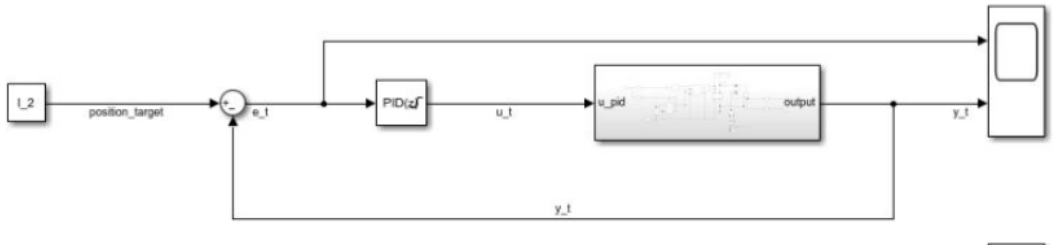


Figure 5.5: Plant single leg

Figure 5.5 shows a single linear actuator system where it can be seen the target value in the right square, the sum block performs the "error", the PID block has been used in a discrete mode to simplify the calculations and at the end the Plant which is shown in Figure 5.6.

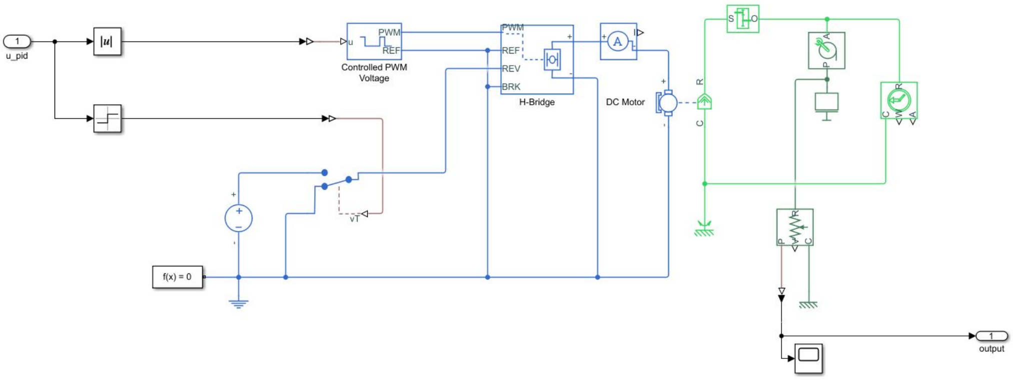


Figure 5.6: Plant

Starting from the left, it can be seen that the actuating signal received by the controller is received by two different blocks. The first at the top serves to maintain only the positive value of this signal to work only on its module.The second instead serves to "extract" the sign of the actuating signal, it will serve the H-bridge to invert the voltage to implement the extension and retraction of the linear actuators. It will be discussed later. At the bottom left you can see a switch used only for simulation purposes, it reacts to the change of sign of the previously transformed signal by causing the required voltage to be changed from positive to negative depending on the sign of the actuating signal.

On the right it can be seen the mechanical part.To simulate our linear actuator, a toothed wheel was inserted to simulate the transmission ratio between the DC motor shaft and the worm screw of the linear actuator whose task is to lengthen or shorten the leg.

### 5.4.2 Components explanation

Figure 5.7 shows the PID component, a specific explanation of how it works is given in section 5.2, and from here the actuating signal is provided.

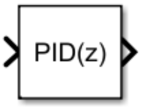
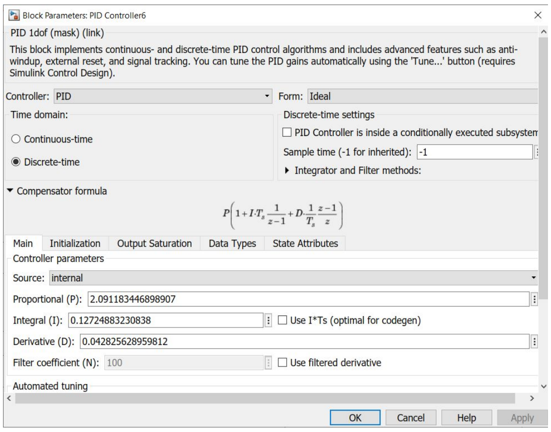


Figure 5.7: Prova

The PID component was set up like this:


Block Parameters: PID Controller6

PID 1dof (mask) (link)

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PID

Form: Ideal

Time domain:

○ Continuous-time

● Discrete-time

Discrete-time settings

□ PID Controller is inside a conditionally executed subsystem

Sample time (-1 for inherited): -1

Integrator and Filter methods:

Compensator formula

$$P\left(1+I\cdot T_s\frac{1}{z-1}+D\cdot\frac{1}{T_s}\frac{z-1}{z}\right)$$

Controller parameters

Source: internal

Proportional (P): 2.091183446898907

Integral (I): 0.12724883230838 □ Use I*Ts (optimal for codegen)

Derivative (D): 0.042825628959812

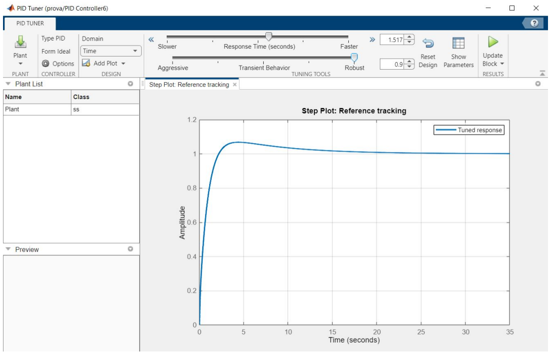
Filter coefficient (N): 100 □ Use filtered derivative

Automated tuning

Figure 5.8: PID settings

Discrete-time, Ideal form and no filter were used. In the main section you can see all the weighting factors that characterize the final answer.

The auto-tuning was used, following these settings:


PID Tuner (prova/PID Controller6)

Type PID

Form Ideal

Domain: Time

Response Time (seconds): 1.517

Transient Behavior: 0.9

Plant List

| Name | Class |
|---|---|
| Plant | ss |

Figure 5.9: PID tuning settings

Figure 5.10 shows the Controlled PVM Voltage, it must set the maximum pvm at which our DC motor will work.

