ESEIAAT

Bachelor's Thesis

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Study of different control methods applied to a self-balancing robot

Degree
Bachelor's degree in Industrial Technology Engineering

Student
Sergio Prieto Molina

Director
Ramon Comasòlivas Font

Delivery date
10/06/2019

For those who persist in their desires

Acknowledgments

This bachelor thesis is not the product of just the effort of one person, it is the culmination of many people who had an impact on me or helped me during my life.

First of all, I would like to express my deepest gratitude to my parents, who have given and continue giving me the emotional and financial support during my years of student. Since I was a child, they made me believe that everything was possible with persistence and motivation. Without them none of what I am now would have been possible.

Secondly, thanks to the UPC Space Program, concretely speaking, to Aldora. They introduced me to the world of drones which in one way or another lead me to the amazing world of programming and robotics.

Finally, thanks to my director of this bachelor thesis, Ramon Comasòlivas and professor Bernardo Morcego, who guided me in critical moments.

Abstract

During the 21st-century robotics has been growing exponentially owing to the huge influence that robots have in our lives. Therefore, due to my interest in this field, this thesis is going to deal with the construction of a two-wheeled robot and the study of different control methods to balance the robot.

A research about control methods will be exposed to let the reader know which control methods are more suitable for our case. Consequently, a mathematical dynamic model of the robot will be represented, this process will allow us to select the ideal methods to control the balancing process.

Once the methods will be selected, simulations will be committed to insure that the methods selected behave as they were predicted. Simulations are based on theory that is quite ideal; therefore, to break the ideality of our models and bring them to reality experiments will be performed to confirm if the models control the desired process.

Finally, with the simulations' results and experiments, a comparison between the selected methods will be done to chose the best method to achieve the desired robot's behavior.

List of Figures

List of Tables

Contents

1 Introduction

Control theory has been with us since the 19th century. One of the first machines to use this subfield of mathematics was the steam engine, this machine used a mechanism formed by two balls to control the velocity of the steam engine. The more velocity of the engine the more spread the balls are, this behavior causes the automatic regulation of the velocity. This machine is shown in the figure 1.1.

Figure 1.1: Steam engine with the two balls system in the center of the picture.
[1]

Nowadays, in some way or another we are constantly dealing with technology which contains either software or hardware based on control theory. A close example of this technology would be the cruise control of our car which regulates the velocity of our vehicle without being constantly pressing the throttle.

Another sophisticated device not included in our daily lives which highly uses control theory is the Falcon 9 rocket. This rocket developed by SpaceX is the first rocket to land automatically modify its trajectory(Figure 1.2)[2].

Due to the sudden apparition of hover-boards on streets(Figure 1.3), the curiosity has been pricking my mind. Therefore this thesis will deal with a basic physical exploration of these devices and a more detailed exploration of the used control methods to balance these inventions.

Figure 1.2: Falcon 9
[3]

Figure 1.3: Hoverboard
[4]

1.1 Aim

The purpose of this bachelor thesis is to build a two wheel robot fully balanced using control theory methods. Behind the main purpose, there is the desire to design the control method which will produce a fast response to disturbances and will maintain a good balance in a two wheel robot. Furthermore, several aims are contemplated inside this thesis, a sample of these aims are displayed below:

- Learn more control theory
- Gain experience with MATLAB in the field of simulation
- Apply control theory to real problems
- Understand the most used control theory controllers
- Model physics systems

1.2 Scope

The scope of this thesis is defined by the following diagram. The first task is to research and study several control theory methods and concepts to tackle the method's design, once this task is done, the differential equations that define the robot's model will be obtained.

Next, several control methods will be selected taking into account the desired robot's behavior, fast response and accuracy. In the case that the control method will not be appropriate, another control method selection will be committed until several methods will obey the behavior mentioned.

Continuing with the scope, the following task is conducting a MATLAB simulation to see if the system response is the adequate taking into consideration the desired robot performance, if the performance of the simulation is not the expected, the selection of the control method will be continued to study new methods. Subsequently, experiments will be done with the methods simulated.

Concurrently, a comparison will be done to establish the differences between the methods selected and finally, the best methods will be selected taking into account their performance.

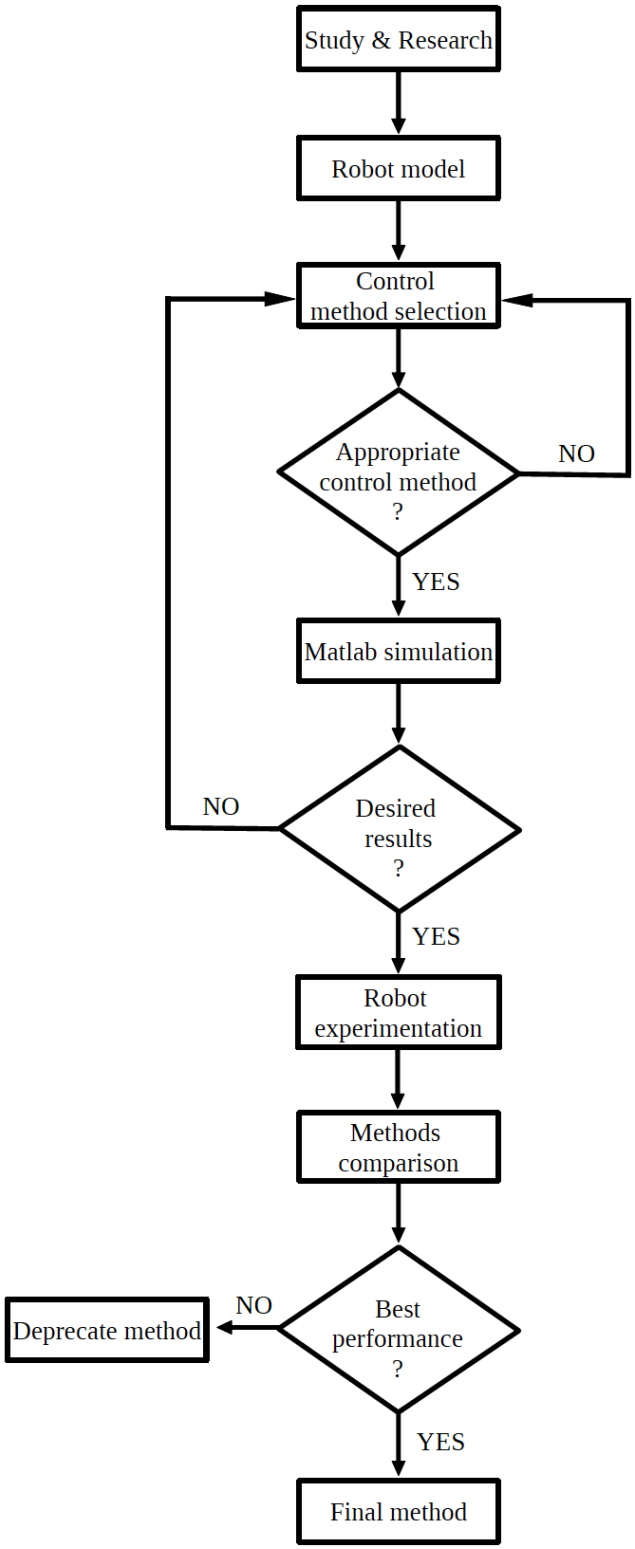


Figure 1.4: Thesis scope

1.3 Requirements

Due to one of the tasks of this project is building a two-wheel robot to experiment with several control theory methods, various parts are needed such as a micro-controller, a gyroscope, two wheels and two electric motors. However, thanks to the automatic department of the ESEIAAT, purchases will not be committed because of the availability of these pieces. Moreover, the MATLAB program will be needed to make the simulations of the methods and the posterior methods experimentation, specifically speaking, the Simulink tool.

