



ECOLE
POLYTECHNIQUE
DE BRUXELLES



UNIVERSITÉ LIBRE DE BRUXELLES

Design and Control of a Didactic Inverted Pendulum Actuated by a Reaction Wheel

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil Biomedical à finalité Instrumentation et Biomécanique

Clara Neuburger

Directeur
Prof. Michel Kinnaert

Co-Promoteur
Laurent Catoire

Service
SAAS

Année académique
2024 - 2025

Title: Design and Control of a Didactic Inverted Pendulum Actuated by a Reaction Wheel

Author: Clara Neuburger

Master in Biomedical Engineering - Major in Instrumentation and Biomechanics

Academic year: 2024-2025

Abstract

Hands-on experience with real prototypes is fundamental in teaching control engineering, helping students understand the gap between theoretical models and their physical implementations. This master thesis supports this approach by developing a reaction wheel-actuated inverted pendulum intended for didactic purposes. The system features a vertical pendulum rod with a motor and sensors mounted on the top extremity. Torque is applied through a spinning reaction wheel, allowing the pendulum to be balanced upright by controlling its angular momentum. The prototype was designed around the motor's torque-to-weight constraint and modelled to capture friction in order to approximate the experimental behaviour. It consists of a brushless DC motor, mounted on 3D-printed structures, that spins the reaction wheel. Sensors monitor the pendulum and motor states, while an Arduino microcontroller and motor driver handle communication and control. Following the identification of key physical parameters, including the motor's transfer function, system inertias, and friction, the model was validated against experimental data. To control the pendulum's motion, two regulation approaches were investigated. First, a Linear Quadratic Regulator was implemented by selecting weighting parameters that reflect the relative importance of system states and the desired balance between performance and control effort. This method was evaluated through Nyquist-based robustness analysis and experimental testing, focusing on recovery angle, disturbance rejection, and estimated power consumption. Then, a cascade PID controller was considered since the fast dynamics of the inner loop allows for a fast disturbance rejection. The controller's robustness was evaluated using Bode diagrams and while its performance was theoretically analysed, the controller could not be tested experimentally due to hardware limitations. The final system provides a stable, compact platform for experimenting with non-linear control and supports practical learning in control system design.

Keywords: Inverted Pendulum, Control Engineering, Linear Quadratic Regulator, Cascade Control, Didactic Device

Table of Contents

Abstracts	II
Abstract	II
Table of Contents	V
List of Figures	VII
List of Tables	VIII
List of Abbreviations	IX
1 Introduction	1
1.1 Motivations	1
1.2 Context	1
1.2.1 Application in spacecrafts control	1
1.2.2 Application in exoskeleton design	1
1.3 Organisation of the document	2
2 State of the Art	3
2.1 System description	3
2.2 Parameters of the system	4
2.3 1 DOF inverted pendulum with reaction wheel	5
2.3.1 Mathematical model	6
2.3.2 Control methods	6
2.3.3 System variables, sensors and actuators	10
2.4 2 DOF inverted pendulum with reaction wheel	10
2.4.1 Mathematical model	10
2.4.2 Control methods	11
2.4.3 System variables, sensors and actuators	13
2.5 Conclusion	13
3 Mathematical model	14
3.1 Motion of the pendulum	14
3.2 Motion of the reaction wheel	16
3.3 State-space	16
4 Design of the inverted pendulum	17
4.1 Development board	17
4.2 Motor and driver	18
4.3 Sensors	21
4.3.1 Inertial Measurement Unit (IMU)	21
4.3.2 Hall effect sensors	22
4.4 3D design and printing	23
4.4.1 Pendulum choice	23
4.4.2 Motor and sensor: mounted structure	23

4.4.3	Driver and microcontroller: base structure	24
4.5	Wiring and final prototype	24
5	Parameter estimation of the inverted pendulum	28
5.1	Pendulum parameters	28
5.1.1	Pendulum inertia	28
5.1.2	Pendulum friction	29
5.2	Motor parameters	30
5.2.1	Reaction wheel inertia	30
5.2.2	Motor friction and constants	31
5.3	Verification of the model	32
6	Control of the inverted pendulum	35
6.1	Dynamic behaviour of the inverted pendulum	35
6.2	LQR control	37
6.2.1	Choice of matrix Q and scalar R	39
6.2.2	Validation of the LQR	41
6.2.3	Theoretical evaluation of the LQR	41
6.2.4	Experimental evaluation of the LQR	43
6.2.5	Motor power consumption	47
6.3	Cascade PID control	48
6.4	Discussion of control strategies	53
7	Conclusion	55
7.1	Further improvements	56
Bibliography		60
Appendices		61
A	Optimisation of motor's transfer function in current	61
B	Optimisation of motor's transfer function in speed	63
C	Experimental identification of the wheel inertia	65
D	Adaptation of parameters for LQR_2	67
E	Simulink simulation for verification	68
F	Simulink simulation of the LQR	69
G	Simulink simulation of the cascade PID	70
H	Technical drawings of the prototype	71

List of Figures

1.1	Linear inverted pendulum model for gait analysis	2
2.1	Inverted pendulum with reaction wheel	3
2.2	Interaction block diagram	4
2.3	Schematic of one-axis inverted pendulum with reaction wheel	4
2.4	Final prototype of 2D inverted pendulum from [39]	6

2.5	Cascade control block diagram	7
2.6	Cascade control block diagram for 2D reaction wheel inverted pendulum [39]	8
2.7	LQG control block diagram for 2D reaction wheel inverted pendulum [39]	9
2.8	Final prototype of 3D inverted pendulum from [7]	10
2.9	Ziegler Nichols tuning method [19]	11
2.10	PID control for pendulum's angle block diagram	12
2.11	PD+P control block diagram	12
3.1	Schematic of one-axis inverted pendulum with reaction wheel	14
4.1	Picture of the Arduino Due [3]	18
4.2	Two motor options by Maxon [25]	19
4.3	Picture of the ESCON 50/5	20
4.4	Picture of the MPU6050 [12]	22
4.5	Hall sensor feedback signals [23]	23
4.6	3D printed mounted structure	24
4.7	3D printed base structure	25
4.8	3D printed support and rectangular holder	25
4.9	Picture of the Reaction Wheen Inverted Pendulum	25
4.10	Final 3D design of the inverted pendulum with reaction wheel	26
4.11	Wiring diagram of the inverted pendulum with reaction wheel	27
5.1	Pendulum oscillation period experiment	29
5.2	Reaction Wheel's radius measurements	30
5.3	Visual estimation of parameters G and τ on Python	31
5.4	Comparison of simulation and experimental of (a) pendulum angle, (b) motor's angular speed, for a command of 1A of 1 second	33
5.5	Comparison of simulation and experimental with updated parameters of (a) pendulum angle, (b) motor's angular speed, for a command of 1A of 1 second	33
5.6	Command sent to the motor for mathematical model verification	33
6.1	Phase portrait of the non linear upright pendulum	36
6.2	Phase portrait of the pendulum in the downward position (a) non linear and (b) linearised around $(x_1, x_2) = (0, 0)$	36
6.3	LQR block diagram	39
6.4	Analysis of pendulum angle, reaction wheel speed and control command for changes in scalar R	40
6.5	Open-loop system with K_{LQR} computed from LQR design [8]	42
6.6	Nyquist plot of the open-loop system	42
6.7	Comparison of Nyquist plots	43
6.8	Comparison of pendulum angle for an initial 5° angle.	45
6.9	Comparison of motor speed for an initial 5° angle.	45
6.10	Comparison of current command sent to the motor for an initial 5° angle.	45
6.11	Experimental prototype behaviour for an initial angle of 11° with regulators LQR_1 and LQR_2	46

6.12	Experimental setup for adding a torque disturbance to the system	47
6.13	Experimental prototype response to a torque τ_d for LQR_1 and LQR_2	47
6.14	Cascade PID configuration	48
6.15	Cascade PID configuration with ESCON closed loop speed control	49
6.16	Visual estimation of parameters G and τ on Python	49
6.17	Root locus of the system transfer function of the system (G_{system}) coupled to a P controller	50
6.18	Root locus of $C \cdot G_{system}$ computed with <code>rlocus()</code> function	51
6.19	Bode plots of the system coupled to a PID	52
6.20	Bode plots of the system coupled to a PID discretised	52
6.21	Disturbance rejection comparison of continuous and discrete (Tustin) systems	53
6.22	Comparison of the system controlled by a LQR and a PID controller under the same disturbance torque	54
A.1	Parameters and cost function change over iteration	61
A.2	Transfer function identification	62
A.3	Transfer function verification	62
B.1	Parameters and cost function change over iteration	63
B.2	Transfer function identification	64
B.3	Transfer function verification	64
C.1	Schematics of experimental setup for verification of wheel inertia	65
D.1	Test of recovery angle of 5°	67
E.1	Simulink simulation for the verification of mathematical model of the pendulum with reaction wheel	68
E.2	Simulink simulation for the verification of mathematical model of the pendulum with reaction wheel, zoom on the mathematical model	68
F.1	Simulink simulation for the simulate the LQR control of the pendulum with reaction wheel	69
G.1	Simulink simulation for the cascade PID control of the pendulum with reaction wheel	70

List of Tables

2.1	Variables, symbols, and units used in the system.	5
2.2	Components comparison summary	13
4.1	Comparison between Arduino Uno and Arduino Due	17
4.2	Comparison of Motor Types [25]	18
4.3	Comparison of EC-flat motors [25]	19
4.4	Characteristics of the Escon 50/5 [23]	20
4.5	Implementation of control of the Escon 50/5 [23]	21

4.6	MPU6050 characteristics [12]	22
4.7	Analog output of the Escon 50/5 [23]	22
4.8	Properties comparison of carbon fibre, aluminium and PLA	23
4.9	Power supplies of the prototype	26
5.1	Weights and lengths of the inverted pendulum as presented in Fig.(3.1)	28
5.2	Parameter values summary	32
5.3	Parameter values adapted to the experimental model	34
6.1	Mean value of $\theta_p/\dot{\theta}_p$ varying R from different initial angles	39
6.2	Characteristics of the pendulum 5° angle recovery	44
6.3	Transfer functions of the system	49
D.1	Parameter values adapted to the experimental model	67

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
BLDC	Brushless Direct Current
CAD	Computed Aided Design
CoM	Centre of Mass
DAC	Digital-to-Analog Converter
DC	Direct Current
DOF	Degree Of Freedom
GM	Gain Margin
HJB	Hamilton-Jacobi-Bellman
IMU	Inertial Measurement Unit
LQE	Linear Quadratic Estimation
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
M3	Metric 3
MIMO	Multiple-Input Multiple-Output
PID	Proportional Integral Derivative
PLA	Polylactic Acid
PM	Phase Margin
PWM	Pulse Width Modulation
RMS	Root Mean Square
RWIP	Reaction Wheel Inverted Pendulum
SAAS	Service d'Automatique et d'Analyse des Systèmes
SRAM	Static Random Access Memory
ZMP	Zero Moment Point

Chapter 1

Introduction

1.1 Motivations

This master thesis aims to develop the design, control and implementation of a didactic device intended for teaching control engineering. The device is an inverted pendulum actuated by a reaction wheel. It is intended to help students learn how to design and test controllers for stabilizing such non-linear unstable system. Through a series of guided exercises, students will explore the system's dynamics and learn to implement effective control strategies to keep the pendulum in its upright position. Working on a tangible device allows students to learn about and encounter model inaccuracies (such as friction or saturation), sensor imperfections like noise, and sampling effects. It also offers the opportunity to verify a physical system experimentally and to better understand the connection between the system's behaviour and its control law.

An inverted pendulum is a well-established didactic device, commercially available from companies specialised in developing didactic devices for control engineering such as Quanser [31] or Acrome robotics [1]. They usually come in the form of rotary or linear inverted pendulum, where the pendulum is stabilized through movement at its base.

The reaction-wheel-actuated inverted pendulum studied in this thesis shares the same non-linear instability but introduces additional challenges: the motor is placed at the top of the pendulum rather than at the base, introducing further design constraints, particularly related to weight distribution and inertia.

1.2 Context

Beyond its pedagogical use in teaching control engineering, the reaction wheel-actuated inverted pendulum is a system used in various real-world applications like spacecraft control or exoskeleton design.

1.2.1 Application in spacecrafts control

To counter disturbances that act on spacecrafts during flights like gravitational perturbations, or solar pressure forces, reaction wheels are often used as part of the attitude control system [9]. On light orbital stations or autonomous spacecrafts like satellites, reaction wheels are used to create a control torque applied to the spacecraft body. These electrically powered reaction wheels allow for a smooth and precise control while reducing overall weight by minimizing fuel usage [9].

1.2.2 Application in exoskeleton design

Another possible application of the inverted pendulum with reaction wheel is related to the design of exoskeletons or bipedal robots. In order to replicate human walking in robots, one must first conduct gait analysis to understand the dynamics of human locomotion. While

machine learning techniques are commonly used to that end, they often lack interpretability and a clear physical relationship to the underlying model. The study conducted by Ya-Fang Ho et al. [14] presents a method based on the inverted pendulum with a reaction wheel model, offering a more intuitive approach. They compare the robot to an inverted pendulum with a Centre of Mass (CoM) constrained at a certain height y_c in Fig. (1.1). Because of the fixed height of the CoM, the pendulum rod exhibits a damping behaviour rather than purely rigid motion.

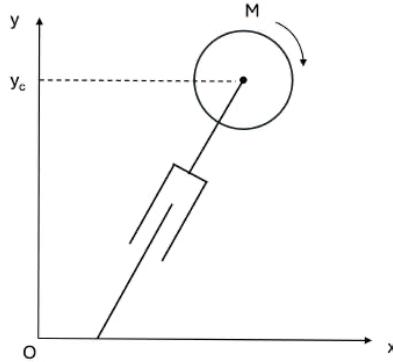


Figure 1.1: Linear inverted pendulum model for gait analysis

The Zero Moment Point (ZMP) is defined as the point on the ground where the total torque due to the weight and the reaction forces equals zero [38], in other words, where the foot of the robot is perfectly balanced. The authors present a parametrized gait generator based on the linear inverted pendulum model; by knowing and predicting the ZMP position, the robot can adjust the position of the CoM to balance the robot.

1.3 Organisation of the document

This report begins with a State of the Art review which will cover mathematical formulations, control strategies and prototypes of two inverted pendulums. It then introduces the mathematical model of the Reaction Wheel Inverted Pendulum (RWIP), followed by a description of the prototype's design. The prototype parameters are experimentally identified and verified before two control strategies are explored and contrasted.

Chapter 2

State of the Art

This chapter presents the State of the Art, beginning with an overview of the system and its key components. Two examples are then examined: one with a single Degree Of Freedom (DOF) and another with two DOFs. For each case, the mathematical formulation, control strategy, and prototyping approach are detailed, highlighting relevant challenges and solutions. Despite the dimensionality change, the approach remains unchanged; moving from one DOF to two DOFs simply involves duplicating the control laws and mechanical setup along a second axis.

2.1 System description

The system studied in this thesis is called an inverted pendulum with reaction wheel. It consists of a rigid pendulum rod placed on a surface which will be stabilized in the upright position using a motor coupled to a wheel placed on top of the pendulum, as represented in Fig. (2.1).



Figure 2.1: Inverted pendulum with reaction wheel

The actuator takes the form of a motor and generates controlled motion based on sensor feedback. These sensors measure angular speed and position of the pendulum and motor, providing real-time data for the control system. The interaction between these components is illustrated in Fig.(2.2).

The goal of this project is to build such a system by selecting sensors and actuators, designing signal conditioning and acquisition stages, and creating the experimental setup using tools like **Fusion360** and a 3D printer. This also involves selecting an appropriate power supply, cable management, modelling the system and implementing a control strategy.

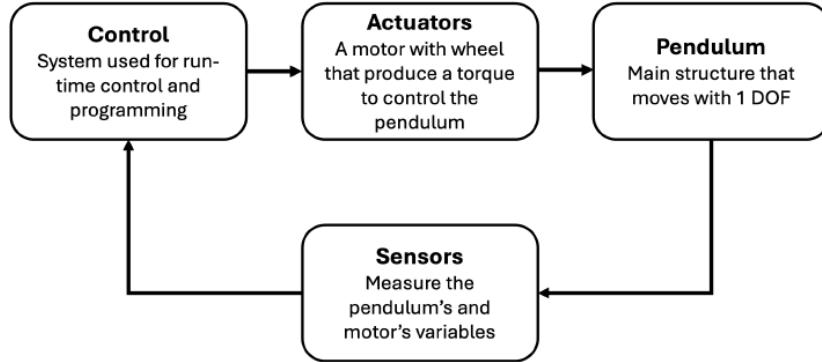


Figure 2.2: Interaction block diagram

This system is stabilized by the torque produced by the motor placed on top of the pendulum. Newton's Second Law of Motion ($\sum \tau = I\alpha$) states that the net torque τ acting on a rotating system is equal to the product of inertia I and the angular acceleration α [21]. In this system, when the pendulum experiences a disturbance that causes it to deviate from its upright equilibrium position, the motor generates a control torque called τ_c . This torque applies a restoring angular acceleration, counteracting the disturbance. The generated torque continuously adjusts the pendulum's orientation, opposing the forces pulling it away from equilibrium.

2.2 Parameters of the system

All variables used in the mathematical description of the model are described in Tab.(2.1). Any term related to the rod will be referred to with the mention "p" for pendulum. Terms that describe characteristics of the reaction wheel will be followed by a "w". The coordinate systems of origin O , O_p and O_w are presented in Fig.(2.3).

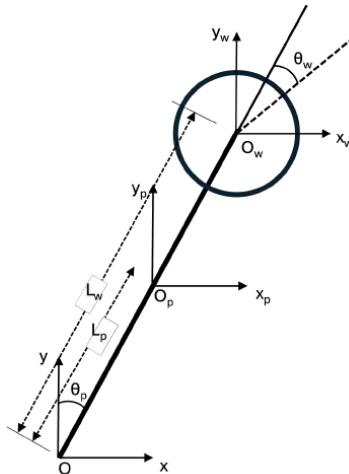


Figure 2.3: Schematic of one-axis inverted pendulum with reaction wheel

Table 2.1: Variables, symbols, and units used in the system.

Description	Symbol	Unit
Motor viscous friction coefficient	b_w	Nms
Pendulum friction coefficient	b_p	Nms
Distance from pivot point to pendulum's centre of gravity	d_M	m
Disturbance torque added to the system	τ_d	Nm
EMF constant of the motor	K_e	Vs
Torque constant of the motor	K_m	Nm/A
Gravitational constant	g	Nm ² /kg ²
Current flowing through the motor	i	A
Inertia of the system with respect to O	I	kg · m ²
Inertia of motor with respect to O	I_m	kg · m ²
Inertia of pendulum with respect to O	I_p	kg · m ²
Inertia of reaction wheel with respect to O_w	I_w	kg · m ²
Parallel combination of I and I_w	\hat{I}	kg · m ²
Coil inductance of the motor	L	H
Distance between origin O and centre of gravity of motor	L_m	m
Distance between origin O and centre of gravity of pendulum	L_p	m
Distance between origin O and centre of gravity of reaction wheel	L_w	m
Mass of motor	M_m	kg
Mass of pendulum	M_p	kg
Mass of reaction wheel	M_w	kg
Centre of gravity of pendulum	O_p	—
Centre of gravity of reaction wheel	O_w	—
Origin of coordinate system (x,y)	O	—
Coil resistance of the motor	R	Ω
Control torque produced by the motor	τ_c	Nm
Voltage command sent to the motor	u	V
Angle between pendulum and vertical axis	θ_p	° (degrees)
Angle of reaction wheel	θ_w	° (degrees)

2.3 1 DOF inverted pendulum with reaction wheel

The first version to analyse is a 1 DOF inverted pendulum developed by Dominik Zaborniak and Krzysztof Patan published in *Electronics*, for a special issue 'State-of-the-Art Research in Systems and Control Engineering' in 2024. The paper describes the mathematical model, the development of the physical prototype and the control methods implemented [39]. The final prototype can be seen in Fig.(2.4)



Figure 2.4: Final prototype of 2D inverted pendulum from [39]

2.3.1 Mathematical model

The author of the article chose a thorough and comprehensive mathematical description of the model, considering friction from the different components, additional external torques that could be applied to the system and the motor's electromagnetic properties. By summing the torques for each part of the system the author derived the following system that describes the angular acceleration of the pendulum ($\ddot{\theta}_p$) and the reaction wheel ($\ddot{\theta}_w$):

$$\begin{cases} \ddot{\theta}_p = \frac{M_{tot}d_M g}{I} \theta_p - \frac{b_p}{I} \dot{\theta}_p + \frac{b_p}{I} \left(b_w + \frac{K_m K_e}{R} \right) \dot{\theta}_w + \frac{1}{I} \tau_d - \frac{1}{I} \frac{K_m}{R_a} u, \\ \ddot{\theta}_w = -\frac{M_{tot}d_M g}{I} \theta_p + \frac{b_p}{I} \dot{\theta}_p - \frac{I + I_w}{II_w} \left(b_w + \frac{K_m K_e}{R} \right) \dot{\theta}_w - \frac{1}{I} \tau_d + \frac{I + I_w}{II_w} \frac{K_m}{R} u. \end{cases} \quad (2.1)$$

With,

$$M_{tot} = M_p + M_w + M_m \quad (2.2)$$

This system is non-linear due to the presence of $\sin(\theta)$. By choosing the operating point $\theta = 0$, which corresponds to the equilibrium position that the system has to reach, the system can be made linear. The rest of the analysis is carried out in state-space format:

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx. \end{cases} \quad (2.3)$$

Where A,B and C are constant matrices and x presents the states of the system $[\theta_p, \dot{\theta}_p, \theta_w, \dot{\theta}_w]^T$

2.3.2 Control methods

The article introduces two control methods: a cascade PID and an LQG regulator, which will be detailed in the following sections.

Cascade PID control

PIDs are one of the most widely used controllers in industrial processes [19]. Initially developed using a trial-and-error approach, this control system consists of three components:

- **P:** Proportional action based on the error.

- **I:** Integrating action that reduces steady-state error but may degrade transient response.
- **D:** Derivative action that improves transient response through anticipation.

The following transfer function can be associated to this regulator:

$$D_{cl}(s) = \frac{U(s)}{E(s)} = k_P + \frac{k_I}{s} + \frac{k_D s}{sT_f + 1} \quad (2.4)$$

Where $U(s)$ is the command and $E(s)$ the error. In a real system, noise is unavoidable on a sensor. Due to the derivative term, high-frequency noise, even at low levels, is amplified, potentially distorting the original signal and affecting system performance. The solution to this issue is to add a low-pass filter that attenuates frequencies above a certain threshold. The denominator $sT_f + 1$ found in equation (2.4) accounts for such low pass filter and its value is to be determined with the rest of the gains k_P , k_I and k_D during the tuning process.