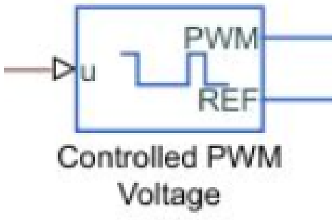
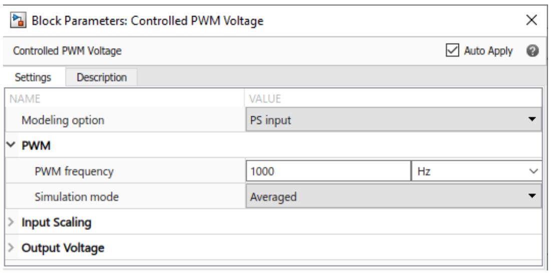


Figure 5.10: Controlled PVM Voltage

Settings:


Block Parameters: Controlled PWM Voltage

Controlled PWM Voltage ☑ Auto Apply

Settings | Description

| NAME | VALUE | |
|---|---|---|
| Modeling option | PS input | |
| PWM | | |
| PWM frequency | 1000 | Hz |
| Simulation mode | Averaged | |
| Input Scaling | | |
| Output Voltage | | |

Figure 5.11: Controlled PVM Voltage settings

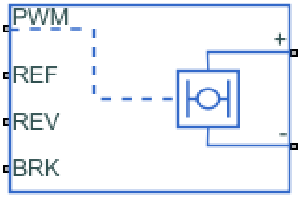
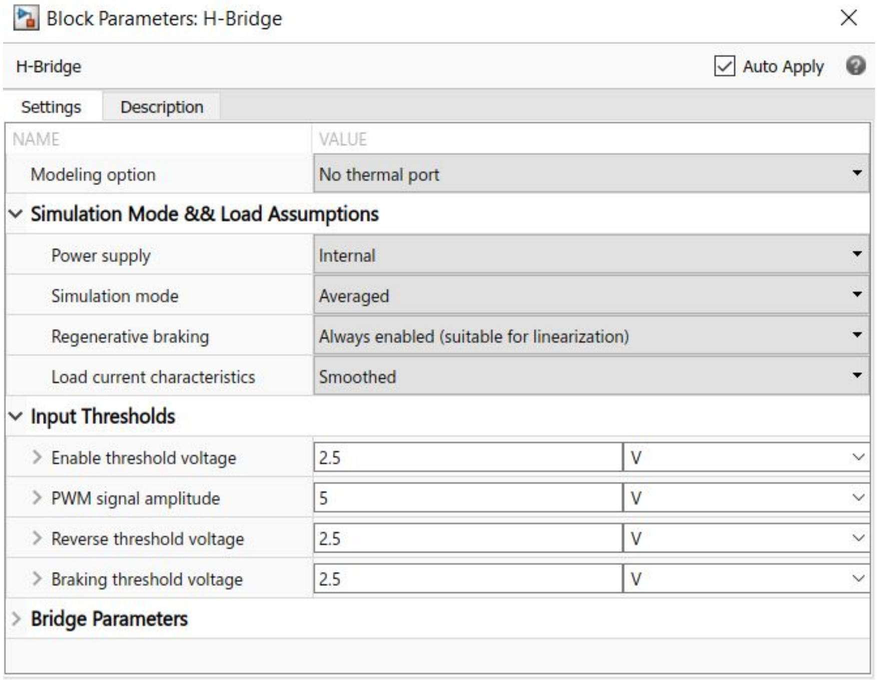


Figure 5.12: H-bridge component

In Figure 5.12 the H-Bridge component is shown, it simulates the H-bridge hardware component that will be implemented. Settings are shown down below.



Block Parameters: H-Bridge

H-Bridge ☑ Auto Apply

| NAME | VALUE |
|---|---|
| Modeling option | No thermal port |
| **Simulation Mode && Load Assumptions** | |
| Power supply | Internal |
| Simulation mode | Averaged |
| Regenerative braking | Always enabled (suitable for linearization) |
| Load current characteristics | Smoothed |
| **Input Thresholds** | |
| Enable threshold voltage | 2.5 V |
| PWM signal amplitude | 5 V |
| Reverse threshold voltage | 2.5 V |
| Braking threshold voltage | 2.5 V |
| **Bridge Parameters** | |

Figure 5.13: H-Bridge settings

Last but not list the DC motor component and its settings

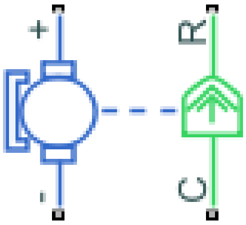
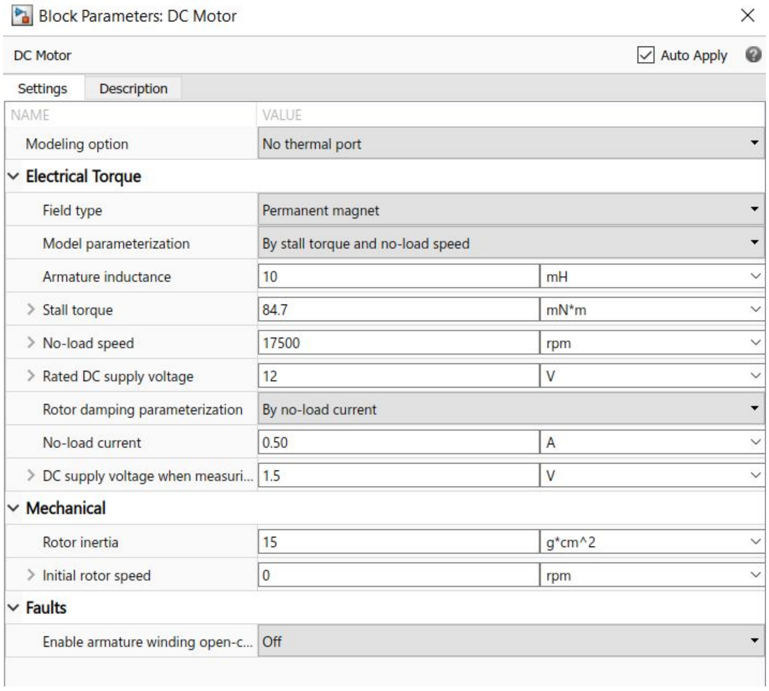


Figure 5.14: Prova



Block Parameters: DC Motor

DC Motor ☑ Auto Apply

| NAME | VALUE | |
|---|---|---|
| Modeling option | No thermal port | |
| **Electrical Torque** | | |
| Field type | Permanent magnet | |
| Model parameterization | By stall torque and no-load speed | |
| Armature inductance | 10 | mH |
| Stall torque | 84.7 | mN*m |
| No-load speed | 17500 | rpm |
| Rated DC supply voltage | 12 | V |
| Rotor damping parameterization | By no-load current | |
| No-load current | 0.50 | A |
| DC supply voltage when measuri... | 1.5 | V |
| **Mechanical** | | |
| Rotor inertia | 15 | g*cm^2 |
| Initial rotor speed | 0 | rpm |
| **Faults** | | |
| Enable armature winding open-c... | Off | |

Figure 5.15: DC motor settings

## 5.5 Simulation results

Here the simple result is depicted, they are expressed for some particular cases such as just rotation, just translation and finally translation and rotation together.