Lent material from ESEIAAT:

- Lego Mindstorms EV3 kit (Figure 1.5)

Software needed:

- MATLAB R2019a student license version

In regards to the nonphysical or software requirements, it is my thought that a minimum of knowledge in control theory and in MATLAB is required to tackle this thesis.

Figure 1.5: Lego Mindstorm EV3 kit
[5]

1.4 Justification

Since the apparition of robots and the incorporation of electronic devices in our days, more and more control theory is being used to ease several capabilities that were unthinkable in the past, a sample of these capabilities is the automatic parking of various car models.

Hence, because of my curiosity for robots and control theory, it is my desire to explore how a hover-board works. The main reason for choosing a hover-board instead of another device, it is owing to the huge influence that this means of transport is causing on our society. Furthermore, developing a small version of a hover-board is relatively easy in comparison to a guided rocket such as the one in Figure 1.2; therefore, a hover-board is an adequate device to work on.

1.5 Utility

The usefulness of this thesis does not reside in direct industrial use. The real utility lies in gathering the knowledge thought at the bachelor's degree in industrial technology engineering and use this knowledge to explore a real problem where multiple subjects are required. Hence, this thesis is useful to apply what has been learned during the degree and delve into control theory concepts that have not been given in the degree.

The following points summarize the utility of this thesis:

- Model real systems
- Obtain the differential equations that describe the physics behavior
- Learn more control theory
- Understand the more used control theory methods
- Apply control theory knowledge to real problems

2 Development

In this section of this bachelor thesis, the reader will be accompanied through the different stages that compose the acquisition of the robot model, the methods selected, the simulations and the experimentation with the selected methods.

The obtaining of the robot model is focused on demonstrating the main equations that have been resolved to obtain the theoretical model and the real model of the robot.

In the second stage of the development section, the controllers will be explained and developed for our case.

Finally, simulations will be committed alongside the development of the controllers to study if the methods selected are appropriate enough to achieve a fast response and a good balance. After the simulation, experiments will be performed to observe the real behavior of the robot.

2.1 Precedents

Nowadays, it is easy to see a variety of two-wheeled devices that maintain the balance during its driving. Regarding their origins, all these devices come from a two-wheeled chair named iBOT(Figure 2.1). iBOT was developed by Dean Kamen with DEKA's partnership during the 90's to assist people who have walking disabilities. The main advantage of this chair in comparison to those which are on the market is that this chair is able to climb sidewalks without any difficulty.[7]

Figure 2.1: Dean Kamen sited in an iBOT
[6]

Thanks to the development of iBOT, Dean Kamen got the necessary inspiration to develop his new creation, the Segway(Figure 2.2) [8]. Since then, various brands have commercialized different variations of the original Segway, a sample of a variation is displayed in Figure 1.3.

Due to the huge impact of these means of transport in our society, several theses have been done focusing on the study of just one control method for these robots. That is where this thesis has its appearance, this thesis will use several control methods and compare them to obtain the best method to control a two-wheeled robot.

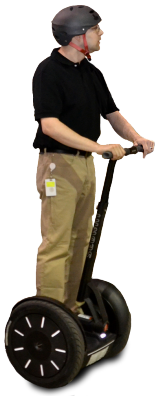


Figure 2.2: Segway
[9]

2.2 Robot physics

To find mathematical algorithms in order to control the robot, a force diagram is presented to understand the physical behavior. Therefore, the following paragraphs will try to elaborate the necessary explanations to show the behavior.

The figure displayed is a simplification of the robot, the circle would be the representation of two wheels in 2D while the body of the robot which is the bar would comprehend the microcontroller EV3, the gyroscope and the motors to move the wheels.

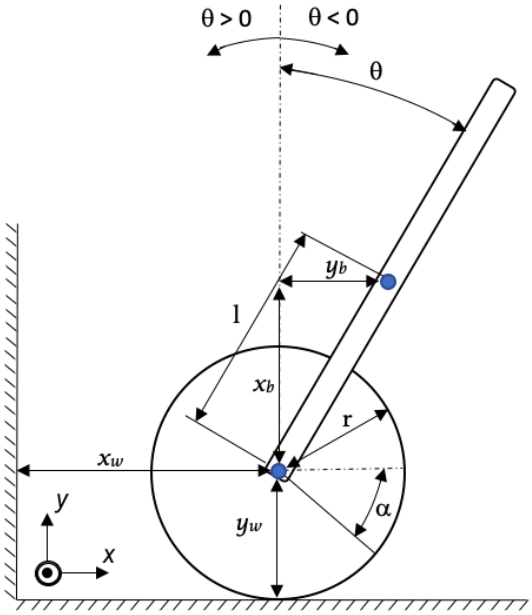
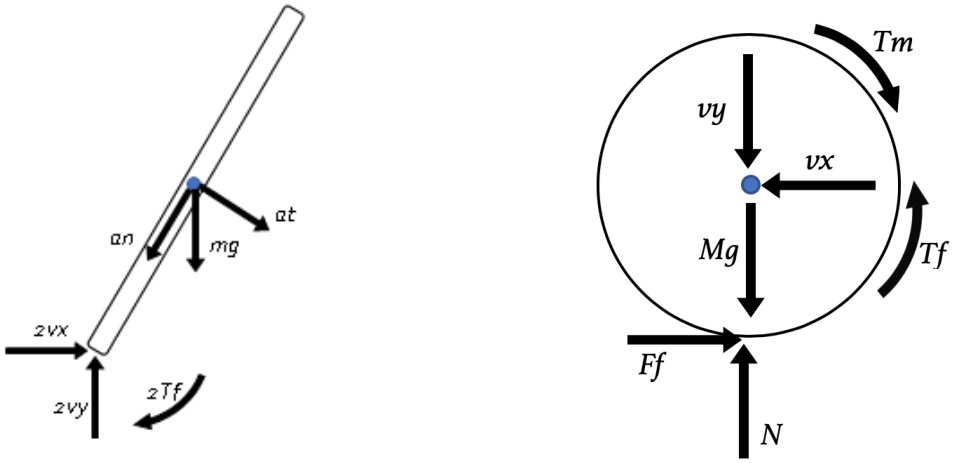


Figure 2.3: Diagram of forces

As it can be seen in the Figure 2.3, θ is the angle rotated by the bar regarding the vertical discontinued line while α is the angle rotated by the wheels with respect to the horizontal discontinued line. The x_w is the absolute horizontal position of the wheel whereas the x_b is the horizontal position of the bar relative to wheels and the y_b is the vertical position of the bar relative to wheels. Finally, the r and l are the radius of the wheel and the length of the bar from the motors axes until the bar's gravity center.



(a) Bar diagram

(b) Wheels diagram

Figure 2.4: Forces diagram

Thanks to figure 2.4, Newton's third law and second law in the horizontal direction, the vertical direction and the summation of torques, we are able to express the forces' diagrams into equations.

$$\sum F_x = m \cdot a_x$$
$$\sum F_y = m \cdot a_y$$
$$\sum T = I \cdot \sigma$$

The following expressions are found using the bar's diagram figure (Figure 2.4a) and taking into account that the bar it in an angular position(θ) which is specified as a negative in radiant.

$$2v_x = m \cdot \ddot{x_b} \tag{2.1}$$

$$-g \cdot m + 2v_y = m \cdot \ddot{y_b} \tag{2.2}$$

$$-2T_f + g \cdot l \cdot m \cdot \sin(\theta) = \left(I_b + l^2 \cdot m\right) \ddot{\theta} - l \cdot \cos\theta \cdot \ddot{x_w} \cdot m \tag{2.3}$$

The next ones are found using the wheels' diagram (Figure 2.4b):

$$F_f - v_x = M \cdot \ddot{x_w} \tag{2.4}$$

$$-M \cdot g + N - v_y = 0 \tag{2.5}$$

$$F_f \cdot r + T_f - T_m = I_w \cdot \ddot{\alpha} \tag{2.6}$$

Regarding the acceleration of the bar in the horizontal direction and in the vertical direction, they can be rewritten as a set of different equations depending on θ and x_w. The movement of the bar is characterized by being a circular movement (Figure 2.5). Hence, there is a tangential acceleration a_t and a normal acceleration an. These two forces can be decomposed in the x axis and in the y axis to find $\ddot{y_b}$ and $\ddot{x_b}$ (expressions 2.7 and 2.8).