Although other articles used simple PID regulators working on the pendulum's angle to control their system [35] [22], it is often insufficient on its own as it cannot manage to properly stabilize the pendulum. This challenge can be addressed by implementing a second PID controller on another variable of the system, such as the reaction's wheel speed or position depending on the sensors. However, a study conducted by Fuat Peker in 2018 on cart inverted pendulums [30], which present very similar dynamics and challenges, recommended drawing inspiration from the system's inherent cascade structure rather than implementing two separate controllers and combining them afterward. A cascade PID structure is known to improve the reduction of possible disturbances [37]. An additional sensor allows to separate fast and slow dynamics by creating a nested loop configuration.

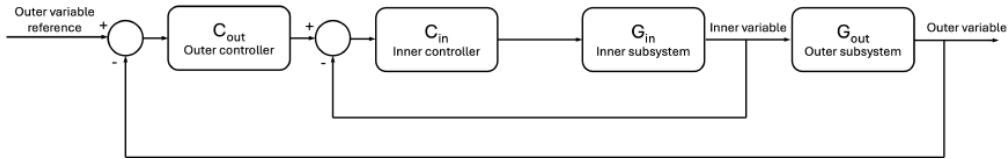


Figure 2.5: Cascade control block diagram

The system is divided into two subsystems as demonstrated in Fig.(2.5): an inner subsystem controlled by an inner controller and an outer system controlled by the outer controller. The advantage of this design lies in the fast dynamics of the inner loop, which allows a faster disturbance attenuation and minimizes the possible effect disturbances before they affect the outer variable [37]. Tuning a cascade PID is more complex than a standard single-loop PID, as it involves first tuning the inner controller (C_{in}), followed by tuning the outer controller (C_{out}) while accounting for the inner controller's influence on the inner loop.

For the reaction wheel inverted pendulum, the aim of the inner loop is to stabilize the pendulum's angle θ_p as quickly as possible. The outer loop controller stabilizes the reaction wheel position θ_w by providing the reference signal $\theta_{p,ref}$ to the inner control loop. The block diagram of the control scheme is illustrated in Fig.(2.6).

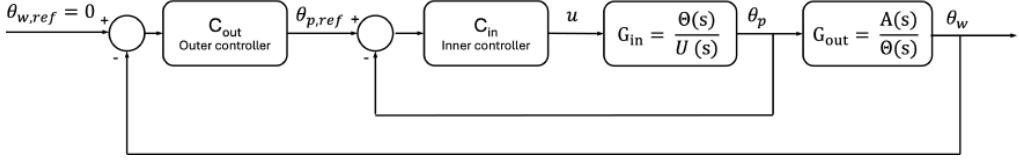


Figure 2.6: Cascade control block diagram for 2D reaction wheel inverted pendulum [39]

The inner subsystem can be represented by the transfer function between the pendulum angle θ and the voltage command u that is set by the inner controller to regulate the system:

$$G_{in} = \frac{\Theta_p(s)}{U(s)} = \frac{a_1 s}{b_0 + b_1 s + b_2 s^2 + b_3 s^3} \quad (2.5)$$

With a_1, b_0, b_1, b_2 and b_3 integers expressed by products and sums of the parameters present in Equation (2.1).

$$\begin{cases} a_1 = -I_w K_m \\ b_0 = -M_{tot} d_M g (R b_w + K_e K_m) \\ b_1 = K_e K_m b_p + R b_w b_p - I_w M_{tot} d_M g \\ b_2 = (I + I_w) (R b_w + K_e K_m) + I_w R b_p \\ b_3 = I I_w R \end{cases} \quad (2.6)$$

The outer loop subsystem transfer function is the ratio between the reaction wheel position θ_w and the inner subsystem variable, the pendulum angle θ_p .

$$G_{out} = \frac{\Theta_w(s)}{\Theta_p(s)} = \frac{a_0 + a_1 s + a_2 s^2}{b_2 s^2} \quad (2.7)$$

With $a_0, a_1, a_2, b_1, b_2, b_3$, and b_4 integers expressed by products and sums of the parameters present in Equation (2.1).

$$\begin{cases} a_0 = M_{tot} d_M g \\ a_1 = -b_p \\ a_2 = -(I + I_w) \\ b_2 = I_w \end{cases} \quad (2.8)$$

The inner loop PID is tuned by analysing the root locus of the subsystem. The root locus is a representation of how the poles of a system move in a complex plane as the gain of the system varies. For a system made of a regulator $D(s)$, a plant $G(s)$ and a sensor $H(s)$, the closed loop of this system for a reference $R(s)$ and an output $Y(s)$ is as follows [19]:

$$\frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)H(s)} \quad (2.9)$$

The characteristic equation of the closed loop, whose roots are the poles of the transfer function is:

$$\begin{aligned} 1 + D(s)G(s)H(s) &= 0 \\ 1 + KL(s) &= 0 \end{aligned} \quad (2.10)$$

The root locus consists of plotting the locus of all possible roots by varying K from 0 to

infinity. The `rltool()` function in Matlab allows users to shape the root locus by adding or removing poles and zeros, helping design a controller that places the system's poles in the left half-plane to ensure stability.

Once the inner loop controller is designed, the loop is closed and treated as a new plant in series with the outer loop. The outer loop controller can then be tuned using the same method.

Linear-Quadratic-Gaussian Control

The second method that is considered in the article [39] is a Linear Quadratic Regulator with a Kalman filter, also called Linear Quadratic Gaussian Controller.

A Linear Quadratic Regulator (LQR) is a control method similar to pole placement but instead of placing the closed-loop poles at suitable locations, the goal is to find the optimal state feedback gain K by choosing characteristics balance between the performance and the effort. To do so, a cost function J is set up that will set the importance of each criterion:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (2.11)$$

By minimizing the cost function J , one can find the optimal control law $u = -Kx$ which will be fed to stabilise the system. Matrix Q from Equation (2.11) accounts for the performance of the system while matrix R accounts for the effort provided by the actuator. For example in satellites in which the actuators are really expensive due to the fuel's price, one might penalize them by setting very high coefficients in the R matrix [10]. The main advantage of LQR compared to the usual pole placement technique is that the choice of coefficients in the matrices R and Q are much more intuitive than placing poles in a complex plan. Such design method is very suitable for dealing with Multiple-Input Multiple-Output (MIMO) system such as this one by being able to handle dynamic coupling between state variables [39].

In addition to the LQR, a Linear Quadratic Estimation (LQE) also called Kalman filter is added to the regulation system. This filter will estimate the states of the system based on noisy and incomplete measures and the state-space model [6]. The Kalman filter not only corrects errors from sensors but also from the model. It can predict the system's state and assess the average error of its estimation. The control system shown in Fig.(2.7) is obtained.

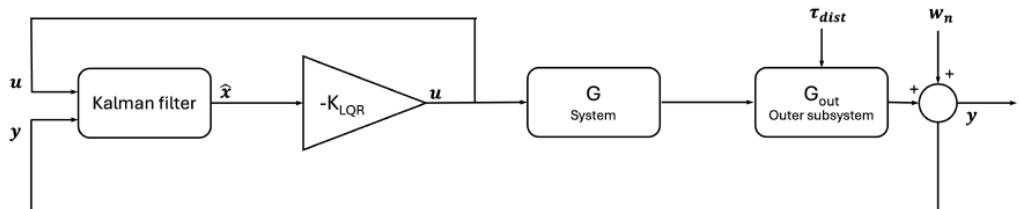


Figure 2.7: LQG control block diagram for 2D reaction wheel inverted pendulum [39]

With such a configuration, the state-space expression of the controlled system becomes:

$$\begin{cases} \dot{x} = Ax + Bu + \tau_{dist} \\ y = Cx + w_n \end{cases} \quad (2.12)$$

Where τ_{dist} is a disturbance vector acting on the pendulum angle, defined as $\tau_{\text{dist}} = [\tau_d, 0, 0, 0]^T$ and w_n is measurement noise.

In the article [39], after discretizing the system, the Kalman filter gain matrix is computed, followed by the estimation of the feedback gain K_{LQR} .

2.3.3 System variables, sensors and actuators

An image of the prototype is presented in Fig.(2.4) at the start of this section. The system has a single actuator to stabilise the pendulum: a gear DC motor Pololu 4752 that weighs around 200g and can produce high torques thanks to the gear system. An H-bridge is used to control the DC motor by reversing its polarity and regulating speed through Pulse Width Modulation (PWM). The configuration also contains three sensors: a rotary encoder (E38S6-C-600B5-26G2) to measure the pendulum's angle θ_p , an encoder inside the motor to measure the motor's angle θ_w and finally an Inertial Measurement Unit (IMU LSM6DS33) to measure the pendulum's angular speed $\dot{\theta}_p$. All of these variables are then retrieved with a STM32 Nucleo-F103RB board that acts as a digital controller. The prototype is mainly made out of wood but some pieces such as the pendulum or the motor holder are 3D printed in Polylactic Acid (PLA).

The most important features of this design are worth highlighting. First, the choice of a relatively heavy motor ensures sufficient torque to effectively stabilize the system, even though it adds to the overall weight. Second, the pendulum's height, while not explicitly stated in the report, is relatively small compared to the size of the reaction wheel. The design choice of a shorter pendulum likely plays a role in keeping the system compact. Finally, regulation relies solely on position, as a design choice was made to exclude motor speed measurement.

2.4 2 DOF inverted pendulum with reaction wheel

The second paper analysed in this literature review is a prototype developed by Petter Brevik in 2017. The report describes the mathematical model, the development of the physical prototype, and the control methods implemented [7].

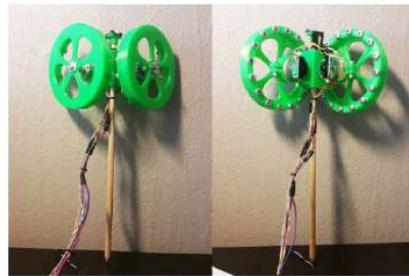


Figure 2.8: Final prototype of 3D inverted pendulum from [7]

2.4.1 Mathematical model

Unlike the mathematical description of the model presented earlier, this article introduces a different computational approach. The model here is much more simplified: it does not

account for external torque, neglects component friction, and also simplifies the operational behaviour of the DC motor. Instead of summing the torques as in [39], the equations of motion are derived here using the Euler-Lagrange method. The system is analysed in 2D as the mathematical model and control laws can just be replicated in the second plane. This method consists of firstly computing the Lagrange operator \mathcal{L} for the system which can be expressed as the difference between kinetic and potential energy.

$$\mathcal{L} = KE - PE \quad (2.13)$$

Then, the operator can be replaced in Equation (2.14) with q_i the generalized coordinates and τ_i the total torque of the coordinates [36].

$$\tau_i = \frac{d}{dt} \left(\frac{\delta \mathcal{L}}{\delta \dot{q}_i} \right) - \frac{\delta \mathcal{L}}{\delta q_i} \quad (2.14)$$

By doing such, the following system is obtained, with θ_p the pendulum angle and θ_w the reaction wheel angle:

$$\begin{cases} \ddot{\theta}_p = \frac{\sin(\theta_p)(m_p L_p^2 + m_w L_w^2)g - K_m u}{m_p L_p^2 + m_w L_w^2 + I_p} \\ \ddot{\theta}_w = \frac{K_m u}{I_w} - \frac{\sin(\theta_p)(m_p L_p^2 + m_w L_w^2)g - K_m u}{m_p L_p^2 + m_w L_w^2 + I_p} \end{cases} \quad (2.15)$$

Similarly to the 1 DOF pendulum's Equation (2.1), the system is non linear and needs to be linearised around the operating point $\theta_p = 0$

2.4.2 Control methods

The article presents two control methods: a simple PID and a dual PID controller. This time, the control is not based on the motor's position but on its speed.

Simple PID controller

The author initially began by tuning the system with a simple proportional (P) regulator applied to the pendulum position's error, using the Ziegler-Nichols method. This tuning approach, designed for P, PI, and PID controllers, involves setting the system in a specific configuration, as shown in Fig. (2.9).

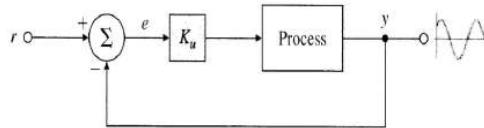


Figure 2.9: Ziegler Nichols tuning method [19]

By first setting the integral and differential gains to 0, the method starts by decreasing the gain K_u (ultimate gain) until the system exhibits stable oscillations. The oscillation period T_u is then measured and these parameters are used to find the coefficients for the

PID controller: [19].

$$\begin{cases} k_P = 0.6K_u \\ k_I = 0.5T_u \\ k_D = 0.125T_u \end{cases} \quad (2.16)$$

However, this tuning method appears questionable, as an inverted pendulum is an unstable system that cannot naturally reach a state of oscillation at the ultimate gain. The author did not provide further details on how the method was implemented. Therefore, this tuning approach will not be considered for the system proposed in this report.

The implemented controlled system block diagram is represented in Fig.(2.10).

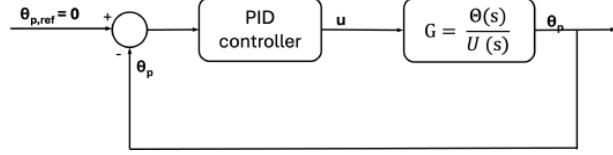


Figure 2.10: PID control for pendulum's angle block diagram

The transfer function between the pendulum's angle θ and the command u is given by the following equation:

$$G = \frac{\Theta}{U} = \frac{a_0}{b_0 + b_2 s^2} \quad (2.17)$$

With a_0 , b_0 , and b_2 integers expressed by products and sums of the parameters present in Equation (2.15).

$$\begin{cases} a_0 = -K_m \\ b_0 = -g(m_p L_p + m_w L_w) \\ b_2 = m_p L_p^2 + m_w L_w^2 + I_p \end{cases} \quad (2.18)$$

Despite this effort, it was observed that the resulting PID controller could not simultaneously stabilize the pendulum and ensure the motor speed returned to zero.

PD + P controller

The second method involved combining a PD controller for the pendulum's angle with a P controller for the reaction wheel's speed. To ensure the reaction wheel's speed converged to zero, the integral term from the original PID controller was removed. The block diagram of such configuration is represented in Fig.(2.11).

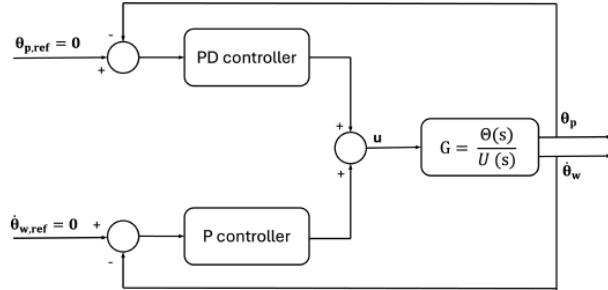


Figure 2.11: PD+P control block diagram

This configuration successfully handled disturbances up to 9°, even when followed by an additional impulse force on the pendulum's body.

2.4.3 System variables, sensors and actuators

For this project, the Mabuchi RS-385PH-16140 motor (brushed DC motor) is chosen. It is a very light motor (82g) which can provide a torque of maximum 9.18mNm. The motor works with a simple driver made of an H-bridge which is linked to an Arduino Uno for control. The author however noticed a voltage mismatch between their H-bridge L298P (rated up to 24V) and the Arduino (rated up to 18V), concluding that the motor would not be performing at its best performance. The setup contains 2 sensors: a single-channel incremental encoders attached to the motor and an Inertial Measurement Unit (IMU) attached at the top of the pendulum which is a 3-axis accelerometer and gyroscope able to measure the pendulum angle and angular velocity. The pendulum measures 30cm and is made out of wood while the other components such as the reaction wheels and the structure to hold the motor and sensors together are 3D printed in PLA.

2.5 Conclusion

A table that summarizes the different characteristics of the prototypes presented previously is available in Tab.(2.2).

Table 2.2: Components comparison summary

Characteristic	2D pendulum	3D pendulum
Motor maximum torque:	0,1765197 N·m	0,00918 N·m
Motor weight:	200g	82g
Pendulum height:	15cm	30cm
Microcontroller:	STM32 Nucleo-F103RB	Arduino Uno
Sensors:	Rotary encoder and IMU	Single Channel encoder and IMU

The main difference between the two motor options is their trade-off between weight and power: one being lightweight with lower torque, and the other offering higher power but increased weight. The optimal choice would be a motor that combines both lightweight construction and high torque performance. By studying the components, a list of criteria for the prototype can be established:

- Motor should have the highest torque constant possible and weigh a maximum of 150g.
- Pendulum's height should be adjustable.
- The structure holding the sensors and motors to the pendulum should be 3D printed and easily removable.

The rest of the criteria and the choice of the components will be described in section 4.

Chapter 3

Mathematical model

3.1 Motion of the pendulum

The model described is represented in Fig.(3.1). As seen in the literature, such systems are typically analysed using either Lagrangian mechanics or the Newton-Euler approach.

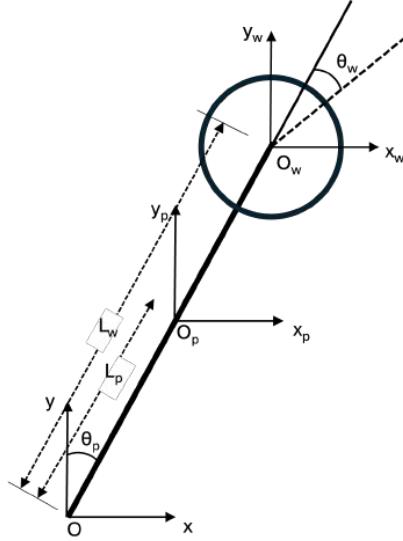


Figure 3.1: Schematic of one-axis inverted pendulum with reaction wheel

The Newton-Euler method is based on a computationally efficient recursive technique that expresses the system as a serial structure but the resulting mathematical model is not expressed in closed form [15]. On the other hand, it is much simpler and intuitive to apply the Lagrange method and it is also more suitable to understand the effects of changes in mechanical parameters. However, since the model is obtained by first computing kinetic and potential energies, it is not computationally efficient. The Lagrange method was chosen because the simplicity of the physical system makes the energy computations straightforward, resulting in a more intuitive modelling approach.

The Lagrange equation is described by Equation (3.1) below with q_i as the generalized coordinates and ξ_i as the generalised force associated with the generalised coordinates q_i [36]. The contributions to the generalised forces are given by the non-conservative forces such as the actuator torques τ_i , the joint viscous friction or external forces.

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \quad (3.1)$$

The Lagrange operator \mathcal{L} can be expressed in terms of the kinetic energy T and potential energy U :

$$\mathcal{L} = T - U \quad (3.2)$$

Equation (3.1) is expressed for the pendulum coordinate θ_p . Expressing the kinetic energy

of the pendulum means taking into account the inertia and the angular velocity of the structure:

$$T = \frac{1}{2} I \dot{\theta}_p^2 \quad (3.3)$$

where I is the system's total inertia. The potential energy U is expressed as follows:

$$U = -K(1 - \cos(\theta_p)) \quad (3.4)$$

with,

$$K = L_g M_{tot} g = \frac{M_p L_p + M_w L_w + M_m L_w}{M_{tot}} M_{tot} g = (M_p L_p + M_w L_w + M_m L_m) g \quad (3.5)$$

where M_{tot} is defined as:

$$M_{tot} = M_p + M_w + M_m \quad (3.6)$$

By computing the derivatives of the Lagrange operator \mathcal{L} , the following expression is obtained:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_p} - \frac{\partial \mathcal{L}}{\partial \theta_p} = I \ddot{\theta}_p - K \sin(\theta_p) \quad (3.7)$$

Equation (3.8) is equal to the generalised forces ξ_i taking into account the torque produced by the motor called control torque τ_c , a possible external disturbance torque τ_d and the pendulum's viscous friction $b_p \dot{\theta}_p$:

$$I \ddot{\theta}_p - K \sin(\theta_p) = \tau_d - b_p \dot{\theta}_p - \tau_c \quad (3.8)$$

The goal is to express the system's dynamics as a function of the control command i , the current sent to the motor. To do so, it is essential to analyse the motor's internal dynamics, particularly how the control torque is generated [32]. The control torque τ_c is the net torque applied by the motor, which is the difference between the electric torque τ_e generated by the motor and the frictional torque $b_w \dot{\theta}_w$ that opposes the motion. The term $b_w \dot{\theta}_w$ represents the damping effect of the motor's internal friction. This relationship is expressed as:

$$\tau_c = \tau_e - b_w \dot{\theta}_w \quad (3.9)$$

The electric torque of the motor can be expressed as the product of the torque constant and the electrical current supplied to the motor:

$$\tau_e = K_m \cdot i \quad (3.10)$$

The final equation that describes the pendulum motion is:

$$\ddot{\theta}_p = \frac{K}{I} \sin(\theta_p) - \frac{b_p}{I} \dot{\theta}_p - \frac{K_m}{I} i + \frac{b_w}{I} \theta_w + \frac{\tau_d}{I} \quad (3.11)$$

The mathematical model is formulated in 2D, but since the motion remains identical in the perpendicular plane, the model can later be extended to 3D.

