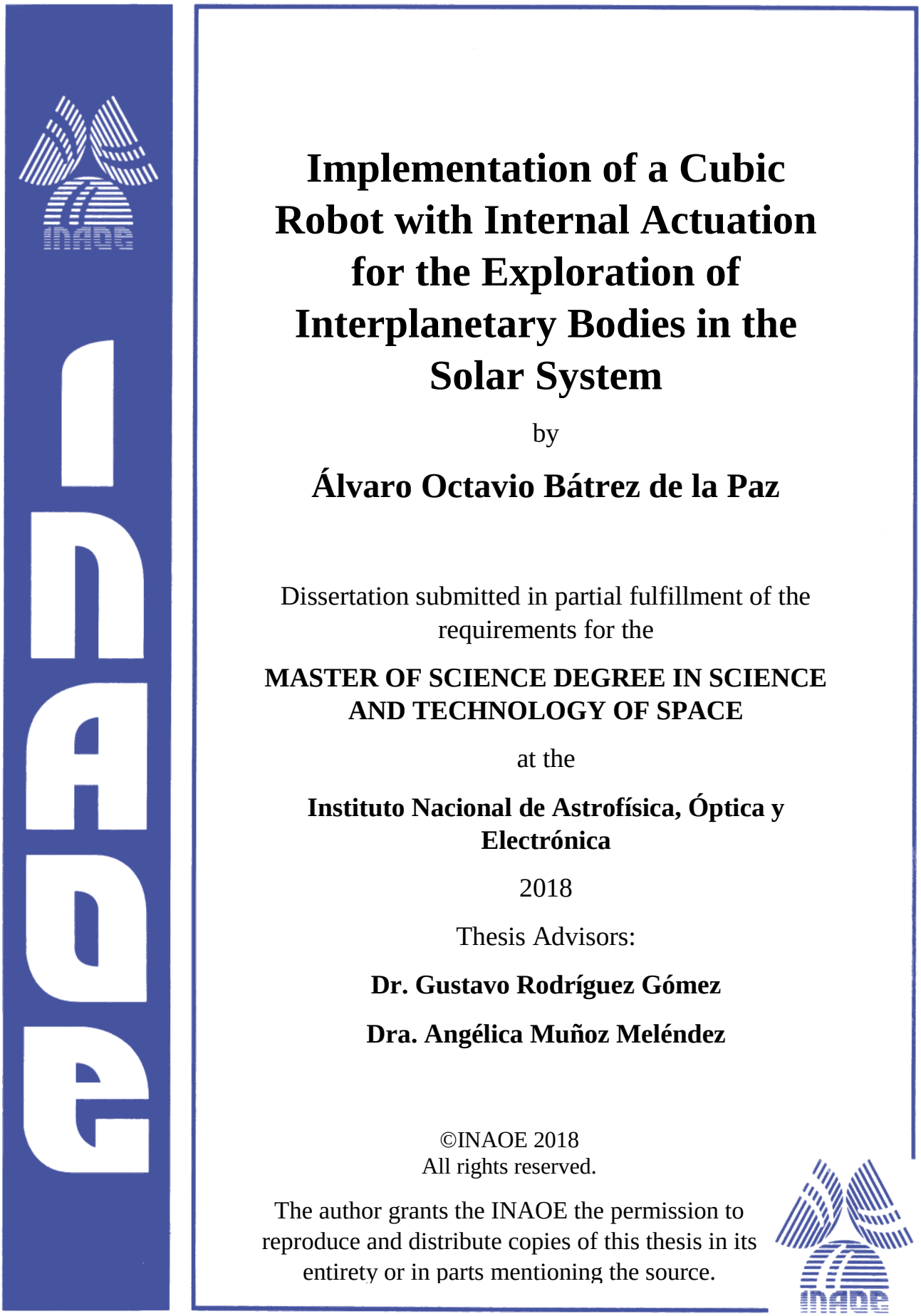


Implementation of a Cubic Robot with Internal Actuation for the Exploration of Interplanetary Bodies in the Solar System

by

Álvaro Octavio Bátrez de la Paz

Dissertation submitted in partial fulfillment of the requirements for the

MASTER OF SCIENCE DEGREE IN SCIENCE AND TECHNOLOGY OF SPACE

at the

Instituto Nacional de Astrofísica, Óptica y Electrónica

2018

Thesis Advisors:

Dr. Gustavo Rodríguez Gómez

Dra. Angélica Muñoz Meléndez

Acknowledgements

My deepest gratitude to my family for their invaluable support to achieve my dreams.

To my thesis advisors and mentors, Dr. Gustavo Rodríguez Gómez and Dra. Angélica Muñoz Meléndez. My thanks also to my thesis committee, Dr. Leopoldo Altamirano Robles, Dr. Rogerio Enríquez Caldera and Dr. Eduardo Morales Manzanares.

I would also like to thank Dr. Gustavo Mendoza Torres for advising me in the construction process of the hopper prototype.

My experience at INAOE was memorable.

Contents

List of Figures

List of Tables

Glossary

aliasing The misidentification of a signal frequency introducing distortion error.

emf The electromotive force is the electrical intensity generated by another form of energy.

flywheel A flywheel is a heavy rotational wheel in a machine that is used to increment its momentum.

gecko The gecko is a lizard who lives in warm climates throughout the world. They range from 1.6 cm to 60 cm. These animals have adhesive toe pads that let them adhere to most surfaces.

H-bridge The H bridge is an electrical circuit often used to allow DC motor to run forward and backward.

NEOs Near Earth Objects are comets and asteroids whose orbits are in the neighborhood of the Earth.

probe Similar to spacecraft.

regolith This term refers to a layer of material covering solid rock, which can come in the form of dust, soil or broken rock.

rover This name is referred to any exploration vehicle designed to move across extraterrestrial lands.

small bodies The term small bodies or low gravity bodies are all the objects in the Solar System that are not planets.

Acronyms

ADC Analog-to-Digital Converter.

CAD Computer-aided Design.

CHopper Comet Hopper.

DAC Digital-to-Analog Converter.

DLR German Aerospace Center.

ESA European Space Agency.

ISAS Institute of Space and Astronautical Science.

ISS International Space Station.

JAXA Japan Aerospace Exploration Agency.

LEMUR 3 Limbed Excursion Mechanical Utility Robot.

LTI Linear Time-Invariant.

MUSES-CN Mu Space Engineering Spacecraft.

NASA National Aeronautics and Space Administration.

PWM Pulse-width Modulation.

SLAM Simultaneous Localization and Mapping.

ZOH Zero Order Hold.

Abstract

The exploration of asteroids and comets will give us clues about the formation of the Solar System, besides it is an important step for space travel. Moving on the surface of this interplanetary objects could be challenging for future robotic explorers due to the weak gravity force. In this work, we present an approach that bases on a new kind of *jumping rovers*, which have internal flywheels in their interior. By slowly spinning up the flywheels and suddenly braking them, it is possible to perform a hop from a few meters until hundreds of them. Unlike similar researches that stop the flywheel instantaneously, we focus on stopping the flywheel by voltage inversion in a short lapse with the aim of modifying the launch angle, a quite useful action over terrains with different degrees of inclination. In this work, we discussed the dynamics of the rover for a 2D model, the speed control of the flywheel and also we conducted experiments in a prototype to validate the effectiveness of the braking and later, we analyze the stopping performance to make simulations of the ballistic trajectory under low gravity conditions.

La exploración de asteroides y cometas nos darán pistas acerca de la formación del Sistema Solar, además de ser un paso importante en la exploración espacial. Moverse sobre la superficie de estos cuerpos interplanetarios podría ser un desafío para futuros exploradores robóticos debido a la débil fuerza de gravedad. En este trabajo, presentamos un enfoque que se basa en un nuevo tipo de robots brincadores, los cuales cuentan con ruedas inerciales en su interior. Al girar lentamente estos volantes y frenarlos repentinamente, es posible realizar un salto desde unos pocos metros hasta cientos de ellos. A diferencia de investigaciones similares que detienen el volante instantáneamente, nosotros nos enfocamos en detener la rueda mediante inversión de voltaje en un lapso corto con el objetivo de modificar el ángulo de lanzamiento, una acción bastante útil en terrenos con diferentes grados de inclinación. En este trabajo, discutimos la dinámica del vehículo para un modelo 2D, el control de velocidad del volante y también realizamos experimentos en un prototipo para validar la efectividad del frenado y, posteriormente, analizamos el rendimiento del frenado para realizar simulaciones de la trayectoria balística bajo condiciones de baja gravedad.

Chapter 1

Introduction

1.1 Motivation

The study of small interplanetary bodies such as asteroids and comets, in addition to other objects like little moons, provides useful information about the origin and evolution of the Solar System [1]. Most of the bodies with low mass are analyzed with ground-based telescopes and in some cases with space telescopes. Since these bodies are covered with a layer of regolith the information about their physical and chemical properties is limited. Also, as the spectrum of these small bodies is almost unavailable, the understanding about their origin an evolution is quite limited. These aspects are pointing to the development of a new generation of space missions whose aim will be collect data *in situ, i.e.* directly from the surface in order to obtain hopefully information about their origin and nature [2].

The analysis of these objects has become relevant to the leading space agencies in the past two decades. There have been interplanetary spaceships such as Dawn, Osiris-REX, Deep Impact, Rosetta, Stardust-NExt, among others [3], which have had an impact, both, in the scientific community as and in popular culture. Similarly, other robotic missions like Curiosity or Cassini-Huygens have been considered historical missions due to the complexity of their objectives. The fascinating discoveries from each of these spacecrafts have transformed our understanding of different bodies in the Solar System. Nevertheless, without the development of visionary technologies that allowed the spacecrafts to upgrade their capabilities, these engineering feats could not have been realized.

1.2 Problem Statement

Although the ability and perseverance of scientists and engineers have shown space exploration as a routine; actually, it represents one of the most challenging ventures in humankind history. Any spaceship, no matter its destination, must fulfill the challenges of traveling very long distances, work correctly during its mission and operate without any control in real time with sporadic transmissions to the Earth. In order to make a wide variety of tasks and simultaneously fulfill the concerning issues of the inhospitable environment in which they are, these space vehicles/robots must be equipped with technological advances in the next categories [1]:

- Minimize the mass and energy demand of the spacecraft and its subsystems.
- Improve their capability for communications.
- Increase their autonomy.
- Have a significant efficiency in the propulsion systems in all the phases of the mission.
- Improve the security that contributes to keep the spacecraft safe in extreme environments.

One of the principal priorities for the National Aeronautics and Space Administration (NASA) and the European Space Agency (ESA) is the study of NEOs. NASA was developing the first robotic spacecraft capable of visiting an asteroid and pick up a stone from the surface, to bring it into a stable orbit around the Moon [4]. This mission was canceled in December 2017. ESA is currently working on the Asteroid Impact Mission [5], that plans to hit a binary asteroid to observe the repercussions of such impact, as part of the European planetary defense program.

The proposal of the National Research Council of the United States about the technological development priorities of NASA published in 2015 [6], has divided these priorities into 15 fields. One of this fields is *Robotics and Autonomous Systems*. Within this field, the *Small-Body and Microgravity Mobility* topics are identified as of high priority.

The principal goal of a rover within a low gravity environment is to provide certain motion autonomy. A rover usually needs actuators and effectors for moving on over almost the total surface of the explored object to have access to places previously determined by scientists and designers. Among scientific

and technical considerations, complex characteristics of the object to be explored such as the gravity force fluctuations, the land irregularities must be taken into account too. The temperature of the environment and the available energy sources. Also, it is important to consider the specific requirements of the mission such as operation time, on-board instrumentation and the equipment needed to collect, keep and protect samples.

The milli-micro gravity force of the small bodies is a constraint for the traditional wheeled rovers. For these reasons, it is convenient to analyze alternative motion strategies that take advantage of low gravity.

There are some difficulties for future low-gravity rovers:

- A robot needs to include some mechanism specifically designed to work in low gravity environments.
- A robot needs an effective locomotion mechanism to be able to traverse uneven surfaces.
- Due to the limitations of real experimentation on our Planet, accurate simulations of the complete functionality are necessary.

Castillo-Rogez et al. [2] have divided into four classes the solutions for mobile platforms for surface exploration:

1. *Thruster Motion.* Mobility by rocket thrusters would land few times to collect soil samples and data using on-board instruments. A drawback of this alternative is the limited number of locations that the spacecraft could visit [2], [7]. However, is faster than other forms of locomotion.

2. *Wheeled Motion.* Wheeled vehicles have been very successful in Martian and Lunar explorations, however, the lack of traction causes that the maximum speed that can be reached is about 1.5 mm/s [8]. Bouncing and dust are other inconveniences for wheeled rovers.

3. *Legged Motion.* Locomotion based on legs is mechanically complicated because it requires an anchoring system and also is slower than the other mentioned solutions. However, a legged robot could reach some unattainable places for mobile robots that do not rely on legs, such as cliffs [9], [10].

4. *Hopping Motion.* This kind of locomotion is based on a robotic platform able to jump. The *hopper* has a minimalistic design with little weight, small dimensions, and low cost. This robot relies on inertial wheels which

are held within the robot. The movement is produced by slowly accelerating these wheels and suddenly stopping them, generating in that way a braking torque and the consequent transfer of energy from the wheels to the chassis that causes the robot jumps [11], [12], [13]. There are two basic categories of hopping locomotion:

(a) *Sticking Mechanism.* The robot has a push system that sticks the surface of the explored object to generate the impulse.

(b) *Moving Inertial Masses.* This system consists of one or more internal masses that spin at a certain velocity, the abrupt change in this velocity generates a momentum that makes the robot jump, turn or flip.

1.3 Proposed Solution

The present investigation focuses on locomotion by jumping using spinning inertial masses. It has the advantage that using a simple mechanism (the actuator is a flywheel attached to a simple DC motor), hopping robots are capable of doing significant traverses, and also, as the actuation is encapsulated internally, there is no risk of stuck by dust [2], [12]. Hockman et al. [14] have established that the best morphology of a *hopper* is a cubic shape since it always offers a launch angle of 45° over flat surfaces.

To perform a jump maneuver a reaction wheel is slowly accelerated and suddenly stopped to generate the momentum that will provide a ballistic trajectory. A hop maneuver is divided into two phases:

1. *Leverage phase.* The time elapsed between the moment when the robot is in a rest position and the time when it entirely has no contact with the ground.

2. *"Fly" phase.* The time elapsed between the moment when the robot leaves the ground and the moment when it falls again.

In recent related works, *hopper* robots have an instantaneous mechanical brake of the inertial wheel that is applied by both, a rubber band [14] and an impact hammer [15]. This approach might be harmful to the robot. In contrast, as the surface of an asteroid or comet is rarely flat, is a motivation for this research to deflect the launch angle depending on the slope of the ground to maximize the hop distance. The application of a controlled-time-braking of the DC motor by voltage inversion over surfaces with different degrees of inclination is proposed.

1.4 Hypothesis

Through the implementation of a control system for decelerating an internal flywheel of a jumping rover, it will be possible to explore the surface of small bodies of the Solar System.

1.5 Objectives and Contribution

The **main goal** of this research is to develop a speed control system for a flywheel of a cubic robot that allows it to jump over the surface of interplanetary bodies with low gravity.

To achieve the main objective, a set of **specific objectives** have been defined as follows:

- To analyze the dynamic properties of the *hopper* robot.
- To implement a control technique that allows stopping the flywheel in a specific time applying controlled braking by voltage inversion.
- To adjust the jump angle to reach the desired distance over sloped surfaces.
- To design a real test system to validate the effectiveness of braking the flywheel by voltage inversion.
- To perform computational simulations of the robot and its trajectory.

In recent related works, the *hopper* robot has an instantaneous mechanical brake that is applied by both, a rubber band [14] and an impact hammer [15]. In contrast, in this research the **main contribution** is the application of controlled braking of the wheels by voltage inversion over surfaces with different degrees of inclination, with the aim of changing the launch angle for always getting the desired displacement regardless the slope of the ground in which the robot is located. This choice aims at protecting eventual damages that might happen to the mechanical components of the robot for a sudden uncontrolled brake. As a first approximation, it has been only considered a 2D ballistic flight and one single flywheel and also has been assumed that the center of mass of the *hopper* is situated at its geometric center.

1.6 Methodology

This section briefly presents the methodology to be used in order to reach the previously defined objective.

1. **Hopping Dynamics**

   The first step is to develop a mathematical model of the relationships among the flywheel, the chassis of the *hopper*, the inclination of the land and the jump trajectory. The appropriate model will be obtained based on the related bibliography.

2. **Adjust the Trajectory for Uneven Terrains**

   The optimal trajectory is at a launch angle of 45° no matter the inclination of the surface. It is possible to change the jumping angle and consequently, the travel distance switching between instant-braking of the flywheel to time-controlled-braking. Through the slope of the surface and the desired distance, it is possible to determinate the angular speed and the lapse in which the inertial mass must stop to always approximate to a 45° launch angle.

3. **Design of the Speed Controller**

   The next step is to design a control system that allows braking the flywheel in the lapse specified above. Good design prevents oscillations, improve both time response and the stability, as well as minimize the steady state error. In this work, it has been applied the state feedback by pole placement technique.

4. **Validation** The last step is to develop a computer simulation to visualize and validate the response of the implemented control technique and the trajectory of the *hopper*; also, it is necessary to build a prototype to perform the necessary experiments to verify the effectiveness of the speed control of the flywheel.

It is almost impossible to hit the target place in the first jump. For this reason, it is necessary to establish a target area instead of an accurate location, and then, carry out a series of jumps that allow the *hopper* to reach the target area.

1.7 Constraints

The slips and bounces of the robot are governed by the friction with the regolith and rocks on the surface of the comets or asteroids. These interactions are beyond the scope of current research. Due to the difficulty of experimentation of

milli-microgravity, the way to validate the results will be focusing on computer simulations.

1.8 Organization of this Document

The rest of this document is organized as follows. The second chapter revises the related work to this research. The third chapter presents all the theoretical foundations required to meet the objectives. The fourth chapter explains in detail the methodology applied to achieve the goals. The fifth chapter focuses on describing the experiments and simulations conducted in this research. The sixth chapter of this document summarizes the results and the feasibility of the proposed approach. In the last chapter are presented the concluding remarks of this research.

Chapter 2

Related Work

So far it does not exist any robot with an efficient locomotion system that enables it to explore an asteroid or comet on which it can be brought. In this chapter it is shown the relevant work to the four classes of mobility platforms mentioned in Chapter 1.

2.1 Rocket Thruster Motion

Figure 2.1: Artist concept of CHopper.

The CHopper mission was a lander proposed by NASA [7], [16]. If it had been built, it would have orbited and landed multiple times on the comet 46p/Wirtanen. This spacecraft had as a primary goal to map the spatial heterogeneity of the surface solids and analyze dust and gas emissions. When the comet had approached to the Sun, the space probe would take off and land several times

to record surface changes as the comet became more active. This mission was canceled in 2012. Figure 2.1 shows an artist concept of CHopper [16].

2.2 Wheeled Motion

Wheeled robots have been very successful in the exploration of the Moon and Mars. Not only NASA is in the way to build rovers, other space agencies and companies have already sent or are developing their space vehicles to send them to our natural satellite or Mars in the next years, for example, ESA's Exomars [17] and ©Google Lunar XPRIZE [18]. Nevertheless, for low gravity objects, the physical conditions are entirely different, and wheeled rovers might not be very efficient. As the energy transmission between the wheel of the robot and the ground depends on friction, the loose and soft soil of the comets or asteroids is not appropriate to transform wheel rotation into forwarding motion without losses. Every disturbance on the road like a rock or gap can lead to a flip of the wheels [12].

To solve these problems NASA and the Institute of Space and Astronautical Science (ISAS) of Japan created a prototype of a rover called MUSES-CN [8] with a mass of 1300 g. The rover consisted of a rectangular body with 14×14×6 cm in dimension with four wheels with a top speed of 1.5 mm/s. The robot had six optical detectors on its external faces to determinate the direction of the Sun. It had a laser range finder, which enables it to determine the range of nearby objects. The rover carried three science instruments: a visual camera, a near infrared spectrometer, and an alpha X-ray spectrometer. The entire robot had being qualified for a temperature range of -180 °C to 110 °C. The MUSES-CN was also designed to be self-righting and was able to operate upside down. The project was canceled by NASA in November 2000 due to budget constraints [19]. Figure 2.2 shows the MUSES-CN nanorover prototype [8].