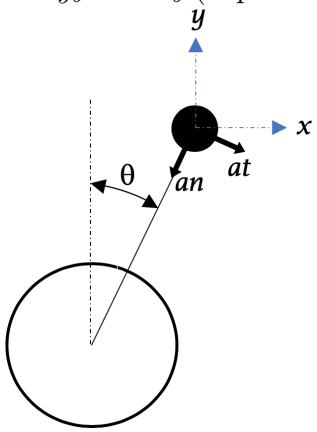


Figure 2.5: Pendulum movement

$$\ddot{x_b} = l \cdot \sin(\theta) \cdot \dot{\theta}^2 + l \cdot \cos(\theta) \cdot \ddot{\theta} + \ddot{x_w} \tag{2.7}$$

$$\ddot{y_b} = l \cdot \sin(\theta) \cdot \ddot{\theta} - l \cdot \cos(\theta) \cdot \dot{\theta}^2 \tag{2.8}$$

Due to the utilization of electric motors to drive the robot, the torque of the motor T_m and the resistance torque of the motor T_f can be modeled using figure 2.6. In our case, the stator would be the bar and the rotor would be the wheels; hence, with this statement, we can proceed to understand the motor modeling.

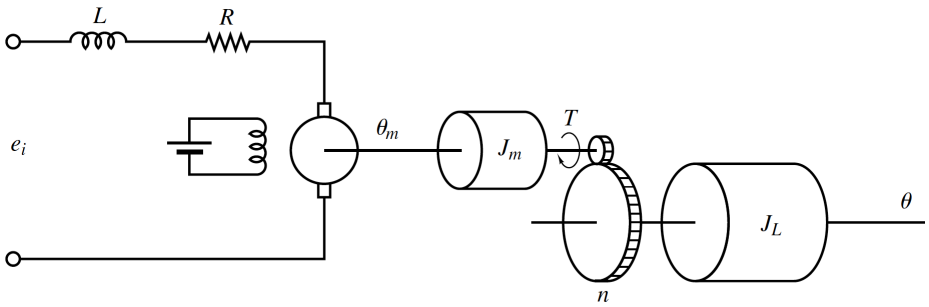


Figure 2.6: Motor system
[10]

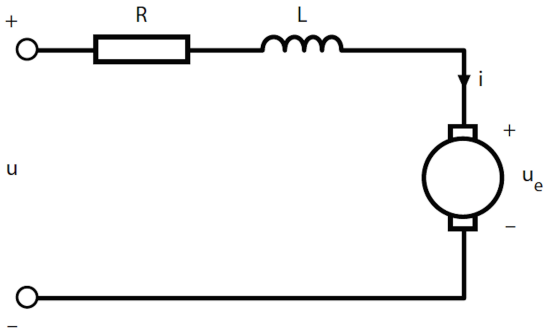


Figure 2.7: Motor simplified
[11]

Motors can be modeled by an inductance (L) due to the magnetic field created by coils, a resistance (R) to the current opposition and a back EMF(Electromotive force)(u_e)[11]. In a more mathematical way, the back EMF can be expressed as the following equation, where the term $(\dot{\alpha} - \dot{\theta})$ is the rotor's velocity with respect to the stator.

$$u_e = K_e(\dot{\alpha} - \dot{\theta}) \tag{2.9}$$

Applying Kirchhoff voltage law, we are able to obtain the mathematical representation of the circuit in figure 2.7[11].

$$u = R \cdot i + L \cdot \frac{di}{dt} + u_e \tag{2.10}$$

If we solve equation 2.10 for the current working under the assumption that the inductance is small due to the utilization of small motors and that the voltage applied to motors is steady, we are able to see that the exponential term is going to rapidly decrease with time. Hence, if the resistance is bigger than the inductance, the exponential term is insignificant and then the expression 2.11 is reduced to the equation 2.12[11].

$$i(t) = \frac{u - u_e}{R} \cdot (1 - e^{\frac{-R \cdot t}{L}}) \tag{2.11}$$

$$i(t) = \frac{u - u_e}{R} \tag{2.12}$$

Using expression 2.9 and expression 2.12, we are able to gather them to obtain the following mathematical equation 2.13[11].

$$i(t) = \frac{u - K_e(\dot{\alpha} - \dot{\theta})}{R} \tag{2.13}$$

Going back to our previous intention, T_m can be modeled into a mathematical expressions that depends on the voltage applied to the motor. Therefore, manipulating the expression 2.13 and 2.14, T_m can be obtained depending on u (Equation 2.15)[11].

$$T_m = K_m \cdot i \tag{2.14}$$

$$T_m = -\frac{K_e \cdot K_m \left(\dot{\alpha} - \dot{\theta}\right)}{R} + \frac{K_m \cdot u}{R} \tag{2.15}$$

In regard to the friction torque, T_f can be expressed as the next expression, where b is a constant[11].

$$T_f = b\left(-\dot{\alpha} + \dot{\theta}\right) \tag{2.16}$$

Although the equations of the motor has been presented, there is another way to obtain the equations which will represent the behaviour of the motors.

To start, a summation of torques in the bar is going to be committed under the hypothesis that there is no torque T_f.

$$\sum T = I\sigma$$

$$K_m \cdot I(s) = I_w \cdot \ddot{\alpha}(s) \tag{2.17}$$

Substituting expression 2.9 into 2.10 and applying the Laplace transform to the resulted equation, equation 2.18 is obtained.

$$U(s) = R \cdot I(s) + L \cdot s \cdot I(s) + K_e \cdot \dot{\alpha}(s) \tag{2.18}$$

With the utilization of equations 2.17, it is possible to substitute $I(s)$ in the equation 2.18 and obtain the following transfer function if the assumption that L is contemptible is done.

$$\frac{\dot{\alpha}(s)}{U(s)} = \frac{K_m}{\frac{R \cdot I_b}{K_e \cdot K_m} \cdot s + 1} \tag{2.19}$$

$$\frac{\dot{\alpha}(s)}{U(s)} = \frac{K}{\tau \cdot s + 1} \tag{2.20}$$

The expression 2.20 can be transformed into the equation 2.21, in which the anti-Laplace transform can be used to obtain expression 2.22.

$$\dot{\alpha}(s) \cdot (\tau \cdot s + 1) = K \cdot U(s) \tag{2.21}$$

$$\ddot{\alpha}(t) \cdot \tau + \dot{\alpha}(t) = K \cdot u(t) \tag{2.22}$$

Solving equation 2.22 for $\ddot{\alpha}$, the angular acceleration of the wheel will be found which will help to find a torque T_w. This torque T_w is the result between torques T_m, T_f and the torque produced by F_f; hence, with the utilization of T_w, there is no need to find T_m, F_f and T_f.

$$\ddot{\alpha}(t) = \frac{K \cdot u(t)}{\tau} - \frac{\dot{\alpha}(t)}{\tau} \tag{2.23}$$

$$T_w = I_w \cdot (\frac{K \cdot u(t)}{\tau} - \frac{\dot{\alpha}(t)}{\tau}) \tag{2.24}$$

Taking into consideration equation 2.24, figure 2.4b is modified into the following ones.