### 5.5.1 Rest Position

The different situations were simulated and visualized, starting from the rest position in which the maximum height of the Stewart platform is 16cm. As far as the code I developed is constructed, to reproduce the correct output simulation the height of the platform has to be inserted in the ”t” translation. For the following implementation, the translation vector will start from 16 and the true translation will be added.

Here’s the code about the working parameters:

```

            t_x = 0;
            t_y = 0;
            t_z = 16;
            t = [t_x; t_y; t_z];

            ah_f = 0;
            bh_f = 0;
            gh_f = 0;


```

A 3D vision has been implemented with *3D calculator-Geogebra*, the dimension is in a scale of 1:3 to represent the small HW model which later will be developed. As both the code and the model show the rest position height is 16cm.

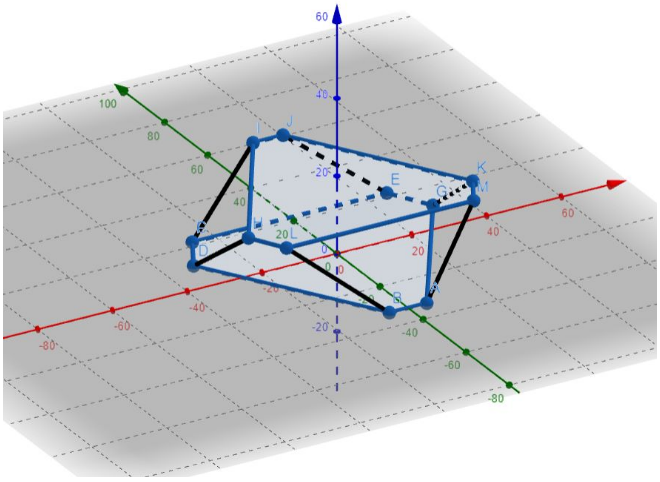


Figure 5.16: Rest position

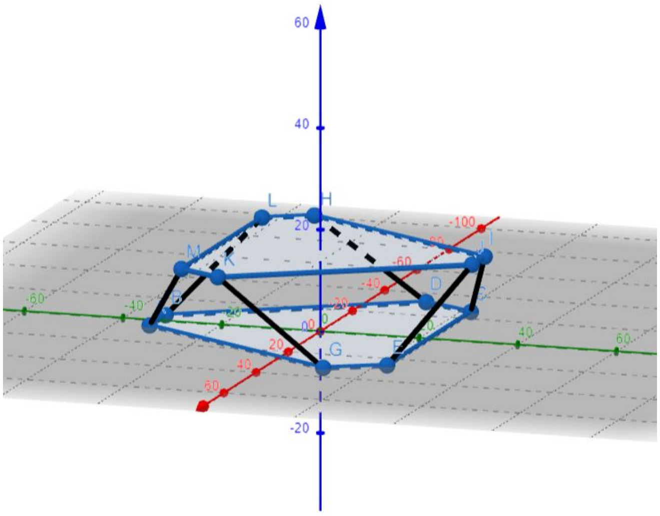
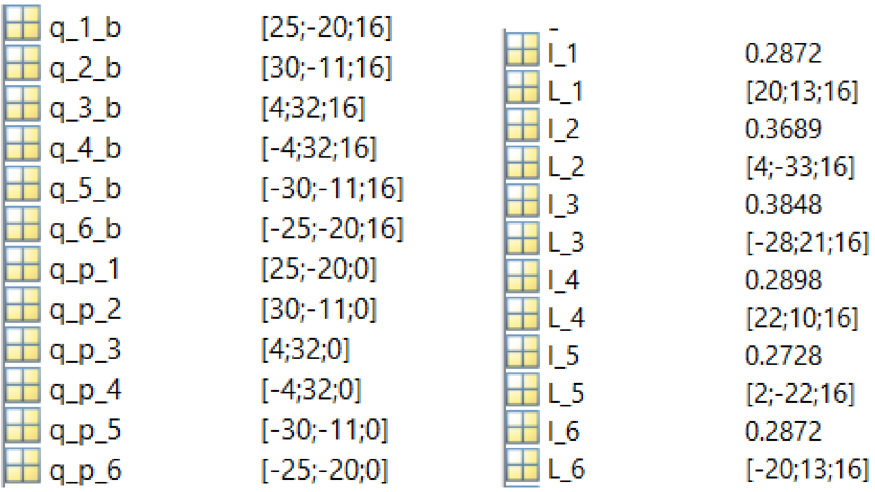


Figure 5.17: Rest position other perspective



| | |
|---|---|
| q_1_b | [25;-20;16] |
| q_2_b | [30;-11;16] |
| q_3_b | [4;32;16] |
| q_4_b | [-4;32;16] |
| q_5_b | [-30;-11;16] |
| q_6_b | [-25;-20;16] |
| q_p_1 | [25;-20;0] |
| q_p_2 | [30;-11;0] |
| q_p_3 | [4;32;0] |
| q_p_4 | [-4;32;0] |
| q_p_5 | [-30;-11;0] |
| q_p_6 | [-25;-20;0] |

(a)

| | |
|---|---|
| - | |
| l_1 | 0.2872 |
| L_1 | [20;13;16] |
| l_2 | 0.3689 |
| L_2 | [4;-33;16] |
| l_3 | 0.3848 |
| L_3 | [-28;21;16] |
| l_4 | 0.2898 |
| L_4 | [22;10;16] |
| l_5 | 0.2728 |
| L_5 | [2;-22;16] |
| l_6 | 0.2872 |
| L_6 | [-20;13;16] |

(b)

Figure 5.18: (a) Joint coordinate (b) Leg values

In Figure 5.18 (a) the Matlab results are printed, those are the joint values concerning the base frame representing the joint coordinates in the space. Those results are the coordinate values represented in Figures 5.16 and 5.17.