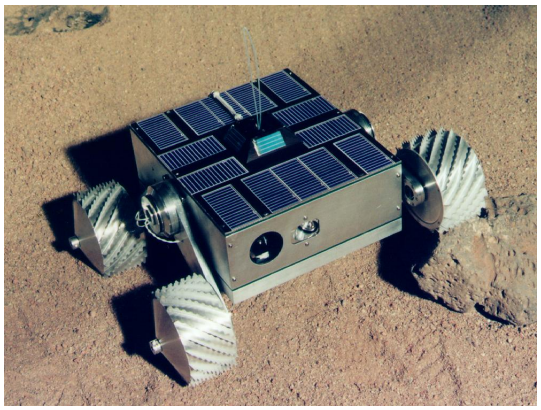


Figure 2.2: MUSES-CN nanorover prototype.

2.3 Legged Motion

Helmick et al. [9] have presented a limbed robot with stable, robust and precise mobility and the capacity of collect samples from the surface of small bodies. Due to the lack of traction on comets and asteroids, the advantages that a robot with extremities and anchorages presents over irregular terrain are:

- The capacity of moving over rough surfaces.
- The ability to strategically choose contact points.
- The multifunctional capabilities of limbs beyond locomotion.

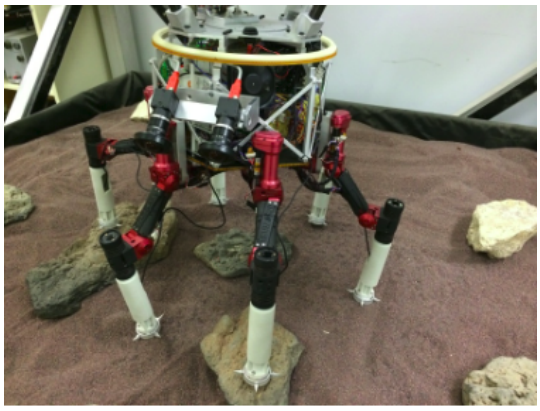


Figure 2.3: The prototype of the limbed robot.

Figure 2.3 [9] shows a robot with limbs designed to minimize unexpected forces that could lose contact with the terrain, change its trajectory, or exceed the escape velocity of small bodies. An important aspect to consider is the power requirement. This walking robot has twelve actuators and can be energized with solar power or batteries.

Figure 2.4 shows a more complex robot than the previous one [10]. The LEMUR 3 was built from two projects, one focused on crawling across the outdoor of the International Space Station (ISS) using gecko-adhesive [20], and the second one focused on climbing vertical cliffs and traversing cave ceilings using microspine grippers [21]. Its design was motivated by a desire of climbing extreme terrains with seven links per extremity. This prototype only needs switching its end effector to climb across icy, sandy or rocky terrains.

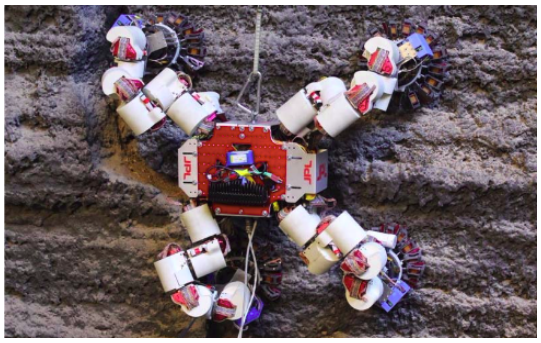


Figure 2.4: The LEMUR 3 robot.

Figure 2.4 presents the LEMUR 3 climbing a rock [10].

2.4 Hopping Motion

In this section, significant research about hopping robots is revised.

Figure 2.5 shows a robot called Cubli [22], designed mainly for terrestrial applications. This robot is based on the principle of an inverted pendulum in a three-dimensional environment. It presents a cubic form design of 15 cm per side and has three internal spinning wheels that provide the necessary torque and momentum to carry out the appropriate movements that enable the robot to reach a balancing position. Cubli uses a nonlinear control approach that makes possible rising from a rest position and balance.

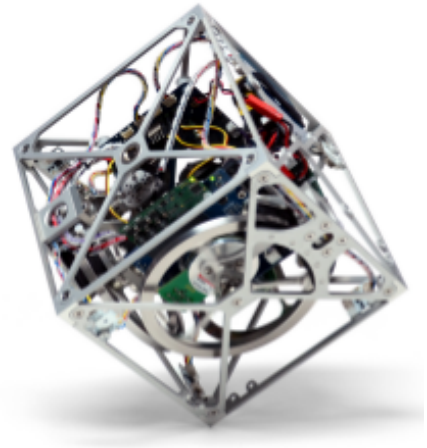


Figure 2.5: The Cubli prototype balancing on one of its corners.

Another related space rover has been presented by ESA and the German Aerospace Center (DLR), with the name of MASCOT [11], [12]. On June 26th,

2018, this lander met the asteroid Ryugu, on board the spaceship HAYABUSA-2 led by the Japan Aerospace Exploration Agency (JAXA) [23]. It uses a locomotion strategy that is similar to the proposed one in this research; however, MASCOT only needs one single mass, that is outside its gravity center, which gives it the possibility of jumping to another location on the surface of an asteroid. MASCOT is unable to focus on a particular place of interest since it has been designed to collect information from random locations.

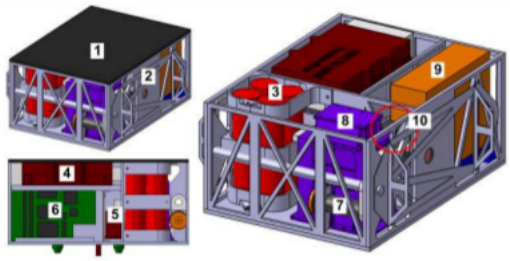


Figure 2.6: MASCOT CAD model.

Figure 2.6 shows the subsystems on board [11].

Kato et al. [24] have presented a small rocket-propelled robot, Figure 2.7, which also has an inertial load within its body, Figure 2.7. This strategy uses a small main rocket engine to lift off from the ground and a second engine to stabilize the traverse trajectory. When the robot is flying, a reaction wheel starts to rotate for adjusting the thrusting direction of both drivers. This strategy needs one shot, and it can be very challenging to control under low gravity.

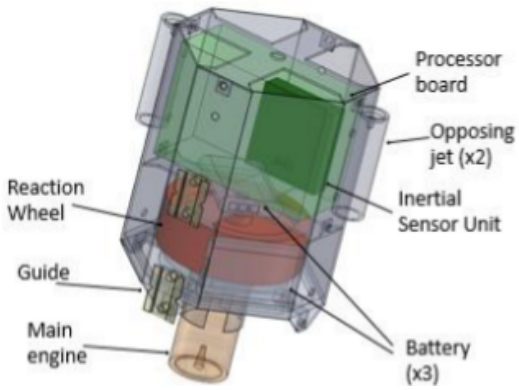


Figure 2.7: CAD model showing the internal parts of the robot.

The research of Hockman et al. [14] is based on the work of Muehlebach and D'Andrea [22] and is adapted to a situation of low gravity. The validation of

the experiments and the response of the algorithms is conducted using a crane type system. This work leaves several open problems, such as the development of realistic contact models of sandy terrain (regolith) found in asteroids; the improvement of the reliability of the maneuvers on rocky terrains, and the development of Simultaneous Localization and Mapping (SLAM) techniques to deal with the complexity of the environment, among others. The work of Hockman et al. is the primary reference of this document. Figure 2.8 [14] shows the prototype and a CAD model of its interior.

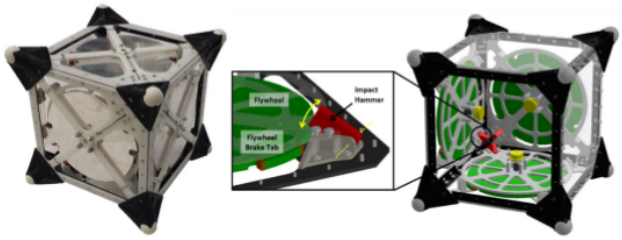


Figure 2.8: Prototype and CAD model showing some internal components of the rover.

As we said before, our research focuses in this last form of locomotion. It is important to recall that the main difference between the related works in the *Hopping Mobility* field and the presented in this document is that, we change the manner to stop the flywheel using controlled braking for avoiding some eventual damage to the structure of the robot.

The next chapter presents the mathematical basis and the theoretical foundations of this work.

Chapter 3

Foundations

In this chapter are presented the necessary theoretical basis of classical mechanics, dynamic systems and modern control theory for a better understanding of this research.

3.1 Mathematical Modeling of Dynamic Systems

In control theory, a mathematical model of a system is the set of differential equations which describes the behavior of the system [25]. Based on the model it is possible to calculate the response of the system variables for any instant of time. Unfortunately, it is almost impossible to make an exact model of a physical system. However, even if a model represents partially the reality, it can be useful if it describes the dominating dynamic properties of the system.

3.1.1 The Inverted Pendulum

In the mathematical representation of the hopper robot, it is necessary to know the model of an inverted pendulum because this model describes the first phase of the jump (when the robot is at rest position until it loses ground contact).

The primary objective of this model is to maintain a bar in a vertical position. For the next analysis, it is considered a problem in two dimensions, in which the pendulum only moves in the horizontal and vertical planes. Assuming that the bar's gravity center is at its geometrical center and it is supplied with a control force $u(t)$ to the cart. A free-body diagram of the inverted pendulum system [26] is shown in Figure 3.1.

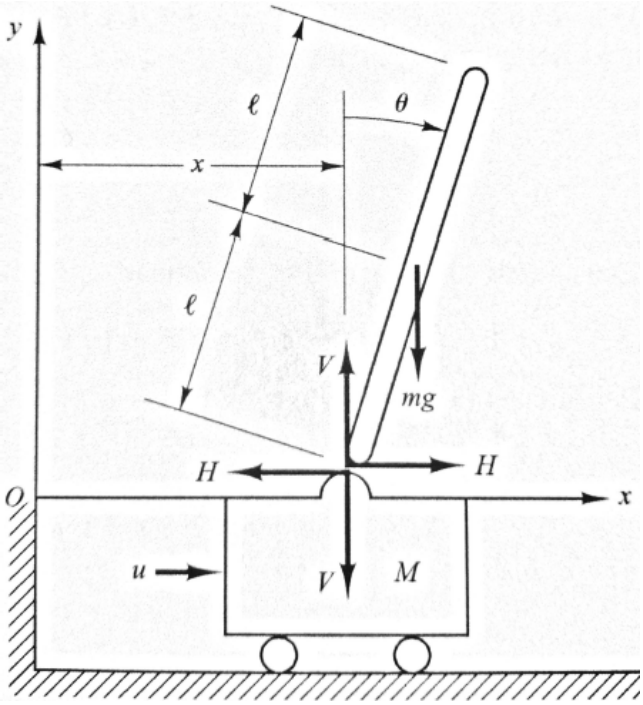


Figure 3.1: The inverted pendulum system.

Symbol	Value	Units
I	Moment of inertia of the bar	$kg \cdot m^2$
θ	Angular position of the bar	rad
$\dot{\theta}$	Angular speed of the bar	rad/s
$\ddot{\theta}$	Angular acceleration of the bar	rad/s^2
l	Half the length of the bar	m
V	Vertical reaction force	N
H	Horizontal reaction force	N
x	Position of the cart	m
$\dot{x}$	Speed of the cart	m/s
$\ddot{x}$	Acceleration of the cart	m/s^2
M	Mass of the cart	kg
m	Mass of the bar	kg
g	Gravity acceleration	m/s^2
b	Friction coefficient	$-$
u	Force applied to the cart	N

Table 3.1: Parameters of the inverted pendulum Figure 3.1.

Be θ the angular position of the bar concerning to the vertical and transforming the gravity center coordinates of the bar (x, y) as Expressions (3.1)

and (3.2)

$$x_{gc} = x + l\sin\theta, \tag{3.1}$$

$$y_{gc} = l\cos\theta. \tag{3.2}$$

The rotational motion of the bar in its gravity center is expressed by Formula (3.3)

$$I\ddot{\theta} = Vl\sin\theta - Hl\cos\theta, \tag{3.3}$$

where I is the inertial moment of the bar.

From the mass of the bar m times the acceleration of x it is possible to obtain the horizontal motion of the gravity center of the bar given by (3.4)

$$m\frac{d^2}{dt^2}\left(x + l\sin\theta\right) = H. \tag{3.4}$$

The vertical movement is described by the mass of the bar times the acceleration in y, minus the gravity force, as in Equation (3.5)

$$m\frac{d^2}{dt^2}\left(l\cos\theta\right) = V - mg. \tag{3.5}$$

The horizontal motion of the cart is denoted by (3.6)

$$M\ddot{x} + b\dot{x} = u - H, \tag{3.6}$$

where M is the mass of the cart and u is the supplied control force. Forces of the cart in the vertical direction are not useful information.

Since the pendulum must remain in vertical position, it is feasible to suppose that θ and $\dot{\theta}$ will be small, so, $\sin\theta \approx \theta$, $\cos\theta \approx 1$; then, the Equations (3.3), (3.4) and (3.5) are respectively linearized in the form of Equations (3.7), (3.8) and (3.9)

$$I\ddot{\theta} = Vl\theta - H, \tag{3.7}$$

$$m\left(\ddot{x} + l\ddot{\theta}\right) = H, \tag{3.8}$$

$$0 = V - mg. \tag{3.9}$$

The Expression (3.10) indicates the horizontal motion of the bar, and can be derived from Equations (3.6) and (3.8)

$$\left(M + m\right)\ddot{x} + b\dot{x} + ml\ddot{\theta} = u. \tag{3.10}$$

The angular motion of the bar, Formula (3.11), can be obtained from Equations (3.7), (3.8) and (3.9)

$$I\ddot{\theta} = mgl\theta - Hl = mgl\theta - l\left(m\ddot{x} + ml\ddot{\theta}\right), \tag{3.11}$$

alternatively expressed by (3.12)

$$\left(I + ml^2\right)\ddot{\theta} + ml\ddot{x} = mgl\theta. \tag{3.12}$$

Consult [26] and [27] for a detailed description of this model.

3.1.2 The DC Motor

As it was mentioned in Section 1.3, the actuator of the *hopper* is a simple DC motor. The motion of the robot depends on the braking force of the motor and the mass of the load J. Develop the appropriate mathematical model of the engine is essential for meeting the desired performance. The electromechanical circuit of a direct-current motor with permanent magnets [28] is referenced in Figure 3.2.

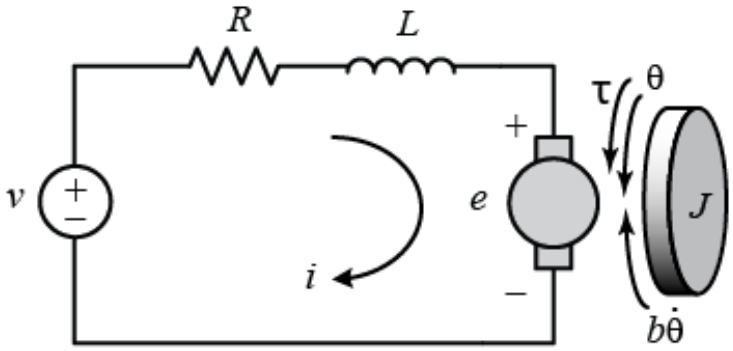


Figure 3.2: The electromechanical circuit of a direct-current motor.

Symbol	Description	Units
v	Input voltage	V
R	Motor winding resistance	Ω
L	Motor winding inductance	H
$\imath$	Armature current	A
J	Moment of inertia of the rotor	$kg \cdot m^2$
ε	Electromotive force of the motor	V
τ	Torque	$N \cdot m$
$\dot{\theta}$	Angular speed of the rotor	rad/s
$\ddot{\theta}$	Angular acceleration of the rotor	rad/s^2
b	Friction coefficient of the rotor	$N \cdot m \cdot s$
K_m	Mechanical constant	$N \cdot m/A$
K_b	Electrical constant	$\frac{V}{rad/s}$

Table 3.2: Parameters of the DC motor.

In Formula (3.13) the torque in the rotor is proportional to the current and the magnetic field force

$$\tau = K_m \imath, \tag{3.13}$$

where K_m is the mechanical constant and is related to the magnetic field force of the motor.

The friction torque T is equal to the friction coefficient of the rotor multiplied by the angular speed, as expressed in (3.14)

$$T = b\dot{\theta}. \tag{3.14}$$

The voltage induced by the motor, also called emf, is a tension proportional to the angular speed of the rotor and the electrical constant K_b, which is also a physical parameter of the motor, this is given by Formula (3.15)

$$\varepsilon = K_b \dot{\theta}. \tag{3.15}$$

Applying the laws of Kirchhoff, it is obtained that, the sum of the voltage drops is equal to the voltage supplied, as shown in Formula (3.16)

$$v = R\imath + L\frac{d\imath}{dt} + \varepsilon. \tag{3.16}$$

Applying Newton's second law, in Expression (3.17) the inertial load of the rotor times the derivative of the angular speed is equal to the sum of all the torques in the rotor

$$J\ddot{\theta} = -T + \tau. \tag{3.17}$$

The mechanical and electrical constants are equivalents $K_m = K_b$, henceforth denoted only by K.

This leads to two differential equations that establish the relationship between the electrical and mechanical behavior of the motor. By substituting the value of $\varepsilon(t)$ in Equation (3.16) it is obtained the description, given by (3.18), of the current through the circuit

$$v = R\imath + L\frac{d\imath}{dt} + K\dot{\theta}, \tag{3.18}$$

moreover, by substituting Equation (3.13) and (3.14) in (3.17), the equation that describes the angular movement of the rotor can be derived as in (3.19)

$$J\ddot{\theta} = -b\dot{\theta} + K\imath \tag{3.19}$$

Consult [25] and [28] for a detailed analysis of this model.

3.2 Modern Control Theory

Control theory has played an essential role in the advancement of science and technology. Practically all new complex technological systems require some or several types of feedback control. Control engineering is a vital part of, among other examples, space vehicles, robots, modern industrial fabrication processes, airplanes, and cars.

At this point, it is important to remark that some basic definitions such as *plant, actuator, sensor, controller, open-loop, closed-loop, reference, control variable, output variable, error, perturbation.* These terms are all taken from [29].

3.2.1 Preliminaries

Laplace Transform

Laplace transform is the representation of a linear ordinary differential equation in an algebraic form [30].

The Laplace transform of a function $f(t)$ is defined as indicated in Formula (3.20)

$$F(s) = \mathcal{L}\left[f(t)\right] = \int_0^\infty f(t)e^{-st}dt, \tag{3.20}$$

where s is the transform variable, generally has complex values and is expressed by $s = \sigma + j\omega$.

The inverse Laplace transform is expressed in (3.21)

$$f(t) = \mathcal{L}^{-1}\left[F(s)\right] = \frac{1}{2\pi j}\int_{\gamma-j\infty}^{\gamma+j\infty} F(s)e^{st}ds, \tag{3.21}$$

where γ is chosen such that the infinite integral converge.

Some useful transform pairs in control problems are:

1. *Constant* (3.22)

$$f(t) = a \Longleftrightarrow F(s) = \frac{a}{s}. \tag{3.22}$$

2. *Exponential function* (3.23)

$$f(t) = e^{-at} \Longleftrightarrow F(s) = \frac{1}{s+a}. \tag{3.23}$$

3. *First-order derivative* (3.24)

$$\mathcal{L}[f'(t)] \Longleftrightarrow sF(s) - f(0). \tag{3.24}$$

4. *Second-order derivative* (3.25)

$$\mathcal{L}[f''(t)] \Longleftrightarrow s^2F(s) - sf(0) - f'(0). \tag{3.25}$$

For a complete list of Laplace transforms see [31] and Appendix A of [26].