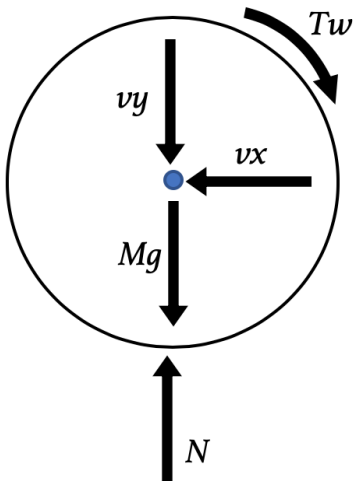


Figure 2.8: Forces diagram using T_w

Although it is considered that T_w is the result of T_m, T_f and the torque produced by F_f, in the summation of forces in the wheel F_f is considered; hence equation 2.6 can be rewritten as the equation 2.24.

2.3 Robot model

Reached this point where all possible unknowns have been found, it is time to start with the mathematical model which will guide us to develop the control algorithms. Since we want to control θ and the position of the robot x_w, the state space representation is going to be used. The main benefit of using this representation over the classic transfer function is that state-space models are represented using matrixes; hence, the system is represented using linear equations which are easy to deal with. Equations 2.25 and 2.26 displays the mathematical representation of the state space model.

$$\dot{x} = A \cdot x + B \cdot w \tag{2.25}$$

$$y = C \cdot x + D \cdot w \tag{2.26}$$

To understand each symbol of the expressions 2.25 and 2.26, we will start with w. w is the system's input which in our case is the voltage variable u. In regard to the x term, this term is a vector that contains variables that define our system, for example, in a normal airplane which consumes kerosene, the mass of the airplane varies during its journey. Therefore, in the case of having an aircraft, the mass would be a variable which would be contained on the x vector, commonly named, state vector.

To find the terms of the expressions 2.25 and 2.26, it is necessary to study the equation 2.3 and equation 2.24. Equations 2.3 and 2.24 are second order equations due to $\ddot{\theta}$ and $\ddot{\alpha}$. Thus, owing to the second derivative of therms θ and $\ddot{\alpha}$, our state vector will be composed by two α terms and two θ terms. However, instead of using the angular position α, the linear position xw will be used.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ x_w \\ \dot{x_w} \end{bmatrix} \tag{2.27}$$

Looking back to expression 2.25, it can be seen that the vector $\dot{x}$ will be the vector 2.27 but

derived.

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x_w} \\ \ddot{x_w} \end{bmatrix} \tag{2.28}$$

To find the equations that represent the state vector $\dot{x}$, equations 2.24 and 2.3 are going to be used. As it has been mentioned, instead of using the angular position of the wheel α, x_w will be used. Therefore, equation 2.23 will be modified to obtain a representation in terms of x_w.

$$\ddot{x_w}(t) = \frac{K \cdot r \cdot u(t)}{\tau} - \frac{\dot{x_w}(t)}{\tau} \tag{2.29}$$

Knowing $\ddot{x_w}$, it is possible to find $\ddot{\theta}$ if $\ddot{x_w}$ is substituted in the expression 2.3.

$$\ddot{\theta} = \frac{-2T_f + g \cdot l \cdot m \cdot \sin(\theta) + l \cdot \cos\theta \cdot \ddot{x_w} \cdot m}{I_b + l^2 \cdot m} \tag{2.30}$$

$$\ddot{\theta} = \frac{-2b\left(-\dot{\alpha} + \dot{\theta}\right) + g \cdot l \cdot m \cdot \sin(\theta) + l \cdot \cos\theta \cdot (\frac{K \cdot r \cdot u(t)}{\tau} - \frac{\dot{x_w}(t)}{\tau}) \cdot m}{I_b + l^2 \cdot m} \tag{2.31}$$

Thinking about the bar movement, due to the movement correction of the motors over the bar, the body is going to oscillate tracing small angles. Therefore, the following hypothesis can be made:

$$\sin(\theta) \approx \theta$$
$$\cos(\theta) \approx 1$$

Furthermore, if notations in the vectors 2.27 and 2.28 are used in equation 2.29 and 2.31, the following equations are obtained.

$$\dot{x_2} = \frac{g \cdot l \cdot m}{I_b + l^2 \cdot m} \cdot x_1 - \frac{2b}{I_b + l^2 \cdot m} \cdot x_2 + \frac{2b \cdot \tau - l \cdot m \cdot r}{r \cdot \tau(I_b + l^2 \cdot m)} \cdot x_4 + \frac{k \cdot l \cdot m \cdot r}{\tau(I_b + l^2 \cdot m)} \cdot u(t) \tag{2.32}$$

$$\dot{x_4}(t) = \frac{K \cdot r \cdot u(t)}{\tau} - \frac{x_4}{\tau} \tag{2.33}$$

With the expressions 2.32 and 2.33 it is possible to find the state space representation.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g \cdot l \cdot m}{I_b + l^2 \cdot m} & -\frac{2b}{I_b + l^2 \cdot m} & 0 & \frac{2b \cdot \tau - l \cdot m \cdot r}{r \cdot \tau (I_b + l^2 \cdot m)} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K \cdot l \cdot m \cdot r}{\tau (I_b + l^2 \cdot m)} \\ 0 \\ \frac{K \cdot r}{\tau} \end{bmatrix} \cdot u(t) \tag{2.34}$$

To find the expression 2.26 it is necessary to know that the available sensors measures are x_2 and x_3. Thus, expression 2.26 is the displayed below.

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \tag{2.35}$$

However, the state space representation 2.34 is represented symbolically, to give numbers it is necessary to find the following parameters specified in equations 2.32 and 2.33.

In order to determine the unknown parameters, several experiments are going to be performed. First, parameters τ and K will be determined applying a step input to a motor, with this experiment data will be recorded using the following model in Simulink.

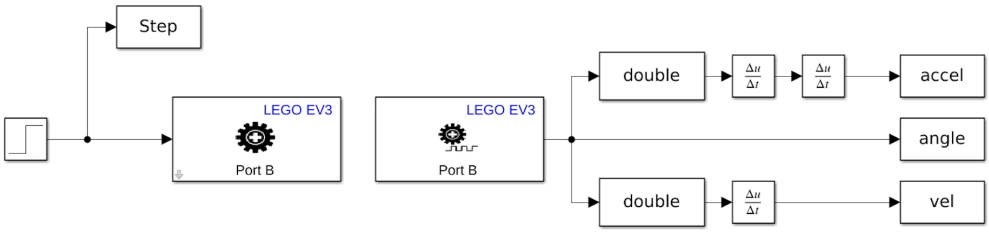


Figure 2.9: Block diagram of the wheel experiment

(a) Motor block (b) Encoder block

Figure 2.10: EV3 blocks

In regards to the EV3 blocks(figure 2.10), concretely figure (a), it needs a signal whose value must be between 100 and -100 whereas block in figure (b) outputs the angular position of the wheel in degrees. Viewing the output from the block (b) in figure 2.11, it can be seen that although the signal comes with noises, the model is a first order model. Hence, it is verified that the L is negligible in equation 2.20 and that the only torque that affects the wheel when a constant voltage is applied is T_m(equation 2.17) . Furthermore, the K value

in the numerator of equation 2.20 is 8 if the graphic 2.11 is observed.