Figure 5.18 (b) instead, shows us the leg length. For future analysis we can state that the maximum starting leg extension is 38 cm while the minimum extension is 28cm.

### 5.5.2 Just Translation

A second simulation has been implemented, a simple translation of 5cm has been executed since the *z component* of the translation vector has been increased to 21cm from 16cm.

```

            t_x = 0;
            t_y = 0;
            t_z = 21;
            t = [t_x; t_y; t_z];

            ah = 0;
            bh = 0;
            gh = 0;



```

The results are shown down below.

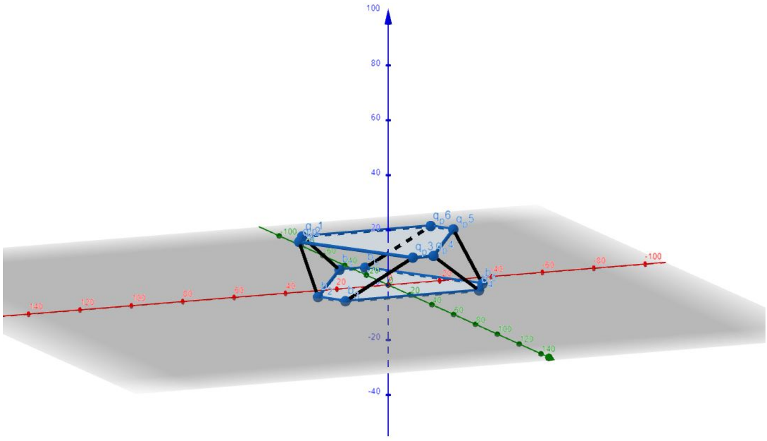
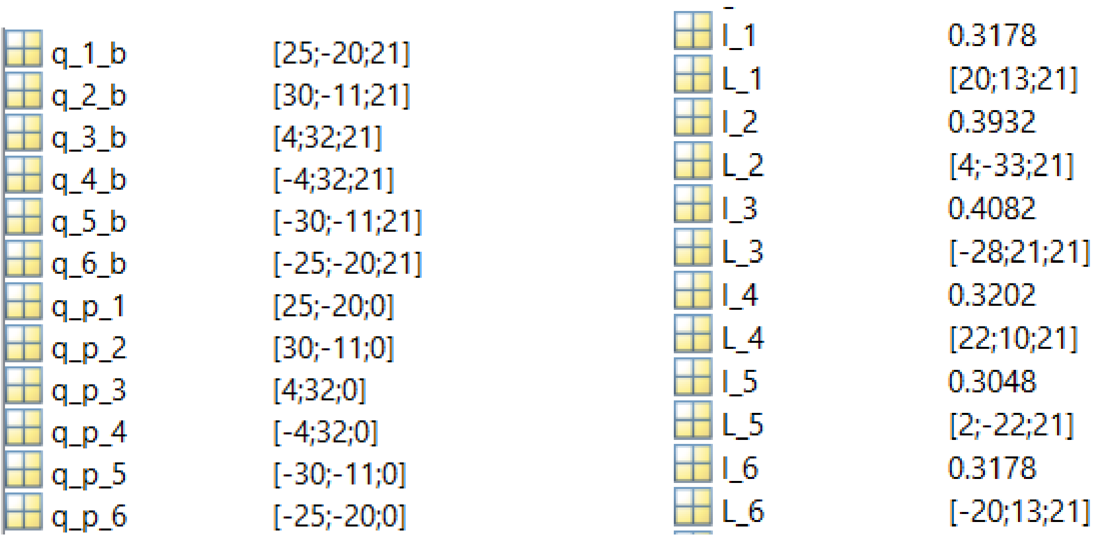


Figure 5.19: Translation only



| | |
|---|---|
| q_1_b | [25;-20;21] |
| q_2_b | [30;-11;21] |
| q_3_b | [4;32;21] |
| q_4_b | [-4;32;21] |
| q_5_b | [-30;-11;21] |
| q_6_b | [-25;-20;21] |
| q_p_1 | [25;-20;0] |
| q_p_2 | [30;-11;0] |
| q_p_3 | [4;32;0] |
| q_p_4 | [-4;32;0] |
| q_p_5 | [-30;-11;0] |
| q_p_6 | [-25;-20;0] |

(a)

| | |
|---|---|
| l_1 | 0.3178 |
| L_1 | [20;13;21] |
| l_2 | 0.3932 |
| L_2 | [4;-33;21] |
| l_3 | 0.4082 |
| L_3 | [-28;21;21] |
| l_4 | 0.3202 |
| L_4 | [22;10;21] |
| l_5 | 0.3048 |
| L_5 | [2;-22;21] |
| l_6 | 0.3178 |
| L_6 | [-20;13;21] |

(b)

Figure 5.20: (a) Joint coordinate (b) Leg values

As depicted in the previous simulation Figure 5.13 (a) shows the joint's coordinate values while 5.13 (b) shows the leg length.

Here we can see that every leg has increased its length as one can imagine. Considering the previous simulation the maximum stroke due to the 5cm translation is 3 cm in leg 6.

Just here graphs results are shown, to depict how the system performs and how the PID controller operates.