Initial and Final Value Theorems These theorems are used to find the values at the extremes $t = 0$ (3.26) and $t = \infty$ (3.27) of a time-domain function without employing the inverse Laplace transform, assuming that initial conditions are 0 [30]:

- *Initial value theorem*

$$\lim_{s\to\infty}[sF(s)] = \lim_{t\to 0} f(t) \tag{3.26}$$

- *Final value theorem*

$$\lim_{s\to 0}[sF(s)] = \lim_{t\to\infty} f(t) \tag{3.27}$$

Transfer Function

The concept of transfer function $G(s)$ is the Laplace transform relation between the output variable $Y(s)$ of a process and its input variable $U(s)$ [26]. Considering a Linear Time-Invariant (LTI) system described as indicated in Formula (3.28)

$$a_0y^{(n)}+a_1y^{(n-1)}+\cdots+a_{n-1}\dot{y}+a_ny = b_0y^{(m)}+b_1y^{(m-1)}+\cdots+b_{m-1}\dot{y}+b_my. \tag{3.28}$$

The Laplace transform of this equation is given in Formula (3.29)

$$a_0s^nY(s) + \cdots + a_{n-1}sY(s) + a_nY(s) = b_0s^mU(s) + \cdots + b_{m-1}sY(s) + b_mY(s). \tag{3.29}$$

A common representation of the transfer function is expressed as in Formula (3.30)

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0s^m + b_1s^{m-1} + \cdots + b_m}{a_0s^n + a_1s^{n-1} + \cdots + a_n}. \tag{3.30}$$

A transfer function has the next properties [30]:

- The denominator of $G(s)$ is the characteristic polynomial of the differential equation.

- The roots $U(s)$ of the characteristic equation are the poles of $G(s)$. Denoted by $U(s) = 0 : p_1, p_2, \cdots, p_n$.
- The term *roots of the characteristic equation* is the same that *poles of the transfer function*.
- The poles reveal the behavior of the model differential equations qualitatively.
- The roots of the polynomial $Y(s)$ are zeros. Denoted by $Y(s) = 0 : z_1, z_2, \cdots, z_m$.

Second-Order Systems

The order of a dynamic system is the highest derivative of its governing differential equation [32] or the highest power of s in the denominator of its transfer function. This section only reviews the second-order system properties.

A second-order differential equation is given by (3.31) [25]

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = f(t). \tag{3.31}$$

The above expression can be written as in Equation (3.32)

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = K\omega_n^2 u(t). \tag{3.32}$$

Its transfer function is indicated in (3.33)

$$G(s) = \frac{1}{ms^2 + bs + k} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{3.33}$$

where ζ is the damping ratio, and ω_n is the natural frequency.

Gain The *gain* is the ratio of the magnitude of the steady state step response to the magnitude of the step input, Equation (3.34)

$$K = \frac{1}{k}. \tag{3.34}$$

Damping Ratio The *damping ratio* ζ is the rate at which an oscillation decay. From (3.33) is defined by (3.35)

$$\zeta = \frac{b}{2\sqrt{km}}. \tag{3.35}$$

Natural Frequency The *natural frequency* ω_n is the frequency in which the system will oscillate when $\zeta = 0$, Equation (3.36). This expression also can be obtained from (3.33)

$$\omega_n = \sqrt{\frac{k}{m}}. \tag{3.36}$$

Poles and Zeros A second-order transfer function has two poles, p_1 and p_2 (3.37). These are the roots the denominator of (3.33)

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}. \tag{3.37}$$

The value of ζ determines if the poles are real or complex conjugate, also, determines the step response of the system.

Classification of Second-Order Systems The second-order systems are classified according to [30] as *overdamped, critically damped, underdamped, undamped* and *unstable*. Figure 3.4 shows this classification with respect to the value of ζ.

Overdamped Systems $\zeta > 1$. The step response has not overshoot. The poles are real and distinct.

Critically Damped Systems $\zeta = 1$. The step response has no overshoot. The poles are real and have the same location.

Underdamped Systems $0 < \zeta < 1$. In this case, the step response has overshoot. The poles are complex conjugate.

The *overshoot* (OS) is the value by which the response exceeds its steady-state value and is given in percentage, as indicated by (3.38)

$$OS = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}. \tag{3.38}$$

From Formula (3.38), ζ is given by (3.39)

$$\zeta = \frac{-\log(OS)}{\sqrt{\pi^2 + \log^2(OS)}}, \tag{3.39}$$

this result is in agreement with Equation (3.35).

The *settling time* (T_s) is the time required to reach a certain percentage of the steady-state value [32]. Common tolerances are

10%	5%	2%	1%
$T_s = \frac{2.3}{\zeta\omega_n}$	$T_s = \frac{3}{\zeta\omega_n}$	$T_s = \frac{3.9}{\zeta\omega_n}$	$T_s = \frac{4.6}{\zeta\omega_n}$

Table 3.3: Common tolerances for settling time.

According to Table 3.3, the *natural frequency* ω_n is given by (3.40)

$$\omega_n = \frac{tolerance}{\zeta T_s}. \tag{3.40}$$

The *rise time* (T_r) is the time required to rise from 10% to 90% of its steady-state value.

Undamped Systems $\zeta = 0$. The step response permanently oscillates at frequency ω_n [25]. The poles are imaginary.

Unstable Systems $\zeta > 1$. The step response always increases over time. The poles are real.

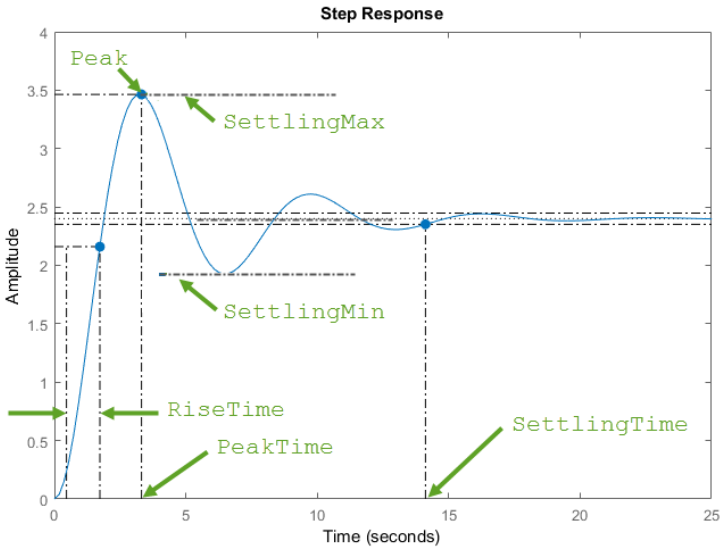


Figure 3.3: Some characteristics of a second-order response.

In Figure 3.3 some characteristics as *peak response*, *rise time*, and *settling time*, among others, of a second-order response are represented in green letters [33].

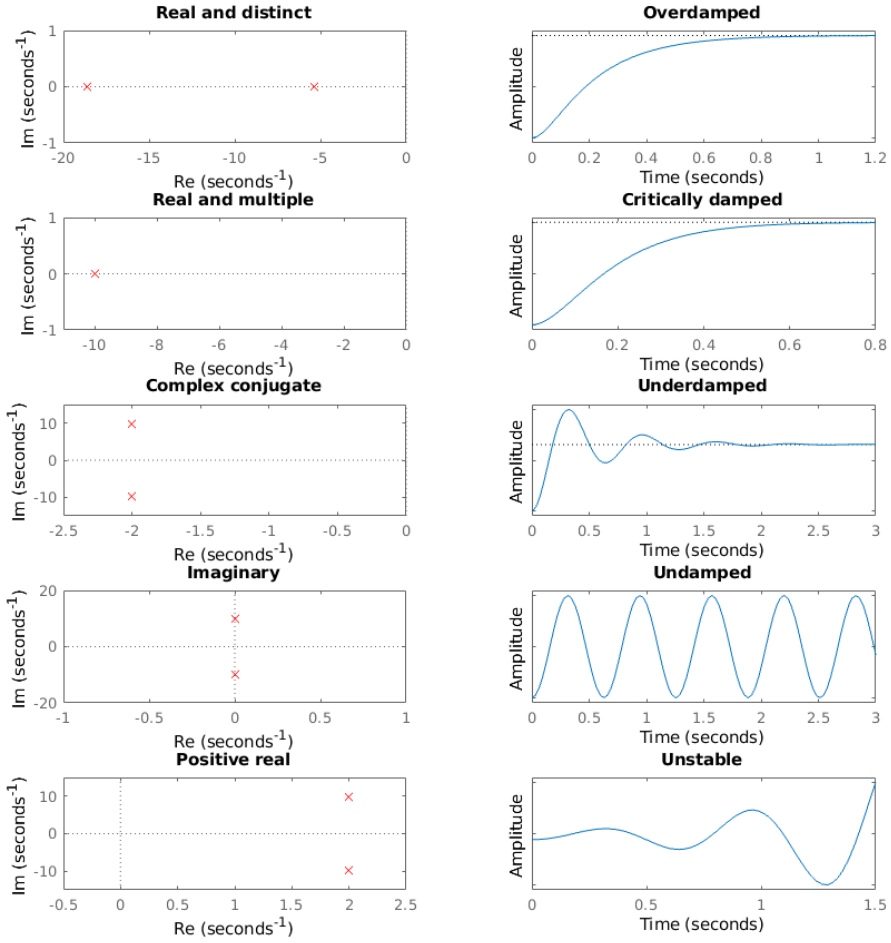


Figure 3.4: Classification of second-order systems by the value of ζ.

3.2.2 State-Space Model

For LTI systems, the state-space equations [30] are a set of differential equations represented by a matrix form. This representation is expressed by (3.41) and (3.42)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \tag{3.41}$$

$$y(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t), \tag{3.42}$$

where $\mathbf{x}(t)$ is the state-variable vector, $u(t)$ is the input and $y(t)$ the output. $\mathbf{A}$ is the process (plant) matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output matrix and $\mathbf{D}$ is the feedback matrix (almost always 0).

Relation of State-Space Models with Transfer Function Models

The Laplace transform of Equations (3.41) and (3.42) are respectively given in Equations (3.43) and (3.44)

$$s\mathbf{X}(s) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}U(s), \tag{3.43}$$
$$Y(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}U(s), \tag{3.44}$$

also, from Equation (3.43) it can be obtained (3.45)

$$\mathbf{B}U(s) = (s\mathbf{I} - \mathbf{A})\mathbf{X}, \tag{3.45}$$

where $\mathbf{I}$ is the identity matrix.

From Equations (3.43) – (3.45) the transfer function can be derived as in (3.46) [26]

$$G(s) = \frac{Y(s)}{U(s)} = \mathbf{C}\left(s\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D}. \tag{3.46}$$

The limitation of transfer function representation becomes helpless for complex systems with multiple inputs and outputs; transfer function matrices can become very complex.

Canonical Forms Considering the system defined by Equation (3.30):

Controllable The matrix representation used in Equations (3.47) and (3.48) is known as *controllable canonical form* [26]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u, \tag{3.47}$$

$$y = \begin{bmatrix} b_n - a_n b_0 & b_{n-1} - a_{n-1}b_0 & \cdots & b_2 - a_2 b_0 & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + b_0 u. \tag{3.48}$$

Observable The next matrix representation, (3.49) and (3.50), is known as *observable canonical form*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & 0 & -a_{n-1} \\ 0 & 1 & \cdots & 0 & -a_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \vdots \\ b_2 - a_2 b_0 \\ b_1 - a_1 b_0 \end{bmatrix} u, \quad (3.49)$$

$$y = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + b_0 u. \quad (3.50)$$

Stability, Controllability, and Observability

Stability For LTI systems, the *stability* is guaranteed only if all the eigenvalues of matrix $\mathbf{A}$ have negative real parts [29]. The systems that meet this feature are *asymptotically stables*. The solutions of this eigenvalues or system's poles are given by (3.51) [34]

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0. \quad (3.51)$$

Controllability A system described by (3.41) and (3.42) is completely controllable, if there exists an input u that can transfer an initial state to any other final state without restrictions, over a finite time interval $t_0 < t < t_f$.

If the system is completely controllable, the matrix (3.52)

$$\mathbf{C}_O = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}, \quad (3.52)$$

must be of rank n [26]. This matrix is commonly called *controllability matrix*.

Observability A LTI system described in Equations (3.41) and (3.42) is entirely observable if every initial state $\mathbf{x}(0)$ can be deduced from knowledge of the output $y(t)$ over a finite time interval $t_0 < t < t_f$.

If the system is completely observable, the matrix (3.53)

$$\mathbf{O}_B = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}, \quad (3.53)$$

must be of rank n. This matrix is commonly called *observability matrix* [26].

3.2.3 Control Design with State Feedback

In modern control design, all the poles are located in closed-loop; however, there is a cost. It is necessary to acquire proper measures of all state variables, or include a state observer in the system if they are inaccessible [26].

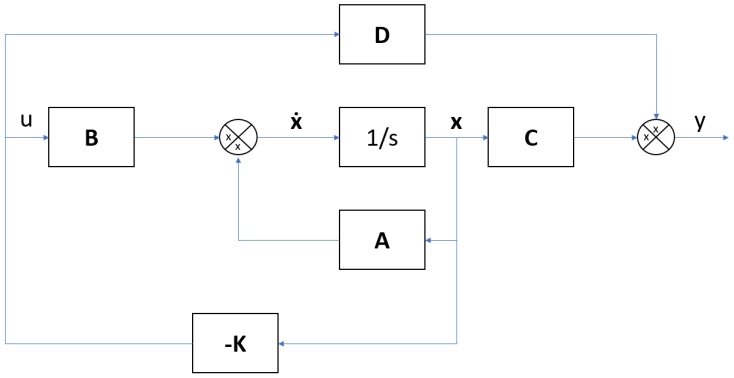


Figure 3.5: Closed-loop blocks diagram.

Considering again the state-space representation, Equations (3.41) and (3.42) respectively

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t),$$
$$y(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t),$$

the control signal is chosen as in Equation (3.54)

$$u(t) = -\mathbf{K}\mathbf{x}(t), \tag{3.54}$$

where $\mathbf{K}$ is the *state feedback gain vector*.

Equation (3.55) is obtained by substituting (3.54) into (3.41)

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(t). \tag{3.55}$$

The solution of $\mathbf{x}(t)$ is given by Expression (3.56)

$$\mathbf{x}(t) = e^{(\mathbf{A}-\mathbf{B}\mathbf{K})t}\mathbf{x}(0). \tag{3.56}$$

First, it is necessary to write the model of the plant, Equation (3.41), in the *controllable canonical form* as in (3.47); by choosing an appropriate $\mathbf{K}$, the matrix $\mathbf{A} - \mathbf{B}\mathbf{K}$ becomes asymptotically stable. The eigenvalues of this matrix are called the regulator poles. The goal is to find $\mathbf{K}$, such that it satisfies all the eigenvalues of this matrix.

Pole Placement

Arbitrary placement of the poles is possible, if and only if, all the system is entirely controllable [26] as is shown in Equation (3.52). Equation (3.37) places the poles for second-order systems.

Determination of K Using Ackermann's Formula There are several methods to obtain $\mathbf{K}$, one of them is a widespread method known as *Ackermann's formula*.

The method of Ackermann is expressed as in (3.57) [30]

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}^{-1} \mathbf{P}(\mathbf{A}), \tag{3.57}$$

being $\mathbf{P}(\mathbf{A})$, Equation (3.58), the polynomial matrix of $\mathbf{A}$ [35]

$$\mathbf{P}(\mathbf{A}) = \alpha_0\mathbf{A}^n + \alpha_1\mathbf{A}^{n-1} + \cdots + \alpha_{n-1}\mathbf{A} + \alpha_n\mathbf{I}, \tag{3.58}$$

and α_n, is the characteristic equation for the poles determined in Equation (3.37).

Tracking Systems

For tracking a reference, it is desired that $y(t) \approx r(t)$ as $t \to \infty$ [36]. Considering the *final value theorem* (3.27), it is possible to obtain (3.59)

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s). \tag{3.59}$$

For a good tracking performance, the transfer function from R(s) over Y(s) should be 1 as indicated in Equation (3.60)

$$sY(s) \approx sR(s), s \to 0 \quad \Rightarrow \quad \left.\frac{Y(s)}{R(s)}\right|_{s=0} = 1. \tag{3.60}$$

One method to reach a proper tracking is to scale the reference input $r(t)$ in the way expressed in (3.61)

$$u(t) = \overline{N}r(t) - \mathbf{K}\mathbf{x}(t), \tag{3.61}$$

where $\overline{N}$ is the extra gain used to scale the transfer function.

By substituting Equation (3.61) into (3.41) and neglecting $\mathbf{D}$ in Equation (3.42) are obtained the Expressions (3.62) and (3.63)

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) + \mathbf{B}\overline{N}r(t), \tag{3.62}$$

$$y(t) = \mathbf{C}\mathbf{x}(t), \tag{3.63}$$

so that the transfer function $G(s)$ is represented as in (3.64)

$$G(s)\overline{N} = \frac{Y(s)}{R(s)} = \mathbf{C}(s\mathbf{I} - (\mathbf{A} - \mathbf{BK}))^{-1}\mathbf{B}\overline{N} \tag{3.64}$$

Considering Equation (3.59) for a step input $y(t) \to constant$ as $t \to \infty$, it is possible to obtain the expression for the extra gain $\overline{N}$ (3.65)

$$\overline{N} = G(0)^{-1} = -(\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{B})^{-1}. \tag{3.65}$$

3.3 Digital Modern Control Theory

A *digital system* or *digital signal* is a sampled system in which there is a quantization process to convert analog to binary signals [37]. With the arrival of the Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC), it is possible to use computers as a control element. Some advantages and disadvantages of continuous-time control and discrete-time control are summarized below:

- *Advantages*
  - The possibility of real-time execution.
  - Flexibility in the implementation, tests, and reprogramming of algorithms.
  - The capability to obtain good results with low-cost microcontrollers.
- *Disadvantages*
  - Due to the sampling procedure, there is a loss of the information.
  - It requires learning a different theory, complementary to continuous systems.
  - It is necessary to bring the analog signal to conditioning for its digital conversion.

As many topics of digital control systems are similar in many ways to continuous control systems, the next section only presents the topics of interest in a summarized way. See [37], [38] and [39] for a detailed explanation of these topics.

3.3.1 Preliminaries

Mathematical Models of Digital Systems

A discrete time function is represented as indicated in Formula (3.66) [37]

$$y = f(k), \tag{3.66}$$

where, k represents an instant in time where the function has some value.

A n-order linear discrete time system can be described by Equation (3.67)

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n) = u(k)+b_1u(k-1)+\cdots+b_mu(k-m),\ n \geq m, \tag{3.67}$$

subject to initial conditions: $\{y(-1), y(-2), \cdots, y(-n), u(-1), u(-2), \cdots, u(-n)\}$, where a and b are constants, $y(k)$ is the *output*, $u(k)$ is the *input*, n and m are the orders.

Sample and Hold Signals

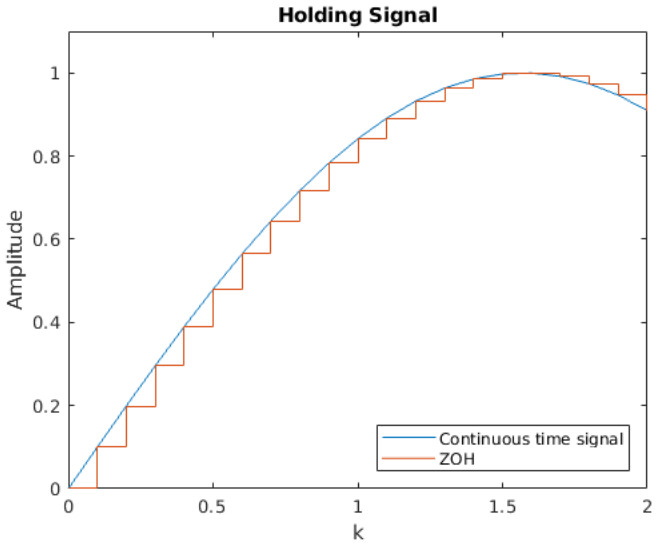


Figure 3.6: ZOH sampling.

Sampling Signals To get a sampled signal $f^*(t)$, it can be represented as the product of a pulse train multiplied by the function $f(t)$ [37], which has samples every T time units as indicated in (3.68)

$$f^*(t) = I(t)f(t). \tag{3.68}$$

The Fourier transform of $f^*(t)$, where $\omega_s = 2\pi/T$, is given by (3.69)

$$F^*(\omega) = \sum_{k=-\infty}^{\infty} C_k F(\omega - k\omega_s). \tag{3.69}$$

Formula (3.69) shows that the transform of the original signal composes the Fourier transform of the sampled signal, but with many copies out of phase in multiples of the sampling frequency ω_s.

Selection of sampling period It is essential to emphasize that to prevent aliasing it is necessary that the sampling signal ω_s is at least twice the frequency of the original signal ω_n as indicated in (3.70)

$$\omega_s \geq 2\omega_n. \tag{3.70}$$

In control theory is a common practice to choose a sampling time between 2 to 3 times the rise time [37].