3.2 Motion of the reaction wheel

The speed of the reaction wheel is expressed as the difference between the wheel inertial speed and the pendulum's speed:

$$\dot{\theta}_w = \dot{\theta}_{w,R} - \dot{\theta}_p \quad (3.12)$$

The wheel's inertial angular acceleration is given by the ratio of the control torque to the wheel's moment of inertia with respect to its centre of gravity:

$$\ddot{\theta}_{w,R} = \frac{\tau_c}{I_{w,R}} \quad (3.13)$$

By deriving Equation (3.12) and replacing with Equations (3.13) and (3.9), the motion of the reaction wheel can be expressed as following:

$$\ddot{\theta}_w = -\frac{K}{I} \sin(\theta_p) + \frac{b_p}{I} \dot{\theta}_p + K_m \frac{I+I_w}{II_w} i - \frac{I+I_w}{II_w} b_w \dot{\theta}_w - \frac{\tau_d}{I} \quad (3.14)$$

3.3 State-space

The final system of equations that describe the motion of the entire system is:

$$\begin{cases} \ddot{\theta}_p = \frac{K}{I} \sin(\theta_p) - \frac{b_p}{I} \dot{\theta}_p - \frac{K_m}{I} i + \frac{b_w}{I} \theta_w + \frac{\tau_d}{I} \\ \ddot{\theta}_w = -\frac{K}{I} \sin(\theta_p) + \frac{b_p}{I} \dot{\theta}_p + K_m \frac{I+I_w}{I \cdot I_w} i - \frac{I+I_w}{I \cdot I_w} b_w \dot{\theta}_w - \frac{\tau_d}{I} \end{cases} \quad (3.15)$$

The obtained equations of motion describe a non-linear system due to the presence of terms such as $\sin(\theta_p)$. To create a linear state-space model, it is necessary to linearize the equations around a specific operating point. By assuming $\sin(\theta_p) \approx \theta_p$, which holds when θ_p is sufficiently small, the system can be approximated as linear. This approximation enables the expression of the state matrix and input matrix for a state space model:

x : state vector	
u : input vector	
y : output vector	$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$
A : state matrix	(3.16)
B : input matrix	
C : output matrix	
D : feed-through matrix	

$$\mathbf{x} = [\theta_p \quad \dot{\theta}_p \quad \dot{\theta}_w]^T, \quad \mathbf{u} = [i \quad \tau_d]^T,$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{K}{I} & -\frac{b_p}{I} & \frac{b_w}{I} \\ -\frac{K}{I} & \frac{b_p}{I} & -\frac{I+I_w}{I \cdot I_w} b_w \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{-K_m}{I} & \frac{1}{I} \\ \frac{I+I_w}{I \cdot I_w} K_m & -\frac{1}{I} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Chapter 4

Design of the inverted pendulum

This section focuses on selecting the components for the inverted pendulum prototype. The system is built around a development board, which is programmed to implement the control method described in Chapter 6. The board regulates the motor's operation by sending voltage signals to a driver, which then adjusts the current supplied to the motor to control the pendulum's position. Additionally, sensors continuously monitor both the pendulum and motor, providing real-time feedback to update the control signal sent to the driver.

4.1 Development board

Choosing an appropriate development board is essential to meet the system's requirements. The main objective is to select a board capable of efficiently handling signal processing tasks while ensuring availability in the lab. Since Arduino boards were already accessible in the SAAS department, the comparison focuses on two models: Arduino Uno R3 and Arduino Due.

Table 4.1: Comparison between Arduino Uno and Arduino Due

Characteristics	Arduino Uno R3 [4]	Arduino Due [3]
Operating voltage [V]	5	3.3
Clock speed [MHz]	16	84
Analogic output	PWM (6)	PWM (12) and DAC (2)
Output voltage [V]	[0,5]	[0.55,2.75]
Resolution [Bits]	8	12
Flash memory [kB]	32	512
SRAM [kB]	2	96

The main selection criteria, summarized in Tab.(4.1), include:

- Operating voltage: Unlike other Arduino models, the Arduino Due runs at 3.3V and applying any voltage above this value would damage the board.
- Clock speed: A higher clock speed enables faster computations and more responsive performance, which is the case for the Arduino Due.
- Analog output: The goal of the analog output is to provide a certain voltage to control an external circuit. Both boards support PWM but the Arduino Due also features a DAC (Digital-to-Analog Converter), providing smoother analog output.
- Memory capacity: A larger flash memory supports more complex code to be stored and executed while Static Random Access Memory (SRAM) affects how much run-time data can be stored [16]. In this project, the flash memory of both Arduino boards is sufficient, but to conduct longer real-time regulations a larger SRAM is preferred.

Both boards were available in the SAAS lab, but the Arduino Due was chosen not only for its DAC output, which the Arduino Uno lacks, but also because it operates at a much higher clock frequency (84 MHz versus 16 MHz). As a result, for the same code, the Arduino Due is capable of faster computations, making it more suitable for higher-frequency regulations. Additionally, the Arduino Due has a larger SRAM (96 kB compared to 2 kB), allowing to store more variables for real time computations.



Figure 4.1: Picture of the Arduino Due [3]

4.2 Motor and driver

The motor serves as the system's actuator, generating the torque required to restore equilibrium. The literature review helped define the key selection criteria: the motor should be as lightweight as possible, given its position at the top of the pendulum, while also providing maximum torque. Maxon motors were selected due to their frequent use in the SAAS lab. This brand offers the advantage of highly detailed and precise datasheets, along with the option to purchase compatible gearboxes, encoders, and drivers as needed.

Table 4.2: Comparison of Motor Types [25]

	DC Motor	BLDC Motor
Advantages	High acceleration thanks to a low mass inertia High efficiency Compact design	More efficient (no brush friction) Longer lifespan High speed capabilities Smoother operation, less noise Usually lighter and smaller
Disadvantages	Brushes wear out, reducing lifespan	Requires a dedicated controller Higher cost compared to brushed motors

The two main types of motors used for this type of application are Direct Current (DC) motors and Brushless Direct Current (BLDC) motors.

A DC brushed motor works by passing electric current through the armature (rotor) windings, creating a magnetic field that interacts with the stator's magnetic field. This interaction generates torque, causing the rotor to spin. The commutator and brushes switch the current direction in the rotor windings, ensuring continuous rotation in the same direction [13]. On the other hand, a Brushless Direct Current motor has a permanent magnet in the rotor and wound coils in the stator (coil wires wrapped around the stator), and requires

a drive circuit instead of brushes and a commutator to achieve rotation [5]. The main advantages and disadvantages of both brushed and brushless DC motors are presented in Tab.(4.2).

As the requirements for the motor include both torque and weight, this led to a closer examination of brushless DC motors, as they offer a good balance between efficiency, power density, and reduced weight compared to brushed alternatives. Maxon motor offers a category of motor called ECX-flat/EC-flat. This type of motor offers an attractive price to performance ratio, a high torques due to external multipole rotor, and a flat and light design meant for limited space [25]. From the catalog were selected two models: the EC 45 flat which is an older version of the EC category and the ECX flat 42S, a more recent version with a higher torque. The main characteristics for both motors are represented in Tab.(4.3). The sensors integrated in the motor used to monitor its speed will be discussed in section 4.3.2.

Table 4.3: Comparison of EC-flat motors [25]

	EC 45 flat	ECX flat 42S
Nominal voltage [V]	24	48
Nominal speed [rpm]	5240	8760
Nominal torque [NmN]	83.4	133
Nominal current [A]	3.52	2.57
Weight [g]	110	106



Figure 4.2: Two motor options by Maxon [25]

Initial simulations were made with values from the literature to evaluate if both motors had the necessary torque to weight ratio to control the inverted pendulum. The simulations involved replicating in **Simulink** the mathematical model with parameter values from the literature and using auto-tuning PID blocks to find control laws.

It was demonstrated that both motors could control the pendulum while the ECX Flat 42S showed superior performances as it provides a bigger torque for approximately the same weight. However, due to significant shipping delays for this motor, it was decided to order the EC flat 45 instead.

In order to control the motor using the microcontroller, a driver needs to be used. A driver serves as the interface between the motor and the microcontroller to control the motor's speed, direction or other parameter. Common drivers include the L298N which has an H-bridge typology. An H-bridge is a simple circuit that contains 4 switching elements and depending on which switches are ON/OFF, the current flows in one direction or the other,

changing the motor's direction. This driver is very cheap and commonly used option but is however not compatible with BLDC motors as they cannot handle 3-phase control [33]. An option compatible with BLDC motors are the Maxon drivers that were specially conceived to work with matching motors. The model that matches the motor EC Flat 45 Flat is the ESCON 50/5 meant to work for voltages up to 50V and permanent current up to 5A.



Figure 4.3: Picture of the ESCON 50/5

Additionally to being very easily connected to the matching motors, this drivers presents 3 operative modes [23]:

- Current controller which compares the actual motor current with the applied set value and dynamically adjusts it in case of deviation.
- Closed loop speed controller that compares the actual speed with the applied set value and dynamically adjusts it.
- Open loop speed controller that feeds the motor with a voltage proportional to the set value. The load changes are compensated with IxR methodology.

Table 4.4: Characteristics of the Escon 50/5 [23]

ESCON 50/5 Motor Driver	
Nominal operating voltage [V]	10 to 50
Maximal output current [A]	15 A (<20 s)
Analog input [V]	-10 to 10
Hall sensor compatibility (y/n)	yes
Encoder compatibility (y/n)	yes
Weight [g]	204

This driver also has the main advantage of being programmable via a USB cable and graphic interface. The controller receives an analog input varying from -10V up to 10V as seen in Tab.(4.4). This voltage is received as reference and the ESCON driver implements a specified regulation to set the current or speed of the motor. The chosen microcontroller Arduino Due is only able to send voltages from 0.55V to 2.75V and by programming the

ESCON, one can associate to each min/max value a command for the motor.

Tab.(4.5) summarizes the implementation on ESCON software for 2 different operation modes: current controller and closed loop speed controller. In each mode, a voltage signal sent by the Arduino is interpreted by the ESCON as a reference for either motor current or speed.

Table 4.5: Implementation of control of the Escon 50/5 [23]

Current control		
Current at	0.55 V	-2.33 A
Current at	2.75 V	2.33 A
Closed loop speed control		
Speed at	0.55 V	-5240 rpm
Speed at	2.75 V	5240 rpm

The operating voltage of the ESCON is decided using the following formula [23]:

$$V_{CC} \geq \left[\frac{U_N}{n_O} \cdot \left(n + \frac{\Delta n}{\Delta M} \cdot M \right) \cdot \frac{1}{0.98} \right] + 1 [V] \quad (4.1)$$

With M [mNm] the operating torque, the operating speed n [rpm], the nominal motor voltage U_N [Volt], the motor no-load speed at U_N , n_0 [rpm] and the speed/torque gradient of the motor $\Delta n/\Delta M$ [rpm/mNm]. By replacing the values from the EC 45 flat datasheet, it was determined that a voltage V_{CC} of at least 22.79 V is necessary to power the motor. A source of 24 V has been selected.

4.3 Sensors

After selecting the system's actuator, it is essential to choose the appropriate sensors. These sensors measure the system's variables, providing the necessary information to compute the control command. Based on the state matrix A presented in section 3.3, the control strategy requires access to the pendulum's angle and angular velocity, as well as the motor's speed. These measurable variables are obtained using two sensors: an Inertial Measurement Unit (IMU) and integrated Hall effect sensors.

4.3.1 Inertial Measurement Unit (IMU)

In order to recover the pendulum angle and angular speed an inertial measurement unit can be used. It consists of a combination of accelerometer and gyroscope that allows to measure and report the angular rate of a structure. The accelerometer measures changes of acceleration in the sensor with respect to the earth while the gyroscope measures changing angular motion. The chosen IMU is the MPU6050 linked to an I2C bus for easier connection with the Arduino microcontroller. The chip is represented in Fig.(4.4) and its main characteristics in Tab.(4.6).

It has a 3 axis gyroscope and accelerometer, allowing to recover the angle and acceleration in the 3 planes (x,y,z). Using the MPU6050 Arduino library from [i2cdevlib](#) [17], it is possible to retrieve the Euler angles of the pendulum with the function `mpu.dmpGetEuler()`. By filtering the angle value and doing a derivative, the pendulum angular velocity can also be determined.



Figure 4.4: Picture of the MPU6050 [12]

Table 4.6: MPU6050 characteristics [12]

Inertia Measurement Unit: MPU6050	
Powering of I2C	3.3V
Dimensions	14 × 21 mm
Axis of angular rate sensor	3 axis
Axis of accelerometer sensor	3 axis

4.3.2 Hall effect sensors

To measure the motor's properties, the EC 45 Flat comes with integrated Hall sensors. Hall sensors detect the magnetic field of a control magnet which is attached to the shaft. The sensors are arranged between the stator teeth and monitor directly the poles of the magnetic ring with offsets of 120°. The Hall sensors evaluate the direction of the magnetic field and generate a high output (5V) if the north pole of the control magnet is close and a low output (GND) if the south pole is close [26].

As represented in Fig.(4.5a), there are 3 Hall effect sensors (blue, green and red). At rotation angle $\Phi = 0$ in Fig(4.5b), the blue Hall sensor has a high output when it sees the north pole and keeps it high for 120°. The green sensor is closer to the south pole and its output is thus low for the next 60°. The red sensor just switched from high to low level and stays low for 180° until the north pole comes back. Thus, by looking at these 3 signals it is possible to know the rotor position within 60°.

The ESCON 50/5 is once again very useful as it can automatically recover the motor's speed and send it via a programmable analog output. The Arduino Due board can recover an analog input range of 0 to 3.3V and Tab (4.7) summarizes the conversion used to recover the motor's speed on the microcontroller.

Table 4.7: Analog output of the Escon 50/5 [23]

Analog output for motor speed		
Voltage at	-5240 rpm	0 V
Voltage at	5240 rpm	3.3 V

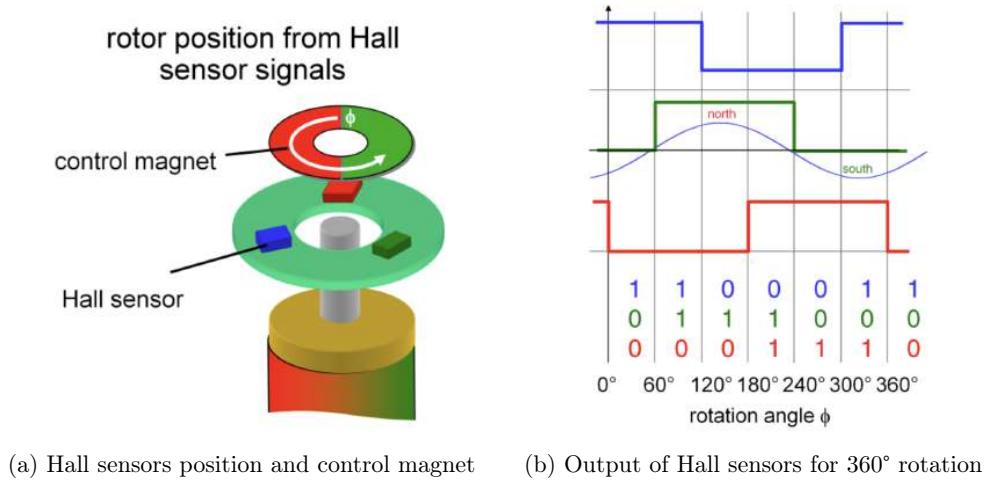


Figure 4.5: Hall sensor feedback signals [23]

4.4 3D design and printing

4.4.1 Pendulum choice

The pendulum is designed as a hollow cylinder on which the motor is mounted. It should have a high stiffness-to-weight ratio and not deform during the pendulum motion. The goal is to find a material with a high Young's modulus and a low density. Based on similar projects the materials represented in Tab.(4.8) were considered.

Table 4.8: Properties comparison of carbon fibre, aluminium and PLA

Property	Carbon fibre [28]	Aluminium [2]	PLA [11]
Density (g/cm ³)	1.79 - 1.99	2.7 - 8	1.3
Young's modulus (GPa)	200 - 500	70 - 200	3 - 4
Cost	\$\$\$	\$\$	\$

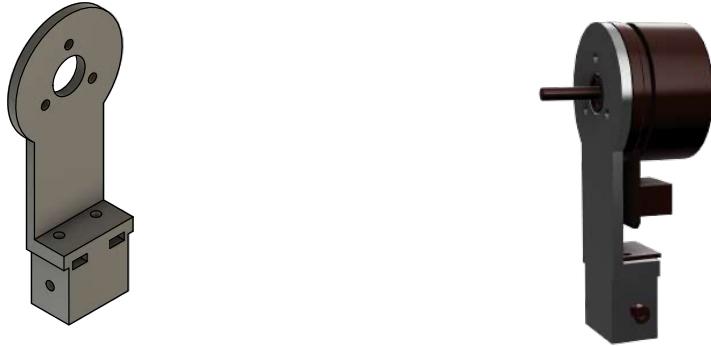
3D printed PLA offers the advantages of being widely available and cost-effective, making it ideal for prototyping. It also allows for the integration of the motor and pendulum into a single printed piece. However, PLA presents a low Young's modulus resulting in limited stiffness. Moreover, achieving high precision 3D printing on small cylinders is very challenging. For these reasons, PLA was not selected. Instead, carbon fibre was chosen for its combination of low density and high Young's modulus, providing both lightweight and structural rigidity. Additionally, suitable carbon fiber rods were already available in the SAAS lab.

The rod is 30cm long to be able to adapt the pendulum's height and has an outer radius of 7.5mm.

4.4.2 Motor and sensor: mounted structure

Besides the pendulum rod, all other components of the prototype were designed on Fusion360 and 3D-printed with PLA in the atelier of the SAAS.

Mounted on top of the carbon fibre rod is a structure that holds the motor and IMU. The upper structure holds the BLDC motor using 3 Metric 3 (M3) screws directly into the



(a) 3D design of the mounted structure (b) Mounted structure with motor and IMU

Figure 4.6: 3D printed mounted structure

motor through the 3D printed structure. The IMU is placed right under the motor and kept in place with 2 M3 screws that are secured with nuts placed within the 3D printed structure. The entire assembly is attached to the top of the pendulum rod via two lateral screws and nuts, one on each side of the rod. The CAD model is shown in Fig.(4.6a). Joined with the motor, M3 screws and nuts and IMU the assembly is shown in Fig.(4.6b).

4.4.3 Driver and microcontroller: base structure

A single base structure is 3D printed to support both the ESCON driver and the Arduino Due as close as possible to the pendulum. A key constraint when designing the prototype is to minimize the length of the IMU cables as the I2C bus used to interface with the sensor can only support short distances reliably (approximately 40cm, found by trial and error). This design constraint makes it essential to keep the microcontroller (connected to the I2C bus with wires) as elevated as possible to minimize cable length. To keep a compact design, the driver and power switch are also tightly integrated around the pendulum base, represented in Fig.(4.8b).

The pendulum itself slots into a rectangular holder presented in Fig.(4.8a) and is secured laterally using two screws. This rectangular holder is connected to two base structures using ball bearings to allow one DOF motion of the pendulum while reducing friction as much as possible. Those structures are represented in Fig.(4.7), they both include a cavity dedicated to the placement of the ball bearings. One of the base structure has a cable holder on top to pass the cables from the pendulum directly to the ESCON.

4.5 Wiring and final prototype

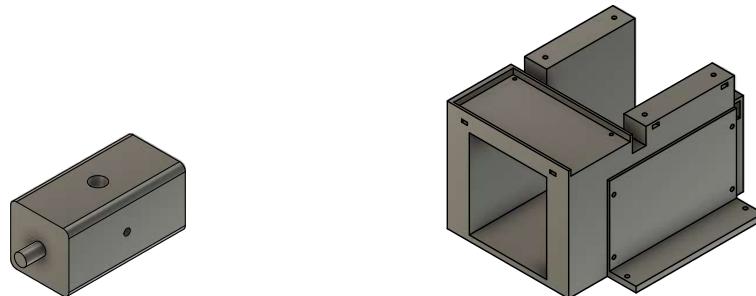
The final prototype represented in Fig.(4.10). It is made of the base structure that contains the Arduino, ESCON and the support to allow the pendulum's motion. The motor mounted on top of the pendulum thanks to the mounted structure is connected to the reaction wheel. The reaction wheel is also 3D printed in PLA and has screws and nuts on its radius to increase its inertia. The wheel is fixed on the motor shaft using 3 lateral screws with nuts embedded in the 3D printed wheel.



(a) 3D design of the base structure with cable holder

(b) 3D design of the base structure

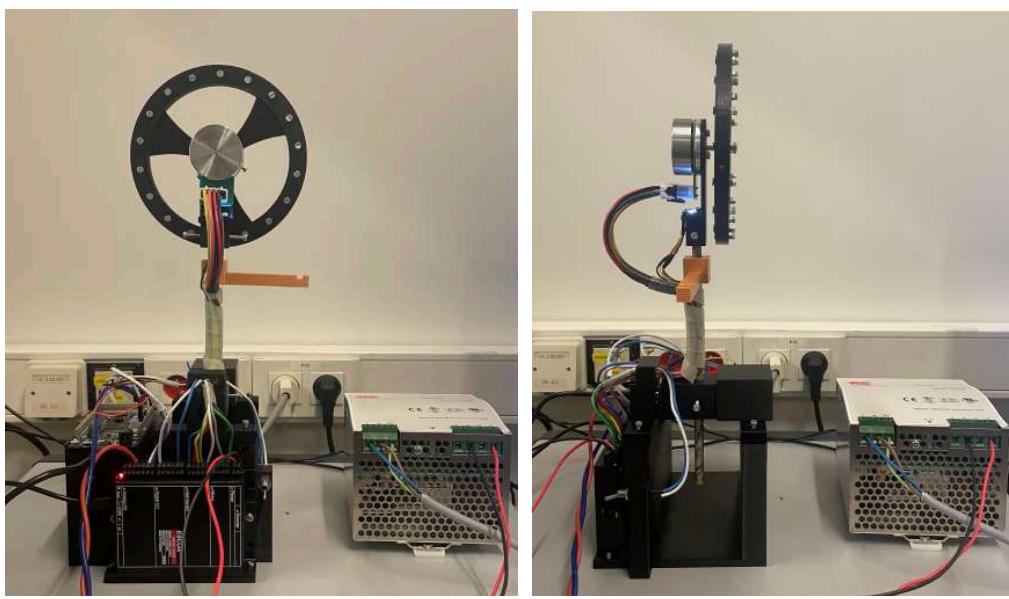
Figure 4.7: 3D printed base structure



(a) 3D design of the rectangular holder

(b) 3D design of the support structure

Figure 4.8: 3D printed support and rectangular holder



(a) Back of the prototype

(b) Side of the prototype

Figure 4.9: Picture of the Reaction Wheen Inverted Pendulum

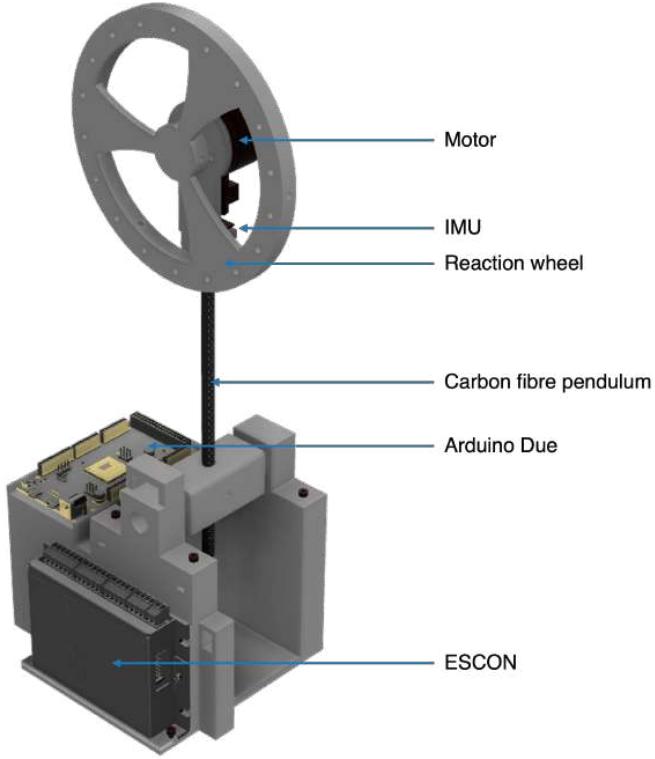


Figure 4.10: Final 3D design of the inverted pendulum with reaction wheel

As shown in Tab.(4.9), the ESCON is powered by a 24V supply, while the Arduino receives power via a USB connection to the computer. The ESCON supplies power to both the motor and the Hall sensor, whereas the Arduino powers the IMU.