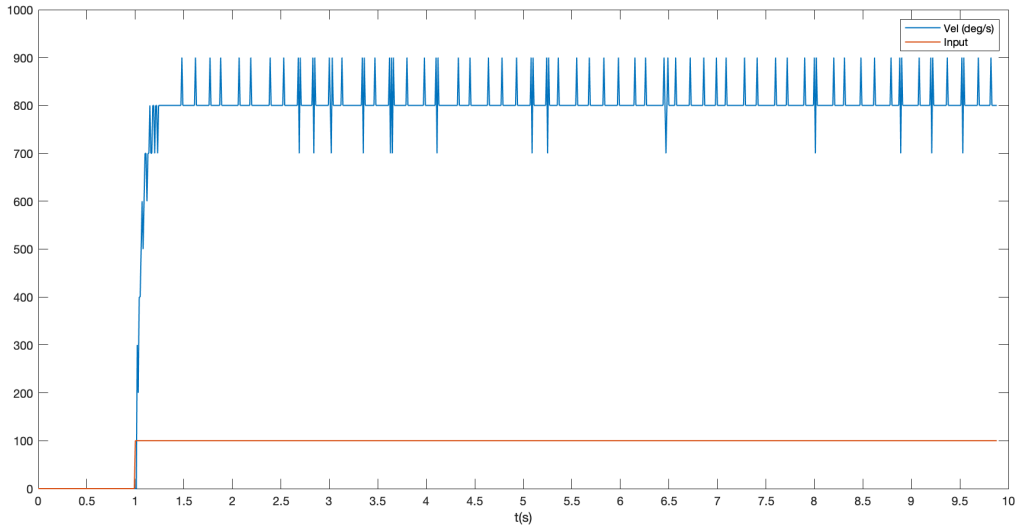


Figure 2.11: Motor experiment with wheel

Due to the model of the motor is of first order, it is possible to obtain τ and K using the output in figure 2.11, but to be more precise, the signal is going to be filtered. Ideally, the velocity output should be without any noises; thus, contemplating figure 2.11, it can be seen that noises have a higher frequency in comparison with the ideal velocity signal. To correct this, a low pass filter will be implemented, but to implement this filter it is necessary to know the cut-off frequency. In order to know the minimum frequency when the filter will start to act, an interval of time which contains the minimum number of oscillations is going to be observed.

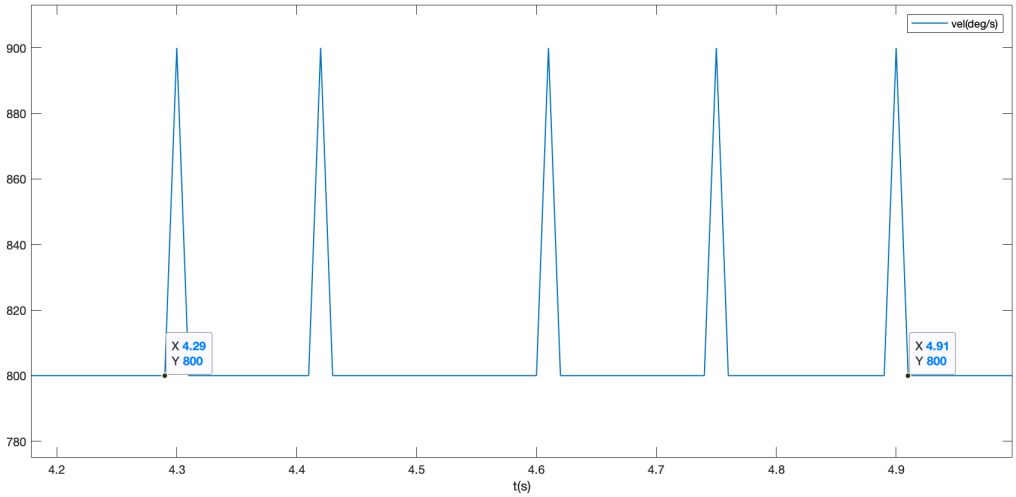


Figure 2.12: Observation of the noise in figure 2.11

With figure 2.12, the cut-off frequency is the number of peaks divided by the period of time in which the noise appears, this gives a cut-off frequency of $f_c = 8.06Hz$. It has been decided that a second order filter will be used instead of a higher order filter or a first order filter. A high order filter will give better results but the design would be more complex, on the other hand, a first order filter is easier to implement but it would give worse results than a higher order filter, in other words, the higher the order the better is the approximation of the output to the ideal output but the complex the calculations are. Therefore, a second order filter will be selected, concretely, a Butterworth second order filter with gain $1(k = 1)$, $a = 1.414$ and $k_o = 1$. Apart from what it has been mentioned, over the different models of second order filters a Butterworth filter has been, this is due to the Butterworth filter with the values specified offer a filter which has almost no dumping at the cut-off frequency.

V_{out}: Ideal output

V_{in}: No filtered signal

$$\frac{V_{out}}{V_{in}} = k \cdot \frac{w_o^2}{s^2 + a \cdot w_o + w_o^2} \tag{2.36}$$

$$w_o = 2 \cdot \pi \cdot f_c \cdot k_o \tag{2.37}$$

$$\frac{V_{out}}{V_{in}} = \frac{2526}{s^2 + 71 \cdot s + 2526} \tag{2.38}$$

NOTE: The final values of the filter 2.38 have been rounded to the lowest integer number to obtain a filter which will reduce noises at less than the cut-off frequency specified.

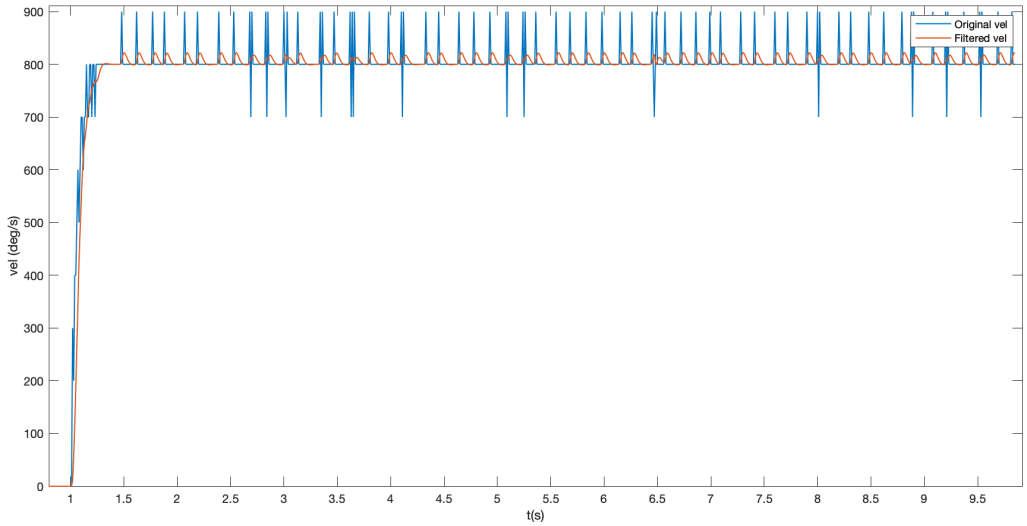


Figure 2.13: Comparison between the filtered velocity and the original velocity

Using the filter 2.38, the filtered output is displayed in figure 2.13 in comparison with the original output.

As it can be seen in graphic 2.13, the filter shits the filtered signal with respect to the original signal. To correct this the filtered signal is going to be shifted until there will be no shift between the original signal and the shifted one.