Holding Signals It is necessary to use *holders* or *extrapolators* (3.71) [39] from a set of samples $f(t_k)$ to rebuild a continuous-time signal from a sampled-signal

$$f(t) = f(t_k),\ t_k \leq t < t_{k+1}, \tag{3.71}$$

obtaining a continuous signal by joining two points through a constant (zero order polynomial). This way is called ZOH. Figure 3.6 shows the effect of ZOH.

Z Transform

The Z transform can be used to solve discrete differential equations in an equivalent form as the Laplace transform and is defined by Formula (3.72) [40]

$$Y(z) = Z[y(k)] = \sum_{k=0}^{\infty} y(k) z^{-k}, \tag{3.72}$$

where z is a complex variable.

The key property of the Z transform is that it allows a solution to discrete differential equations

$$Z[f(k-1)] = z^{-1} F(z). \tag{3.73}$$

For a complete table and more characteristics of Z transform see [38].

Transfer Function

A discrete linear differential equation (3.67), also can be represented as in Equation (3.74)

$$a_n y(k)+a_{n-1}y(k+1)+\cdots+a_0 y(k+n) = b_m u(k)+b_{m-1}u(k+1)+\cdots+b_0 u(k+m). \tag{3.74}$$

The Z transform has the same role in discrete systems that the Laplace transform has [38]. To get the Z transform to each term it is necessary to apply Equation (3.73) into Equation (3.74)

$$a_0 z^n Y(z)+a_1 z^{n-1}Y(z)+\cdots+a_n Y(z) = b_0 z^m U(z)+b_1 z^{m-1}U(z)+\cdots+b_m U(z). \tag{3.75}$$

The general discrete transfer function $H(z)$ is given by (3.76)

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}, \tag{3.76}$$

moreover, if $n \leq m$, Equation (3.76) can be used to write the above as a ratio of polynomials in z, this is expressed as in (3.77)

$$H(z) = \frac{b_0 z^m + b_1 z^{m-1} + \cdots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_n}. \tag{3.77}$$

The properties of the transfer function are the same that were revised in the continuous-time domain in Section 3.2.1.

3.3.2 Digital State-Space Model

The discrete state-space equations and continuous state-space equations are similar in several ways. The present and the next sections only overhaul the fundamental similarities and differences to implement the state feedback control method into a computational algorithm.

Discretization of Continuous State-Space Model

Given the continuous state-space model, Equations (3.41) and (3.42)

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u,$$

$$y = \mathbf{C}\mathbf{x} + \mathbf{D}u,$$

preceded by a ZOH, with a sample time T_s, and defining the matrices $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ as the discrete equivalents of matrices $\mathbf{A}$ and $\mathbf{B}$ [39], Expressions (3.78) and (3.79) are obtained

$$\mathbf{\Phi} = e^{\mathbf{A}T_s}, \tag{3.78}$$

$$\mathbf{\Gamma} = \int_0^{Ts} e^{\mathbf{A}\eta T_s} d\eta \mathbf{B}. \tag{3.79}$$

Then the discrete state-space equivalent will be (3.80) and (3.81)

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k), \tag{3.80}$$

$$y(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{J}u(k), \tag{3.81}$$

being k a instant of time when the function takes a value, and $k+1$ the value at the subsequent instant of time.

The relation between discrete state-space model and discrete transfer function and the canonical forms are the same as in continuous time (3.43)–(3.50).

Stability, Controllability, and Observability

Stability A discrete LTI system described by Equations (3.80) and (3.81) is *asymptotically stable*, if and only if, all the eigenvalues of $\mathbf{\Phi}$ are located in the interior of the unitarian circle $\lambda_i < 1$ [37].

Controllability The deduction of the discrete controllability matrix is analogous to Equation (3.52), substituting $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ into $\mathbf{A}$ and $\mathbf{B}$, respectively, the discrete controllability matrix stays as in (3.82) [37]

$$\mathbf{C}_O = \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Phi\Gamma} & \cdots & \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \end{bmatrix}, \tag{3.82}$$

Observability As in the same way above, the discrete observability matrix, given by Equation (3.83), is analogous to its continuous matrix, Equation (3.53). It just needs to replace $\mathbf{A}$ with $\mathbf{\Phi}$ and $\mathbf{C}$ with $\mathbf{H}$

$$\mathbf{O}_B = \begin{bmatrix} \mathbf{H} \\ \mathbf{H\Phi} \\ \mathbf{H\Phi}^2 \\ \vdots \\ \mathbf{H\Phi}^{n-1} \end{bmatrix}, \tag{3.83}$$

3.3.3 Digital Control Design with State Feedback

This section is similar to Section 3.2.3. The most important things are summarized for a digital state feedback application. More details can be found in [37] and [38].

The discrete control signal $u(k)$ is chosen considering the discrete state-space Equations (3.80) and (3.81) as (3.84)

$$u(k) = -\mathbf{K}\mathbf{x}(k). \tag{3.84}$$

By substituting Equation (3.84) into Equation (3.80), it is obtained the Expression (3.85)

$$\mathbf{x}(k+1) = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}(k). \tag{3.85}$$

As it was done for the continuous-time system, it is necessary to write the model of the plant in *controllable canonical form*. Verify that the ranks of the discrete *controllability* and *observability* matrices are acceptable. The goal is still to find some value of $\mathbf{K}$, such that it satisfies all the eigenvalues of the matrix $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$.

Pole Placement

For an arbitrary placement of the poles, the rank of $\mathbf{C}_O$, Equation (3.82), must be the same as the order of the system. For a second-order system, the location of the poles are defined by (3.37) [25]

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}.$$

It is necessary to move these poles to the *Z-plane* as in (3.86) [37]

$$p_z = e^{pT_s}. \tag{3.86}$$

Determination of K Using Ackermann's Formula The method of Ackermann also works for discrete systems and is expressed as indicated in Formula (3.87) [38]

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Phi}\mathbf{\Gamma} & \cdots & \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \end{bmatrix}^{-1} \mathbf{P}(\mathbf{\Phi}), \tag{3.87}$$

defining $\mathbf{P}(\mathbf{\Phi})$ as in (3.88)

$$\mathbf{P}(\mathbf{\Phi}) = \alpha_0\mathbf{\Phi}^n + \alpha_1\mathbf{\Phi}^{n-1} + \cdots + \alpha_{n-1}\mathbf{\Phi} + \alpha_n\mathbf{I}. \tag{3.88}$$

Tracking Systems

The solution for a good tracking performance for a discrete state feedback control is scaling the reference input $r(k)$ as in Section 3.2.3. This scaling is indicated in Equation (3.89)

$$u(k) = \overline{N}r(k) - \mathbf{K}\mathbf{x}(k). \tag{3.89}$$

By substituting Equation (3.89) into (3.80) and equating $\mathbf{J} = 0$ in Equation (3.81), the Expressions (3.90) and (3.91) are obtained

$$\mathbf{x}(k+1) = (\mathbf{\Phi} - \mathbf{\Gamma K})\mathbf{x}(k) + \mathbf{\Gamma}\overline{N}r(k), \tag{3.90}$$

$$y(k) = \mathbf{H}\mathbf{x}(k). \tag{3.91}$$

$\overline{N}$ is defined by (3.92) [41]

$$\overline{N} = (\mathbf{H}(\mathbf{I} - (\mathbf{\Phi} - \mathbf{\Gamma K}))^{-1}\mathbf{\Gamma})^{-1}. \tag{3.92}$$

In this thesis, Sections 3.1.1 and 3.1.2 are used to analyze the properties of a single jump of the *hopper*. Section 3.2 is employed for the required simulations and Section 3.3 is used for perform real experiments in our prototype test-bed.

In the next chapter the methodology of this research is discussed.

Chapter 4

Methodology

This chapter describes the detailed solution of the problem, summarized in Chapter 1, of how the cubic robot can jump to a specific place over the surface of a comet or asteroid. Figure 4.1 shows the reaction forces and torque, as well as its ballistic trajectory [14].

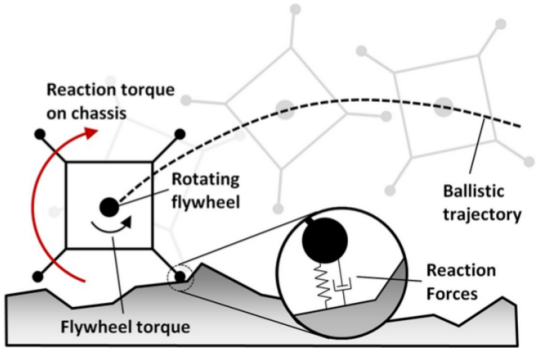


Figure 4.1: By rotating a flywheel, reaction forces make the rover jump.

Key Features

- The milligravity is exploited to generate big traverses with a low energy cost.
- The *hopper* robot has three orthogonal flywheels in its interior.
- As a first approximation, in this research, it is only considered a 2D ballistic flight and one single flywheel, and it also assumes that the center of mass of the *hopper* is at its geometric center. The implication of a movement in 3 dimensions will extend the analysis of the ballistic trajectory to a vectorized way. That analysis is left out for future work.
- To realize a jump maneuver a reaction wheel is slowly accelerated and suddenly stopped to generate the momentum that will propel a ballistic trajectory.

- A hop maneuver consists of two phases.
  - *Leverage phase.* The time elapsed between the moment when the robot is in a rest position and the time when any of the spikes have no contact with the ground.
  - *"Fly" phase.* The time elapsed between the moment when the platform leaves the ground and the moment when it falls again.

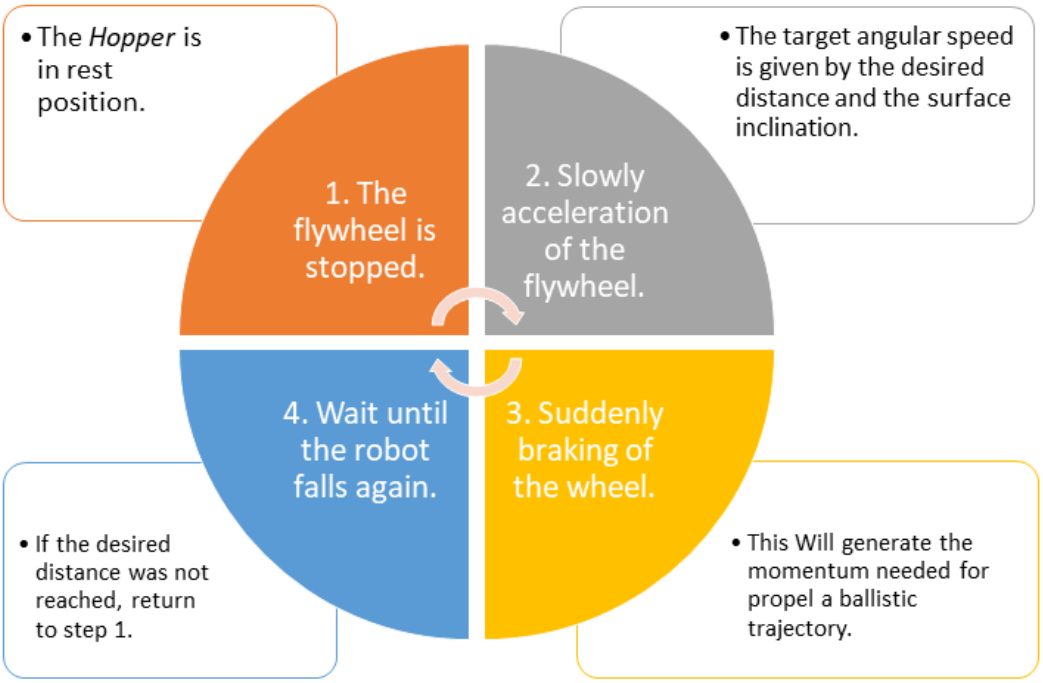


Figure 4.2: Diagram of a complete jump maneuver.

In Figure 4.2 it is explained in more detail how a complete maneuver of the robot is made to perform a jump to the target area. Table 4.1 describes the parameters of the *hopper* for a 2D model.

1. Consider the target distance d_{obj} and the surface inclination β.
2. Slowly spinning up the flywheel with motor torque $\tau < \tau_{min}$, Equation (4.3), to prevent the robot loses contact with the ground in advance.
3. d_{obj} and β give the target angular speed ω_f, Section 4.3; once it is reached stop the flywheel.
4. Stop the motor applying a voltage inversion brake at a specific time. This time is also related to d_{obj} and β.

5. Initiate the ballistic trajectory and wait until the platform falls again, Figure 4.1.

6. Verify if d_{obj} was reached; otherwise, make another jump repeating the complete process.

4.1 Hopping Dynamics

For a 2D model and just one flywheel, the robot is represented as a circle with four rigid bars attached to it, Figure 4.3 and Table 4.1 [14]. At the center of mass, there is a DC motor with an inertial wheel that provides a torque to the robot.

It is assumed that there are no slips on the edge of the spikes. The slips and bounces are governed by the friction with the regolith and rocks on the surface of the comets or asteroids. These interactions are beyond the scope of this research.

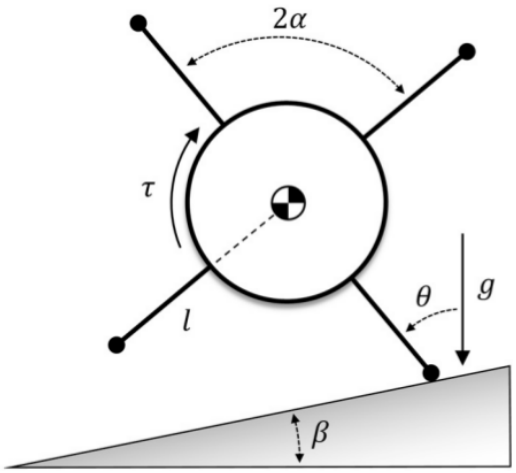


Figure 4.3: Robot model in 2D.

Symbol	Value	Units
θ	Hopper's angle	degrees
β	Surface slope	degrees
l	Spike's length	m
I_f	Flywheel's rotational inertia	kgm^2
I_p	Platform's rotational inertia	kgm^2
τ	Flywheel's torque	Nm
2α	Angle between spikes	degrees
ω_f	Flywheel angular velocity	rad/s
g	Gravity acceleration	m/s^2
m_p	Platform's mass	kg

Table 4.1: Parameters of the *hopper* model in 2D.

The surface inclination β will be positive for counterclockwise angles and negative for clockwise angles.

4.1.1 Instant Braking of the Flywheel

For the leverage phase, the equations of motion are based on the inverted pendulum model, Equations (3.10) and (3.12). For this analysis the second one is considered, as represented in (4.1)

$$\left(I + ml^2\right)\ddot{\theta} + ml\ddot{x} = mgl\theta. \tag{4.1}$$

By substituting $ml\ddot{x} = \tau$, the inertia $I = I_p$, and the total mass of the platform $m = m_p$, we have the expression for the angular velocity of the pendulum $\ddot{\theta}$ in (4.2)

$$\ddot{\theta} = \frac{m_p gl \sin\theta - \tau}{I_p + m_p l^2}. \tag{4.2}$$

Allen et al. [13] have derived that the minimum torque to initiate a rotation from rest is given by (4.3)

$$\tau_{min} = m_p gl \sin(\alpha + \beta), \tag{4.3}$$

also, the cited authors have defined the energy transfer ratio as Equation (4.4)

$$\eta = \frac{E^-}{E^+} = \frac{I_f}{I_p + m_p l^2}, \tag{4.4}$$

where E^- is the energy just before actuation (flywheel kinetic energy), and E^+ is the energy just after actuation (platform kinetic energy) [14].

By combining the angular momentum of the flywheel to the angular momentum of the platform about a spike, it is obtained Expression (4.5)

$$I_f \omega_f = \dot{\theta}(I_p + m_p l^2), \tag{4.5}$$

also, assuming an immediate momentum transfer as in (4.6)

$$v_h = l\dot{\theta}, \tag{4.6}$$

by substituting Equation (4.5) into Equation (4.4) it is obtained the resulting hop velocity (4.7)

$$v_h = \eta \omega_f l. \tag{4.7}$$

The hop angle θ_h is obtained only by the geometry of the robot α and the slope of the surface β. For horizontal terrains $\beta = 0^o$, this angle is always 45^o, as indicated by Equation (4.8)

$$\theta_h = \alpha + \beta. \tag{4.8}$$

It is important to remind that, this is the primary reason because the hopper is a cube instead of another polyhedron.

The formula of projectile movement, (4.9), gives the horizontal distance

$$d_h = \frac{v_h^2 \sin(2\theta_h)}{g}. \tag{4.9}$$

From Equations (4.4)–(4.9) it is possible to obtain the flywheel speed required to cover a horizontal distance, as given in (4.10)

$$\omega_f = \sqrt{\frac{d_h g}{\eta^2 l^2 \sin(2(\alpha + \beta))}}. \tag{4.10}$$

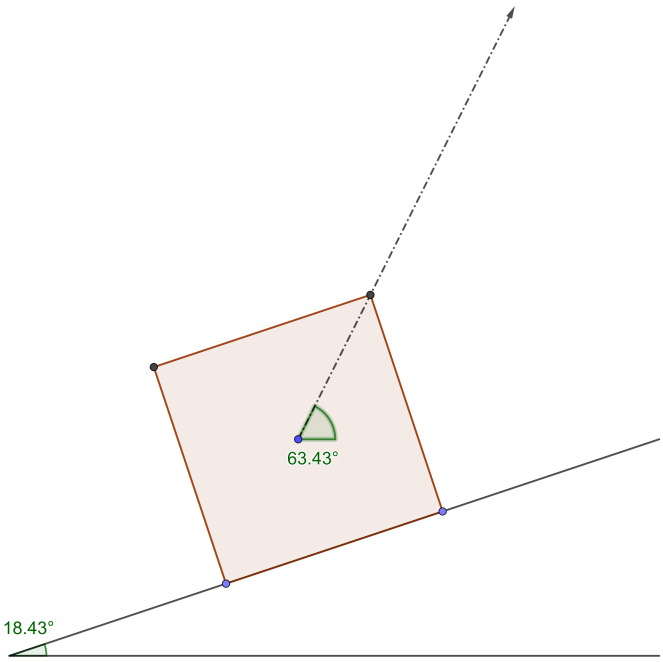


Figure 4.4: By stopping the flywheel instantaneously, the launch angle (represented as a green cone in the figure) will always be 45º plus the inclination of the surface 18.43º in the figure. Equation (4.8).

4.2 Proposed Solution

In this section, a solution for controlling jumps of the *hopper* cube is presented in detail. First, as the surface of an asteroid or comet is rarely flat, is a motivation of this research to deflect the launch angle depending on the slope of the ground to maximize the hop distance, as shown in Figure 4.5; for this, it will be necessary to extend the levering phase. Second, as sudden braking by impact of the inertial wheel might be harmful to the robot, we propose to stop the flywheel by voltage inversion. This approach also decreases the weight of the robot.

4.2.1 Controlled Braking of the Flywheel

Since the momentum transfer is no longer instantaneous [14], it is considered that the braking torque $\overline{\tau}$ and the angular momentum of the flywheel are related by (4.11)

$$\overline{\tau}\Delta t = I_f \omega_f, \tag{4.11}$$

where Δt is the braking time.

For big jumps, it is assumed that, $\overline{\tau} \gg m_p g l \sin\theta$, so Equation (4.2) can be approximated by (4.12)

$$\ddot{\theta} \approx \frac{-\tau}{I_p + m_p l^2}. \tag{4.12}$$

The new expressions will be: the launch velocity v_h, Equation (4.13), remains equal as described in Equation (4.7)

$$v_h = \frac{I_f \omega_f l}{I_p + m_p l^2}; \tag{4.13}$$

the launch angle θ_h can be determined by (4.14) [14]

$$\theta_h = \alpha + \beta - \frac{\eta I_f \omega_f^2}{2\overline{\tau}}; \tag{4.14}$$

and the horizontal distance d_h, (4.15), is given by the projectile formula as in Equation (4.9)

$$d_h = \frac{v_h^2 \sin(2\theta_h)}{g}. \tag{4.15}$$

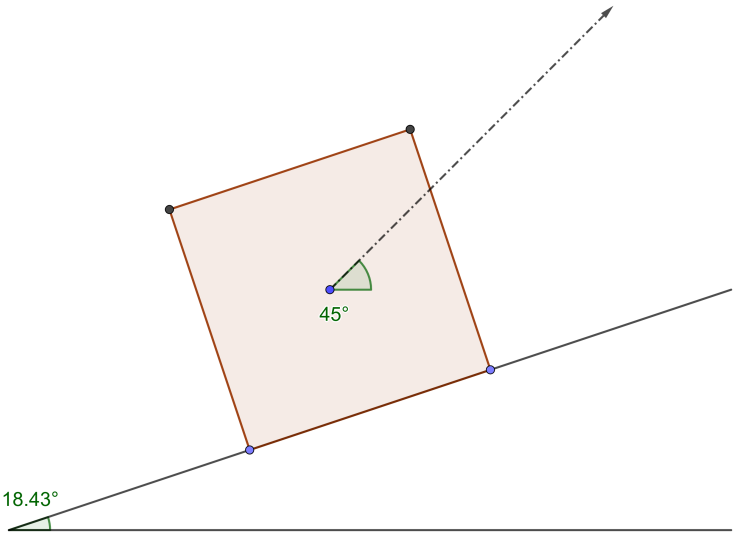


Figure 4.5: By a controlled braking torque $\overline{\tau}$ of the flywheel, the launch angle can be deflected. The goal is to approach the launch angle to 45° for maximizing the jump distance (4.14).