Table 4.9: Power supplies of the prototype

Power supplies	
ESCON	24V Power supply
Arduino	5V from USB cable on PC
IMU	3.3V from Arduino
Motor	24V from ESCON
Hall sensors	5V from ESCON

The wiring diagram of the inverted pendulum with reaction wheel is illustrated in Fig.(4.11).

Finally, all the technical drawings of the CAD models are presented in Appendix H.

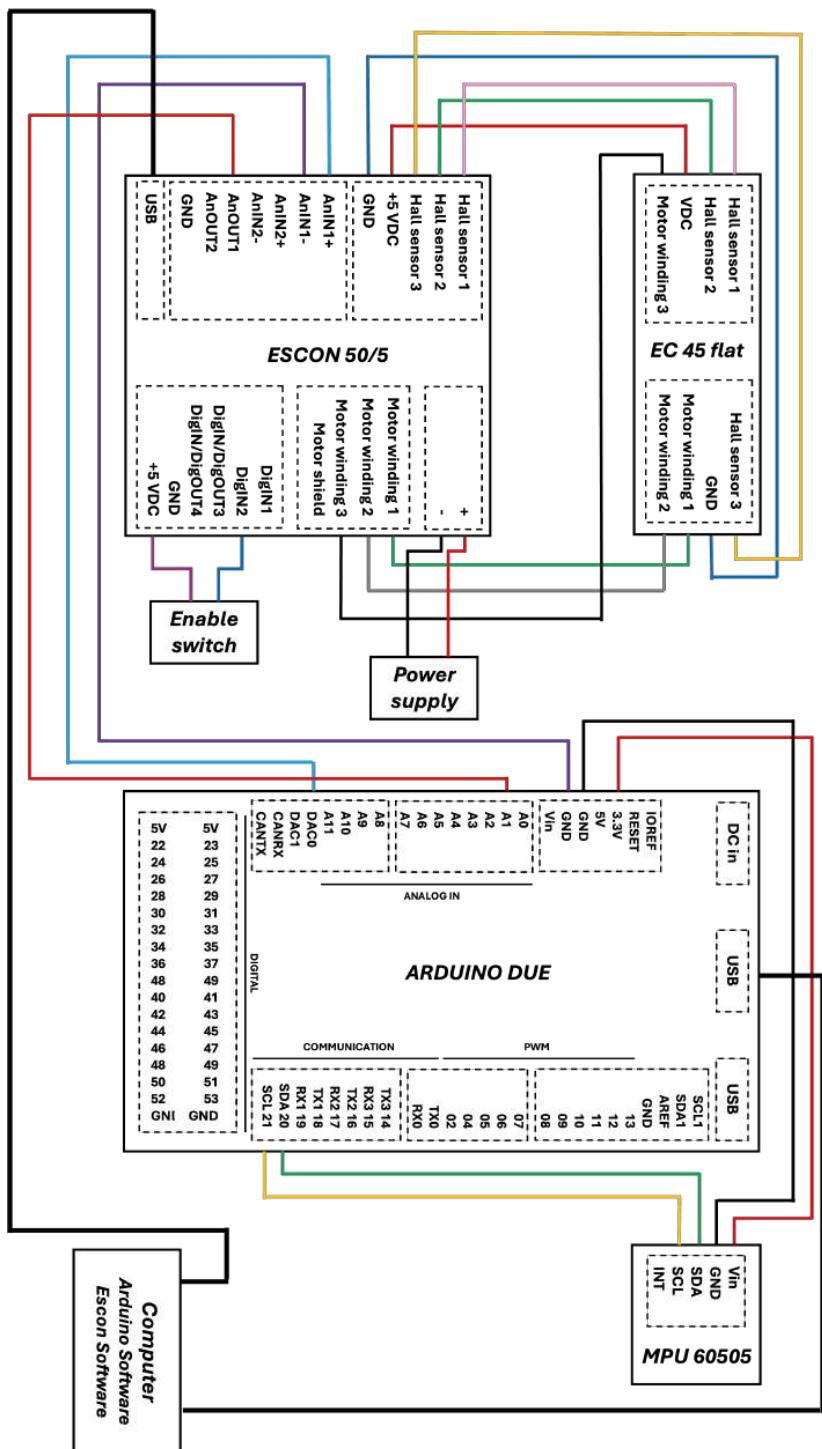


Figure 4.11: Wiring diagram of the inverted pendulum with reaction wheel

Chapter 5

Parameter estimation of the inverted pendulum

This short chapter is dedicated to estimating the parameter values of the prototype previously presented in chapter 4. The analysis is based on the mathematical description developed in Equation (3.15). Firstly, each component of the prototype is weighted and measured:

Table 5.1: Weights and lengths of the inverted pendulum as presented in Fig.(3.1)

Mass of the pendulum rod	M_p	6.5 g
Mass of the wheel	M_w	74 g
Mass of the motor	M_m	155 g
Height of the pendulum rod	L_p	20 cm
Height of the motor shaft	L_m	26 cm
Height of the reaction wheel	L_w	26 cm

Simple parameters such as the variable K is estimated using values from Tab.(5.1):

$$K = (M_p L_p + M_w L_w + M_m L_w)g = 0.597 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-2} \quad (5.1)$$

5.1 Pendulum parameters

Pendulum's parameters such as its inertia and friction must be determined experimentally due to the complexity of the prototype's shape.

5.1.1 Pendulum inertia

The pendulum's inertia can be computed considering the prototype as a simple pendulum of mass m concentrated at length L from the origin, making a small angle approximation of $\sin(\theta_p) \approx \theta_p$: [39]

$$\frac{d^2\theta_p}{dt^2} = \left(\frac{-mgL}{I} \right) \theta_p \quad (5.2)$$

The solution of such differential equation is as following:

$$\theta_p(t) = \Theta \cos(\omega t + \phi) \quad (5.3)$$

And the period of such system is thus defined as:

$$T = 2\pi \sqrt{\frac{I}{mgL}} \quad (5.4)$$

Reorganising Equation (5.4) gives:

$$I = \frac{T^2 mgL}{4\pi^2} \quad (5.5)$$

With the motor blocked and the pendulum placed in downward position and oscillating it at a small angle to compute the period T , it is thus possible to approximate I the system's total inertia. This experiment is represented in Fig.(5.1).

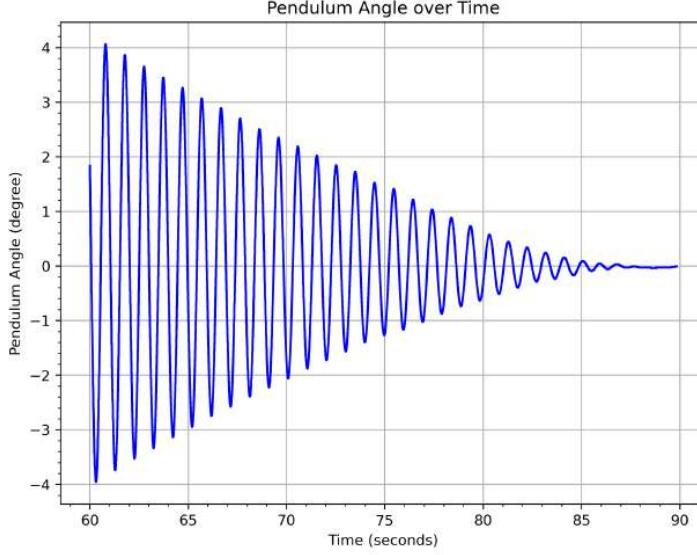


Figure 5.1: Pendulum oscillation period experiment

The process was repeated five times, and the average period computed with a Python code is 0.98 second, leading to an inertia of $I = 0.1435 kg \cdot m^2$.

5.1.2 Pendulum friction

To estimate the pendulum's friction coefficient b_p , the pendulum is put downside and the motor is blocked. By neglecting the motor and reaction wheel, the following equation is obtained:

$$\ddot{\theta}_p = \frac{-K}{I} \sin(\theta_p) - \frac{b_p}{I} \dot{\theta}_p \quad (5.6)$$

Using the K value previously found, in order to verify the inertia value computed using the period, the least squares method is used to compute I and b_p . Using the least squares method, the goal is to solve for p the following system :

$$Y = Xp \quad (5.7)$$

Which can also be defined as:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix} = \begin{bmatrix} \sin(\theta_1) & \dot{\theta}_1 \\ \sin(\theta_2) & \dot{\theta}_2 \\ \vdots & \vdots \\ \sin(\theta_N) & \dot{\theta}_N \end{bmatrix} \begin{bmatrix} -\frac{K}{I} \\ -\frac{b_p}{I} \end{bmatrix} \quad (5.8)$$

As the matrix X is not squared, the system is solved using the pseudo-inverse:

$$p = X^\dagger Y = (X^T X)^{-1} X^T Y \quad (5.9)$$

By identification, the pendulum friction b_p and the system inertia I can be recovered starting from value K :

$$\begin{cases} I = \frac{-K}{p_1} \\ b_p = -p_2 I \end{cases} \quad (5.10)$$

This method allows both to compute the pendulum friction but also to verify the inertia computed previously. The Python algorithm computed an inertia $I = 0.014 \text{kg} \cdot \text{m}^2$ and a friction coefficient of $b_p = 0.0314 \text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$. The inertia computed with the least squares method indeed confirms the result of $I = 0.1435 \text{kg} \cdot \text{m}^2$ obtained in section 5.1.1.

5.2 Motor parameters

5.2.1 Reaction wheel inertia

The reaction's wheel inertia is estimated using usual algebraic calculations around axis z . The wheel is made of a 3D printed structure that weights $M_{disc} = 50\text{g}$ that can be estimated to a hollow cylinder of internal radius $r_i = 60\text{mm}$ and external radius $r_o = 75\text{mm}$ as seen in Fig.(5.2). On this cylinder are placed 18 screws and nuts to increase inertia. They are placed at a distance $r_{weight} = 67.5\text{mm}$ from the centre of the wheel and each weight $M_{weights} = 1.3\text{g}$.

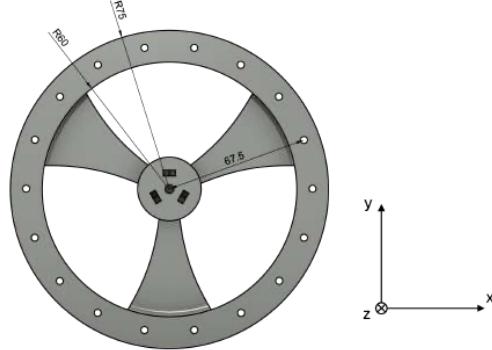


Figure 5.2: Reaction Wheel's radius measurements

The inertia of the wheel without the weights can be estimated to:

$$I_{disc} = \frac{1}{2} M_{disc} (r_i^2 + r_o^2) = 2.33 \cdot 10^{-4} \text{kg} \cdot \text{m}^2 \quad (5.11)$$

The inertia for one weight, considered a simple point mass, is estimated to:

$$I_{weight} = M_{weight} r_{weight}^2 = 5.92 \cdot 10^{-6} \text{kg} \cdot \text{m}^2 \quad (5.12)$$

Finally, the reaction wheel inertia value is:

$$I_w = I_{disc} + 18I_{weight} = 3.39 \cdot 10^{-4} \text{kg} \cdot \text{m}^2 \quad (5.13)$$

5.2.2 Motor friction and constants

To identify the motor parameters, the pendulum is blocked at a zero angle. The pendulum's motion equation from Equation (3.15) thus becomes:

$$\ddot{\theta}_w = K_m \hat{I}i - b_w \hat{I}\dot{\theta}_w \quad (5.14)$$

With \hat{I} as following:

$$\hat{I} = \frac{I + I_w}{II_w} \quad (5.15)$$

The transfer function between the input current and the motor speed is then obtained by applying Laplace transforms to differential Equation (5.14):

$$\frac{\dot{\Theta}_w(s)}{I(s)} = \frac{\frac{K_m}{b_w}}{\frac{s}{\hat{I}b_w} + 1} = \frac{G}{\tau s + 1} \quad (5.16)$$

With,

$$\begin{cases} G = \frac{K_m}{b_w} \\ \tau = \frac{1}{\hat{I}b_w} \end{cases} \quad (5.17)$$

The motor transfer function is approximated to a first order transfer function and the parameters G and τ can be identified by analysing the motor's step response. G is the gain of the system, defined as:

$$G = \frac{\dot{\theta}_{w,final} - \dot{\theta}_{w,init}}{i_{final} - i_{init}} \quad (5.18)$$

A step response is done starting at $i_{init} = 0A$ for input to $i_{final} = 0.3A$. Variable τ is the time for which the system response reaches 63.2% of its final value. A first visual estimation of parameters G and τ is made using a **Python** code for visualisation. The step response is illustrated in Fig.(5.3).

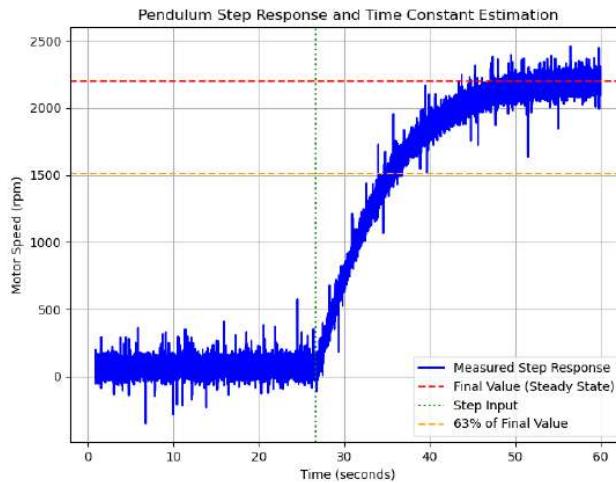


Figure 5.3: Visual estimation of parameters G and τ on Python

The estimated values of the gain and time constants are then used as input of an identification **Matlab** code. The results and details of this identification algorithm are presented in Appendix A.

The final transfer function for the motor's response in current is:

$$\frac{\dot{\Theta}_w(s)}{I(s)} = \frac{7475}{7.611s + 1} \quad (5.19)$$

From parameters G and τ , the motor's torque constant K_m and viscous friction b_w can be estimated using formulae (5.17).

5.3 Verification of the model

All the computed parameters are available in Tab.(5.2).

Table 5.2: Parameter values summary

Parameters		
Pendulum Inertia	I	$0.14 \text{ kg} \cdot \text{m}^2$
Pendulum friction	b_p	$0.0314 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
Reaction Wheel Inertia	I_w	$3.39 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$
Reaction Wheel friction	b_w	$5.5 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
Torque Constant	K_m	0.4116 mNm/A

In order to validate the mathematical model, **Matlab** and **Simulink** simulations must be done. As the system is unstable, it will not have the expected behaviour starting from upwards position unless stabilised with a regulator. The model is thus analysed in the hanging position: a command in current of 1A is sent for a duration of 1 second and the behaviour of the simulation and experimental models are compared. The **Simulink** schematics of this simulation is presented in Fig.(E.1) of the Appendix E.

When plotting the initial comparison shown in Fig.(5.4) it was noticed that there are some significant differences in the period of oscillation and amplitude of the response. The difference in period can be explained by the fact that the inertia of the pendulum I was computed without wires disturbing its path; once the motor is connected, it brings heavy wires with it that disturbs the pendulum's movement. To achieve a better superposition of the simulation with the experimental, the inertia I is reduced to match the oscillation frequency and the pendulum friction b_p is decreased. The difference in pendulum friction could be explained by the change in setup configuration and wire positioning between the setup upside down and upright.

Additionally, there still is a difference in amplitude between the experiment and the stimulation: the **Simulink** code computes a much smaller amplitude. To address this, further investigation focused on the reaction wheel's inertia I_w , which was suspected to have been underestimated. In Appendix C are detailed the investigations conducted.

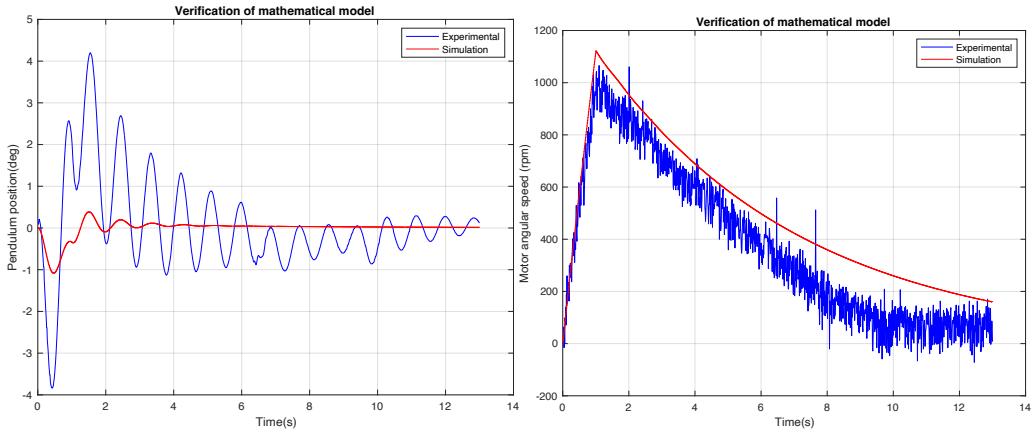


Figure 5.4: Comparison of simulation and experimental of (a) pendulum angle, (b) motor's angular speed, for a command of 1A of 1 second

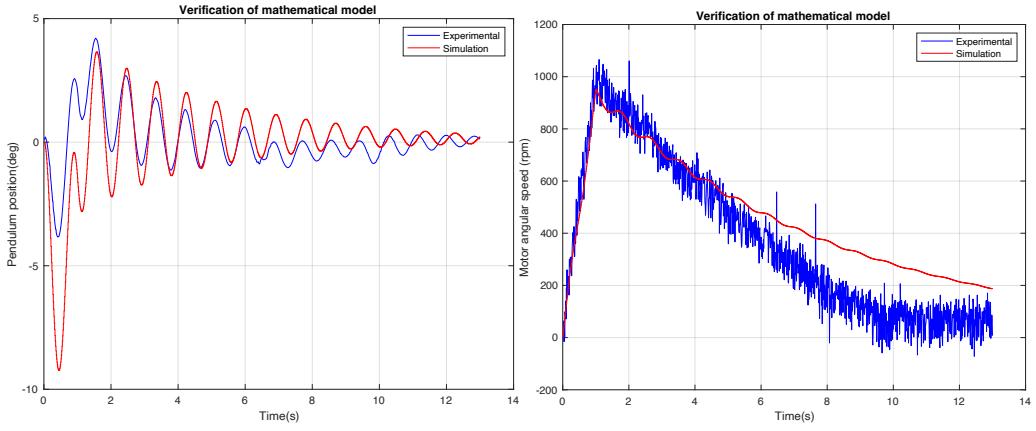


Figure 5.5: Comparison of simulation and experimental with updated parameters of (a) pendulum angle, (b) motor's angular speed, for a command of 1A of 1 second

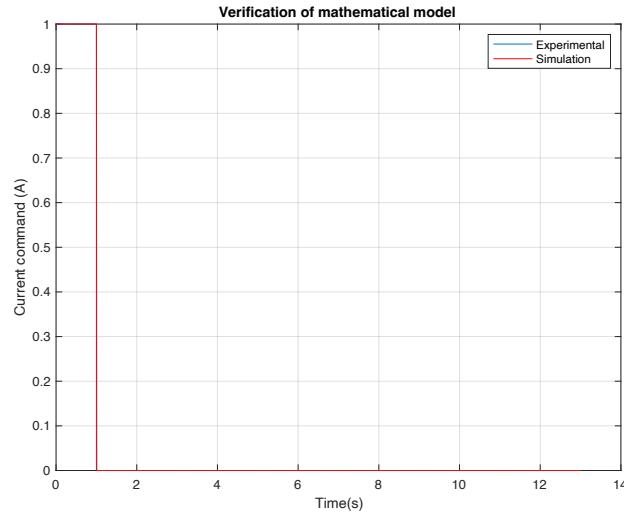


Figure 5.6: Command sent to the motor for mathematical model verification

Tab.(5.3) summarises the parameters once they have been adapted to better fit the experimental model. Considering the updated parameters, in Fig.(5.5) are plotted the pendulum angle and motor's speed response (experimental and simulation) for the command presented in Fig.(5.6).

Table 5.3: Parameter values adapted to the experimental model

Parameters		
Pendulum Inertia	I	$0.012 \text{ kg} \cdot \text{m}^2$
Pendulum friction	b_p	$0.0114 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
Reaction Wheel Inertia	I_w	$3 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$
Reaction Wheel friction	b_w	$4 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
Torque Constant	K_m	2.99 Nm/A

Chapter 6

Control of the inverted pendulum

This chapter is dedicated to designing the control laws for the inverted pendulum. Firstly, the dynamic behaviour of the inverted pendulum is analysed, then a Linear Quadratic Regulator is investigated and evaluated. Finally, a cascade PID controller is studied and its performances are compared to the LQR.

The objective of the control law implemented is to keep the pendulum in the upright position while rejecting disturbances. The control methods will be evaluated based on the following criteria:

- **Recovery angle:** maximal initial angle position in which the pendulum is placed for which the actuator is able to bring the angle back to 0° .
- **Maximum disturbance torque:** highest torque applied to the pendulum that the actuator can counter without the system losing its upright equilibrium.
- **Motor power consumption:** power consumption of the motor computed during the time of the experiment.