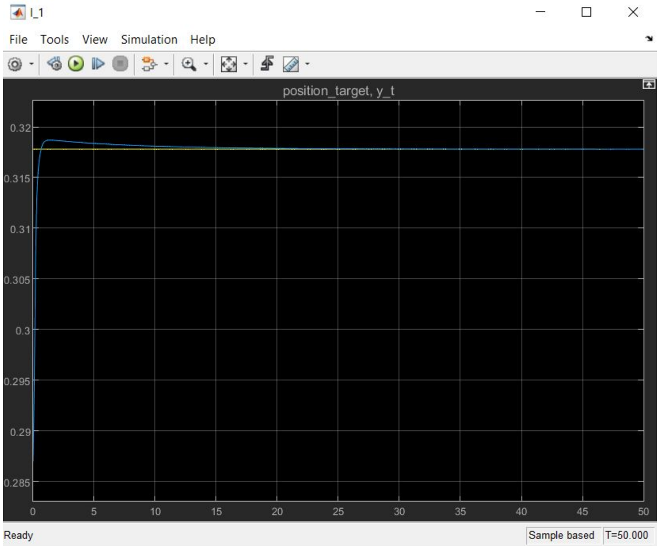


Figure 5.21: Linear actuator 1

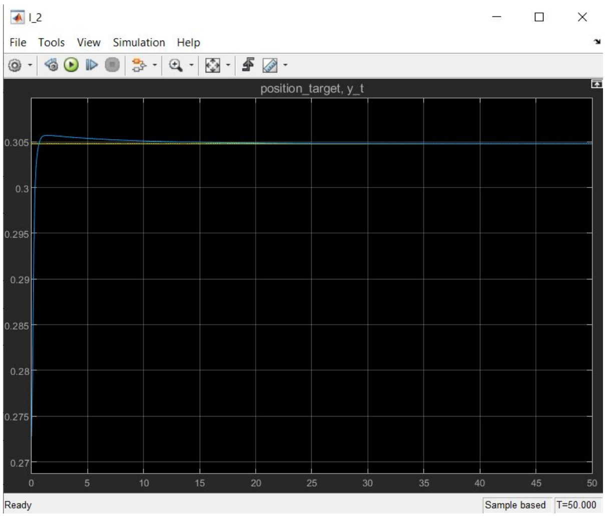


Figure 5.22: Linear actuator 2

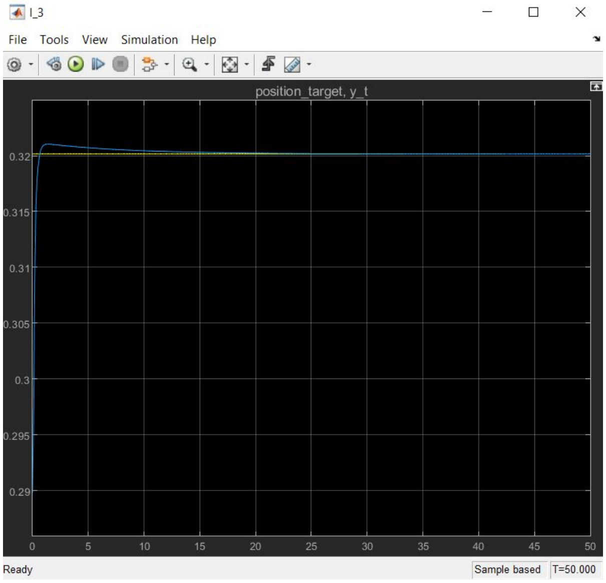


Figure 5.23: Linear actuator 3

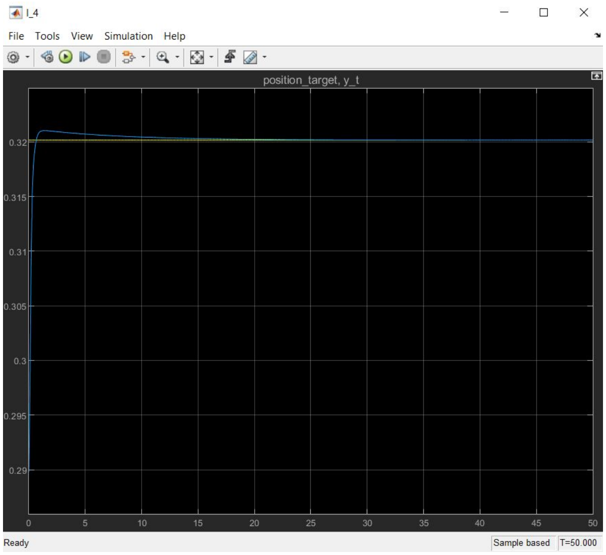


Figure 5.24: Linear actuator 4

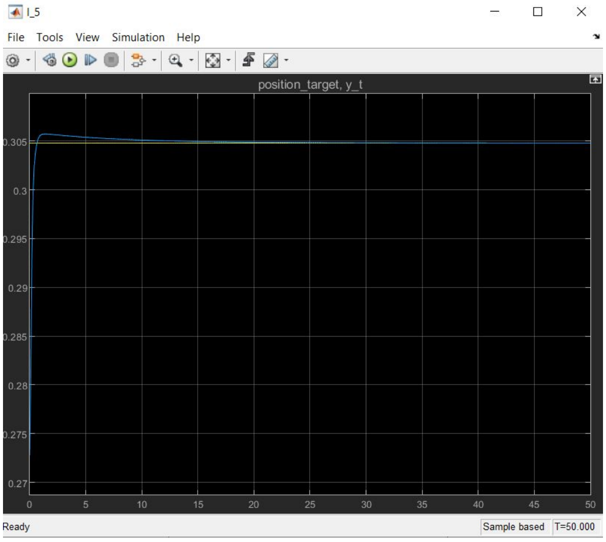


Figure 5.25: Linear actuator 5

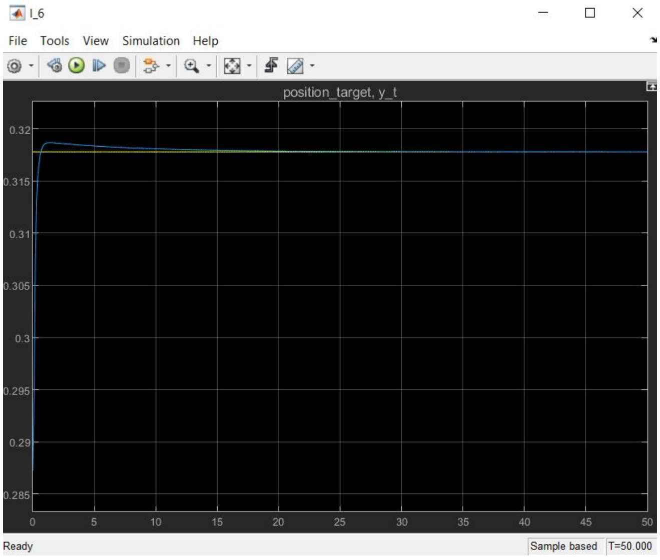


Figure 5.26: Linear actuator 6

### 5.5.3 Rotation about X and Translation

The third simulation has been developed under the assumption of translation ( 5 cm as before ) and rotation around the $x$ axis of $\pi/8$ . A semi-rotated view is given in figure 5.27 a while a side-view is presented in Figure 5.28.

```

                t_x = 0;
                t_y = 0;
                t_z = 21;
                t = [t_x; t_y; t_z];

                ah = pi/8;
                bh = 0;
                gh = 0;