Braking torque, launch angle and jump distance relationships

To better understand the implementation of a controlled braking of the inertial wheel, some numerical simulations illustrating the relationships between the braking torque $\overline{\tau}$, the launch angle θ_h, and the jump distance d_h, have been generated.

Specifications Taking into account an angular speed of the flywheel of $\omega_f = 3500\ rpm$ or $366.5\ rad/s$, a surface slope of $\beta = 15^{o}$, and the physical characteristics of the robot summarized in Table 5.1.

Description The objective of these simulations is to know how θ_h and d_h change in function of $\overline{\tau}$. For these examples $\overline{\tau}$ ranges from $10^{-2}\ Nm$ to $10^{1}\ Nm$.

Expected Results It is expected that the maximum jump distance and a launch angle of 45^{o} can be found at the same magnitude of the braking torque.

Obtained Results Figure 4.6 shows how the launch angle changes, while the braking torque increases. In the same way, Figure 4.7 illustrates the relationship between the jump distance and the braking torque. Both images, demonstrate that the application of a torque of $\overline{\tau} = 0.03\ Nm$ (red dotted line) under a gravity acceleration of 77 $\mu m/s^2$, enables the reaching of a maximum distance about $d_h = 102.03\ m$ and a launch angle of $\theta_h = 45^{o}$.

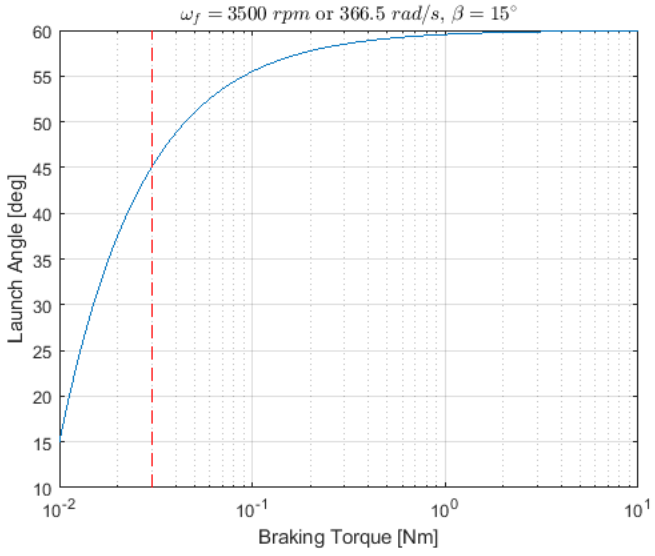


Figure 4.6: Resulting θ_h as function of $\overline{\tau}$.

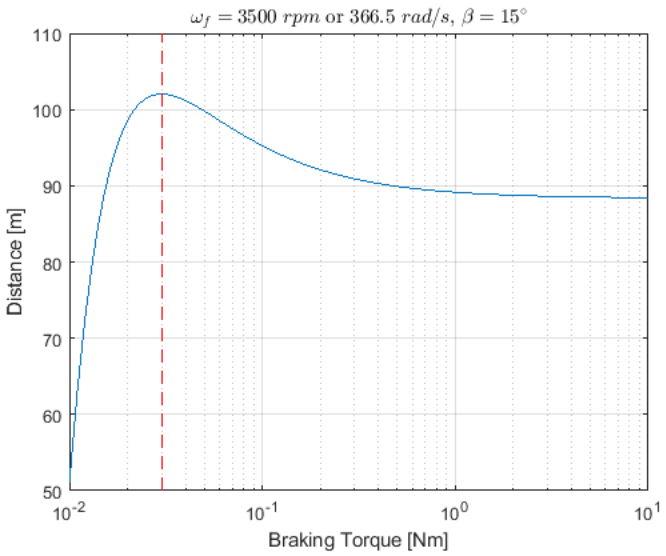


Figure 4.7: Resulting d_h as function of $\overline{\tau}$.

4.3 Adjusting the Trajectory for Uneven Terrains

The angular speed ω_f cannot be calculated directly, as we did in Equation (4.10). It will be necessary to obtain it numerically making a table which contains the target angular speed, the braking time in function of the distance and the surface inclination. Applying Equations (4.3), (4.11), (4.13), (4.14) and (4.15) it is possible to generate this table, as well as other characteristics such as the minimum speed-up time of the flywheel, the launch velocity and the "fly time". The algorithm to write this table is presented in Section 5.1.

4.4 Design of the Controller

The next step is to develop an angular speed controller that enables the robot to brake the motor from ω_f to 0, in a time Δt. Stop the flywheel in a short lapse can be challenging for a small robot, in which the power consumption and the reduced space that is available internally are factors to consider. In this work, it is implemented a state feedback control by pole placement, based on continuous time for the simulations and on discrete time for real experiments.

As we see previously, the actuator of the hopper consists of an inertial load or flywheel attached to a DC motor. For developing a state feedback control it is essential to get the model of the motor. This model is derived in Section 3.1.2. Henceforth, to simplify the notation of the angular speed $\dot{\theta}$ will be denoted as

ω_f. The equations that describe its behavior are given in (4.16) and (4.17)

$$v(t) = R\imath(t) + L\frac{d\imath(t)}{dt} + K\omega_f(t), \tag{4.16}$$

$$J\dot{\omega_f}(t) = -b\omega_f(t) + K\imath(t). \tag{4.17}$$

Applying the Laplace Transform to above equations, it is possible to obtain (4.18) and (4.19)

$$V(s) = RI(s) + LsI(s) + K\Omega_f(s), \tag{4.18}$$

$$Js\Omega_f(s) = -b\Omega_f(s) + KI(s), \tag{4.19}$$

the transfer function can be obtained by Equation (4.20)

$$\frac{\Omega_f(s)}{V(s)} = \frac{K}{(R + Ls)(Js + b) + K^2}. \tag{4.20}$$

where the output angular speed $\Omega_f(s)$ can be controlled through the input voltage $V(s)$. This is a second-order transfer function, some of its properties are reviewed in Section 3.2.1.

Taking the angular speed $\omega_f(t)$ and the circuit current $\imath(t)$ from Equations (4.16) and (4.17) as the output state variables, Equations (4.21) and (4.22) are indicated as

$$\frac{d\omega_f(t)}{dt} = -\frac{b}{J}\omega_f(t) + \frac{K}{J}\imath(t), \tag{4.21}$$

$$\frac{d\imath(t)}{dt} = \frac{v(t)}{L} - \frac{R}{L}\imath(t) - \frac{K}{L}\omega_f(t). \tag{4.22}$$

From (4.21) and (4.22), the state-space representation is given by (4.23) and (4.24)

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v(t), \tag{4.23}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t), \tag{4.24}$$

where the state vector $\mathbf{x}(t) = \begin{bmatrix} \omega_f(t) & \imath(t) \end{bmatrix}^T$ and $\dot{\mathbf{x}}(t) = d\mathbf{x}(t)/dt$. R is the armature resistance, L is the electric inductance, K is the back electromotive force (emf), J is the inertial load on the rotor, b is the viscous motor friction, $v(t)$ is the input voltage.

4.4.1 State Feedback by Pole Placement

For continuous LTI systems, the state representation is reviewed in Section 3.2.3, Equations (3.41) and (3.42)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t),$$

$$y(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t),$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are the state-matrices from Equations (4.23) and (4.24). $\mathbf{D}$ is equal to 0.

For tracking a reference $r(t)$, in this case, the desired angular speed $\omega_f(t)$ of the inertial wheel, the control signal can be selected as in Equation (3.61)

$$u(t) = -\mathbf{K}\mathbf{x}(t) + \overline{N}r(t).$$

For discrete LTI systems, the state representation was reviewed in Section 3.3.3, Equations (3.80) and (3.81)

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k),$$

$$y(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{J}u(k).$$

The control signal for tracking the discrete reference in an instant of time k is given by Equation (3.89)

$$u(k) = -\mathbf{K}\mathbf{x}(k) + \overline{N}r(k).$$

$\overline{N}$ is the extra gain needed to compensate the steady state error. For a continuous time is given as in Equation (3.65)

$$\overline{N} = (\mathbf{C}(-\mathbf{A} + \mathbf{B}\mathbf{K})^{-1}\mathbf{B})^{-1},$$

and for discrete time, Equation (3.92),

$$\overline{N} = (\mathbf{H}(\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K})^{-1}\mathbf{\Gamma})^{-1},$$

where, in this case, $\mathbf{I}$ is the 2×2 identity matrix. For both, $\mathbf{K}$ is a 1×2 matrix that can be found by Ackermann's formula, Section 3.2.3 and 3.3.3. The matrices $\mathbf{\Phi}$, $\mathbf{\Gamma}$, and $\mathbf{H}$ are the discrete equivalents of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively, and have the same size. Section 3.3.2 describes the process of discretization.

The parametrized values of the dynamics of the robot and the flywheel and the control properties are reviewed in the next chapter.

Chapter 5

Implementation

THE implementation has been divided into two phases:

1. **The viability of an effective, fast motor braking by voltage inversion.** Real experimentation with a DC motor, speed sensor, energy supply, motor driver and a microcontroller.

2. **The parabolic flight of the robot.** The simulation of the trajectory of the *hopper* under the gravity of the asteroid Itokawa.

5.1 Deflecting the Launch Angle of the *Hopper*

The first step for realizing a jump to a desired distance d_h is getting the angular speed of the flywheel ω_f and the braking time Δt to deflect the launch angle of the robot θ_h, this with the aim of maximizing the jump distance of the robot despite the inclination of the ground. The parameters required to find ω_f and Δt are shown in Table 5.1.

The model of the *hopper* is a $10 \times 10 \times 10$ cm cube with an inertial wheel of aluminum with a diameter of 6 cm and 76 g of weight. The total weight of the robot is about 1.5 kg since current 1U cubesats are around that size [42].

Symbol	Description	Value
g	Itokawa's mean gravity acceleration	77 $\mu m/s^2$
β	Surface inclination	-30°< β <30°
α	Half the angle between spikes	45°
l	Spike's length	0.071 m
m_p	*Hopper*'s mass	1.5 kg
m_f	Flywheel's mass	0.076 kg
I_p	*Hopper*'s rotational inertia	25e-4 kgm^2
I_f	Flywheel's rotational inertia	3.42e-5 kgm^2

Table 5.1: Parameters required to calculate the ballistic trajectory.

Algorithm 1 presents the procedure to calculate both, the angular speed and the braking time. For the simulations it has been selected the asteroid Itokawa with dimensions of $535 \times 294 \times 209$ m [43]; also this asteroid has been selected because most of the estimated population of NEOs are under 1 km of diameter [44]. It is essential to remind that these specifications are merely speculative and the robot could be bigger and heavier depending on the on-board instruments and the tasks that the robot will perform.

Algorithm 1 Algorithm to find ω_f and Δt, given β

Require: Parameters of the robot, g, β

$\omega_f \leftarrow 1$:*max speed of the motor*
$\overline{\tau} \leftarrow 0.001 : 100$
for $i \leftarrow 1 : length(\omega_f)$ **do**
    $v_h(i) \leftarrow l\eta\omega_f(i)$
    $\theta_h(i) \leftarrow \alpha - (\eta I_f \omega_f(i)^2)/(2\overline{\tau} + \beta)$
    $d_h(i) \leftarrow max(v_h^2 \sin(2\theta_h)/g)$
    $\Delta t(i) \leftarrow I_f \omega_f(i)/\overline{\tau}(i)$
end for

The MATLAB code of Algorithm 1 is presented in Appendix A.1.

Apart from the aforementioned parameters d_h, ω_f, Δt, and θ_h; other parameters are also important to know the ideal trajectory of the jump. The speed-up time T_{min} is the minimum time to reach the objective angular speed of the flywheel from rest position. L_s is the velocity at which the *hopper* will be fired or launch speed. T_f is the time the robot takes to reach its destination or "fly" time.

Distance	$\mathbf{d_h}$**[m]**	**5**	**10**	**30**	**50**	**70**	**100**
Angular speed	ω_f[rad/s]	81.3	115.0	198.9	256.7	303.7	363.0
Speed-up time	T_{min}[s]	375.0	531.0	919.0	1186.0	1403.0	1677.0
Braking time	Δt[s]	2.78	1.97	0.97	0.80	0.69	0.56
Launch speed	L_s[cm/s]	2.0	2.8	4.8	6.2	7.3	8.8
Launch angle	θ_h[degrees]	43.0	43.0	46.0	45.0	44.0	45.0
Fly time	T_f[s]	347.0	491.0	900.0	1139.0	1335.0	1610.0

Table 5.2: Surface slope $\beta = 20^{\text{o}}$.

Distance	$\mathbf{d_h}$**[m]**	**5**	**10**	**30**	**50**	**70**	**100**
Angular speed	ω_f[rad/s]	83.8	118.4	205.0	264.7	313.2	374.4
Speed-up time	T_{min}[s]	612.0	865.0	1497.0	1933.0	2287.0	2733.0
Braking time	Δt[s]	0.0	0.0	0.0	0.0	0.0	0.0
Launch speed	L_s[cm/s]	2.0	2.9	5.0	6.4	7.6	9.1
Launch angle	θ_h[degrees]	35.0	35.0	35.0	35.0	35.0	35.0
Fly time	T_f[s]	302.0	427.0	739.0	954.0	1128.0	1349.0

Table 5.3: Surface slope $\beta = -10^{\text{o}}$.

Tables 5.2 and 5.3 summarize the calculations for the jump parameters for, respectively, a surface slope of 20º and a slope of -10º. Both tables show representative distances from 5 m to 100 m. It is worth to remark that for negative angles on inclination the braking time is about $\Delta t \approx 0$ s.

As it can be seen in Tables 5.2 and 5.3, the longer the distance d_h, the faster the launch speed L_s. Greater distances than 100 m are achievable, however, approaching the escape velocity could be dangerous. For Itokawa, the escape velocity is about 11.28 cm/s [43]. For example, in Tables 5.2 and 5.3 the maximum launch speed L_s are 8.8 cm/s and 9.1 cm/s respectively.

Once the necessary parameters to perform a successful jump have been obtained, the next step is to verify the viability to stop the flywheel in the specific Δt time using voltage inversion. To control the speed of the flywheel it is necessary to develop the model of the motor and find its parametrized state matrices. This model is derived in Section 3.1.2.

5.2 Find the Parameters of the DC Motor

Simulink Simscape Electronics™ R2017b was used to estimate the values b, R, K and L described in Table 3.2; J was estimated by the weight and size of the flywheel. Figure 5.1 presents the model of a DC motor assembling the inherent

electrical (resistance, inductance, emf) and mechanical (shaft inertia, friction) characteristics of the motor; also, the model contains an H-bridge, a Pulse-width Modulation (PWM) voltage source, a current sensor and a rotational motion sensor.

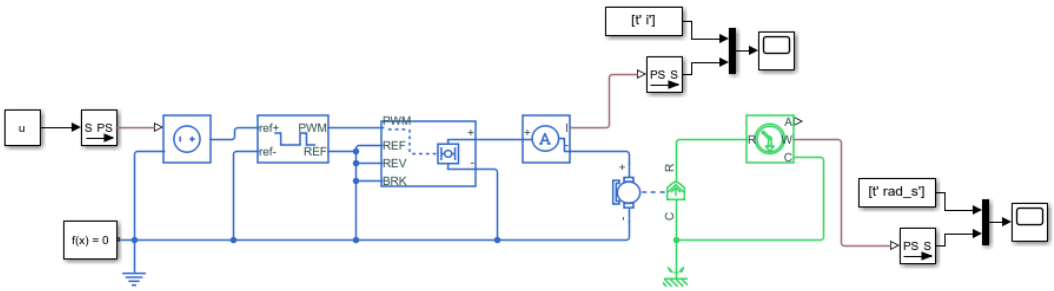


Figure 5.1: Model of a direct-current motor in Simscape^{TM}.

Using measured data from a real system is possible to get the parameters of the motor. The experiments were conducted using an Arduino board MKR1000 [45], an H-bridge BTS7960B [46], a current sensor ACS712 [47] and a DC motor with rotary optical encoder. Figure 5.2 shows the components of the cubic robot. A DC power supply of 24 V — 2 A was used.

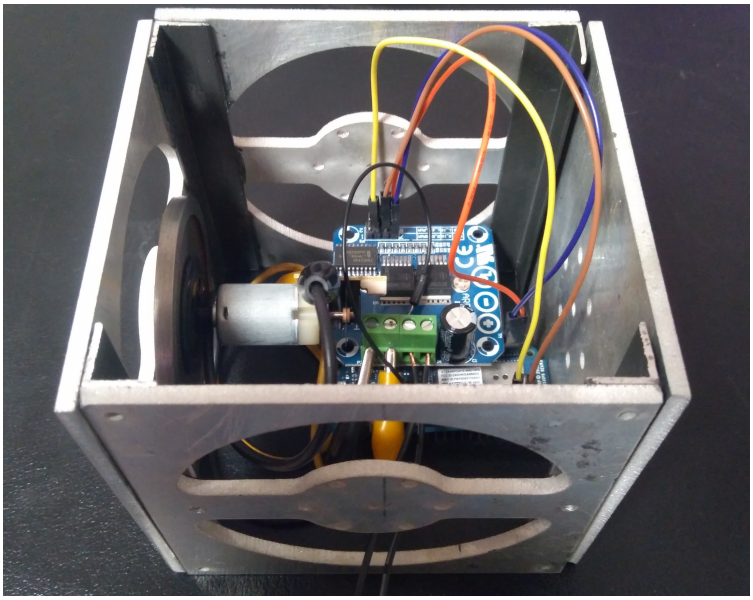


Figure 5.2: Prototype used for the experiments.

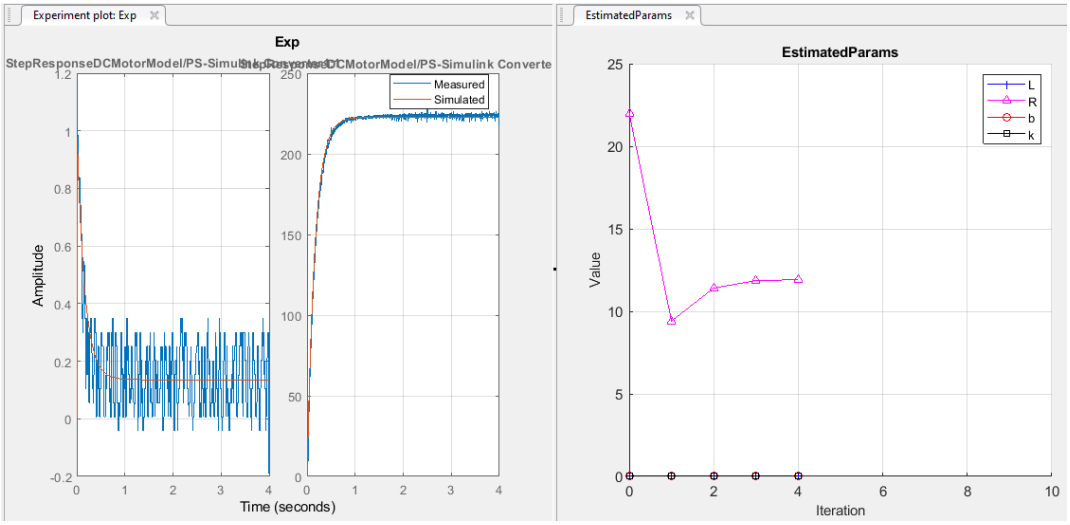


Figure 5.3: Matching measured and simulated data of the DC motor model in Simulink Parameter Estimation™ R2017b. In the right plot the values of L, b and K are quite smaller than R, for that reason are a the bottom of the picture.