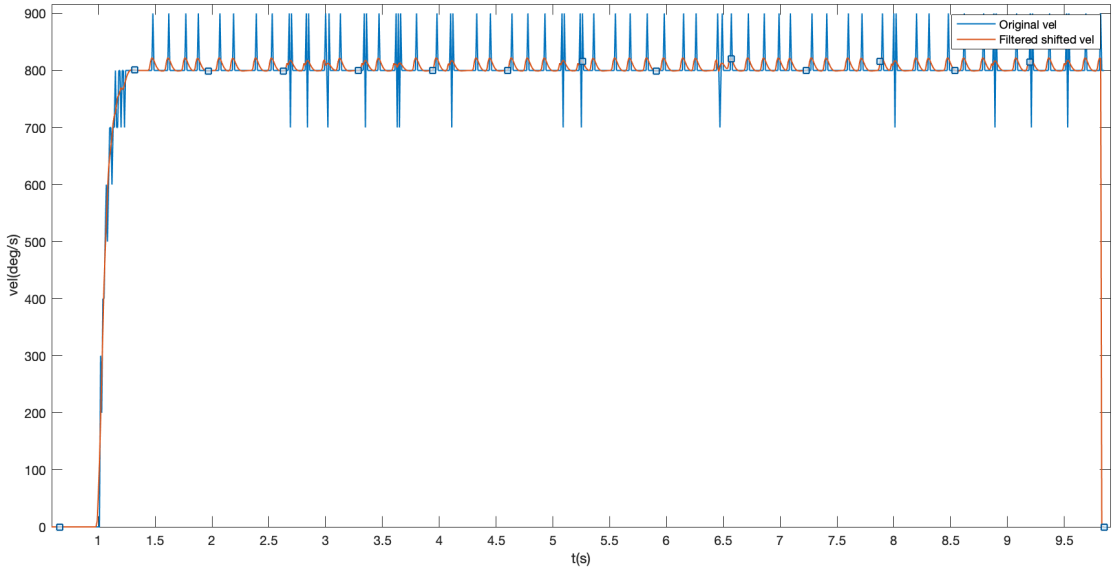


Figure 2.14: Filtered shifted signal in the experiment with wheel

Nevertheless, the model displayed in equation 2.20 has several differences from the model which can be obtained with figures 2.14. The model in the expression 2.20 represents the angular velocity in radiant per second with respect to the input voltage, while the model which can be obtained with figure 2.14 would also represent the angular velocity in degrees per second with respect to a number between 100 and -100. Despite these differences, the model obtained through graphic 2.14 is completely valid to represent the state space model. Hence, the only change in the state space representation is in the expression 2.34, where a constant has to be multiplied to each term of the matrix A or B that has the K term. The constant to multiply has the following value

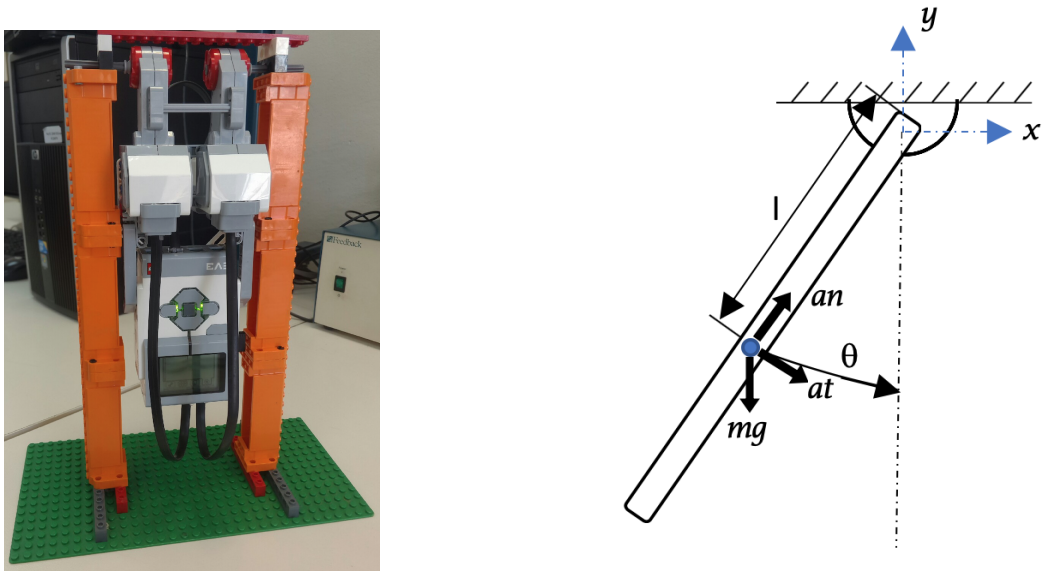
$$K_c = \frac{\pi}{180}$$

To obtain τ, the time has been found in graphic 2.14 for a velocity of 505.6 deg/s.

$$\tau = 0.0546s \tag{2.39}$$

Concerning the inertia of the bar I_b, another experiment will be performed(figure 2.15a). This experiment will consist in detaching the wheels from the robot and put the body suspended

in the air form the axes where wheels are attached, with this kind of pendulum, the angular velocity and time will be measured during the pendulum's oscillations to obtain the Ib using equation 2.43.



(a) EV3 pendulum

(b) Pendulum

Figure 2.15: Pendulums

The equation 2.43 has been obtained through analyzing figure 2.15b and using equation 2.40 and 2.41.

$$a_t = \ddot{\theta} \cdot l \tag{2.40}$$

$$\sum T = I\sigma \tag{2.41}$$

$$m(\ddot{\theta} \cdot l^2 \cdot \cos\theta^2 + \ddot{\theta} \cdot l^2 \cdot \sin\theta^2 + g \cdot l \cdot \sin\theta) = (I_b + m \cdot l^2) \cdot \ddot{\theta} \tag{2.42}$$

$$m \cdot (\frac{g \cdot l \cdot \sin\theta}{\ddot{\theta}}) = I_b \tag{2.43}$$

With these equations and the data obtained from Simulink, it is possible to obtain the inertia I_b. But first, the mass of the pendulum and the distance l are going to be measured. The mass is going to be quantified with a digital weighing scales while the l is going to be acquired placing what it is thought to be the center of mass of the bar and move the robot's

body until the robot will be balanced, the following image displays the experiment done.

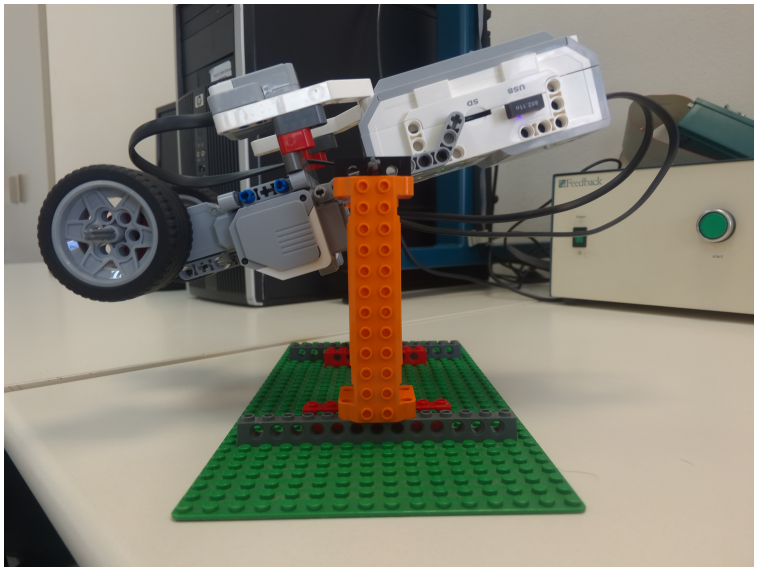


Figure 2.16: Experiment to obtain the l

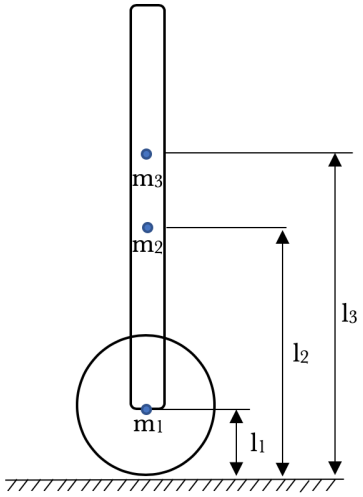


Figure 2.17: Simplification of the robot in figure 2.16

The length obtained through the experiment in figure 2.16 is the equivalent to the length l_2. As the length l is desired to use the equation 2.43, the value of l_3 must be found. Using the following expression the value of l_3 can be calculated if the masses of the wheels and the bar are known, fortunately this is an easy task if a weighting scale is used.