```

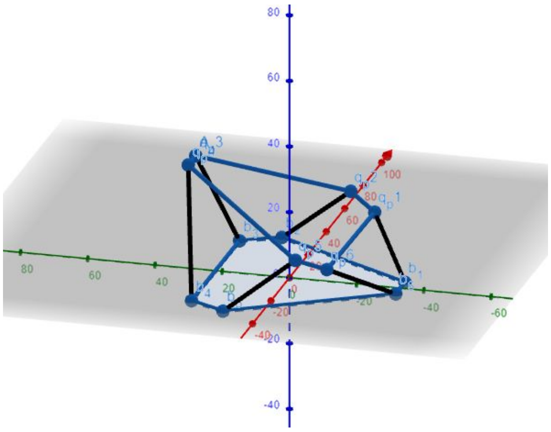


Figure 5.27: Stewart translation and rotation about x

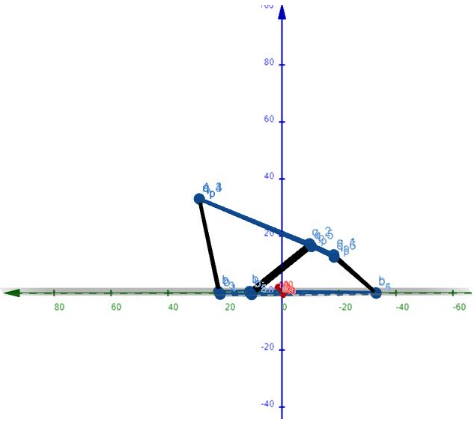
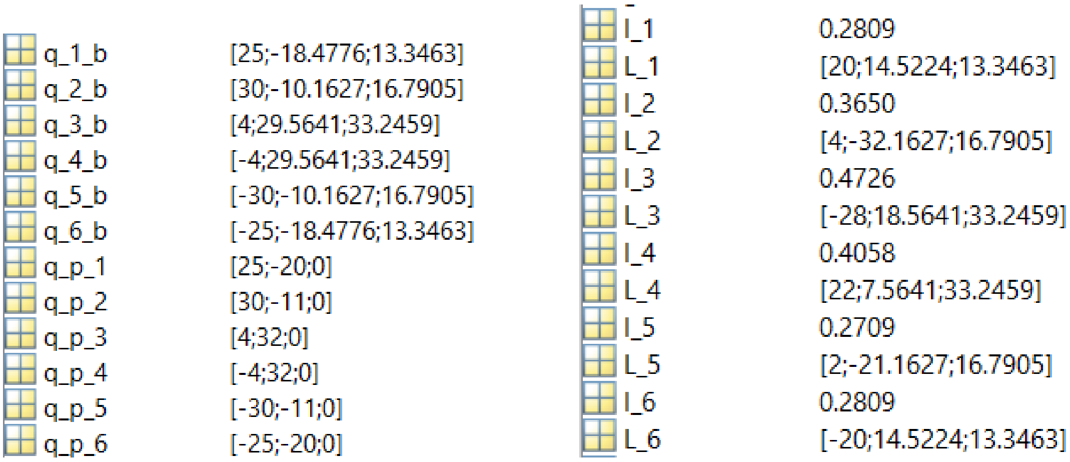


Figure 5.28: Stewart translation and rotation about x, side view

Looking at Figure **??** we can see that the plane formed by the moving platform joints is perfectly flat meaning that the inverse kinematics works well and it is representative of what could happen in real life, with real hardware.



```
q_1_b        [25;-18.4776;13.3463]
q_2_b        [30;-10.1627;16.7905]
q_3_b        [4;29.5641;33.2459]
q_4_b        [-4;29.5641;33.2459]
q_5_b        [-30;-10.1627;16.7905]
q_6_b        [-25;-18.4776;13.3463]
q_p_1        [25;-20;0]
q_p_2        [30;-11;0]
q_p_3        [4;32;0]
q_p_4        [-4;32;0]
q_p_5        [-30;-11;0]
q_p_6        [-25;-20;0]
```

(a)

```
l_1      0.2809
L_1      [20;14.5224;13.3463]
l_2      0.3650
L_2      [4;-32.1627;16.7905]
l_3      0.4726
L_3      [-28;18.5641;33.2459]
l_4      0.4058
L_4      [22;7.5641;33.2459]
l_5      0.2709
L_5      [2;-21.1627;16.7905]
l_6      0.2809
L_6      [-20;14.5224;13.3463]
```

(b)

Figure 5.29: (a)Joint coordinate (b) Leg values

As in the previous simulation, here in figure 5.29 we can see the joint coordinates values (Figure 5.29 (a)) and the leg length values (Figure 5.29 (b)). It can be observed as the results follow what the simulation hypothesis has guessed, the joints which are in the rotation direction go down while the opposite ones go up, due to the rotation around the x-axis. Here the maximum length reached is 40cm while the lowest value reached from at least one joint is 27 cm.

### 5.5.4 Rotation about Z and Translation

The last simulation has been done acting a translation (of 5cm) and a rotation around the *z axis* of $\pi/18$. The rotation of $\pi/18$ has been chosen to show how the platform work under both high and low values of rotation angles.

Down below are shown the parameters that are used for this simulation:

```

                t_x = 0;
                t_y = 0;
                t_z = 21;
                t = [t_x; t_y; t_z];

                ah = 0;
                bh = pi/18;
                gh = 0;


```

Results shown with *3D calculator-Geogebra*:

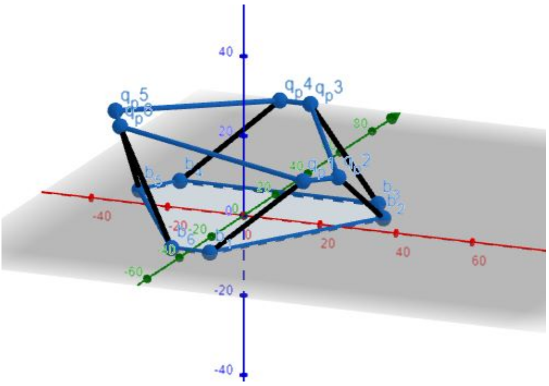
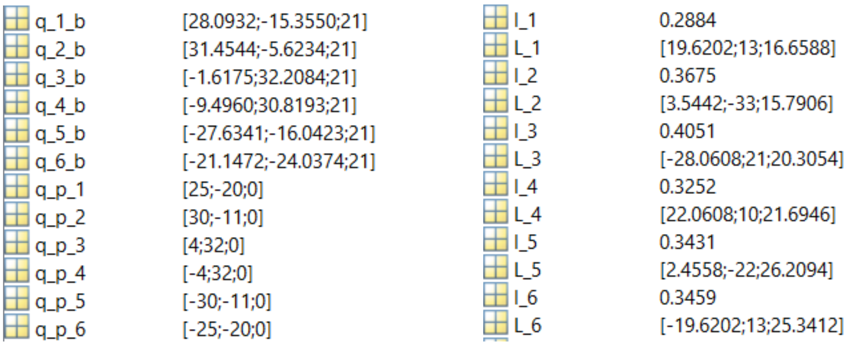


Figure 5.30: Prova



| | |
|---|---|
| q_1_b | [28.0932;-15.3550;21] |
| q_2_b | [31.4544;-5.6234;21] |
| q_3_b | [-1.6175;32.2084;21] |
| q_4_b | [-9.4960;30.8193;21] |
| q_5_b | [-27.6341;-16.0423;21] |
| q_6_b | [-21.1472;-24.0374;21] |
| q_p_1 | [25;-20;0] |
| q_p_2 | [30;-11;0] |
| q_p_3 | [4;32;0] |
| q_p_4 | [-4;32;0] |
| q_p_5 | [-30;-11;0] |
| q_p_6 | [-25;-20;0] |

(a)

| | |
|---|---|
| l_1 | 0.2884 |
| L_1 | [19.6202;13;16.6588] |
| l_2 | 0.3675 |
| L_2 | [3.5442;-33;15.7906] |
| l_3 | 0.4051 |
| L_3 | [-28.0608;21;20.3054] |
| l_4 | 0.3252 |
| L_4 | [22.0608;10;21.6946] |
| l_5 | 0.3431 |
| L_5 | [2.4558;-22;26.2094] |
| l_6 | 0.3459 |
| L_6 | [-19.6202;13;25.3412] |

(b)

Figure 5.31: (a)Joint coordinate (b) Leg values

Figure 5.31 (a) shows the joint coordinate values, no particular characteristics have to be discussed here. Figure 5.31 (b) shows the legs length.

### 5.5.5 General results

All three simulations confirmed the expected results, reporting reliable and predictable values.

Here below there is a table in which every leg (numbered from 1 to 6 as shown along the paper) is shown their stroke during every simulation (also numbered from 1 to 3 referring to 1 for "Just translation" with 2 for "Rotation about X and Translation" and 3 for "Rotation about Z and Translation"), both positive and negative.

| LEG/SIMULATION | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 3.06 cm | -0.63 cm | 0.12 cm |
| 2 | 2.43 cm | -0.39 cm | -0.14 cm |
| 3 | 2.34 cm | 8.78 cm | 3.62 cm |
| 4 | 3.04 cm | 1.16 cm | 5.33 cm |
| 5 | 3.2 cm | -0.19 cm | 7.03 cm |
| 6 | 3.06 cm | -0.63 cm | 5.87 cm |

Table 5.1: Stroke for Legs and Simulations

As the table shows in simulation number 1, the one in which just the translation has been implemented has only a positive stroke, due to the same vertical direction that every leg follows.

Regarding the second simulation instead, it can be seen that, even if along the z-axis some joints reach 13 cm and 16 cm the overall stroke of their leg has a very small negative stroke, ranging from -0.19 cm to -0.63 cm. The same consideration can be made for the last simulation

# Chapter 6

# Conclusions

This research aimed to identify an effective soft docking system for future Artemis missions. Based on previous work, my research and based on the simulation proposed it can be concluded that a Stewart platform based on six linear actuators can be a suitable structure to safely apply all the strict requests the field of space travel requires. A good hardware pick has been done and the PID controller results are optimal for the final goal.

# Appendix A

# A more complete PID explanation

Among all the possible types of controllers, the PID is the one I've chosen because of its simplicity in terms of implementation, code and practicality. Since the space applications need low velocity due to the strict precision required no other control other than the ones the PID controller provides was needed.

As the first thing to define there's the *Plant*, this is the system that we want to control, or the system whose behaviour we want to affect. The input into the plant is the actuated signal and the output is the controlled variable. Different industries refer to these signals by various names, so you might hear them called something else like plant input and plant output. The basic idea of a control system is to figure out how to generate the appropriate actuated signal, the input, so that our system will produce the desired controlled variable, the output. The output we want goes under various names, here command or command variable will be used (it can also be setpoint, the reference, or the desired value ).

The output of the system is called feedback control value, this signal is compared to the command variable to see how far off the system is from where we want it to be. This difference between the two is the error term. If the output was exactly what we commanded it to be, then the error would go to zero and that is what we want zero error. With a controller we take this error term and convert it into suitable actuator commands so that over time the error is driven to zero.

The simplest version of a PID controller is the Proportional one, also called Proportional Controller and abbreviated as "P" controller, it can be said that it keeps track only the present error value and, multiplying it by a particular value which we will call "weight" we have the actuating signal which will diminish with time by its definition do to the error becoming smaller and smaller.

In a more sophisticated case like a quadcopter we want to fly, the actuating signal made up only with a proportional controller doesn't appear good because, right when it would reach the objective the proportional weight would become zero and so the actuating signal lets the drone would fall back to earth, so we will need a constant speed just to keep it steady floating in the air, this situation can lead to errors called steady-state error. To prevent those types of errors we let the controller use past information and the way we do it is by using an integrator path in our controller that's added alongside the proportional path. An integrator sums up the input signal over time keeping a running total. Now if the system gets to a steady-state below the desired value the error term is nonzero and when a nonzero value is integrated, the output will increase and as long as there's an error in our system, the integral output will continue to change, and this increased value will increase the actuating signal, which in the quadcopter case will end up increasing the propeller speed.