Steps to estimate the parameters:

1. In the microcontroller, develop a speedometer and an ammeter to measure their responses to a 12 V step input.
2. Import the measurement data of the angular speed ω_f of the DC motor and the electrical current $\imath$ of the circuit.
3. In the Simulink model give approximate values to b, R, K, and L.
4. Using Simulink Parameter Estimation™, tune these parameters automatically.

On the left of Figure 5.3 is presented the measured and simulated responses of $\imath$ and ω_f, respectively. On the right is shown the tuning of b, K, L, and R.

The obtained values of the electrical and mechanical parameters of the DC motor are presented in Table 5.4.

Symbol	Description	Value	Units
J	Moment of inertia of the flywheel	3.42e-5	$kg\ m^2$
b	Friction coefficient of the rotor	2.20e-5	$N\ m\ s$
K	Back-emf constant	47.96e-3	$\frac{V}{rad/s}$
L	Motor winding inductance	7.75e-3	H
R	Motor winding resistance	11.36	Ω

Table 5.4: Estimated parameters of the motor.

5.3 State Feedback Controller by Pole Placement

In Section 4.4 it was proposed a state feedback control by the pole placement technique to develop an angular speed controller, based on continuous time for simulations and on discrete time for real experiments.

Taking into consideration Equations (4.23) and (4.24), the parametrized state-space representation rounded by 2 decimals can be represented as Equations 5.1 and (5.2), respectively

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -0.64 & 1402.21 \\ -6.19 & -1466.54 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 129.04 \end{bmatrix} v(t), \tag{5.1}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t), \tag{5.2}$$

where the state vector $\mathbf{x}(t) = \begin{bmatrix} \omega_f(t) & \imath(t) \end{bmatrix}^T$ and $\dot{\mathbf{x}}(t) = d\mathbf{x}(t)/dt$.

5.3.1 Simulation of the State Feedback Controller

As a first approximation to verify the simulated performance of the controller, a step input with a reference of $\omega_f = 280\ rad/s$, an *overshoot* of $OS = 4\%$ and a *settling time* equal to $S_t = 0.5\ s$ has been selected.

The controllable canonical form of the Equations (5.1) and (5.2) are given by (5.3) and (5.4)

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -9622 & -1467 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(t), \tag{5.3}$$

$$y(t) = \begin{bmatrix} 1.81\text{e}5 & 0 \end{bmatrix} \mathbf{x}(t), \tag{5.4}$$

Stability, Controllability, and Observability

The eigenvalues of the matrix $\mathbf{A}$ in Equation (5.3) are negatives $\lambda_1 = -6.59$ and $\lambda_2 = -1460.60$, therefore, the motor system is stable.

The controllability matrix for the DC motor system is given by (5.5)

$$\mathbf{C}_O = \begin{bmatrix} 0 & 1 \\ 1 & -1467.19 \end{bmatrix}. \tag{5.5}$$

The rank of the matrix $\mathbf{C}_O = 2$; therefore, the system is entirely controllable.

The observability matrix for the motor system is given by (5.6)

$$\mathbf{O}_B = \begin{bmatrix} 1.81\text{e}5 & 0 \\ 0 & 1.81\text{e}5 \end{bmatrix}. \tag{5.6}$$

The rank of the matrix $\mathbf{O}_B = 2$; therefore, the system is completely observable.

Pole Placement

As it was mentioned in Section 3.2.1, the DC motor model can be analyzed as an LTI second-order system. The value of the *damping ratio* in (5.7) is given by Equation (3.39)

$$\zeta = 0.72, \tag{5.7}$$

also, the *natural frequency* in Equation (5.8) with a 2% tolerance is obtained through Equation (3.40)

$$\omega_n = 10.90. \tag{5.8}$$

In Equation (3.37) is given the way to find the poles of a second-order system. For the actual design have the values of (5.9)

$$\mathbf{p} = \begin{bmatrix} -7.8 + \imath 7.61 & -7.8 - \imath 7.61 \end{bmatrix}. \tag{5.9}$$

Determination of K

Using Ackermann's Formula, revised in Section 3.2.3, it is possible to get the values of the gain matrix $\mathbf{K}$, Equation (5.10)

$$\mathbf{K} = \begin{bmatrix} -9503.10 & -1451.59 \end{bmatrix}. \tag{5.10}$$

Tracking a Referenced Angular Speed

As it was mentioned in Equations (3.61)–(3.63), for tracking a referenced ω_f it is necessary to add an extra gain $\overline{N}$ to reduce the steady-state error. Equation (3.65) shows the method of getting this constant. For the current system is expressed in Equation (5.11)

$$\overline{N} = 1523.16. \tag{5.11}$$

The control signal will be selected as in Equation (3.61).

Obtained Results Figure 5.4 shows a simulation of the behavior of the angular speed of the motor. The design parameters $\omega_f = 280\ rad/s$, $OS = 4\%$ and $S_t = 0.5\ s$ are successfully accomplished. Figure 5.5 shows the simulated input power supply. The simulations have been done using MATLAB® Control System Toolboox™ R2017b.

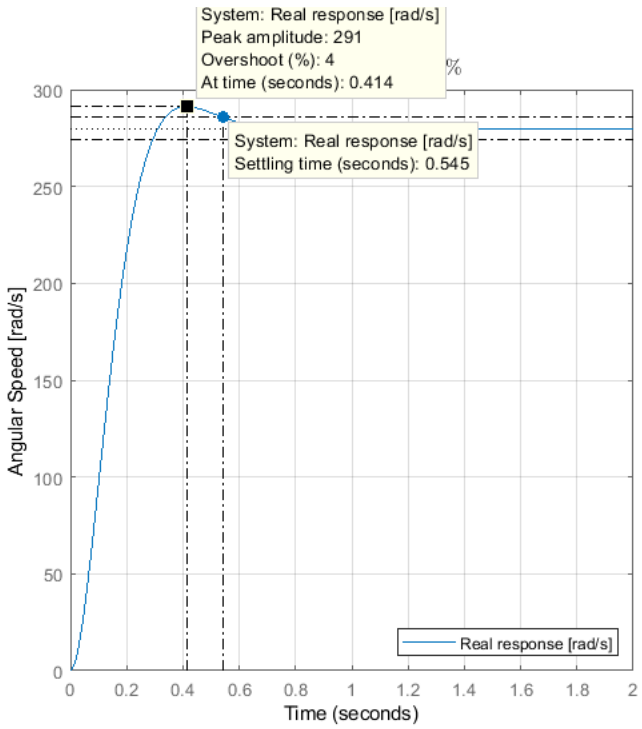


Figure 5.4: Simulated step response characteristics.

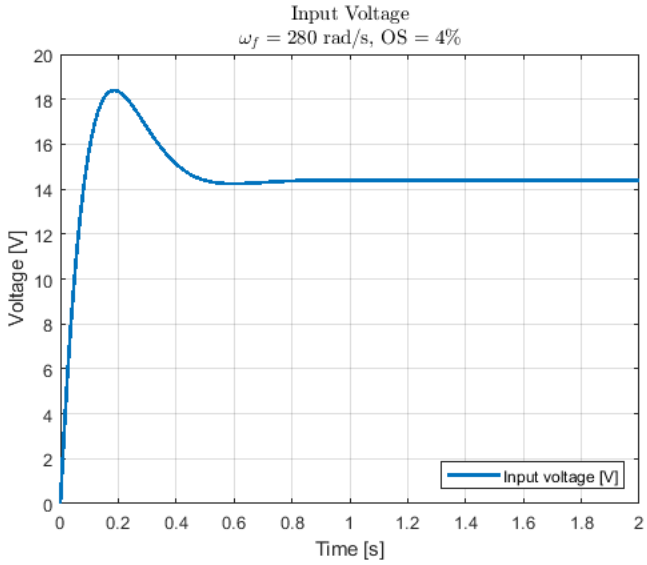


Figure 5.5: Simulated input voltage.

5.3.2 Real Performance of the State Feedback Controller

Once the simulated response has been verified, it is necessary to perform real experiments to compare them and check that the control model are feasible. As in above section, the reference is a step input of $\omega_f = 280\ rad/s$, an *overshoot* of $OS = 4\%$ and a *settling time* equal to $S_t = 0.5\ s$.

The parametrized discrete state-space representation from Equations (5.3) and (5.4) with a sample time $T_s = 0.0005\ s$ rounded by 2 decimals is expressed in Equations (5.12) and (5.13)

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 3.54\text{e-}4 \\ -3.41 & 0.48 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 9.93\text{e-}8 \\ 3.54\text{e-}4 \end{bmatrix} v(k), \tag{5.12}$$

$$y(k) = \begin{bmatrix} 1.81\text{e}5 & 0 \end{bmatrix} \mathbf{x}(k). \tag{5.13}$$

The discretization of a continuous time system is reviewed in Section 3.3.2.

Stability, Controllability and Observability

The system is stable since both eigenvalues of the discrete matrix $\mathbf{\Phi}$ are between $-1 < \lambda_{1,2} < 1$. $\lambda_1 = 0.99$ and $\lambda_2 = 0.48$.

The controllability matrix for the DC motor system is expressed by (5.14)

$$\mathbf{C}_O = \begin{bmatrix} 9.93\text{e-}8 & 2.25\text{e-}7 \\ 3.54\text{e-}4 & 1.69\text{e-}4 \end{bmatrix}. \tag{5.14}$$

The rank of the matrix $\mathbf{C}_O = 2$; so, the system is completely controllable.

The observability matrix for the motor system is given by (5.15)

$$\mathbf{O}_B = \begin{bmatrix} 1.81\text{e}5 & 0 \\ 1.81\text{e}5 & 64.08 \end{bmatrix}. \tag{5.15}$$

The rank of the matrix $\mathbf{O}_B = 2$; also, the system is completely observable.

Pole Placement

For second-order LTI discrete systems, the *damping ratio* and the *natural frequency* have the same values as in the continuous time domain. The previous values of $\mathbf{p} = -7.8 \pm 7.61\imath$ will be moving to the *Z-plane* using Equation (3.86). The values of these poles are indicated in (5.16)

$$\mathbf{p}_z = \begin{bmatrix} 0.98 + \imath 4.35\text{e-}3 & 0.98 - \imath 4.35\text{e-}3 \end{bmatrix}. \tag{5.16}$$

Determination of K

Ackermann's Formula also works for discrete systems. Applying the methodology presented in Section 3.3.3 the gain matrix $\mathbf{K}$, Equation (5.17), can be calculated for the current DC motor system

$$\mathbf{K} = \begin{bmatrix} -7407.48 & -1361.40 \end{bmatrix}. \tag{5.17}$$

Tracking a Referenced Angular Speed

Also, for tracking a referenced ω_f in discrete systems, it is necessary to add an extra gain $\overline{N}$ to reduce the steady-state error. The control signal is selected as in Equation (3.89), where $\overline{N}$ has been calculated from Equation (3.92). Equation (5.18) shows its value

$$\overline{N} = 81.71. \tag{5.18}$$

Obtained Results The behavior of the real response (black line) acting similarly to the predicted response (red line), with a relative error of 1.43%. The discrete input voltage (blue line) is shown in Figure 5.7, and the differences between the simulated input voltage and the input voltage delivered by the Arduino board is about 0 V. As it can be seen in Figure 5.6, a digital implementation of a state feedback controller is necessary for a good performance of the code running in a microcontroller.

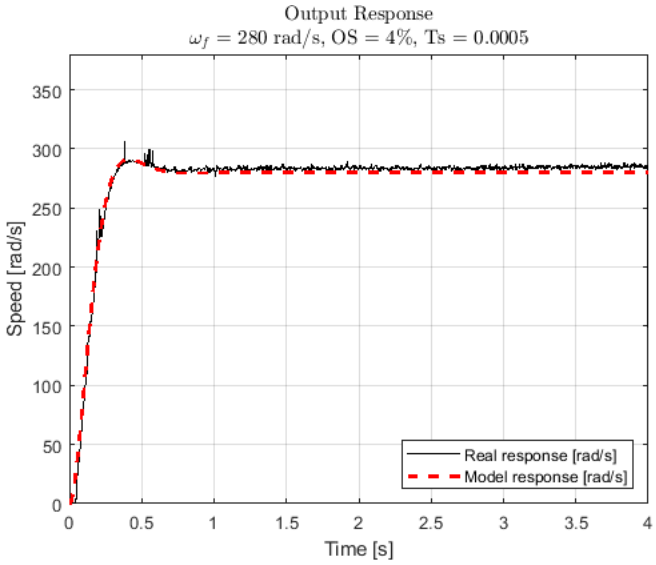


Figure 5.6: Measured and simulated speed responses of the motor.

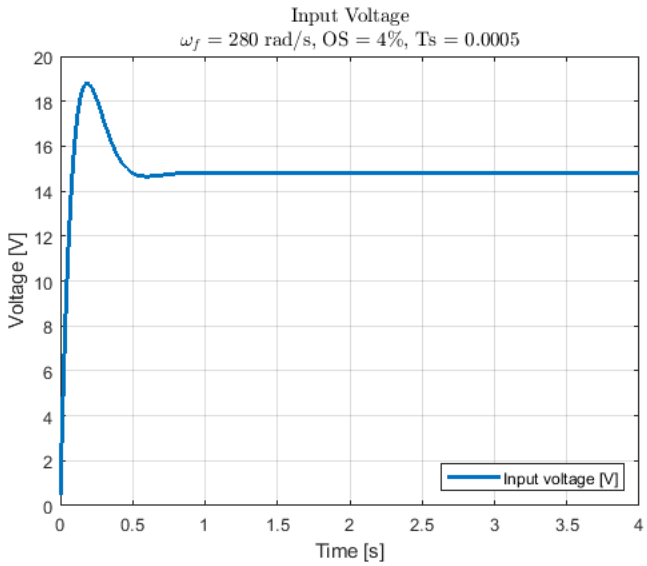


Figure 5.7: Real input voltage of the motor.

5.3.3 Applying the Braking by Voltage Inversion

Algorithm 2 presents the way to control the braking of the flywheel by voltage inversion using state feedback by pole placement. In Appendix A.2 there is an example code for the Arduino MKR1000 board with design parameters of: $\omega_f = 198.758\ rad/s$, $\Delta t = 0.261\ s$, $T_s = 0.0005\ s$, $OS = 0.0001\%$, and $S_t = 0.1\ s$. It is worth to mention that the values of the matrices $\mathbf{A}$ and $\mathbf{B}$, and the gains $\mathbf{K}$, and $\overline{N}$, change respecting ω_f and Δt.

Algorithm 2 DC motor braking: state-space method

Require: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{K}$, $\overline{N}$, ω_f, $\Delta t, T_{min}, T_s$.

$r \leftarrow 0$
$\delta_1 \leftarrow \omega_f * T_s / T_{min}$
$\delta_2 \leftarrow \omega_f * T_s / \Delta t$
if $clock = T_s$ **then**
  if $r < \omega_f$ **then**
    $r \leftarrow r + \delta_1$
  end if
  if $r = \omega_f$ **then**
    $r \leftarrow r - \delta_2$
    if $r \leq 0$ **then**
      $r \leftarrow 0$
    end if
  end if
  $u \leftarrow -\mathbf{K}\mathbf{x}_k + \overline{N}r$
  $\mathbf{x}_{k+1} \leftarrow \mathbf{A}\mathbf{x}_k + \mathbf{B}r$
  $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1}$
  $microcontroller \leftarrow u$
end if

In this chapter the implementation of the proposed technique of control that will allow the *hopper* to stop the flywheel from a certain ω_f in a Δt time was discussed. To achieve the expected performance of the speed controller (state feedback by pole placement) we owed know *a priori* the real parameters of the actuator of the prototype.

The results of the braking and their simulated jumps will be presented in the next chapter.

Chapter 6

Results

BELOW are presented the summary of the resultant ballistic trajectories of the *hopper* robot derived from the application of the real braking of the inertial wheels from the implementation of Algorithm 2.

6.1 Experimental Results and Simulations

For the sake of clarity, the test that were conducted are enumerated consecutively from 1 to 14. For each test, a brief description of the test and its goal, as well as the expected and the obtained results are included.

6.1.1 Braking Responses

Test 1. Speed control of the DC motor and its input voltage signal

Description This first experiment verified the efficiency of the speed controller. The speed of the DC motor (black line) must decrease uniformly following, as close as it can, the simulated model (red dotted line); also, the input signal (blue line) must be within the allowed ranges ($\pm 24\ V$).

Expected Results Taking in consideration an objective distance of $d_h = 70\ m$ and a surface inclination of $\beta = 15^{o}$, the angular speed of the flywheel needs to stop from $\omega_f = 303.69\ rad/s$ to $0\ rad/s$ in a time $\Delta t = 0.50\ s$.

Obtained Results The upper plot of Figure 6.1 shows an example of the implementation of the braking response. The blue line in Figure 6.1 is the input voltage applied to the DC motor. For more aggressive braking, the voltage inversion also will be faster and with more negative amplitude. The real angular speed was $\omega_f = 308.8\ rad/s$ and the braking time was $\Delta t = 0.52\ s$, this gives a parabolic "flight" with a distance equal to $d_h = 72.4\ m$.

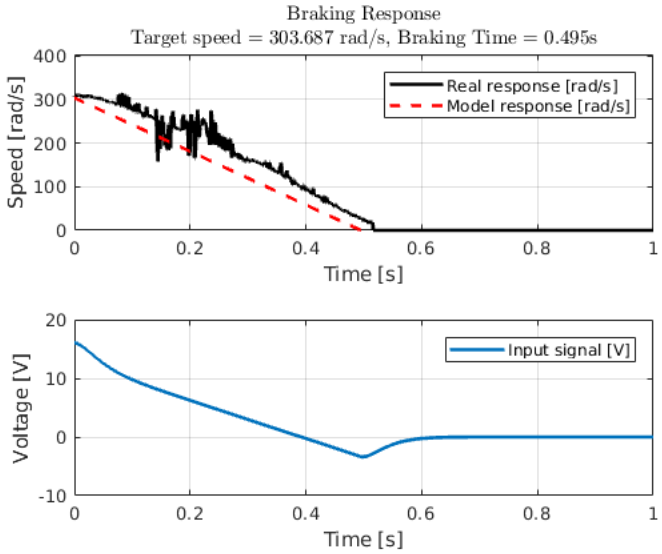


Figure 6.1: Example of the braking response of the flywheel. For the motor used in the prototype some vibration is presented at a range close to 200 rad/s, It is recalling that this vibration was imperceptible and do not change the performance of the results.

Test 2–7. Overview of some speed control responses for a wide variety of target distances and surface inclinations

Description The objective for the next tests is checking the effectiveness of the speed control of the motor by voltage inversion under several angular speeds and braking times.

Expected Results It is expected that the real responses (black lines) follow the model responses (red lines). Table 6.1 summaries the objective parameters.

Test	$\omega_\mathbf{f}$ $[rad/s]$	β $[degrees]$	$\mathbf{\Delta t}$ $[s]$	$\mathbf{d_h}$ $[m]$
2	81.16	0	0.00	5
3	114.88	10	0.98	10
4	198.76	5	0.26	30
5	199.49	30	1.37	30
6	256.67	20	0.80	50
7	389.98	-15	0.00	100

Table 6.1: Ideal braking responses under several conditions and expected jump distances.