$$l_3 = \frac{l_2 \cdot m_2 - 2 \cdot l_1 \cdot m_1}{m_3} \tag{2.44}$$

$$M{=}m_1{=}0.023\text{Kg}$$

$$m{=}m_3{=}0.526\text{Kg}$$

$$m_2{=}0.572\text{Kg}$$

$$l_2{=}0.113\text{m}$$

$$l_1{=}\text{r}{=}0.028\text{m}$$

$$l{=}l_3\text{-}l_1{=}0.120\text{-}0.028{=}0.092\text{m}$$

Once l and masses m and M are obtained, the I_b can be found using the following Simulink set of blocks to store the acceleration, the velocity and the angle of the pendulum. Nevertheless, a filter must be implemented due to the noise which affects the acceleration. The filter used is a Butterworth second order filter with a cut-off frequency of 5Hz, gain $1(k = 1)$, $a = 1.414$ and $k_o = 1$.

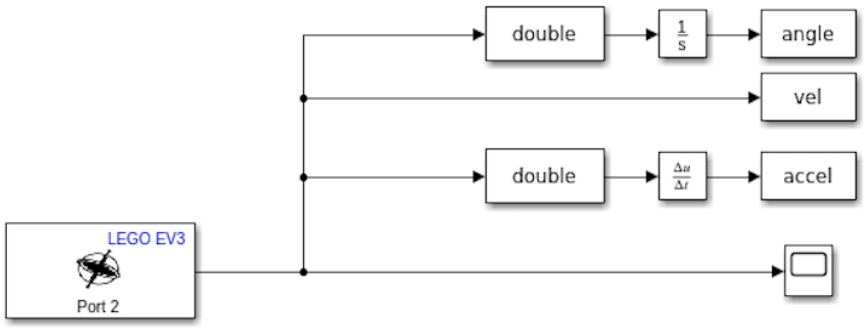


Figure 2.18: Set of blocks to obtain the data from the pendulum

Once the acceleration has been filtered and shifted, the angle vector and the acceleration vector obtained in the experiment are going to be used to calculate a vector which will give us the inertia of the bar during the whole experimentation, this inertia will be obtained using equation 2.43.

The useful data to calculate the inertia is the data between the interval of time 2.10 s and 2.20 s, this interval is the appropriate because at approximately 2.00 s the pendulum has a an acceleration of 0 rad/s^2 and the θ is 31.64°, which would give an indeterminate inertia. On the other hand, when the pendulum approximates to $\theta = 0°$(t=2.30) the acceleration is also 0 which produces also an indeterminate value of the inertia, in our case due to the frequency in which the gyroscope takes the information, it is technically difficult to be precise enough to get an acceleration of 0 rad/s^2 and an angle of 0°; however, it still can be seen that the inertia tends to infinity when the pendulum reaches the $\theta = 0°$ position. Moreover, it does not matter if the graphic displays a positive and a negative inertia, this is due to the corrections that have been done to the data obtained from the incremental encoder, in other words, due to having an incremental encoder, the encoder initially give a $\theta = 0°$ when in the

representation(2.15b), it would start from the position $\theta = 31,64°$. Thus, the acceleration and the $\sin\theta$ will not have the same sign during the experimentation but this does not affect when selecting the value of the inertia. Furthermore, the data from time 2.10 s to 2.20 s is selected to avoid the error accumulation in comparison with the other intervals of time. Taking into account what it has been mentioned, the values of the inertia $I_b = 0.025\ kg \cdot m^2$.

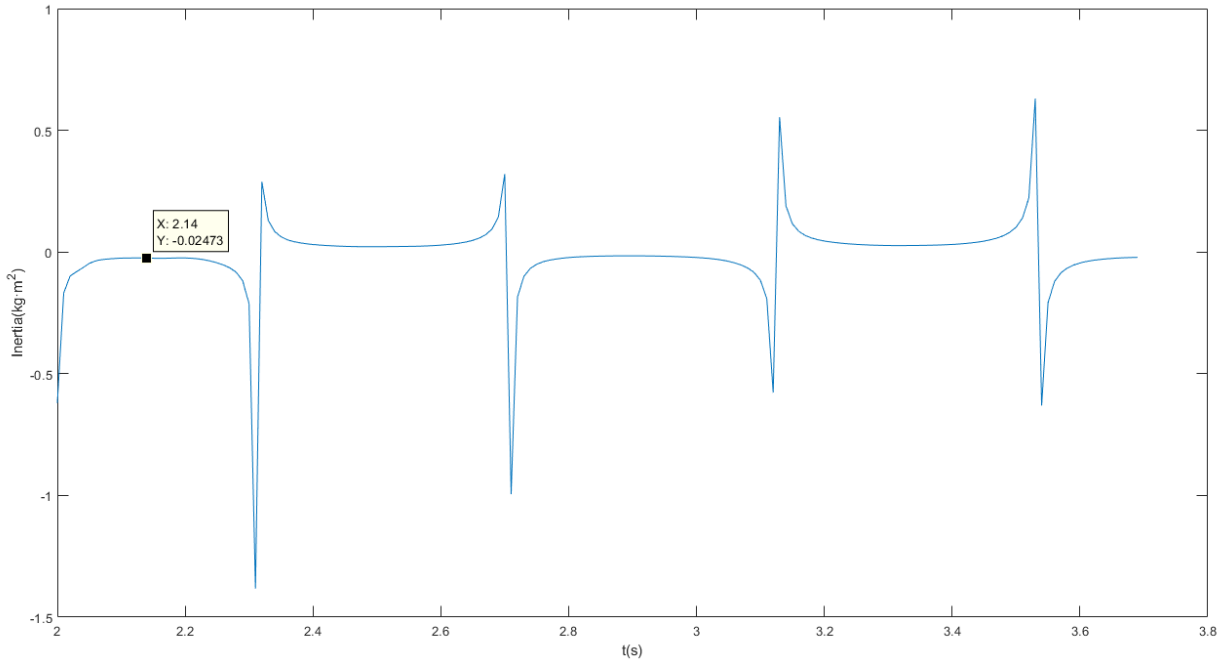


Figure 2.19: Values of the inertia vector

Regarding T_f, it includes the unknown constant b which has to be estimated. To estimate the parameter b an experiment is going to be committed, this experiment will consist on setting the wheel at the maximum velocity while the velocity is measured. When the maximum velocity will be reached the power to the motor will be cut off to make the wheel decelerate mainly by the torque T_f. With the data obtained, the following equation(2.45) will give the unknown parameter b for an interval of velocities; thus, a polynomial regression will be done to estimate an equation which will give the parameter b depending on the velocity of the wheel.

$$\Delta E_c = T_f \cdot \Delta\alpha = b \cdot (\dot{\alpha} - \dot{\theta}) \cdot \Delta\alpha \tag{2.45}$$

$$\frac{1}{2} \cdot M \cdot r^2 \cdot (\dot{\alpha_1}^2 - \dot{\alpha_2}^2) = b \cdot (\dot{\alpha} - \dot{\theta}) \cdot \Delta\alpha \tag{2.46}$$