6.1 Dynamic behaviour of the inverted pendulum

To properly understand the dynamic behaviour of a non linear system such as the inverted pendulum, a simplified version of the inverted pendulum model can be considered in Equation (6.1). It considers only two state variables $x \in \mathbb{R}^2$: the pendulum angle $x_1 = \theta_p$ and the pendulum angular speed $x_2 = \dot{\theta}_p$. That way, the pendulum dynamics can be plotted in 2D in the (x_1, x_2) plane [18].

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ \frac{K}{I} \sin x_1 - \frac{b_p}{I} x_2 \end{pmatrix} \quad (6.1)$$

The vector field of the Equation (6.1) is defined as:

$$\frac{dx}{dt} = F(x) \quad (6.2)$$

$F(x)$ is a vector that defines the velocity of the state x . When plotting these vectors in the plane (x_1, x_2) , a visual representation of the dynamic of the system is obtained. This type of plot is called phase portrait: for a set of initial conditions, the solution of the differential equation is plotted in the plane. By following each velocity arrow $F(x)$, the resulting trajectory appears on the plane [18]. The phase portrait of the upright pendulum described in Equation (6.1) is plotted in Fig.(6.1).

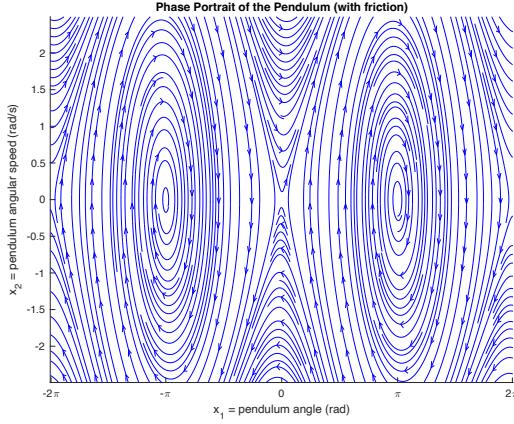


Figure 6.1: Phase portrait of the non linear upright pendulum

Points $(x_1, x_2) = (\pi, 0)$ are equilibrium states: "An equilibrium state is a particular solution of the differential equation that does not depend on time" [19]. In Fig.(6.1) one can see that around these coordinates, the velocity arrows tend to bring the state back to the equilibrium point.

Analysing the phase portrait can be useful to analyse linearisation of a system. The point $(x_1, x_2) = (0, 0)$ is chosen for linearisation of the system, corresponding to the target upright position. The phase portrait of the system with linearisation and without are presented in Fig.(6.2). The pendulum is unstable at this equilibrium point and the linearisation is a good approximation around the point $(x_1, x_2) = (0, 0)$. However, for larger angles, specifically around 30° ($\pi/6$ rad), where the norm of the difference in the vector fields exceeds 0.1, the behaviour of the linearized system starts to diverge from the non-linear system.

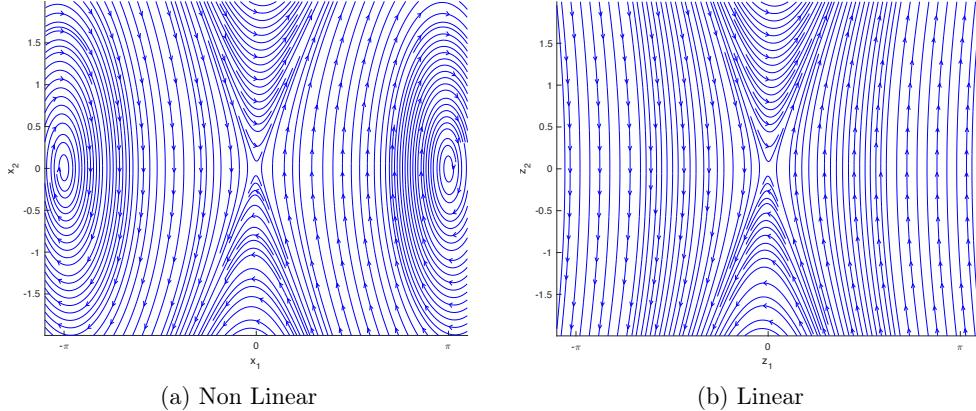


Figure 6.2: Phase portrait of the pendulum in the downward position (a) non linear and (b) linearised around $(x_1, x_2) = (0, 0)$

These 2D plots are useful for a visual interpretation of the physical behaviour of the system and to understand the impact of linearisation, but do represent a simplified version of the real inverted pendulum. To confirm the instability of the linearised system, the eigenvalues of the matrix A of the state-space description 3.3 can be computed.

$$\text{Spec}(A) = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{0, -8.3386, 7.1601, -0.1282\} \quad (6.3)$$

The eigenvalue λ_3 has a real positive part $\Re(\lambda_3) > 0$ and the system is thus unstable. Since the upright equilibrium at $(x_1, x_2) = (0, 0)$ is inherently unstable, any small perturbation causes the pendulum to diverge from this point. Therefore, the objective of the control strategy is to stabilize the pendulum at this unstable equilibrium by applying a control input that counteracts the system's natural tendency to fall away from the upright position. The design and analysis of such a controller relies on the linearized model around this point.

Before implementing a control method, one needs to check if the system is controllable. A system is controllable if it allows to bring the state from an initial value $x(0) = x_0$ to a final value $x(T) = x_1$ using an appropriate function $u(t)$ for $0 \leq t \leq T$. A system is completely controllable if the matrices A ($n \times n$) and B ($n \times r$) of the state space follow the following property [8]:

$$\text{Rank}[B \ AB \ \dots \ A^{n-1}B] = n \quad (6.4)$$

In this case, the rank computed in Matlab is indeed equal to $n = 3$, confirming the system's controllability.

6.2 LQR control

For an unstable, linearised system, classical control methods might not be very effective. Classical control methods involve stabilising the plant based on certain criteria and objectives like a desired transient response, bandwidth or stability margins. For linear high order systems, linear optimal control methods are recommended. Instead of following criteria (e.g. rise or settling time, robustness margins), the objective is to find the best possible control law based on the mathematical model of the system [8].

There are three possible procedures for linear optimal control:

- Full-state feedback design: all states are measured and available for feedback.
- State estimator design: some values of the states need estimation as they cannot be measured but the other measurements are available. The state estimator is used in addition to a full-state feedback design.
- Controller reduction: the concern is to approximate a complicated state estimate feedback controller obtained from the 2 previous steps by a simpler one. [8]

A full state feedback design is implemented as all the states (pendulum angle, velocity and motor speed) can be measured using the sensors of the prototype.

As explained in the State of the Art section, the Linear Quadratic Regulator (LQR) is a method of optimal control. The following development based on the book of *Introduction to Optimal Control* [27] provides a more detailed explanation of the conception and theory behind the LQR. For a linear time invariant system, with x the state of the system and u the control vector, the cost function $J(x, u)$ is expressed as follows [27]:

$$\begin{cases} \dot{x} = Ax + Bu \\ J(x, u) = \int_0^\infty (x^T Q x + u^T R u) dt \end{cases} \quad (6.5)$$

With,

$$\begin{cases} Q = Q^T \geq 0 \\ R = R^T > 0 \end{cases} \quad (6.6)$$

The optimal cost function can be expressed as the minimum of the cost function $J(x, u)$:

$$J^*(x, t) = \min_{u \in \mathbb{R}^2} J(x, u) \quad (6.7)$$

The Hamilton-Jacobi-Bellman (HJB) equation is defined as:

$$-\frac{\partial J^*(x, t)}{\partial t} = \min_{u \in \mathbb{R}^2} \left(F(x, u) + \left(\frac{\partial J^*(x, t)}{\partial x} \cdot \dot{x} \right) \right) \quad (6.8)$$

With,

$$J(x, u) = \int_0^\infty F(x, u) dt \quad (6.9)$$

A candidate solution for the optimal cost is expressed as following:

$$J^*(x, t) = x^T Px \quad (6.10)$$

With P a diagonal positive definite matrix ($P = P^T \geq 0$).

The partial derivatives of the optimal cost function are:

$$\begin{cases} \frac{\partial J^*}{\partial t} = x^T \dot{P} x \\ \frac{\partial J^*}{\partial x} = 2Px \end{cases} \quad (6.11)$$

The HJB Equation (6.8) can thus be written as:

$$-\frac{\partial J^*(x, t)}{\partial t} = -x^T \dot{P} x = \min_{u \in \mathbb{R}^2} (x^T Q x + u^T R u + 2x^T P(Ax + Bu)) \quad (6.12)$$

The minimum of the right-hand term is obtained by setting its partial derivative to zero:

$$\frac{\partial}{\partial u} (x^T Q x + u^T R u + 2x^T P(Ax + Bu)) = 0 \quad (6.13)$$

Leading to the solution:

$$u^* = -R^{-1}B^T Px \quad (6.14)$$

With u^* the optimal input that minimizes the quadratic cost of Equation (6.12).

Considering the property that P is symmetric¹, Equation (6.12) becomes:

$$-x^T \dot{P} x = x^T Q x - x^T P B R^{-1} B^T P x + x^T (P A + A^T P) x \quad (6.15)$$

Simplifying x^T and x the following equation called Riccati equation is obtained:

$$-\dot{P} = A^T P + P A - P B R^{-1} B^T P + Q \quad (6.16)$$

The control vector can be expressed using the Riccati equation:

$$u = -(R + B^T P B)^{-1} B^T P A x = -K_{LQR} x \quad (6.17)$$

A Linear Quadratic Regulator is implemented by choosing matrices Q and R . The matrix $Q \in \mathbb{R}^{3 \times 3}$ accounts for the performance of the system while $R \in \mathbb{R}$ accounts for the effort

¹The anti-symmetric part N of a matrix has a null contribution to a quadratic form $x^T N x = -x^T N x = 0$

provided by the actuator. The choice of matrices Q and R is crucial to obtain the proper regulation of the plant. Matrix K_{LQR} is called the gain of the LQR controller and can be computed using Matlab function `lqr()` that takes as input the state-space matrices A and B and the chosen matrices Q and R .

The block diagram of a LQR controller is presented in Fig.(6.3).

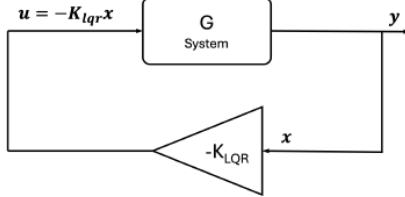


Figure 6.3: LQR block diagram

6.2.1 Choice of matrix Q and scalar R

Matrix Q is a 3×3 diagonal matrix that is multiplied by the state vector in the cost function. Each diagonal coefficient q_{ij} with $i = j$ of the matrix accounts for the importance accorded to a certain state during the regulation. By doing a simulation of the pendulum upside down, setting Q to the identity matrix and doing logarithmic variations of R , it is possible to estimate the coefficient between the pendulum angle and angular velocity. Scalar R was sampled 50 times with values spaced logarithmically between 10^6 to 10^{10} . Starting from angles of 5° , 10° , 15° and 20° , the dynamic behaviour is simulated and the mean of the ratio $\theta_p/\dot{\theta}_p$ is computed for each value of R . These values are exposed in Tab.(6.1).

Table 6.1: Mean value of $\theta_p/\dot{\theta}_p$ varying R from different initial angles

Initial angle $\theta_{p,0}$	$ \frac{\theta_p}{\dot{\theta}_p} $	R variation
5°	0.0324	$[10^6; 10^{10}]$
10°	0.0237	$[10^6; 10^{10}]$
15°	0.0107	$[10^6; 10^{10}]$
20°	0.0304	$[10^6; 10^{10}]$

As the matrix Q is multiplied by the square of the state vector in the cost function, the squared mean of the values is taken as the coefficient. The final coefficient is $\alpha = 5.9 \cdot 10^{-4}$ with $q_{11} = \alpha \cdot q_{22}$.

The base coefficients of matrix Q need to be selected on how much importance is given to each parameter. The priority in the regulation is on the pendulum angle so a coefficient of 100 is put in q_{11} . The same coefficient is chosen for the pendulum's angular velocity, multiplied by the coefficient α . For the motor's speed, a coefficient of 50 is set in q_{33} to

counteract external forces that disturb the pendulum.

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 \times \alpha & 0 \\ 0 & 0 & 50 \end{bmatrix} \quad (6.18)$$

The scalar R accounts for the use of the actuator. Since the saturation of the actuator is not taken into consideration in the explicit design of an LQR, it is indirectly addressed by tuning the R value. To that end, a simulation is done on Simulink to analyse the pendulum response when starting from a 5° angle. The gain K_{LQR} is computed for the matrix Q of Equation 6.18 and for $R = 10^6, \dots, R^{10}$. The results of this experiment are shown in Fig.(6.4).

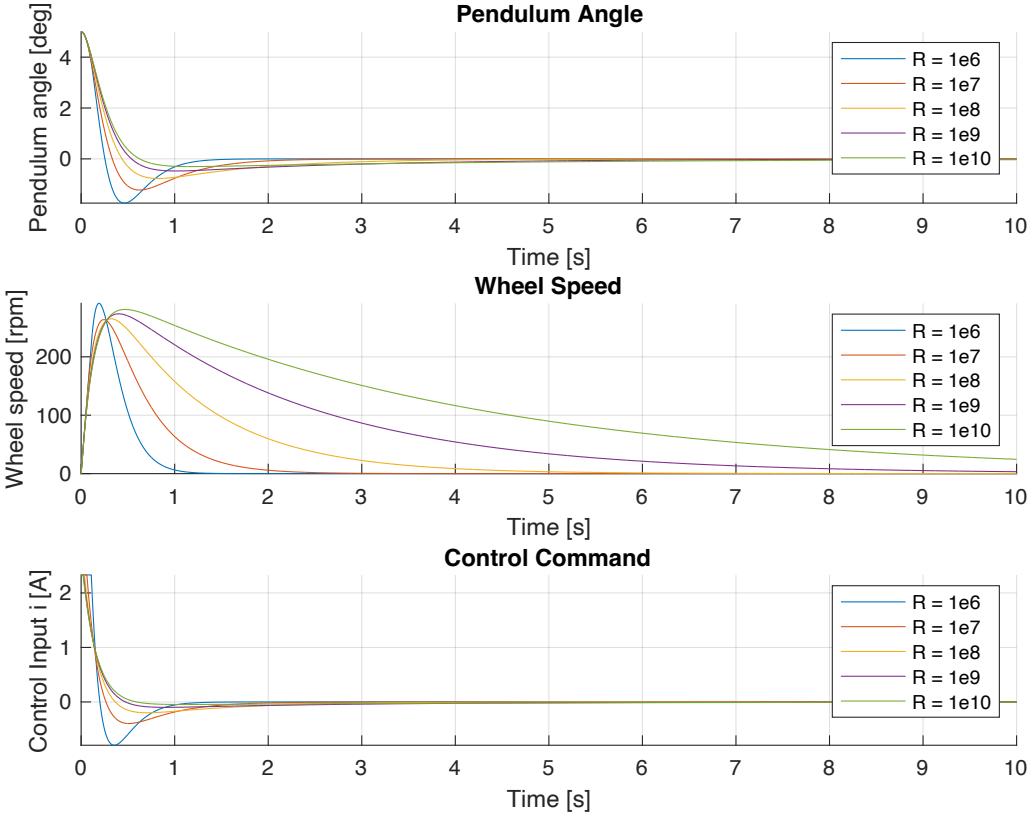


Figure 6.4: Analysis of pendulum angle, reaction wheel speed and control command for changes in scalar R

It can be observed that for smaller values of R , the system shows a more aggressive behaviour with a command at saturation (2.33 A) lasting longer which brings the pendulum back to 0° faster but also involves a bigger overshoot shown both in command and in pendulum angle. On the other hand, a higher R value will prolong the settling time but creates a smaller overshoot. To avoid long saturations of the actuator but still achieving a small settling time, the value of 10^7 is chosen for scalar R .

6.2.2 Validation of the LQR

Firstly, the stability of the controlled system is evaluated by computing the poles of the closed-loop system. The following development is taken from the book *Linear Optimal Control Systems* [20] and illustrates how the selection of Q and R matrices influences the placement of the closed-loop poles. In the analysis presented in the book, a Single Input Single Output system is presented with regulator characteristics $Q = 1$ and $R = \rho \cdot N$ with $N = 1$. The poles of the open-loop system are $\pi_i, i = 1, \dots, n$. For $\rho \rightarrow \infty$, the n closed-loop characteristic values approach the numbers $\hat{\pi}_i, i = 1, \dots, n$, where:

$$\hat{\pi}_i = \begin{cases} \pi_i & \text{if } \Re(\pi_i) \leq 0, \\ -\pi_i & \text{if } \Re(\pi_i) > 0. \end{cases} \quad (6.19)$$

The open-loop poles presented in Equation (6.3) are as follows:

$$p_1 = 0, \quad p_2 = -8.3386, \quad p_3 = 7.1601, \quad p_4 = -0.1282 \quad (6.20)$$

The closed-loop system with an LQR becomes:

$$\dot{x} = (A - BK)x \quad (6.21)$$

The closed-loop poles corresponding to the eigenvalues of $(A - BK)$ are:

$$p_1 = -0.4269, \quad p_2 = -8.5314, \quad p_3 = -7.1687, \quad p_4 = -2.0474 \quad (6.22)$$

After closing the loop, all the poles of the system have a negative real part, meaning the system indeed becomes stable. As stated in Equation (6.19), the closed-loop poles indeed approach the values of the open-loop poles, and the unstable pole p_3 is mirrored in the left half-plane.

6.2.3 Theoretical evaluation of the LQR

To assess the robustness of the LQR controller, Gain Margin (GM) and Phase Margin (PM) are computed based on its open-loop negative feedback frequency response. A Gain Margin of a system is defined as the amount by which the loop gain can be changed until the system becomes unstable. The Phase Margin of a system is the phase delay that can be added to the loop without losing stability. [8]

Chapter 5, *Properties of Regulator Systems with a Classical Control Interpretation*, in the book *Optimal Control* [8], presents a theoretical demonstration of the expected values of the GM and PM for closed-loop single-input systems designed using LQR techniques. It states the following properties:

1. The Gain Margin of such system is infinite.
2. The system has a downside margin of $\frac{1}{2}$ in worst-case scenario.
3. The Phase Margin is at least 60 deg.

To examine the robustness characteristics of the LQR, the open-loop with negative feedback scheme presented in Fig.(6.5) is analysed.

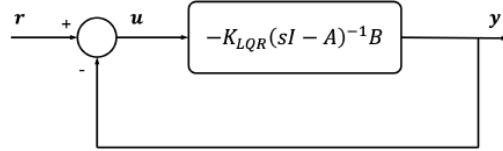


Figure 6.5: Open-loop system with K_{LQR} computed from LQR design [8]

The GM and PM can be analysed by looking at the Nyquist plot of the open-loop system displayed in Fig.(6.6). A Nyquist plot is a curve in the complex plane obtained from the complex values of $-K_{LQR}(j\omega I - A)^{-1}B$ as ω varies through the real numbers from $-\infty$ to $+\infty$ [8].

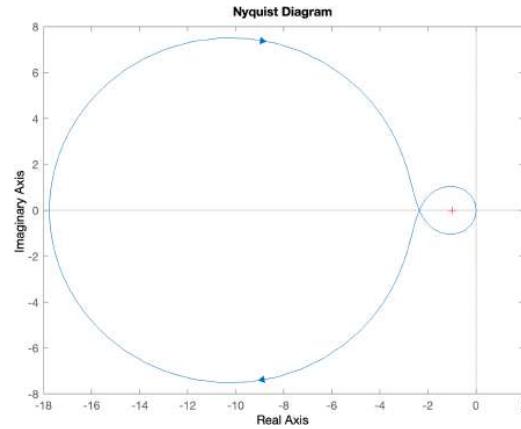


Figure 6.6: Nyquist plot of the open-loop system

Systems regulated by optimal controllers are known to be asymptotically stable². This property entails that the Nyquist plot encircles the point $-1 + j0$ in the anti-clockwise direction, with the number of encirclements corresponding to the number of poles of the transfer function $-K_{LQR}(j\omega I - A)^{-1}B$ that lie in the right-half plane. The downside margin can be checked by examining the Nyquist plot of the open-loop system with a gain $\gamma=0.4$. In this case, the system is no longer asymptotically stable and the curve does not encircle the red dot shown in Fig.(6.7a) in the counter-clockwise direction. The Gain Margin is verified using the `margin()` function from Matlab and is found to be -7.46 dB ($\simeq 0.42$).

The phase margin of the system is the amount of negative phase shift that must be introduced without gain increase to make the part of the Nyquist plot that corresponds to $\omega \geq 0$ pass through $-1 + j0$ [8]. Fig.(6.7b) shows the Nyquist plot for the system with a phase shift of 61.1° , corresponding to the Phase Margin. The curve indeed passes through the point $-1 + j0$ which confirms a PM of 61.1° .

²If initial conditions are close, the solutions converge toward each other as time approaches infinity [34]

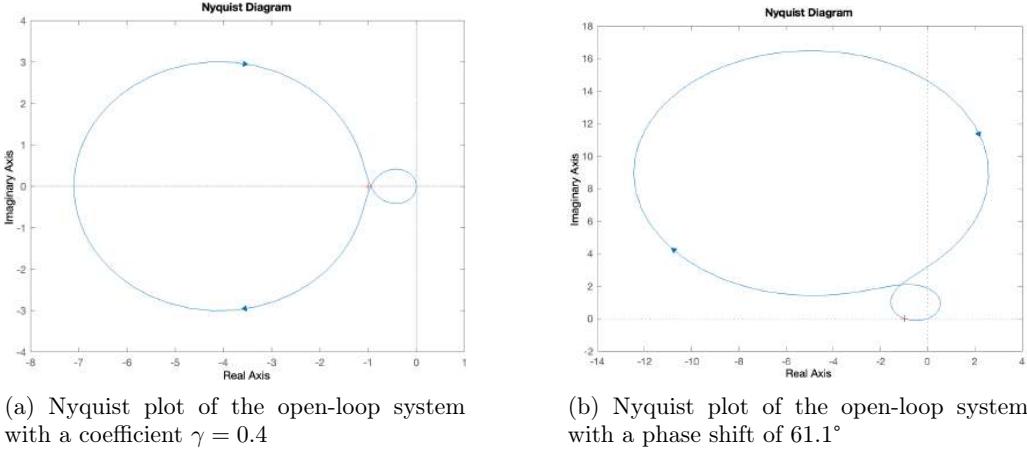


Figure 6.7: Comparison of Nyquist plots

The properties 1 and 2 can be verified using algorithm 1: for a selected range of gain γ , the poles of the closed-loop system are computed and the smallest gain for which the system remains stable is displayed. The algorithm returns a value of $\gamma = 0.42$, which is very close to the worst case scenario of $1/2$.