Obtained Results As in Figure 6.1, the different plots in Figure 6.2 illustrate the responses of the controlled braking of the inertial wheel. Several inclinations

and target distances have been selected to illustrate the behavior of the motor braking. The red lines indicate the ideal controlled speed of the motor and how it decreases from a designated ω_f to 0 in a Δt time. Figure 6.2 shows the behavior of the tests proposed in Table 6.1. Table 6.2 makes a comparison of the obtained results. The relative error in Table 6.2 is obtained from the Formula (6.1)

$$Error = \frac{|expected\ response - real\ response|}{|expected\ response|} \times 100. \tag{6.1}$$

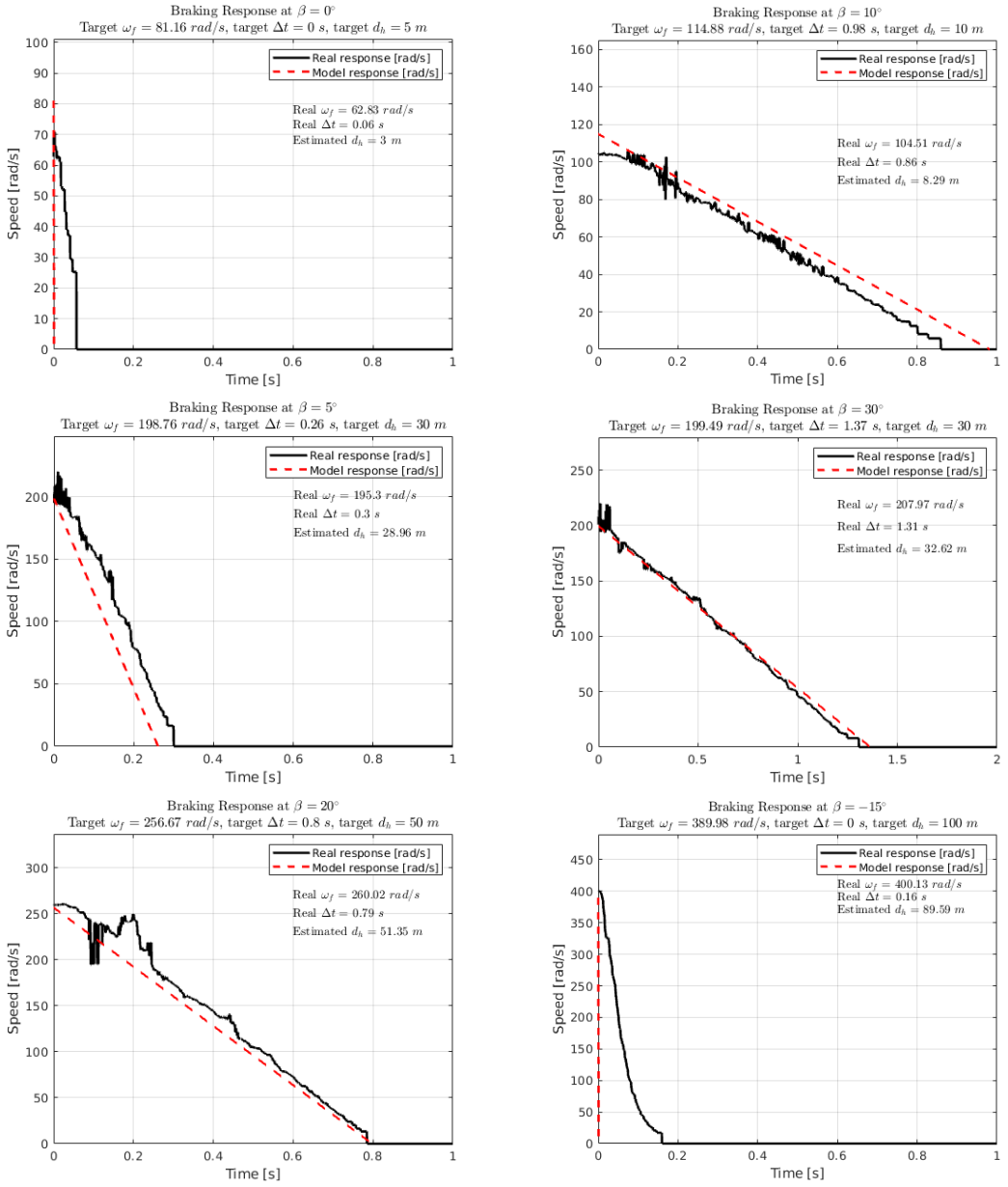


Figure 6.2: Examples of the application of the braking by voltage inversion and the estimated distance that the robot could reach.

Slope	Parameter	Exp. Response	Real Response	Error
0º	d_h $[m]$	**5**	**3.00**	40.00 %
	ω_f $[rad/s]$	81.16	62.83	
	Δt $[s]$	0.00	0.06	
10º	d_h $[m]$	**10**	**8.29**	17.10%
	ω_f $[rad/s]$	114.88	104.51	
	Δt $[s]$	0.98	0.86	
5º	d_h $[m]$	**30**	**28.96**	3.47%
	ω_f $[rad/s]$	198.76	195.3	
	Δt $[s]$	0.26	0.30	
30º	d_h $[m]$	**30**	**32.62**	8.73%
	ω_f $[rad/s]$	199.49	207.97	
	Δt $[s]$	1.37	1.31	
20º	d_h $[m]$	**50**	**51.35**	2.7%
	ω_f $[rad/s]$	256.67	260.02	
	Δt $[s]$	0.80	0.79	
-15º	d_h $[m]$	**100**	**89.59**	
	ω_f $[rad/s]$	389.98	400.13	10.41%
	Δt $[s]$	0.00	0.16	

Table 6.2: Comparison between the simulated and resultant responses of Figure 6.2.

6.1.2 Jump Results

To validate the resultant parabolic trajectories of the rover, the surface slope was divided each 5º from 30º to -30º; for every angle, representative distances have been selected (5 m, 10 m, 30 m, 50 m, 70 m, and 100 m); for each of these distances, 14 repetitions of the braking of the DC motor and their consequent simulated jumps under the gravity of the asteroid Itokawa were conducted. Given a total of 1092 experiments carried out.

In this research, three surface inclinations have been chosen for the presentation of the results: 15º, 0º, and -15º. From these angles, have been plotted three target distances: 5 m, 50 m, and 100 m, in which the robot has reached, respectively, the worst, the best and an average performance according to the landing point and the original objective distance. As we said before, we do not considering bouncing.

Test 8. 50 m target

Description In this test, have been conducted a series of 14 jump simulations for each of the three selected inclinations to reach a 50 m target. Every simulated jump has been obtained from a real braking experiment of the motor.

Expected Results It is expected that for a 50 m target distance, each landing will be within a 5 m tolerance area.

Obtained Results Figure 6.3 offers a perspective from above of the target distance $d_h = 50\ m$ ($\circ$) with a 10% error tolerance and the place where everyone of the 14 simulations for each selected inclination touched the ground ($\times$) of the asteroid. Figure 6.3 scores the best results with a 50 m range with a relative error of 2.4%.

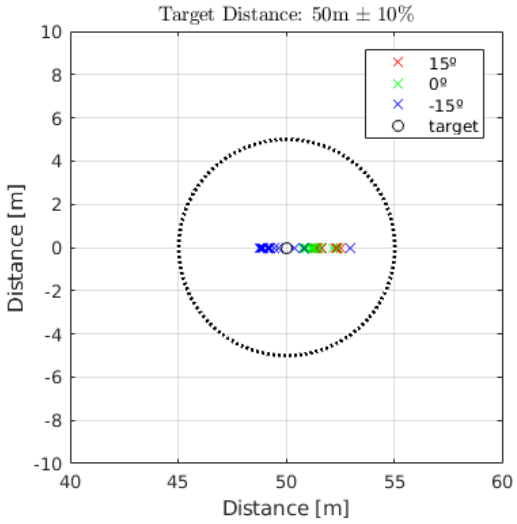


Figure 6.3: Best performance: 50 m.

Test 9. 100 m target

Description In this test, have been conducted a series of 14 jump simulations for each of the three selected inclinations to reach a 100 m target. Every simulated jump has been obtained from a real braking experiment of the motor.

Expected Results It is expected that for a 100 m target distance, each landing will be within a 10 m tolerance area.

Obtained Results Figure 6.4 offers a perspective from above of the target distance $d_h = 100\ m$ ($\circ$) with a 10% error tolerance and the place where everyone of the 14 simulations for each selected inclination touched the ground ($\times$) of the asteroid. Figure 6.4 presents the average results in the 100 m range with a relative error of 5.9%.

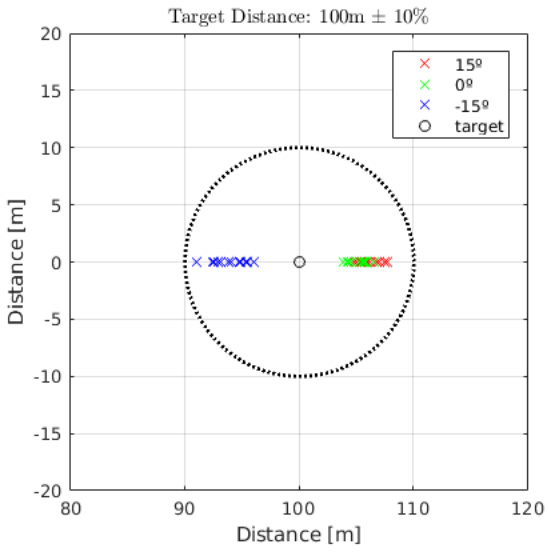


Figure 6.4: Average performance: 100 m.

Test 10. 5 m target

Description In this test, have been conducted a series of 14 jump simulations for each of the three selected inclinations to reach a 5 m target. Every simulated jump has been obtained from a real braking experiment of the motor.

Expected Results It is expected that for a 5 m target distance, each landing will be within a 0.5 m tolerance area.

Obtained Results Figure 6.5 offers a perspective from above of the target distance $d_h = 5\ m$ ($\circ$) with a 10% error tolerance and the place where everyone of the 14 simulations for each selected inclination touched the ground ($\times$) of the asteroid. Figure 6.5 records the worst results in the 5 m range with a relative error of 24.4%.

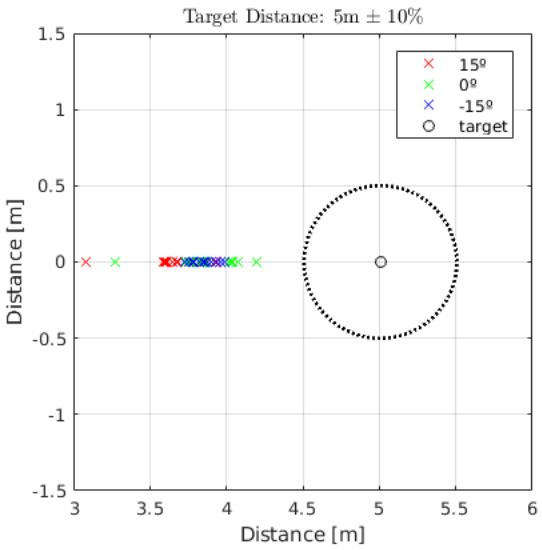


Figure 6.5: Worst performance: 5 m.

6.1.3 Average Trajectories

Tests 11–13. 5 m, 50 m, and 100 m trajectories.

Description Apart from knowing the landing site of the robot, it is necessary to know its "fly" trajectory. In Tests 11–13, the average trajectories of the simulated jumps presented in Section 6.1.2 for the 5 m, 50 m, 100 m targets, as well as the ideal path are displayed.

Expected Results The ideal trajectory is indicated by the dotted black line. It is expected that the average paths marked in red, green and blue colors ends up in the same place.

Obtained Results Figures 6.6 and 6.7 show the average parabolic trajectories for the 50 m, 100 m, and 5 m travels displayed in Figures 6.3, 6.4 and 6.5, respectively, as well as the ideal flight. Figure 6.6 and left plot in Figure 6.7 present acceptable results, since average landings are within the 10% tolerance error. Right plot in Figure 6.7 scores bad results because the average landings are far away from the target point.

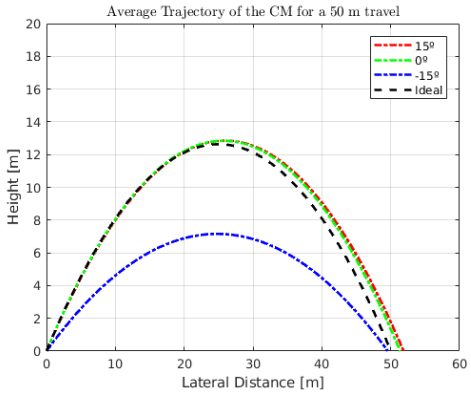


Figure 6.6: Average trajectories for a 50 m courses.

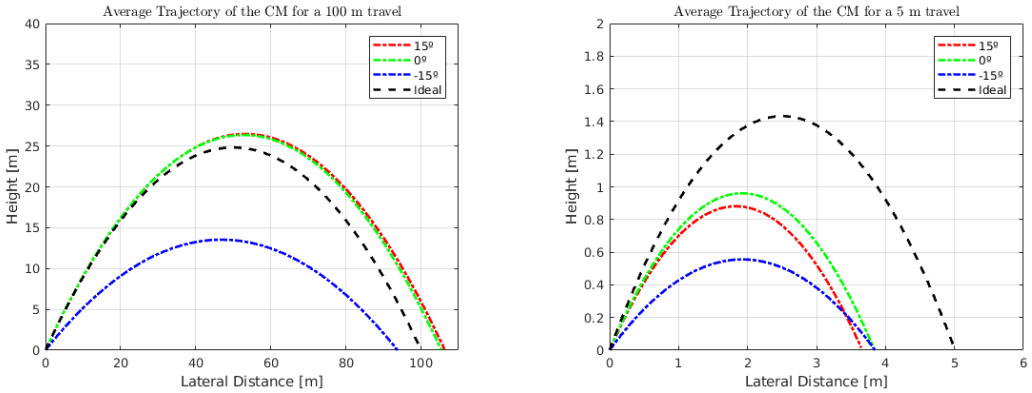


Figure 6.7: Average trajectories for 100 m and 5 m courses.

6.1.4 Consecutive Jumps

Test 14. Big distances: 385 m

Description As it has been said in Section 5.1, greater distances than 100 m are possible, but reaching the escape velocity is a considerable risk to the *hopper*. To prevent this, a series of jumps are needed for the *hopper* to go to the desired place.

Expected Results For a $d_h = 385$ m with 5 m error tolerance, three jumps of 100 m and one jump of 85 m are needed. It is expected the the *hopper* lands within the tolerance area.

Obtained Results In Figure 6.8 are shown the trajectories of the four jumps needed for a target distance of 385 m over flat terrain. In the same way, as in the previous section, the parabolic flights are simulated from the braking

responses of the flywheel under the gravitational pull of the asteroid Itokawa. Table 6.3 presents the results of these jumps.

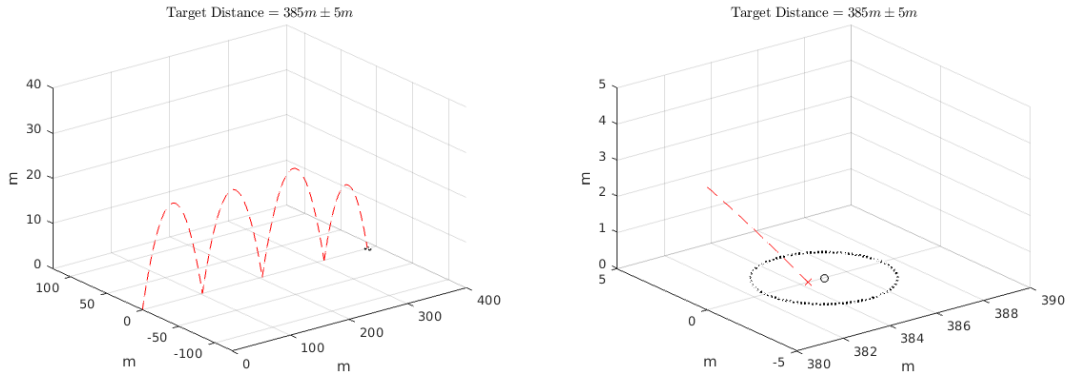


Figure 6.8: Demonstration of an ideal path to reach the objective target of 385 m with a 5 m of tolerance over flat terrain.

Target Distance $[m]$	**Resultant Distance** $[m]$	**Error** $[\%]$
100	102.41	2.41
100	102.18	2.18
100	105.67	5.67
75	73.95	1.40
385	**384.33**	0.17

Table 6.3: Simulated jumps of the *hopper*. The accumulation of the resultant distances reduced the length of the last jump.

6.2 Summary of the Results

In this part, the summarized jump results described in Sections 6.1.2 and 6.1.3 are presented. In Section 6.2.1, 6.2.2, and 6.2.3 the results are presented according to surface inclinations. In Section 6.2.4 the results are presented according to target distances.

6.2.1 Surface Inclination $\beta = 15^{o}$

From Table 6.4 it can be seen that the best response is in the 50 m range with a relative error of 3.7% and the worst is in the 5 m distance with a relative error of 27.2%.

Target Distance	Mean Distance	Std Deviation	Relative Error
5 m	3.6 m	$\pm$0.2 m	**27.2%**
10 m	8.5 m	$\pm$0.2 m	**15.5%**
30 m	35.7 m	$\pm$1.6 m	**19.1%**
50 m	51.9 m	$\pm$0.4 m	**3.7%**
70 m	73.6 m	$\pm$0.2 m	**5.1%**
100 m	106.3 m	$\pm$0.9 m	**6.2%**

Table 6.4: Summary of the results of $\beta = 15^{o}$.

6.2.2 Surface Inclination $\beta = 0^{o}$

From Table 6.5 it can be noticed that the best response is in the 50 m range with a relative error of 2.7% and the worst is in the 5 m distance with a relative error of 23%.

Target Distance	Mean Distance	Std Deviation	Relative Error
5 m	3.8 m	$\pm$0.2 m	**23.0%**
10 m	8.5 m	$\pm$0.2 m	**14.8%**
30 m	31.4 m	$\pm$1.9 m	**4.8%**
50 m	51.4 m	$\pm$0.5 m	**2.7%**
70 m	73.0 m	$\pm$0.4 m	**4.2%**
100 m	105.3 m	$\pm$1.0 m	**5.3%**

Table 6.5: Summary of the results of $\beta = 0^{o}$.

6.2.3 Surface Inclination $\beta = -15^{o}$

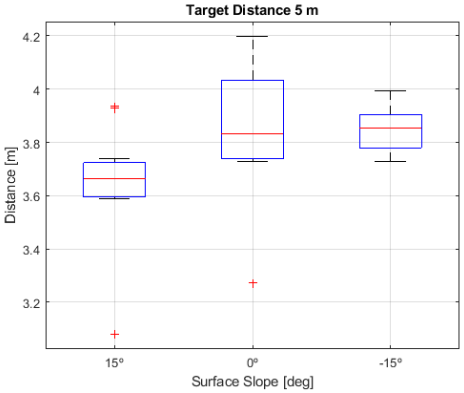
From Table 6.6 it can be noticed that the best response is in the 50 m range with a relative error of 0.8% and the worst is in the 5 m distance with a relative error of 23.1%.

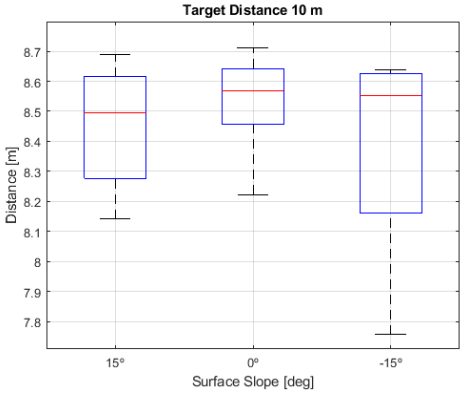
Target Distance	Mean Distance	Std Deviation	Relative Error
5 m	3.9 m	$\pm$0.1 m	**23.1%**
10 m	8.4 m	$\pm$0.3 m	**16.1%**
30 m	34.1 m	$\pm$2.2 m	**13.7%**
50 m	49.6 m	$\pm$1.2 m	**0.8%**
70 m	66 m	$\pm$0.7 m	**5.7%**
100 m	93.8 m	$\pm$1.4 m	**6.2%**

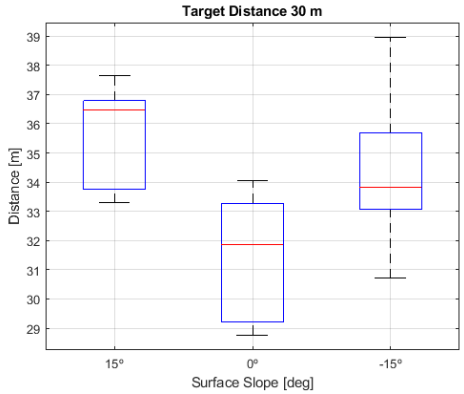
Table 6.6: Summary of the results of $\beta = -15^{o}$.