Adding the integrator component can result, in some cases, in another error because the integral has been summing values for longer than necessary for some reasons. In this case to diminish the actuating signal the integral sum would need to become negative and if it happens after the reference the system could get over it.

to deal with that there's a simple way which is to add a derivative path which can "predict the future" and respond to how fast we're closing in on our goal. A derivative produces a measure of the rate of change of the error that is how fast the error is growing or shrinking Basically, our controller is using changing error to determine that we are closing in on our goal way too fast and then prematurely slowing down the actuating signal.

Just like a PID controller has been created, it is a versatile controller that uses the present error, the past error, and a prediction of the future error to calculate the appropriate actuator commands. It is able to do it using a particular "block" that uses the peculiar characteristics of a simple gain, derivatives and integrals. These three blocks each contribute some amount to the overall output of the controller. And is the designer's objective to decide how to weigh each contribution.

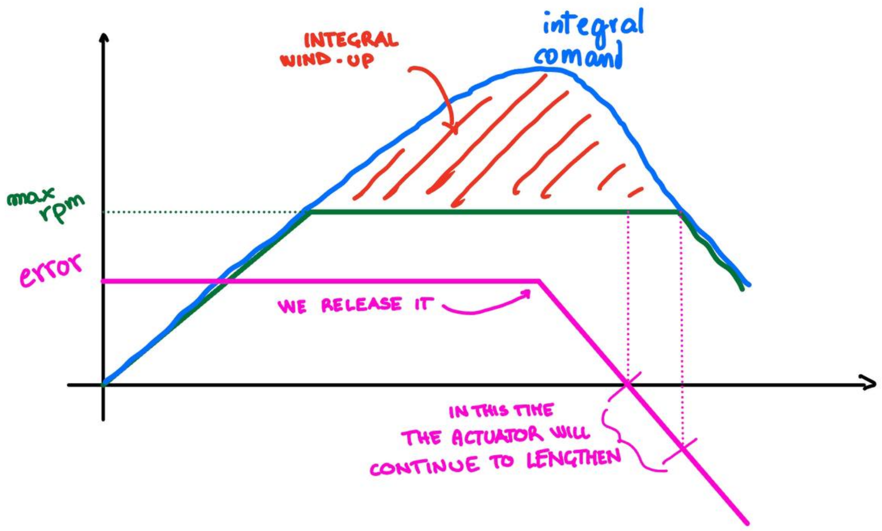


Figure A.1: graph showing integral wind-up

# Acknowledgement

These thanks want to start from those people without whom I could never have achieved all this, they go to those people who have made all this possible, they go to those people who have trusted me since day one, believing in my abilities by paying and financing those my abilities so that one day I could get to where I am now. My thanks start from those people who, faced with some of my failures, believed my words when I said "but don't worry, I know how to go on" without showing a minimum of doubt (at least they didn't let me notice it). My greatest thanks go to those people who raised me and made me ready to face a university of this kind, who raised me with values that allowed me to conscientiously face these, apparently long but actually very short, five years .My thanks go to those people who have never made me miss anything in life, these people are my parents, Diana and Dario, Dario and Diana.

However, my thanks for my family don't end there, because my brothers Chiara and Gabriele also put up with me at home. They know that I only studied during the sessions and as a result they only saw me a few times during those times and for dinner. During that period I always left the bedroom in a mess, I was never there to help around the house and it was almost as if my house were a hotel... there have obviously always been arguments, never lacking, but they have never made me difficult to pass the exams, they never had a negative impact on the progress of my exam session.

Thanks go to grandparents Ivani and grandparents Mariangeli, this thanks goes to them simply for being and having been my grandparents.

I greet and thank grandmother Mariangela and grandfather Carlo, who certainly saw me from up there and helped me in difficult moments, without them I would not be the person I am today. I thank my grandmother Ivana who, for each of my exam sessions, took a walk towards the church of San Grato in the mountains to wish me good luck.One could underestimate the importance of such small gestures but they made me feel unbeatable, I was never alone to face the exams. I greet and thank my grandfather Giò, Giuanin, because in his old-fashioned being he has always flattered me with his healthy disbelief in front of any of my goals achieved.He got excited when he saw me grow as an engineer and he got excited when I graduated for the first time, and he was excited at the idea that I was about to finish my master's too.. hello to you too grandpa.

I also want to thank my cousins and uncles, for raising me by reminding me of the importance of humility and simply making my life fuller by being there.A big greeting and thanks goes to you aunt Laura, who in your own way has always been close to me and who, before saying goodbye to us all, managed to be there until the last of my exams, you also managed to give me advice for my first contract at work, you never stopped helping.

Last but not least, I thank my aunt Sara, who has given me all her energy and soul since I was little, making herself available for anything, from the house to advice and even... reproaches.

After my family, one expects friends to be thanked, but I am particular.. in my life I have met many people and many have left. Many people have been important even if today I don't see them anymore, many people even though I don't see them anymore I still consider them important and I would give them my life.As I said, I am particular, there are so many people who have made me who I am that I would just risk forgetting someone. No one helped me with my thesis but everyone helped me grow into who I am, everyone made me the person I am. So thank you, friends, acquaintances and people who may have made even just one day of my life unforgettable.

Surely, however, I would like to thank the CCG, because I have never felt so appreciated and perhaps loved as by them. They are the first people who built that soft bed of life called "certainty of being appreciated".. it's something I had never heard until a few years ago, and it's something that fills me inside.

I would like to thank my supervisor Professor Marcello Chiaberge for giving me this splendid opportunity and the engineers Andrea Merlo and Paolo Bocchiardi for following me in Thales Alenia Space.

Finally, thanks go to myself, who have always believed in myself and have made some choices, which certainly weren't the best, but they are choices I'm proud of.

# Bibliography

[1] Marco Gaino Davide Petrillo Alessandro Cavinato. "Felds Experiment: A new flexible soft docking concept". In: *ResearchGate* (2015).

[2] *Space rendezvous*. 2022. URL: `https://en.wikipedia.org/wiki/Space_rendezvous`.

[3] *Gemini Spacecraft*. URL: `http://www.geminiguide.com/Systems/docking.html`.

[4] *H-bridge*. URL: `https://en.wikipedia.org/wiki/H-bridge#:~:text=A%20H%2Dbridge%20is%20an,to%20run%20forwards%20or%20backwards.`.

[5] International partner members of the ISS. "International Docking System Standard (IDSS) Interface Definition Document (IDD)". In: (2016).

[6] *Space rendezvous*. 2022. URL: `https://https://en.wikipedia.org/wiki/Simulink`.