This algorithm can also be used to verify that when indefinitely increasing the gain K_{LQR} by a factor γ , the system remains stable: the gain margin is infinite. However, in practice, the maximum usable gain is limited by actuator saturation.

Algorithm 1 LQR Gain Scaling Stability Check

Require: State-space matrices A , B and LQR gain vector K_{LQR}

Ensure: Gain margin lower bound (smallest γ for which system remains stable)

```

1: Define a set of scaling factors  $\gamma \in [1, 0.1]$ 
2: for all  $\gamma$  in decreasing order do
3:    $L \leftarrow \gamma \cdot K_{LQR} \cdot (sI - A)^{-1} \cdot B$  Open-loop transfer function
4:    $T \leftarrow \text{feedback}(L, 1)$  Closed-loop system
5:   Compute poles of  $T$ 
6:   if all real parts of poles  $< 0$  then
7:     System is stable for this gain
8:   else
9:     System is unstable; record  $\gamma$  as lower gain margin
10:    break
11:   end if
12: end for

```

6.2.4 Experimental evaluation of the LQR

To evaluate the LQR, the regulation is implemented in an **Arduino** code and tested on the prototype. A demonstration of the regulated pendulum is shown in the following video: <https://www.youtube.com/shorts/VWwU4DCKMfk>.

In parallel, the regulator is tested in a **Simulink** simulation presented in Appendix F. Two regulators named LQR_1 and LQR_2 are compared:

- LQR_1 is designed using the initial model parameters used in the mathematical verification. The gain of the regulator is:

$$K_{LQR,1} = [-0.6862 \quad -0.0939 \quad -0.0032] \quad (6.23)$$

- LQR_2 is based on an updated set of parameters identified from experimental results obtained with LQR_1 . The details of how each parameter was adapted is available in Appendix D. The gain of the regulator is:

$$K_{LQR,2} = [-0.6538 \quad -0.0763 \quad -0.0032] \quad (6.24)$$

Both experimental and simulation setups start with a pendulum angle of 5° and the pendulum angle, reaction wheel speed and current command responses are monitored. It can be seen in Fig.(6.8a) that the experimental pendulum reacts with the same overshoot as the simulation but much faster. The dynamic of the motor shown in Fig.(6.9a) also seems to fit with the initial simulation. However, the experimental command in Fig.(6.10a) sent to the motor has a much bigger amplitude during the overshoot (-2A versus -0.4A) to recover the pendulum. As a consequence, the pendulum overshoots in the opposite direction, requiring an additional corrective input to bring it back to the upright position at 0° .

The issue might lie in an error in dimensioning the parameters of the system. To that end, it was decided to try a different set of parameters in which the pendulum inertia I is reduced to $0.11 \text{ kg} \cdot \text{m}^2$, the pendulum friction b_p is increased to $0.03 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$ and the time constant of the motor is changed to $T = 6.8\text{s}$. The results of this new regulator LQR_2 are presented in Fig.(6.8b), Fig.(6.9b) and Fig.(6.10b).

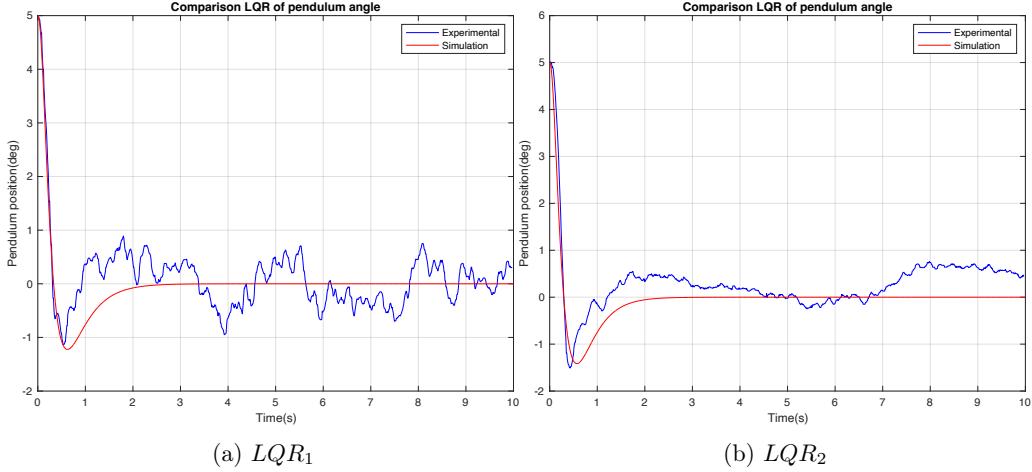
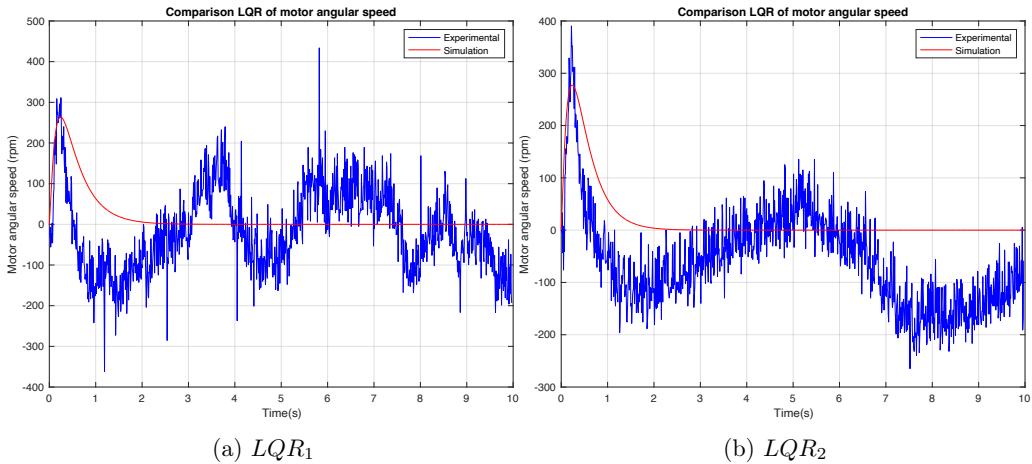
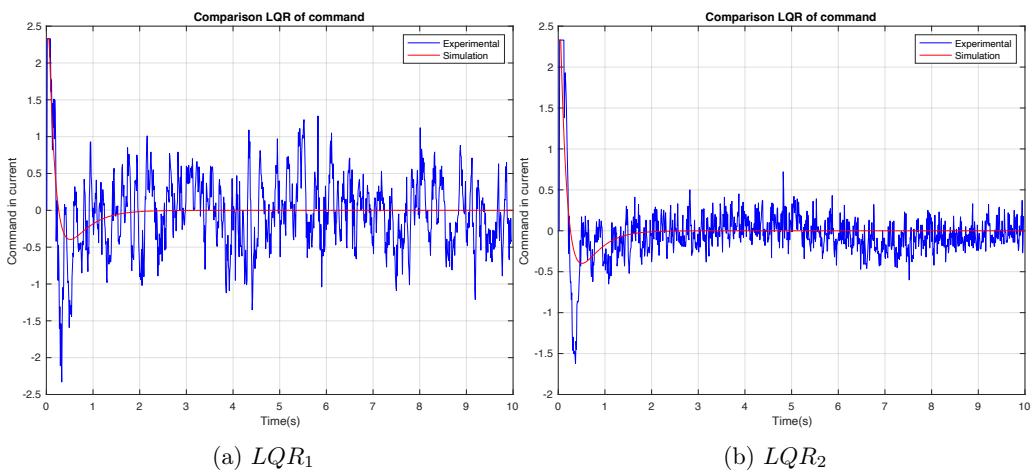
A comparison of 2 characteristics is present in Tab.(6.2): the rise time and the overshoot. The rise time is the time it takes for the response to rise from 10% to 90% of the way from the initial angle value (5°) to the final one (0°) [19]. The overshoot is a percentage of the signal exceeding the target value of 0° . The first LQR has a longer rising time of 0.21 second compared to 0.17 second for the second one, but a smaller overshoot of 22.6% compared to 30.2%.

While LQR_2 responds more aggressively, it also introduces a larger deviation from the target in the first second of the experiment. In contrast, LQR_1 applies a less aggressive control action, resulting in reduced overshoot but a slightly slower response.

It can also be observed that LQR_2 achieves a more stable behaviour in the remaining seconds of the simulations, with smaller control inputs and reduced oscillations in the pendulum angle. For both simulations, the cables, positioned on the positive angle side, exert tension that draws the pendulum slightly in that direction.

Table 6.2: Characteristics of the pendulum 5° angle recovery

Characteristic	Value for LQR_1	Value for LQR_2
Rise time	0.21s	0.17s
Overshoot	22.6%	30.2 %

Figure 6.8: Comparison of pendulum angle for an initial 5° angle.Figure 6.9: Comparison of motor speed for an initial 5° angle.Figure 6.10: Comparison of current command sent to the motor for an initial 5° angle.

As previously mentioned, the LQR is evaluated based on a set of criteria: the maximal recovery angle, the force which can be applied to the system and the motor power consumption. In the following sections, each of those criteria will be analysed for the 2 controllers implemented LQR_1 and LQR_2 .

Recovery angle

The recovery angle is the maximal initial pendulum angle for which the motor is able to bring back to upright position. It was found experimentally that an angle 11° is the maximal angle that can be recovered for both regulators. Beyond this angle, the pendulum overshoots excessively and ultimately falls to the opposite side. The experiment conducted for an 11° recovery angle is plotted in Fig.(6.11).

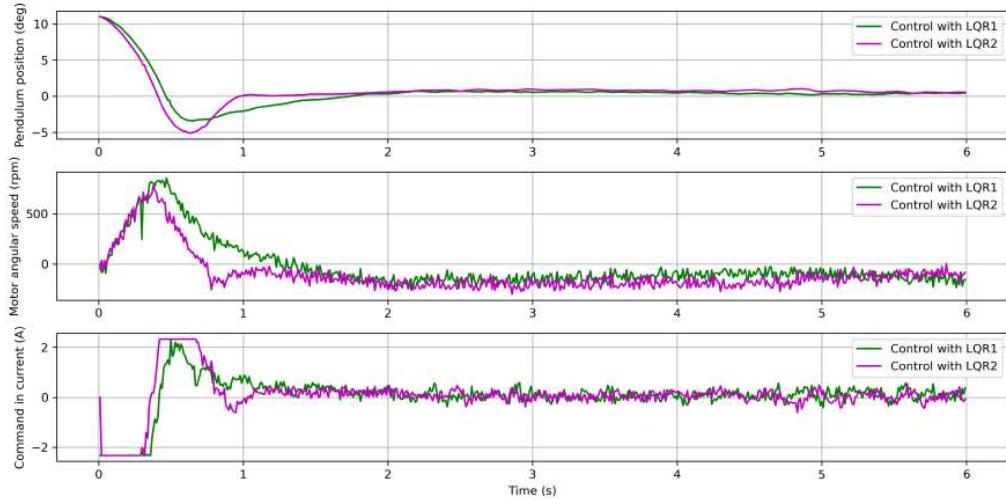


Figure 6.11: Experimental prototype behaviour for an initial angle of 11° with regulators LQR_1 and LQR_2

This observation further confirms the earlier analysis: LQR_2 exhibits a more aggressive response, characterized by a larger overshoot and faster return to the upright position. However, this comes at the cost of a longer time spent in current saturation at 2.33A .

Distrubance torque

The system's response is evaluated based on a torque applied on the pendulum. A 3D-printed PLA piece was designed to be placed on the pendulum axis and host various weights as shown in Fig.(6.12). A weight of 93g is placed on this structure, leading to a disturbance torque τ_d of:

$$\tau_d = m \cdot g \cdot d = 93 \cdot 10^{-3} \cdot 9.81 \cdot 65 \cdot 10^{-3} = 59.30145 \text{ mNm} \quad (6.25)$$

With d the length of lever arm of value 65mm , m the mass of the suspended weight, and g the gravitational acceleration of 9.81m/s^2 .

Bellow in Fig.(6.13) is the pendulum response to such a torque applied around second 4 and left on the pendulum for the rest of the experiment.



Figure 6.12: Experimental setup for adding a torque disturbance to the system

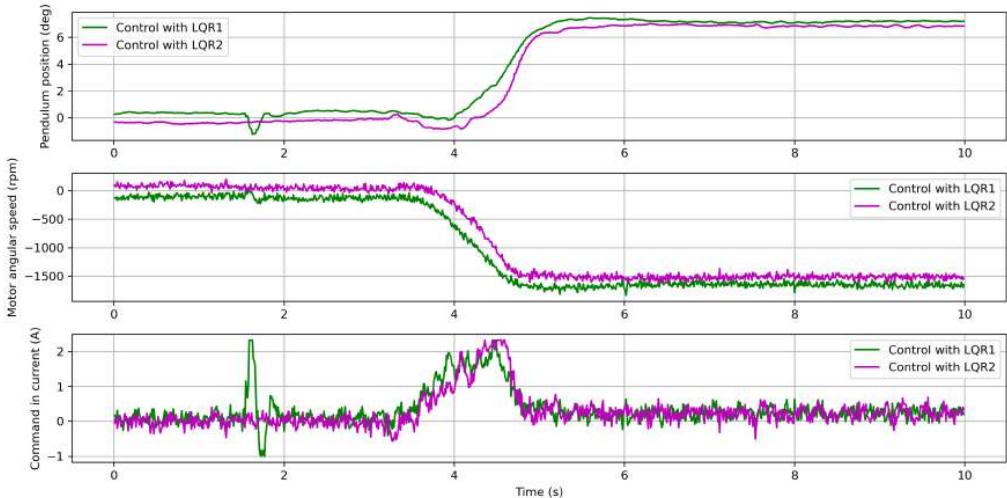


Figure 6.13: Experimental prototype response to a torque τ_d for LQR_1 and LQR_2

With the weight placed on the negative angle side, both regulators succeed in stabilizing the pendulum. However, due to the resulting torque, the pendulum settles at a tilted equilibrium around 6.5° to counterbalance the added load.

The test was also conducted with smaller weights of 9g and 47g and a similar behaviour was observed. However for the 9g weight, the pendulum remained at a 0° angle during the time of the experiment. Tests with heavier weights were not conducted to prevent potential damage to the prototype.

6.2.5 Motor power consumption

Another indicator of the regulator performance is to compute the motor's average power consumption during the time of the experiment. The electrical power of the motor is written as [24]:

$$P_{el} = P_{mech} + P_J = \frac{\pi}{30000} M \cdot n + R i_{motor}^2 \quad (6.26)$$

With,

$$\begin{cases} M = k_m \cdot i_{motor} \\ n = k_n \cdot U_{ind} \end{cases} \quad (6.27)$$

Constants k_m , k_n and R are found in the motor's datasheet while the current i_{motor} and the voltage induced in the windings U_{ind} must be measured.

However, the voltage U_{ind} could not easily be measured on the prototype because of restricted access to the windings so it was decided to compute the average current sent to the motor during the recovery angle experiment for both regulators. While the average current command sent to the motor is not a direct measure of power, it provides a useful indication of the system's electrical consumption and the effort required from the actuator during regulation. Given that the current alternates between positive and negative values during the experiment, the root mean square (RMS) current was computed instead of a simple mean. This provides a more accurate estimate of the effective current and, by extension, the actuator's electrical effort. The RMS current for LQR_1 is equal to 0.528A, while the one for LQR_2 is worth 0.371A. This suggests that LQR_2 provides a more balanced and energy-efficient actuator effort over time compared to LQR_1 , for the same experimental conditions.

Overall, both regulators successfully stabilize the pendulum under identical conditions. However, LQR_2 demonstrates a faster and more aggressive response, achieving quicker stabilization at the cost of longer current saturation. In contrast, LQR_1 exhibits a slower response with reduced overshoot, resulting in smoother but more sustained control effort, as reflected by its higher RMS current. This implies that LQR_2 offers a better trade-off between performance and energy efficiency. These observations also underline the importance of the mathematical model used for controller design, as slight variations can lead to significantly different control behaviours.

6.3 Cascade PID control

The cascade PID control method presented in the State of the Art section works by creating a fast-response inner loop that controls the motor speed and an outer loop that returns the pendulum angle back to 0° . This configuration is represented in Fig.(6.14) and the corresponding transfer functions are detailed in Tab.(6.3).

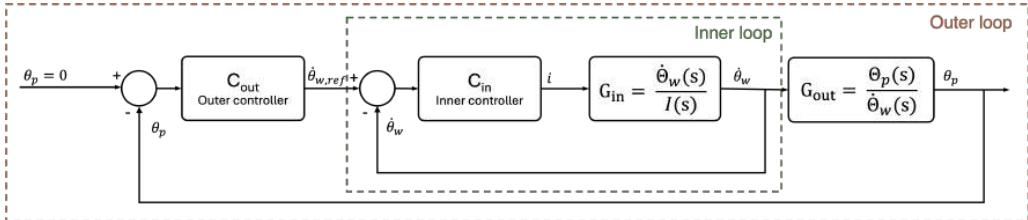


Figure 6.14: Cascade PID configuration

Table 6.3: Transfer functions of the system

Outer loop controller	$C_{out} = \frac{\dot{\Theta}_{w,ref}(s)}{E_p(s)}$
Pendulum angle - Motor angular velocity	$G_{out} = \frac{\Theta_p(s)}{\dot{\Theta}_w(s)}$
Inner loop controller	$C_{in} = \frac{\dot{\Theta}_w(s)}{E_w(s)}$
Motor angular velocity - Current	$G_{in} = \frac{\dot{\Theta}_w(s)}{I(s)}$

Where $\dot{\Theta}_w$ represents the motor angular velocity, Θ_p the pendulum angle, E_p the error between the actual pendulum angle and the reference angle of 0° , E_w the error between the actual motor speed and the reference speed and finally I the current sent to the motor. Instead of computing a regulator for the inner loop, the closed loop speed control of the ESCON is used. By comparing the actual speed to the reference value, the speed is dynamically adjusted.

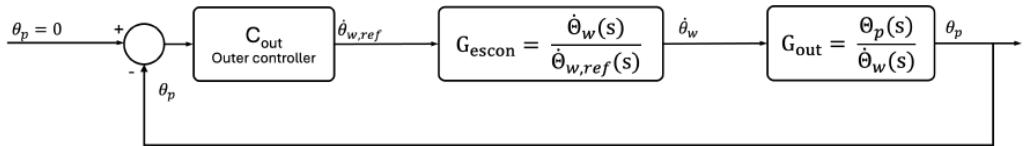
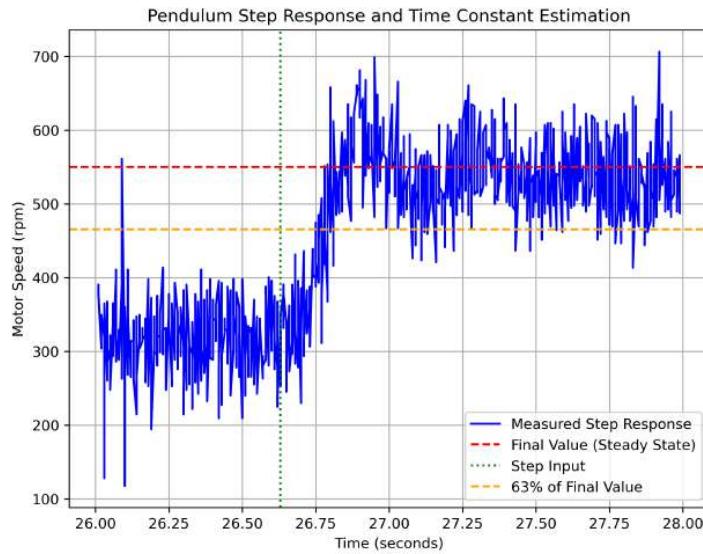


Figure 6.15: Cascade PID configuration with ESCON closed loop speed control

The motor is thus modulated in speed and no longer in current and the transfer function of C_{in} and G_{in} combined called G_{escon} is computed by doing a step response in speed of the motor. A step response of command 450 rpm to 675 rpm is illustrated in Fig.(6.16).

Figure 6.16: Visual estimation of parameters G and τ on Python

Using the same optimisation algorithm as presented for current in section 5.2.2, the following transfer function is obtained. The details of the optimisation are available in Appendix B.

$$G_{escon} = \frac{\dot{\Theta}_w(s)}{\dot{\Theta}_{w,ref}(s)} = \frac{1.049}{0.2s + 1} \quad (6.28)$$

The outer loop transfer function that links the motor's angular velocity to the pendulum's angle can be recovered using Eq.(3.15):

$$G_{out} = \frac{\Theta_p(s)}{\dot{\Theta}_w(s)} = \frac{s}{-\hat{I}s^2 + (\frac{b_p}{\hat{I}} - b_p\hat{I})s + K\hat{I} - \frac{K}{\hat{I}}} = \frac{s}{-2.63s^2 - 1.43s + 89.167} \quad (6.29)$$

The transfer function of the system is defined as the product of the motor's transfer function (G_{escon}) and the transfer function between the pendulum angle and the motor angular velocity (G_{out}):

$$G_{system} = G_{escon} \cdot G_{out} = \frac{-1.049s}{0.9333s^3 + 6.76s^2 - 29.33s - 199} \quad (6.30)$$

The system has an unstable pole at $s = 5.5043$, two poles in the left plane $s = -7.7472$ and $s = -5$, and a zero at the origin. In Fig.(6.17) is plotted the root locus of G_{system} with a simple P regulator. Such regulator is not enough to bring back the positive pole to the left plane of the root locus. As a result, a PID controller was selected, in line with the approaches presented in the State of the Art.