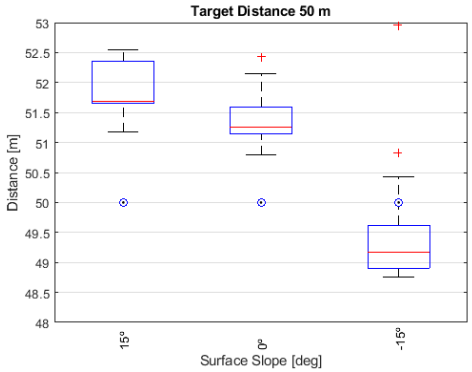
6.2.4 Target Distances 5 m, 10 m, 30 m, 50 m, 70 m, 100 m

Within each box, the red line marks the median, the bottom and top edges of the box indicates the percentile 25 and 75, respectively. The outliers are represented by (+). All the plots show the surface inclinations of 15°, 0°, and -15° in the horizontal axis. In the vertical axis, the set of distances obtained through each jump are displayed. Table 6.7 summarizes these datasets.









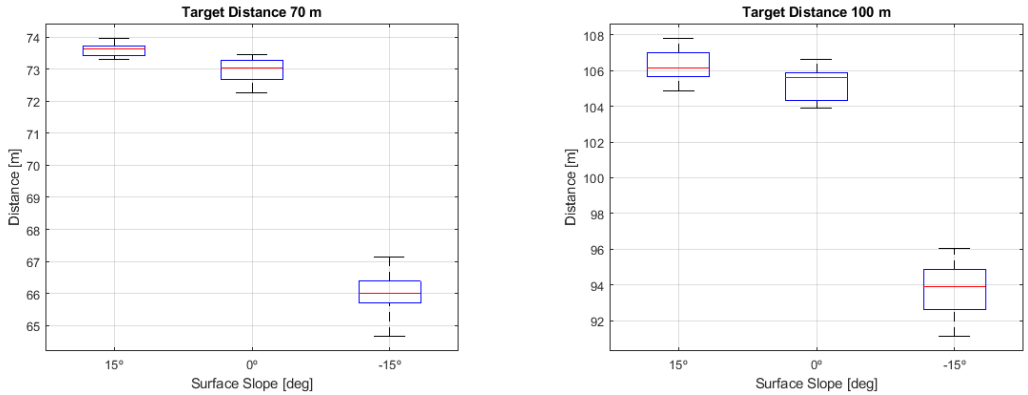


Figure 6.9: Box plot representation by target distance.

Target Distance	Slope	Median Distance	Max Distance	Min Distance
5 m	15°	3.7 m	3.9 m	3.1 m
	0°	3.8 m	4.2 m	3.3 m
	-15°	3.9 m	4.0 m	3.7 m
10 m	15°	8.5 m	8.7 m	8.1 m
	0°	8.6 m	8.7 m	8.2 m
	-15°	8.5 m	8.6 m	7.8 m
30 m	15°	36.5 m	37.6 m	33.3 m
	0°	31.9 m	34.0 m	28.8 m
	-15°	33.8 m	39.0 m	30.7 m
50 m	15°	51.7 m	52.5 m	51.2 m
	0°	51.3 m	52.4 m	50.8 m
	-15°	49.2 m	53.0 m	48.7 m
70 m	15°	73.6 m	74.0 m	73.3 m
	0°	73.0 m	73.5 m	72.3 m
	-15°	66.0 m	67.1 m	64.7 m
100 m	15°	106.1 m	107.8 m	104.9 m
	0°	105.6 m	106.6 m	103.9 m
	-15°	93.9 m	96.0 m	91.1 m

Table 6.7: Summary of the results according to target distances.

Chapter 7

Concluding Remarks

THIS section summarizes the main contributions of this research as well as the possible future related research.

7.1 Overview

A new kind of space robots with locomotion based on jumps developed in this research provides a new way for the exploration of small bodies of our Solar System, taking advantage of the low gravity of these interplanetary objects.

This robot has a cubic small shape with no external actuators. The necessary momentum for executing a jump is provided by internal inertial wheels. This approach consist in three basic steps: 1) develop the mathematical model of the dynamics of the robot; 2) implement a speed controller for the inertial wheels in a prototype of the robot; and 3) with the results of the experiments in the prototype simulate the trajectory of the robot over some asteroid of interest.

This research focuses on the speed control of one inertial wheel, precisely on the braking of one single flywheel and its relationship with the parabolic movement of the *hopper*.

7.2 Conclusions

Considering the examples given in Section 6.1.1 it is possible to observe the behavior of the braking responses.

- Due to the inertia of the flywheel used in the prototype of this research, stopping angular speeds higher than 350 rad/s in a short braking time is difficult to achieve, but it is still a good trade-off.

- The response of the DC motor achieved the expected behavior provided that the angular speeds is in between of 150 rad/s and 350 rad/s.
- For angular speeds lower than 150 rad/s, the objective angular speed is not reached for the DC motor and its response does not fit the expected results causing the biggest relative errors.

The jump distance results from the response of the DC motor. Section 6.2 indicates that large courses $d_h > 30\ m$ have less than 10% of error, but lower distances have greater errors; however, this approach of locomotion has the advantage that if the rover does not reach the indicated place in the first jump, it can perform a series of jumps to reach a given location.

7.3 Contributions

1. The main contribution has been to control the inertia of the flywheels in such a way that the launch angle of the jump can be deflected in an effective way to reach a given location regardless the surface inclination on which the robot is located. It is worth to mention that in related works, the braking of the flywheel is instantaneous thus giving always the same launch angle.
2. A prototype of the *hopper* was built to verify the efficiency of the proposed speed controller of the flywheel.
3. Computer simulations were developed that emulate the parabolic jumps of the rover under milli-microgravity environments according to the real responses of the braking of the inertial wheel.

7.4 Future Work

For future work, it will be necessary to extend the analysis to the 3 degrees of freedom of the *hopper*. Traditionally the DC motors are considered as LTI models, but in practice, controlling high and low velocities could be a nonlinear problem, for this reason, develop a system capable of controlling nonlinear systems is important. Also, it will be essential to study the bounce over different kind of soils and consider irregular gravity fields.

Bibliography

[1] Space Study Board of the National Research Council, "Vision and Voyages for Planetary Science in the Decade 2013-2022," National Academy of Sciences, 2011. [Online]. Available: https://solarsystem.nasa.gov/docs/131171.pdf

[2] J. C. Castillo-Rogez, M. Pavone, J. A. Hoffman, and I. A. Nesnas, "Expected science return of spatially-extended in-situ exploration at small solar system bodies," in *Aerospace Conference, 2012 IEEE*. IEEE, 2012, pp. 1–15.

[3] "Solar Systems Missions," National Aeronautics and Space Administration, 2015. [Online]. Available: https://www.nasa.gov/content/solar-missions.list

[4] "Asteroid Redirect Mission," National Aeronautics and Space Administration, 2018. [Online]. Available: https://www.nasa.gov/mission_pages/asteroids/initiative/index.html

[5] A. F. Cheng, J. Atchison, B. Kantsiper, A. S. Rivkin, A. Stickle, C. Reed, A. Galvez, I. Carnelli, P. Michel, and S. Ulamec, "Asteroid impact and deflection assessment mission," *Acta Astronautica*, vol. 115, pp. 262–269, 2015.

[6] "NASA Technology Roadmaps TA 4: Robotics and Autonomous Systems," National Aeronautics and Space Administration, 2015. [Online]. Available: https://www.nasa.gov/offices/oct/home/roadmaps/index.html

[7] B. C. Clark, J. M. Sunshine, M. F. A'Hearn, A. L. Cochran, T. L. Farnham, W. M. Harris, T. J. McCoy, and J. Veverka, "Comet Hopper: A Mission Concept for Exploring The Heterogeneity of comets," National Aeronautics and Space Administration, Lockheed Martin, Univ. of Maryland, College Park, Univ. of Texas, Austin, Univ. of California, Davis, Smithsonian Institution, Cornell Univ., Ithaca, 2008. [Online]. Available: http://www.lpi.usra.edu/meetings/acm2008/pdf/8131.pdf

[8] B. H. Wilcox and R. M. Jones, "The MUSES-CN nanorover mission and related technology," in *Aerospace Conference Proceedings, 2000 IEEE*, vol. 7. IEEE, 2000, pp. 287–295.

[9] D. Helmick, B. Douillard, and M. Bajracharya, "Small body surface mobility with a limbed robot," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on.* IEEE, 2014, pp. 2341–2348.

[10] A. Parness, N. Abcouwer, C. Fuller, N. Wiltsie, J. Nash, and B. Kennedy, "Lemur 3: A limbed climbing robot for extreme terrain mobility in space," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on.* IEEE, 2017, pp. 5467–5473.

[11] C. Dietze, F. Herrmann, S. Kuß, C. Lange, M. Scharringhausen, L. Witte, T. van Zoest, and H. Yano, "Landing and mobility concept for the small asteroid lander MASCOT on asteroid 1999 JU3," in *International Astronautical Congress*, 2010.

[12] F. Herrmann, S. Kuß, and B. Schäfer, "Mobility challenges and possible solutions for low-gravity planetary body exploration," in *11th Symposium on Advanced Space Technologies in Robotics and Automation: ASTRA 2011*, April 2011. [Online]. Available: http://elib.dlr.de/69905/

[13] R. Allen, M. Pavone, C. McQuin, I. A. Nesnas, J. C. Castillo-Rogez, T.-N. Nguyen, and J. A. Hoffman, "Internally-actuated rovers for all-access surface mobility: Theory and experimentation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 5481–5488.

[14] B. J. Hockman, A. Frick, R. G. Reid, I. A. Nesnas, and M. Pavone, "Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies," *Journal of Field Robotics*, vol. 34, no. 1, pp. 5–24, 2017.

[15] ——, "Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies," in *Conference on Field and Service Robotics*, 2015.

[16] "NASA Goddard Managed Comet Hopper Mission Selected for Further Study," National Aeronautics and Space Administration, 2011. [Online]. Available: https://www.nasa.gov/centers/goddard/news/releases/2011/11-033.html

[17] "Robotic Exploration of Mars," European Space Agency, 2017. [Online]. Available: http://exploration.esa.int/mars/

[18] "Google Lunar XPRIZE," ©XPRIZE Foundation, 2017. [Online]. Available: https://lunar.xprize.org/

[19] "NASA Cancels Miniature Rover for Joint Japan-U.S. Asteroid Mission," National Aeronautics and Space Administration, 2000. [Online]. Available: https://www.jpl.nasa.gov/news/news.php?feature=4864

[20] H. Jiang, E. W. Hawkes, C. Fuller, M. A. Estrada, S. A. Suresh, N. Abcouwer, A. K. Han, S. Wang, C. J. Ploch, A. Parness *et al.*, "A robotic device using gecko-inspired adhesives can grasp and manipulate large objects in microgravity," *Sci. Robot.*, vol. 2, p. eaan4545, 2017.

[21] A. T. Asbeck, S. Kim, A. McClung, A. Parness, and M. R. Cutkosky, "Climbing walls with microspines," in *IEEE ICRA*, 2006.

[22] M. Muehlebach and R. D'Andrea, "Nonlinear Analysis and Control of a Reaction-Wheel-Based 3-D Inverted Pendulum," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 235–246, 2017.

[23] "At Asteroid Ryugu, Japan's Hayabusa 2 Spacecraft Preps for Exploration," ©Scientific American, Springer Nature America, Inc., NY, US, 2018. [Online]. Available: https://blogs.scientificamerican.com/observations/at-asteroid-ryugu-japans-hayabusa-2-spacecraft-preps-for-exploration/?WT.mc_id=send-to-friend

[24] H. Kato, N. Tanishima, K. Yanagase, T. Tsumaki, and S. Mitani, "Distance control of rocket-propelled miniature exploration robot," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 6397–6403.

[25] F. Haugen, *Dynamic Systems: Modeling, Analysis and Simulation*, 1st ed. Tapir Academic, 2004.

[26] K. Ogata, *Ingenieria de Control Moderna*, 5th ed. Pearson Educación, 2010.

[27] B. Messner, D. Tilbury, R. Hill, and J. D. Taylor, "Control Tutorials for MATLAB® and Simulink® - Inverted Pendulum: System Modeling," Univ. of Michigan, Carnegie Mellon Univ., Univ. of Detroit Mercy, 2017. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling

[28] ——, "Control Tutorials for MATLAB® and Simulink® - DC Motor Speed: System Modeling," Univ. of Michigan, Carnegie Mellon Univ., Univ. of Detroit Mercy, 2017. [Online].

Available: http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling

[29] R. Morales and R. Ramírez, *Sistemas de Control Moderno. Volumen I: Sistemas de Tiempo Contínuo*, 1st ed. Editorial Digital del Tecnológico de Monterrey, 2013, vol. 1.

[30] P. C. Chau, *Process Control: A First Course with MATLAB*, 1st ed. Cambridge University Press, 2002.

[31] H. P. Hsu, *Señales y Sistemas*, 2nd ed. McGraw-Hill, 2013.

[32] B. Messner, D. Tilbury, R. Hill, and J. D. Taylor, "Control Tutorials for MATLAB® and Simulink® - Introduction: System Analysis," Univ. of Michigan, Carnegie Mellon Univ., Univ. of Detroit Mercy, 2017. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemAnalysis

[33] "Rise Time, Settling Time, and Other Step-Response Characteristics," The MathWorks, Inc., 2017. [Online]. Available: https://www.mathworks.com/help/control/ref/stepinfo.html

[34] E. W. Weisstein, "Eigenvalue," Wolfram Research, Inc., 2017. [Online]. Available: http://mathworld.wolfram.com/Eigenvalue.html

[35] ——, "Matrix Polynomial," Wolfram Research, Inc., 2017. [Online]. Available: http://mathworld.wolfram.com/MatrixPolynomial.html

[36] J. How and E. Frazzoli, "16.30 Feedback Control Systems: Lecture 9 Full-state Feedback Controller," Massachusetts Institute of Technology: MIT OpenCourseWare, 2010. [Online]. Available: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec11.pdf

[37] R. Fernández, *Análisis y Diseño de Sistemas de Control Digital*, 4th ed. McGraw-Hill Education, 2013.

[38] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Addison Wesley Longman, 1998.

[39] K. J. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design*, 1st ed. Prentice-Hall, 1984.

[40] E. W. Weisstein, "Z-Transform," Wolfram Research, Inc., 2017. [Online]. Available: http://mathworld.wolfram.com/Z-Transform.html

[41] *Control System Toolbox^{TM}Reference*, R2018a ed., The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, 2018, p. 2-200.

[42] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Space Research*, vol. 47, no. 4, pp. 663–684, 2011.

[43] "JPL Small-Body Database Browser: 25143 Itokawa (1998 SF36)," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Cal., 2003. [Online]. Available: https://ssd.jpl.nasa.gov/sbdb.cgi

[44] A. W. Harris and G. D'Abramo, "The population of near-earth asteroids," *Icarus*, vol. 257, pp. 302–312, 2015.

[45] "ARDUINO MKR1000 WIFI," ©Arduino, Turin, Italy, 2018. [Online]. Available: https://store.arduino.cc/usa/arduino-mkr1000

[46] *BTS 7960B High Current PN Half Bridge NovalithIC^{TM}*, Infineon Technologies AG, München, Germany, 2004.

[47] *Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor*, Allegro MicroSystems, Inc., Worcester, Massachusetts, U.S.A, 2007.

Appendices

Appendix A

A.1

MATLAB® algorithm to find ω_f and Δt as well as other characteristics of the parabolic flight of the *hopper*.

```
close all; clear, clc
% Parameters of the robot
g = 77e-6;
a = pi/4;
mp = 1.5;
mf = 0.076;
h = 0.1;
r = 0.03;
Ip = mp * (2 * h^2) / 12;
If = mf * r^2 / 2;
l = sqrt(2 * h^2) / 2;
n = If / (Ip + mp * l^2);
% Surface inclination
inclinacion = 0;
b = deg2rad(inclinacion);
% rpm of the flywheel from 1 until the top speed of the motor
rpm = 1:5000;
wf = rpm * pi / 30;
% Torque from 10E-3 Nm until 10E2 Nm
tau = 0.001:0.001:100;
% Operations to find the maximum distance and the required braking
% time
nn = length(wf);
dMax = zeros(1,nn);
deltaT = zeros(1,nn);
thetaH = zeros(1,nn);
for i = 1 : nn
    vh(i) = l * n * wf(i);
    thetah = a - (n * If * wf(i)^2) ./ (2 * tau) + b;
    for k = 1 : length(thetah)
        if thetah(k) < b
            thetah(k) = b;
        end
```

```
    end
    dh = vh(i)^2 * sin(2*thetah) / g;
    [dMax(i),pos] = max(dh);
    deltaT(i) = If*wf(i)/tau(pos);
    thetaH(i) = a - (n * If * wf(i)^2) / (2 * tau(pos)) + b;
end
% Code to find the acceleration of the flywheel, the speed up time
% and the fly time of the parabolic trajectory
tauMin = mp * g * l * sin(a + b);
acel = tauMin / If;
tA = wf / acel;
tV = dMax ./ (vh .* cos(thetaH));
% Table with all the obtained values
tablaCompleta = [dMax' rpm' wf' tA' deltaT' vh' rad2deg(thetaH)'...
    tV'];
% Table with distances each 10 m
contador = 5;
for i = 1 : nn
    if tablaCompleta(i,1) > contador && contador == 5
        tabla = tablaCompleta(i,:);
        contador = contador + 5;
    elseif tablaCompleta(i,1) > contador
        tabla = [tabla; tablaCompleta(i,:)];
        contador = contador + 10;
    end
end
save('tabla.mat','tabla')
```

A.2

Arduino code for controlling the speed of the DC motor through state feedback by pole placement.

```
// Variables
bool IRSensor = LOW, IRSensor1 = LOW;
int v, rpm, diffTime, i = 0, interval, stopTime;
unsigned long pulseTime = 0, pulseTime1 = 0;
unsigned long sampleTime = 0, sampleTime1 = 0;
double ref = 0.0, T_min = 2.0, u, dr1, dr2;
// Angular speed, braking time, sample time
double vel = 198.758, Bt = 0.261, Ts = 0.0005;
// State matrices and gains
double A[2][2] = {{ 0.999780133410321 , 4.893277156134714e-04 },
{ -0.784246890026860 , 0.961578374365074 }};
double B[2] = { 2.415148963049716e-07 , 8.614647027497394e-04 };
double K[2] = { -7.407483727521837e+03 , -1.361396558743178e+03 };
double G = 0.012238219193920;
double x[2], x1[2];
// Setup function
```

```
17 void setup() {
18   diffTime = Ts * 1e6;
19   dr1 = vel * Ts / T_min;
20   dr2 = vel * Ts / Bt;
21   stopTime = 6000 + Bt / Ts;
22   pinMode(3, OUTPUT);
23   pinMode(4, OUTPUT);
24   analogWriteResolution(12);
25   Serial.begin(115200);
26   while (!Serial) {};
27 }
28 // Loop function
29 void loop() {
30   speedometer();
31   sampleTime = micros();
32   if (sampleTime - sampleTime1 >= diffTime) {
33     if (i <= 4000) {
34       ref = ref + dr1;
35     } else if (i > 4000 && i <= 6000) {
36       ref = vel;
37     } else if (i > 6000 && i <= stopTime) {
38       ref = ref - dr2;
39     } else {
40       ref = 0.0;
41     }
42     i++;
43     controlSignal();
44     Serial.println(rpm);
45     sampleTime1 = sampleTime;
46   }
47 }
48 // Speedometer function
49 void speedometer() {
50   IRSensor = digitalRead(A5);
51   if (IRSensor == HIGH && IRSensor1 == LOW) {
52     pulseTime = micros();
53     interval = (pulseTime - pulseTime1);
54     rpm = 1.5e6 / interval;
55     pulseTime1 = pulseTime;
56   }
57   IRSensor1 = IRSensor;
58 }
59 // Control signal function
60 void controlSignal() {
61   u = -K[0] * x[0] - K[1] * x[1] + G * ref;
62   if (u > 24.4) {
63     u = 24.4;
64   } else if (u < -24.4) {
65     u = -24.4;
66   }
67   v = u * 4095 / 24.4;
68   x1[0] = A[0][0] * x[0] + A[0][1] * x[1] + B[0] * ref;
```

```
69    x1[1] = A[1][0] * x[0] + A[1][1] * x[1] + B[1] * ref;
70    x[0] = x1[0];
71    x[1] = x1[1];
72    if (v >= 0) {
73      analogWrite(3, v);
74      analogWrite(4, 0);
75    } else {
76      v = -v;
77      analogWrite(3, 0);
78      analogWrite(4, v);
79    }
80  }
```