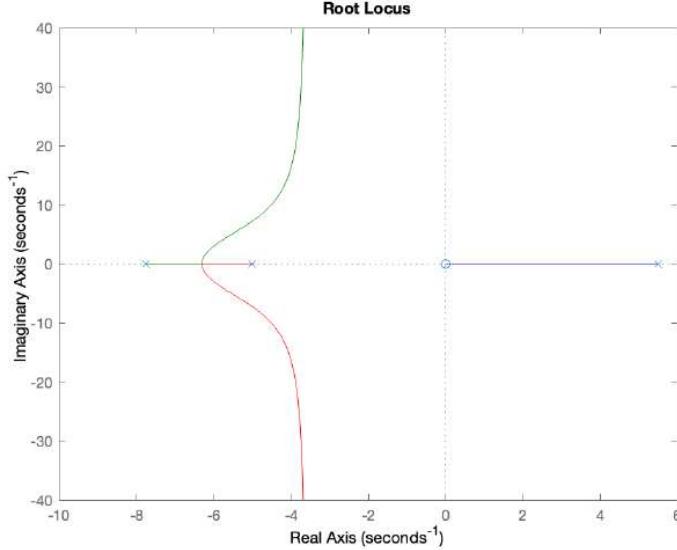


Figure 6.17: Root locus of the system transfer function of the system (G_{system}) coupled to a P controller

The PID was tuned by using the `rltool()` from Matlab. This function creates an interface in which the user can add/remove poles and zeros on the root locus of the system. By modifying the root locus in this way, it can be shifted into the left half of the complex plane, thereby stabilizing the system. To that end, the poles $p = 0$ and $p = -54526$ and zeros $z = -17.6229$ and $z = -12.6996$ are added to create the following control transfer

function:

$$C(s) = \frac{-42.18s^2 - 1279s - 9440}{0.0001834s^2 + s} \quad (6.31)$$

The root locus of the open-loop $C \cdot G_{\text{system}}$ is plotted in Fig.(6.18).

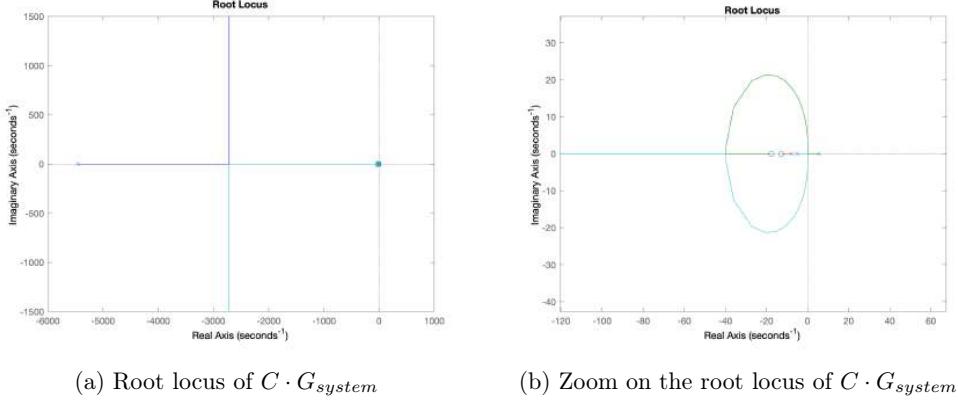


Figure 6.18: Root locus of $C \cdot G_{\text{system}}$ computed with `rlocus()` function

An issue however with regulator $C(s)$ is that when combined with the plant transfer function G_{system} , the following transfer function is obtained:

$$\begin{aligned} C(s) \cdot G_{\text{system}} &= \frac{44.25s^3 + 1342s^2 + 9903s}{0.0001712s^5 + 0.9345s^4 + 6.755s^3 - 29.37s^2 - 199s} \\ &= \frac{44.25s^2 + 1342s + 9903}{0.0001712s^4 + 0.9345s^3 + 6.755s^2 - 29.37s - 199} \end{aligned} \quad (6.32)$$

As seen in the equation above, a pole-zero cancellation occurs at the origin due to the plant's zero and the controller's integrator pole. Such simplification often leads to a loss of observability and, in practice, do not result in perfect cancellation due to model inaccuracies [19]. However, since stability could not be achieved without the integrator, it was retained, with awareness of the potential issues it may introduce.

The Gain Margin and Phase Margin of the system can be analysed by plotting the Bode diagrams in Fig.(6.19). The Gain Margin of the open-loop system is -30.25dB and the Phase Margin is 64.33°. With a similar algorithm to algorithm 1, the poles of the open-loop system can be computed to verify stability with a coefficient γ multiplying the controller. It was observed that below a coefficient of 0.03 (-30dB) the PID could not stabilize the system and increasing γ above values of 1 introduced an oscillatory behaviour in the system.

The robustness margins also need verification when the regulator is implemented in **Arduino**, meaning a sampling frequency is chosen. The bandwidth of the system coupled with the PID controller is 66 rad/s (computed with `bandwidth()` function from **Matlab**) which corresponds to 10.6Hz. Following Nyquist theorem, the sampling frequency should be of at least twice the system bandwidth, 21.2Hz. A sampling frequency of 100Hz was chosen, which is well above the required minimum. By adding a delay in the open-loop as described in Equation (6.33), the Bode diagrams of the discretized system are plotted in Fig.(6.20).

$$H(s) = C \cdot G_{\text{system}} \cdot e^{-0.01s} \quad (6.33)$$

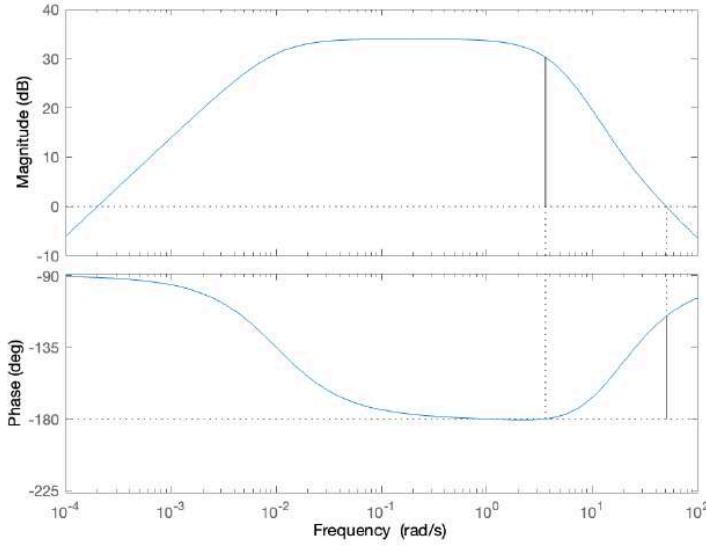


Figure 6.19: Bode plots of the system coupled to a PID

The Gain Margin is now equal to -26.24dB instead of -30.25dB , and the Phase Margin is reduced to 35.3° instead of 64.3° . While these changes indicate a reduction in robustness margins due to discretization, the regulator still ensures acceptable stability. Reducing the sampling period to 0.001s for example can further align the discrete-time performance with the continuous-time design.

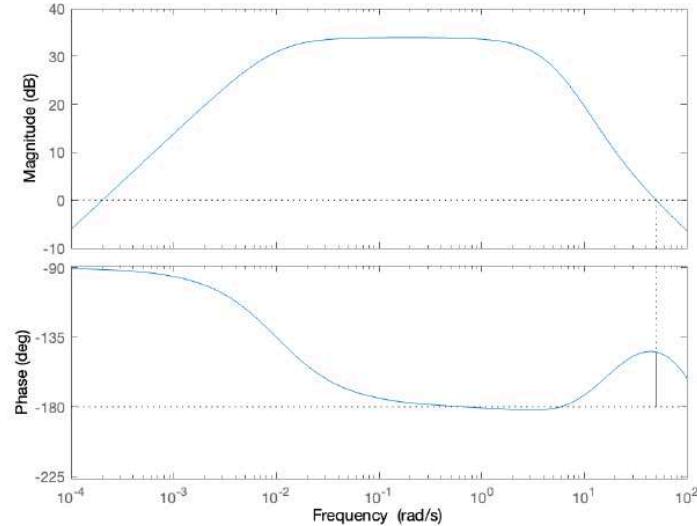


Figure 6.20: Bode plots of the system coupled to a PID discretised

The PID controller operates by acting on the error in the pendulum angle over time. To implement it digitally, the controller must be discretized. To that end, the Tustin discretization method available in **Matlab** is often used and is known to preserve stability [19].

The system was discretized using the `c2d()` function. Both the continuous and discretized systems were subjected to the same disturbance input, and the corresponding responses

are compared in Fig.(6.21). The two curves overlap throughout the duration of the experiment, indicating that the discrete controller replicates the behaviour of the continuous system and effectively rejects disturbances.

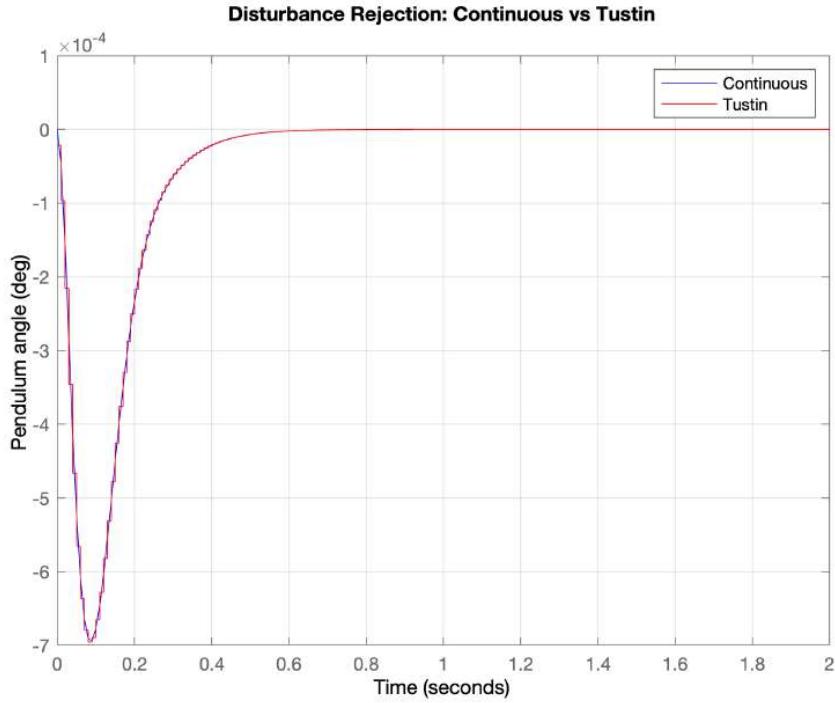


Figure 6.21: Disturbance rejection comparison of continuous and discrete (Tustin) systems

Based on this simulation and the computed robustness margins, it can be concluded that the simulated cascade PID regulator is capable of stabilising the pendulum in the upright position.

In practice, the cascade PID regulator was unable to stabilise the pendulum due to irregularities in the sampling frequency. Despite implementing a wait loop in the Arduino code to enforce a fixed sampling period, the actual sampling remained non-uniform.

6.4 Discussion of control strategies

The performances of both controllers were compared by subjecting the simulation systems to the same torque disturbance τ_d . The LQR simulation was conducted using LQR_2 presented in section 6.2.4.

As seen in Fig.(6.22), the PID system responds significantly faster. This is because it generates a speed command using ESCON controller, which includes a complex internal speed regulation loop. In contrast, the LQR outputs a current command, which results in slower dynamics.

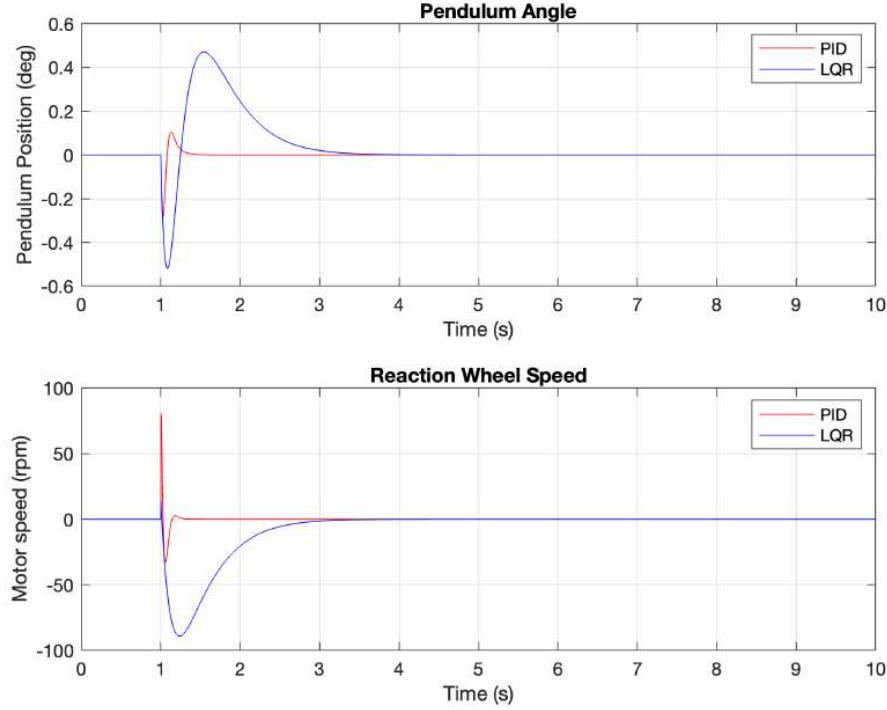


Figure 6.22: Comparison of the system controlled by a LQR and a PID controller under the same disturbance torque

The faster response of the PID controller leads to a smaller deviation in the pendulum angle and a quicker return to the upright position compared to the LQR. However, the simplification of the pole-zero placement in the outer-loop makes the controller's behaviour potentially unpredictable when implemented on the physical system. Additionally, its strong dependence on the sampling frequency for discretization rendered it unsuitable for reliable implementation on the Arduino platform. The cascade PID also relies on the ESCON's internal speed regulation, which limits visibility into the full control process and reduces its educational value.

The LQR, on the other hand, offers a slower response but demonstrates promising performance when implemented on a real plant, achieving recovery from angles up to 11° and withstanding constant disturbances of up to 59mNm.

Chapter 7

Conclusion

This project focused on the design, construction, and control of a didactic reaction wheel actuated inverted pendulum for teaching control engineering. The work began with a review of similar academic projects, analysing their prototypes, mathematical models, and control strategies.

A mathematical model was then developed, complex enough to incorporate system frictions and non-linearities, with the goal of achieving a close match with experimental behaviour. The prototype was then designed drawing from literature, supervisor input, and what was already available at the SAAS lab. Careful consideration went into the mechanical design, as placing the motor at the top of the pendulum made the overall stability highly sensitive to design choices. Several elements were intentionally made adaptable, such as the pendulum height and the reaction wheel's inertia, to allow for experimentation and tuning of model parameters. Component selection was interdependent, driven by the torque-to-weight ratio of the motor and was supported by initial simulations.

Once the prototype was assembled, its physical parameters, such as lengths, masses, inertias, friction coefficients, and motor gains, were experimentally identified to complete and validate the model numerically. To assess how well the model represented the real prototype, its output was compared with experimental data under identical input conditions.

Two control strategies were then explored: an optimal Linear Quadratic Regulator and a cascade PID controller. The LQR was implemented by selecting appropriate weighting matrix Q and scalar R , computing the optimal gain, and closing the loop accordingly. A theoretical robustness analysis was carried out using Nyquist plots, along with experimental evaluations including the recovery angle, response to disturbances, and power consumption (estimated via the average motor current).

Moreover, a cascade PID controller was investigated. The cascade architecture allows to quickly regulate the motor's speed in the inner loop before stabilising the pendulum angle in the outer loop. Its theoretical robustness margins were analysed using Bode diagrams, but experimental implementation was limited by non-uniform sampling on `Arduino`. Additionally, the use of the ESCON driver's built-in speed control in the inner loop reduced the pedagogical value of the cascade PID approach, as it hides part of the regulation design from students.

The outcome of this project is a working reaction-wheel pendulum that not only stabilizes effectively using linear quadratic control but also offers a compact and functional prototype. While both cascade PID control and LQR were investigated, only the LQR approach resulted in a fully functional system, offering both a successful stabilization strategy and a clear, structured learning experience. The system serves as a practical platform for exploring the challenges of controlling non-linear, unstable systems and contributes to hands-on learning in control engineering.

7.1 Further improvements

Despite these achievements, there remain several opportunities for further enhancement and refinement. The next section outlines key improvement ideas.

Firstly, the motor could be more fully exploited: saturation was set to the nominal current to ensure safety, but increasing this limit would allow for stronger control actions. To protect the circuit from potential voltage spikes caused by back electromotive force (back-EMF) during sudden torque changes, a capacitor could be added across the motor driver's power input [29]. An additional refinement would be to formally characterize the motor's dead zone. Although it seemed negligible in practice, evaluating it could improve the model's fidelity and, once integrated into the control law, enhance the regulator's performance.

From a modelling perspective, a mismatch was observed between experimental results and simulation outputs that could not be resolved through parameter tuning alone. A deeper analysis could involve modelling the pendulum as a non-rigid body, allowing the system to capture the vibrations induced by strong, sudden torque inputs.

Regarding control strategies, the cascade PID controller faced practical challenges due to non-uniform sampling on the **Arduino**, limiting its experimental evaluation. Implementing the PID controller in **Matlab**, where a consistent sampling rate can be ensured, could enable experimental validation. Additionally, from a pedagogical standpoint, removing the ESCON driver's built-in speed control would allow students to directly implement and experiment with motor speed regulation and implement the cascade architecture entirely. Similarly, the current LQR could be extended to an LQG (Linear Quadratic Gaussian) controller, introducing students to state observers concepts.

Finally, to create a more impressive and portable prototype, the system could be expanded to use two motors, enabling 3D stabilization of the pendulum. This would eliminate the need for the current mechanical base, making the system easier to transport and set up while maintaining similar control laws in both directions.

The **Matlab**, **Arduino** codes, **Simulink** simulations and **Fusion360** files developed during this project are available in the GitHub: <https://github.com/ClaraNeuburger/Master-thesis-Inverted-Pendulum-with-Reaction-Wheel>

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author used ChatGPT in order to generate short `Python` code snippets for data visualisation of results from text files recovered from `Arduino`, and to improve the clarity and precision of written content.

After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Bibliography

- [1] Acrome, *Linear inverted pendulum*, 2025. [Online]. Available: <https://acrome.net/product/linear-inverted-pendulum>.
- [2] A. Ambroziak and M. Solarczyk, “Application and mechanical properties of aluminium alloys”, in Jan. 2018, pp. 525–528, ISBN: 978-1-138-05045-7. DOI: 10.1201/9781315166605-121.
- [3] Arduino, *Arduino® DUE product reference manual*. [Online]. Available: <https://docs.arduino.cc/resources/datasheets/A000062-datasheet.pdf>.
- [4] Arduino, *Arduino® UNO R3 product reference manual*. [Online]. Available: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>.
- [5] ASPINA Group, *What is a brushed dc motor?* [Online]. Available: <https://eu.aspinagroup.com/en/learning-zone/columns/what-is/031/>.
- [6] F. L. Bars, *Observateurs et filtre de kalman*, Lecture Notes, Course at ENSTA Bretagne.
- [7] P. Brevik, “Two-axis reaction wheel inverted pendulum”, *NTNU*, 2017. [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2451060>.
- [8] A. Brian and M. John, *Optimal Control - Linear Quadratic Methods*. Prentice-Hall International Editions, 1989.
- [9] M. V. Burakov, I. G. Krivolapchuk, V. F. Shishlakov, and D. O. Yakimovskii, “Fuzzy control of spacecraft reaction wheel”, *Gyroscopy and Navigation*, vol. 10, no. 4, pp. 339–345, 2019, ISSN: 2075-1109. DOI: 10.1134/S2075108719040072. [Online]. Available: <https://doi.org/10.1134/S2075108719040072>.
- [10] B. Douglas, *Linear quadratic regulator*, Video on the Matlab youtube account, 2019. [Online]. Available: https://www.youtube.com/watch?v=E_RDCF01Jx4.
- [11] S. Farah, D. G. Anderson, and R. Langer, “Physical and mechanical properties of pla, and their functions in widespread applications — a comprehensive review”, *Advanced Drug Delivery Reviews*, vol. 107, pp. 367–392, 2016, PLA biodegradable polymers, ISSN: 0169-409X. DOI: <https://doi.org/10.1016/j.addr.2016.06.012>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169409X16302058>.
- [12] Farnell, *Sen0142 - capteur breakout, capteur mems gyroscopique et accéléromètre 6 axes mpu-6050*. [Online]. Available: <https://be.farnell.com/fr-BE/dfrobot/sen0142/6-dof-sensor-arduino-board/dp/3769961>.
- [13] P. Henneaux, *Syllabus électricité appliquée (ELEC-H3001)*, Lecture Notes, Presses universitaires de Bruxelles (PUB), 2023.
- [14] Y.-F. Ho, T.-H. S. Li, P.-H. Kuo, and Y.-T. Ye, “Parameterized gait pattern generator based on linear inverted pendulum model with natural zmp references”, *The Knowledge Engineering Review*, vol. 32, e3, 2017. DOI: 10.1017/S0269888916000138.
- [15] B. Innocenti, *Robot dynamics (MATH-H509)*, Lecture Notes, Presses universitaires de Bruxelles (PUB), 2025.

- [16] B. José and C. Taddy, *Arduino memory guide*, 2023. [Online]. Available: <https://docs.arduino.cc/learn/programming/memory-guide/>.
- [17] jrowberg, *Github of i2cdevlib*. [Online]. Available: <https://github.com/jrowberg/i2cdevlib>.
- [18] A. Kar and M. Richard, in *Feedback System - An Introduction for Scientists and Engineers SECOND EDITION*, Princeton University press, 2015, ch. 4. [Online]. Available: https://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-dynamics_04Jun15.pdf.
- [19] M. Kinnaert, *Automatique (ELEC-H3002)*, Lecture Notes, Presses universitaires de Bruxelles (PUB), 2023.
- [20] H. Kwakernaak and R. Sivan, *Linear optimal control systems*, eng. New York: Wiley-Interscience, 1972, ISBN: 0471511102.
- [21] P. Lambert, *Mécanique rationnelle II (MECA-H200)*, Lecture Notes, Presses universitaires de Bruxelles (PUB), 2021.
- [22] Lim, Yon Yaw, Hoo, Choon Lih, and Felicia Wong, Yen Myan, “Stabilising an inverted pendulum with pid controller”, *MATEC Web Conf.*, vol. 152, p. 02009, 2018. DOI: 10.1051/matecconf/201815202009. [Online]. Available: <https://doi.org/10.1051/matecconf/201815202009>.
- [23] Maxon, *Escon overview*. [Online]. Available: <https://www.farnell.com/datasheets/1860324.pdf>.
- [24] Maxon, *Formulae handbook*, Formulae for DC and EC maxon motors.
- [25] Maxon, *Product range 2024-25*, Maxon selection guide catalog.
- [26] MaxonAcademy, *Maxon ec motor - an introduction to brushless dc motors*. [Online]. Available: https://www.maxongroup.com/medias/sys_master/8803451338782.pdf?attachment=true.
- [27] T. Miquel, “Introduction to Optimal Control”, Master, Lecture, ENAC, France, Oct. 2022. [Online]. Available: <https://cel.hal.science/hal-02987731>.
- [28] S. A. Mirdehghan, “1 - fibrous polymeric composites”, in *Engineered Polymeric Fibrous Materials*, ser. The Textile Institute Book Series, M. Latifi, Ed., Woodhead Publishing, 2021, pp. 1–58, ISBN: 978-0-12-824381-7. DOI: <https://doi.org/10.1016/B978-0-12-824381-7.00012-3>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128243817000123>.
- [29] Nanotec Electronic GmbH & Co. KG, *Application note – back-emf protection circuits*, 2022. [Online]. Available: https://www.nanotec.com/fileadmin/files/Application_Notes/back-emf/Back_EMF_Protection_Application_Note_EN_V1.0.0.pdf?1656012519.
- [30] F. Peker, I. Kaya, E. Cokmez, and S. Atic, “Cascade control approach for a cart inverted pendulum system using controller synthesis method”, in *2018 26th Mediterranean Conference on Control and Automation (MED)*, Zadar, Croatia: IEEE, 2018, pp. 1–6. DOI: 10.1109/MED.2018.8442453.
- [31] Quanser, *Rotary inverted pendulum*, 2025. [Online]. Available: <https://www.quanser.com/products/rotary-inverted-pendulum/>.

- [32] C. Richard, *Brushless, permanent-magnet motors and controllers*. Oxford science publication, 1998, ch. 5.
- [33] STMicroelectronics, *Datasheet - l298 - dual full-bridge driver*. [Online]. Available: https://www.st.com/en/motor-drivers/l298.html?ecmp=tt9470_gl_link_feb2019&rt=ds&id=DS0218#documentation.
- [34] L. Trajković, “5 - nonlinear circuits”, in *The Electrical Engineering Handbook*, W.-K. CHEN, Ed., Burlington: Academic Press, 2005, pp. 75–81, ISBN: 978-0-12-170960-0. DOI: <https://doi.org/10.1016/B978-012170960-0/50008-6>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780121709600500086>.
- [35] U. Tugcular and M. Ucar, *Control of an inverted pendulum with a reaction wheel*, May 2019. DOI: [10.13140/RG.2.2.20782.74565](https://doi.org/10.13140/RG.2.2.20782.74565).
- [36] A. Türkmen, M. Y. Korkut, M. Erdem, Ö. Gönül, and V. Sezer, “Design, implementation and control of dual axis self balancing inverted pendulum using reaction wheels”, in *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, 2017, pp. 717–721.
- [37] R. Vilanova and O. Arrieta, “Pid tuning for cascade control system design”, Jun. 2008, pp. 001 775–001 778. DOI: [10.1109/CCECE.2008.4564850](https://doi.org/10.1109/CCECE.2008.4564850).
- [38] M. Vukobratovic and B. Borovac, “Zero-moment point - thirty five years of its life.”, *I. J. Humanoid Robotics*, vol. 1, pp. 157–173, Mar. 2004. DOI: [10.1142/S0219843604000083](https://doi.org/10.1142/S0219843604000083).
- [39] D. Zaborniak, K. Patan, and M. Witczak, “Design, implementation, and control of a wheel-based inverted pendulum”, *Electronics*, vol. 13, no. 3, 2024, ISSN: 2079-9292. DOI: [10.3390/electronics13030514](https://doi.org/10.3390/electronics13030514). [Online]. Available: <https://www.mdpi.com/2079-9292/13/3/514>.

Appendix A

Optimisation of motor's transfer function in current

Via an iterative process, the algorithm will try to minimize a cost function. The cost function is defined as the sum of the errors squared (also called Least Squared):

$$C = \sum_{i=1}^n e_i^2 \quad (\text{A.1})$$

The error is defined as the difference between the real value and the estimated value:

$$e_i = \dot{\theta}_{w,real} - \dot{\theta}_{w,estimation} \quad (\text{A.2})$$

Starting with a first estimation of the parameters, the cost function is computed and the parameters are adjusted to try and minimise this cost function. The Fig.(A.1) shows the changes in the parameters and the value of the cost function during the iteration. The search stops once the cost function become smaller than 10^{-10} .

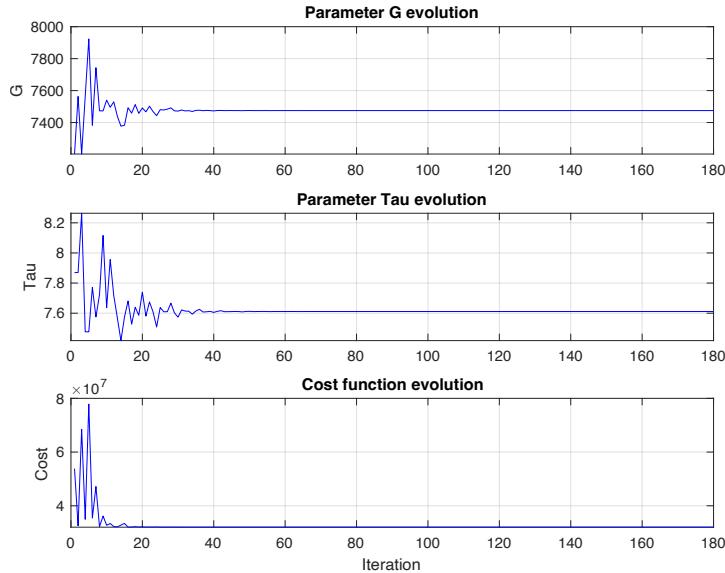


Figure A.1: Parameters and cost function change over iteration

The following parameters are obtained:

$$\begin{cases} G = 7475 \\ \tau = 7.611 \end{cases} \quad (\text{A.3})$$

The simulated transfer function superposed to the real prototype data is represented in Fig.(A.2).

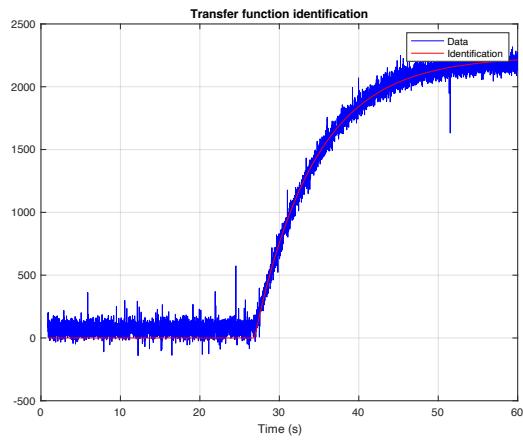


Figure A.2: Transfer function identification

To verify the transfer function obtained, a new step response is made varying the current between 0.2A and 0.3A represented in Fig.(A.3).

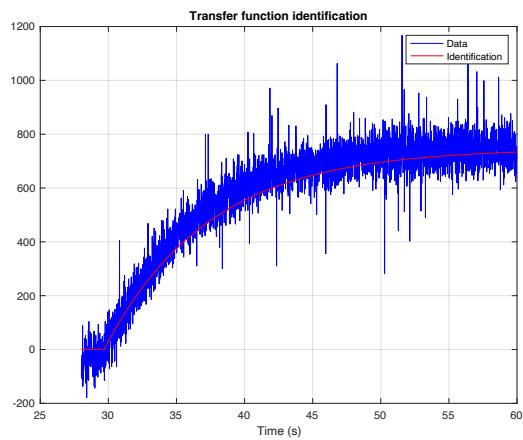


Figure A.3: Transfer function verification

Appendix B

Optimisation of motor's transfer function in speed

The same way as for the current transfer function, an algorithm is used to fine tune the gain and time constant estimations for the speed control of the motor.

The Fig.(B.1) shows the changes in the parameters and the value of the cost function during the iteration. The search stops once the cost function become smaller than 10^{-10} .

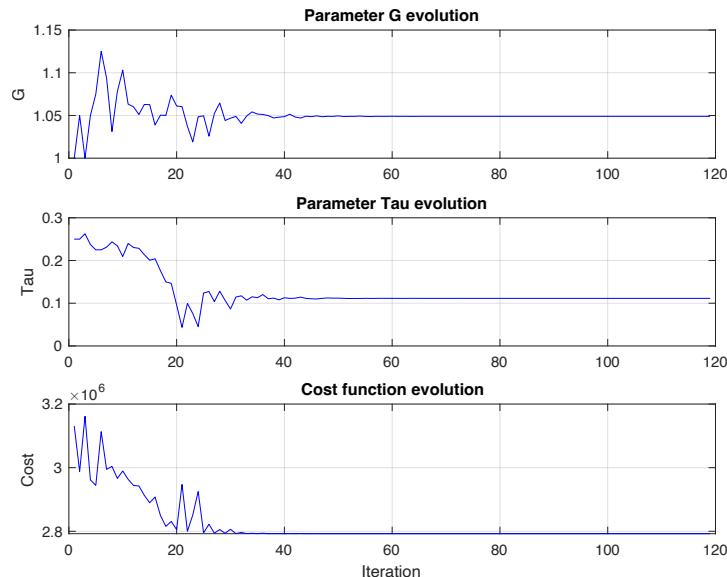


Figure B.1: Parameters and cost function change over iteration

The following parameters are obtained:

$$\begin{cases} G = 1.049 \\ \tau = 0.2 \end{cases} \quad (\text{B.1})$$

The simulated transfer function superposed to the real prototype data is represented in Fig.(B.2).

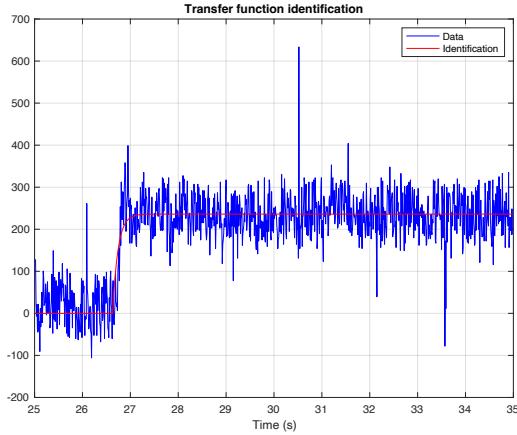


Figure B.2: Transfer function identification

To verify the transfer function obtained, a new step response is made varying the current between 450rpm and 900rpm represented in Fig.(B.3). The y-axis of the plot is normalized, with the step magnitude corresponding to the 450rpm variation.

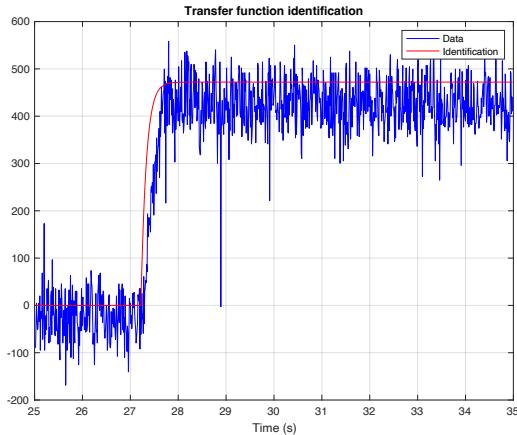


Figure B.3: Transfer function verification

One can notice that with the ESCON speed regulation, the motor's speed response is much faster than with the current regulation with a time constant of 0.2 second compared to the 7 seconds time constant of the current regulation.

Appendix C

Experimental identification of the wheel inertia

The reaction wheel inertia I_w computed by algebraic calculations can be verified experimentally. The pendulum is blocked in the upward position and the system presented in Fig.(C.1) is established. A mass m is suspended with a wire on the reaction wheel. To

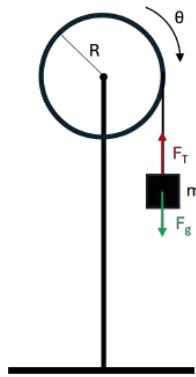


Figure C.1: Schematics of experimental setup for verification of wheel inertia

find the formula that expresses inertia, the sum of torques and forces of the system is done as following:

$$\begin{cases} \sum \tau = I_w \alpha \\ \sum F = F_g - F_T = ma = mR\alpha \end{cases} \quad (\text{C.1})$$

With I_w the wheel inertia, τ the torques of the system, α the angular acceleration of the wheel, m the mass suspended and R the radius of the wheel. The inertia is expressed as:

$$I = \frac{Rm(g - R\alpha)}{\alpha} \quad (\text{C.2})$$

The equation of motion of the wheel is expressed as following:

$$\Delta\theta = \omega\Delta t + \frac{1}{2}\alpha\Delta t^2 \quad (\text{C.3})$$

The angular acceleration can finally be expressed as a fraction that links $\Delta\theta$ the angle difference during the experiment and Δt^2 the elapsed time:

$$\alpha = \frac{2\Delta\theta}{\Delta t^2} \quad (\text{C.4})$$

The experiment was conducted by placing the IMU on the wheel and measuring angle and time with a live Arduino code. For a change in angle of 180° , the recorded time is 0.57

seconds:

$$I_w = \frac{Rm(g - R \frac{2\Delta\theta}{\Delta t^2})}{\frac{2\Delta\theta}{\Delta t^2}} = \frac{75 \cdot 10^{-3} \cdot 9.25 \cdot 10^{-3} (9.81 - 75 \cdot 10^{-3} \frac{2\pi}{0.57^2})}{\frac{2\pi}{0.57^2}} = 2.99 \cdot 10^{-4} kg \cdot m^2 \quad (C.5)$$

The experimental inertia value computed is very close to the inertia estimated analytically of $3.39 \cdot 10^{-4} kg \cdot m^2$.

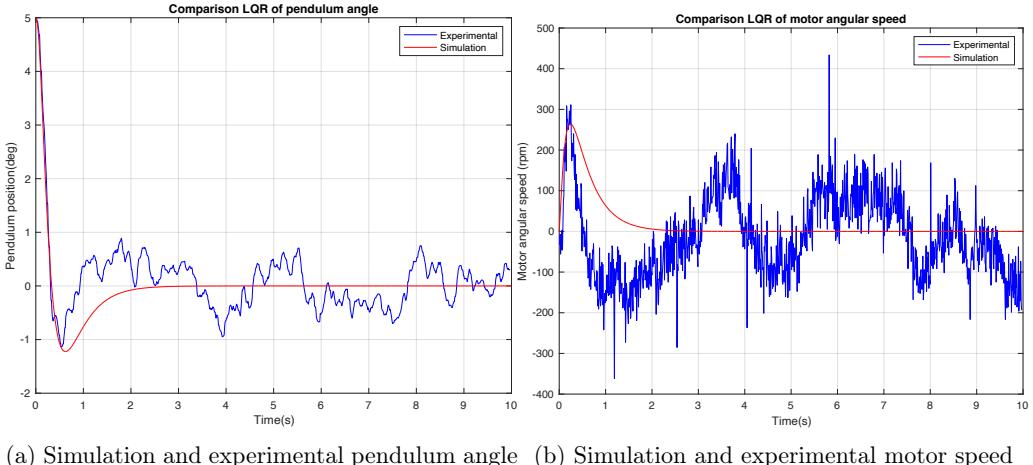
As the experimental data aligns with the analytical one, there should be another reason for this underestimation of I_w .

The motor's rotor inertia could be taken into consideration. In the datasheet of the EC 45 Flat [25], the rotor inertia has a value of $I_{rotor} = 135 g cm^2 = 1.35 \cdot 10^{-5} kg \cdot m^2$. When summed with the inertia I_w , it only slightly increases the value, thus not fully accounting for the higher inertia value required to match the experimental results in the simulation.

It is likely that other parameters were overlooked or underestimated during the modelling process. Still, the value of I_w was adjusted to better match the experimental response in simulation.

Appendix D

Adaptation of parameters for LQR_2



(a) Simulation and experimental pendulum angle (b) Simulation and experimental motor speed

Figure D.1: Test of recovery angle of 5°

An update of the parameters was performed by comparing the pendulum angle and motor's speed in response to a 5° initial angle. While the experimental and simulation angle and speed match during the first half-second, they diverge right after: the experimental pendulum reacts faster than the simulated one. To address this, the pendulum inertia was slightly increased and the friction was reduced to accelerate the simulated pendulum response.

Additionally, to reduce the difference in command between the simulation and experimental, the time constant of the motor was slightly decreased to 6.8 seconds. The change in values is summarized in Tab.(D.1).

Table D.1: Parameter values adapted to the experimental model

Parameter	LQR1	LQR2
Pendulum Inertia I	$0.012 \text{ kg} \cdot \text{m}^2$	$0.011 \text{ kg} \cdot \text{m}^2$
Pendulum friction b_p	$0.0314 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$	$0.0114 \text{ kg} \cdot \text{m}^2 \cdot \text{s}^{-1}$
Motor time constant τ	7.6 s	6.8 s

Appendix E

Simulink simulation for verification

Bellow is presented the **Simulink** setup used to verify the mathematical model of the inverted pendulum. For this simulation, the pendulum is placed in the upside down position, the command is an impulse of 1A lasting 1 second, there are no additional disturbances added to the system.

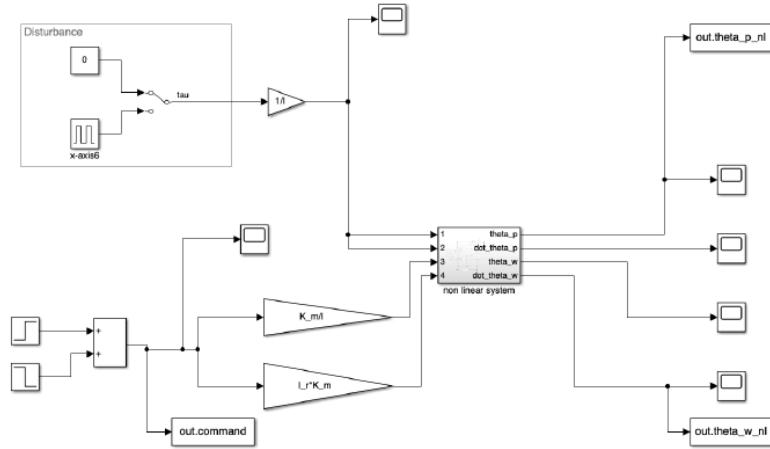


Figure E.1: Simulink simulation for the verification of mathematical model of the pendulum with reaction wheel

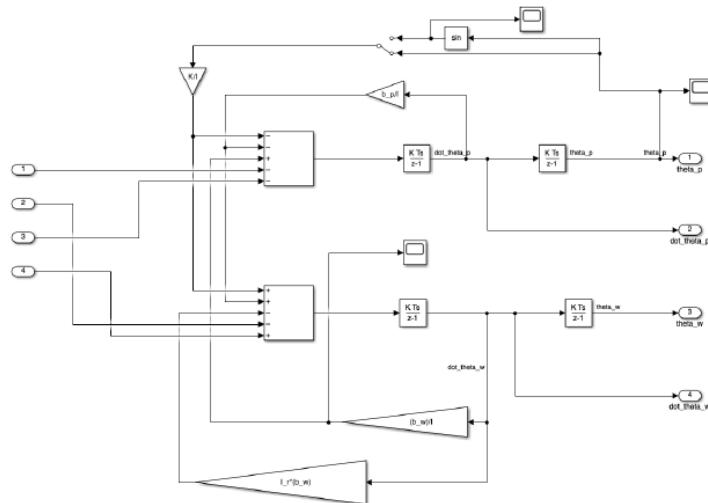


Figure E.2: Simulink simulation for the verification of mathematical model of the pendulum with reaction wheel, zoom on the mathematical model

Appendix F

Simulink simulation of the LQR

Bellow is presented the **Simulink** setup used to simulate the LQR control of the inverted pendulum. For this simulation, the pendulum is placed in the upright and a starting angle can be implemented or disturbances can be added to analyse their impact on the system.

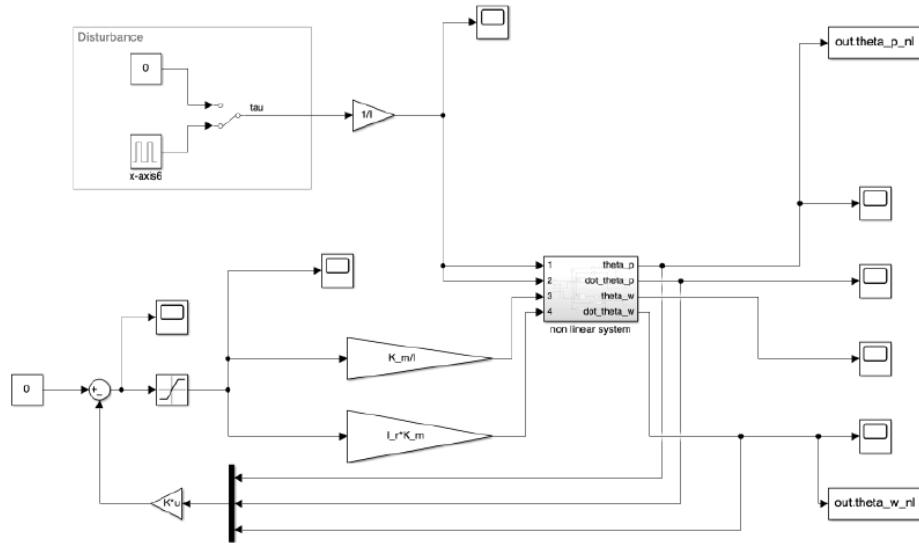


Figure F.1: Simulink simulation for the simulate the LQR control of the pendulum with reaction wheel

Appendix G

Simulink simulation of the cascade PID

Bellow is presented the **Simulink** setup used to simulate the PID control of the inverted pendulum. For this simulation, the pendulum is placed in the upright and disturbances can be added to analyse their impact on the system.

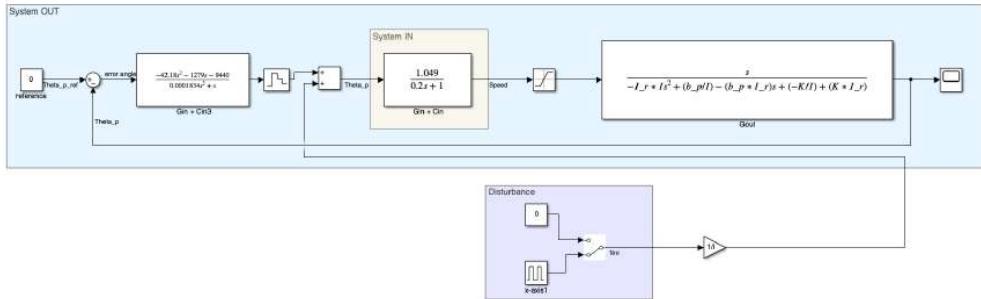


Figure G.1: Simulink simulation for the cascade PID control of the pendulum with reaction wheel

Appendix H

Technical drawings of the prototype