To calculate $\dot{\alpha}$, an average will be done between the velocity $\dot{\alpha_1}$ and the velocity $\dot{\alpha_2}$ and theta in the experiment is 0 due to the bar is not moving; thus, the final equation used to

calculate b in different intervals is the following one 2.48. However, this experiment is not going to be done due to the limited time to do this bachelor thesis.

$$\frac{1}{2} \cdot M \cdot r^2 \cdot (\dot{\alpha_1^2} - \dot{\alpha_2^2}) = b \cdot \frac{\dot{\alpha_1} - \dot{\alpha_2}}{2} \cdot \Delta\alpha \tag{2.47}$$

$$b = \frac{\frac{1}{2} \cdot M \cdot r^2 \cdot 2(\dot{\alpha_1} + \dot{\alpha_2})}{\Delta\alpha} \tag{2.48}$$

Thus, if this experiment is not committed, it is not possible to estimate b, then, T_f is not going to be considered in the state space representation, which changes the equation 2.34 into the representation 2.49.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g \cdot l \cdot m}{I_b + l^2 \cdot m} & 0 & 0 & -\frac{l \cdot m}{\tau(I_b + l^2 \cdot m)} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K \cdot K_c \cdot l \cdot m \cdot r}{\tau(I_b + l^2 \cdot m)} \\ 0 \\ \frac{K \cdot K_c \cdot r}{\tau} \end{bmatrix} \cdot u(t) \tag{2.49}$$

Using the values found through experimentation, the representation 2.49 takes the following values.

$$r\text{=}0.028 \text{ m}$$

$$K_c\text{=}\frac{\pi}{180} \text{ rad/deg}$$

$$I_b\text{=}0.025 \ kg \cdot m^2$$

$$K\text{=}8 \ deg/s$$

$$g\text{=}9.8 \ m/s^2$$

$$l\text{=}0.092 \text{ m}$$

$$m\text{=}0.526 \text{ kg}$$

$$M\text{=}0.023 \text{ kg}$$

$$\tau\text{=}0.0546 \text{ s}$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 16.1 & 0 & 0 & -30.1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -18.3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.118 \\ 0 \\ 0.0716 \end{bmatrix} \cdot u(t) \tag{2.50}$$

Getting deep into our systems model, as it can be observed, our system oscillates due to the imaginary part of the conjugate couple of poles. Moreover, the robot is unstable as the dominant pole is located at the real positive axis.

$$poles = |Is - A| = s\,(s + 18.3)\,(s - 4.01)(s + 4.01) \tag{2.51}$$

2.4 State space concepts

To create controllers which will stabilize the robot several concepts have to be explained. In the state space representation there are two basic concepts, the first concept is the **controllability** of a system. A system is controllable when there is the possibility to change the values of the state vector from t_o to any desired value in t_f. The second concept is the observability, a system is observable if it is possible to determine the state vector in a time t_o with the output value y and the input w during a time interval[10].

Mathematically speaking, there is a matrix called the **controllability matrix**[10] which is defined as the following expression, where n is the rows or the columns of the A matrix. This matrix is useful because it indicates if the system is controllable when the determinant of the controllability matrix is different than 0.

$$[B|AB|\cdots|A_{n-1}B] \tag{2.52}$$

Using the state space representation 2.50, it is possible to calculate the controllability matrix for the particular robot built in this thesis, which determinant is different than 0; thus the system is controllable.

$$[B|AB|\cdots|A_{n-1}B] = \begin{bmatrix} 0 & 0.118 & -2.15 & 41.4 \\ 0.1176 & -2.15 & 41.4 & -757 \\ 0 & 0.0716 & -1.31 & 24.0 \\ 0.0716 & -1.31 & 24.0 & -440 \end{bmatrix} \tag{2.53}$$

$$0.0184 = det([B|AB|\cdots|A_{n-1}B]) \tag{2.54}$$

Regarding **observability**, it can be known using the **observability matrix**, which is defined

as the next expression.

$$\begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ \cdot \\ CA_{n-1} \end{bmatrix} \tag{2.55}$$

With this matrix, it is possible to know if the system is observable whether the determinant is not 0. Nevertheless, in this case the determinant is 0 because the observability matrix is not squared.

$$\begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ \cdot \\ CA_{n-1} \end{bmatrix} = \begin{bmatrix} 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0.0016 & 0 & 0 & -0.0030 \\ 0 & 0 & 0 & 0.0001 \\ 0 & 0.0016 & 0 & 0.0551 \\ 0 & 0 & 0 & -0.0018 \\ 0.0259 & 0 & 0 & -1.06 \\ 0 & 0 & 0 & 0.0335 \end{bmatrix} \tag{2.56}$$

$$0 = det(\begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ \cdot \\ CA_{n-1} \end{bmatrix}) \tag{2.57}$$

Another concept which is useful to design state space models is the concept of duality. To explain this concept it is necessary to display the following systems [10].

System S1

$$\dot{x} = A \cdot x + B \cdot w \tag{2.58}$$

$$y = C \cdot x \tag{2.59}$$

System S2

$$\dot{z} = A^* \cdot z + C^* \cdot v \tag{2.60}$$

$$n = B^* \cdot z \tag{2.61}$$

Here the symbol * indicates the conjugate transpose of a matrix. Looking back at the duality, "the principle of **duality** states that the system S1 is completely state controllable (observable) if and only if system S2 is completely observable (state controllable)" [10].

2.5 Control methods

There are many ways to obtain the desired behavior of our robot: pole placement, quadratic optimal methods, robust control and many more. In this bachelor thesis, a linear quadratic regulator(LQR) controller and a controller based on the pole placement method are going to be used.

2.5.1 Pole placement

Pole placement is a control method that is based on selecting the position of the system's closed loop poles by using a matrix K. Hence, to use this method it is compulsory to have a controllable system, because if not we will not be able to pass from the state vector in t_o to the state vector desired ($x_1 = 0, x_2 = 0, x_3 = 0$ and $x_4 = 0$) in a time t_f.

For a system that does not take external inputs, the representation of our system in the block diagram representation would be the following one, where the input vector u is -K·x and the letters A, B, C and D are the corresponding matrices of our state space model.

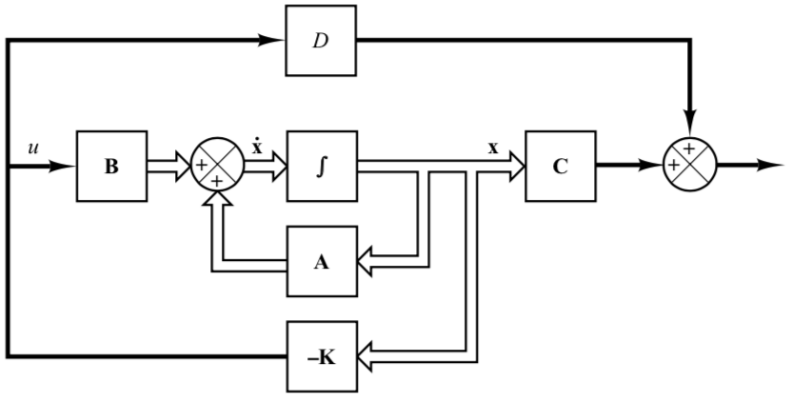


Figure 2.20: Pole placement
[10]

The objective in the pole placement method is to find the appropriate matrix K. To find K we are going to study the mathematical representation of the model displayed in figure 2.20.

$$\dot{x} = A \cdot x + B \cdot (-K \cdot x) \tag{2.62}$$

$$y = C \cdot x + D \cdot (-K \cdot x) \tag{2.63}$$

$$\dot{x} = (A - B \cdot K) \cdot x \tag{2.64}$$

$$y = (C - D \cdot K) \cdot x \tag{2.65}$$

In the state space representation, the poles of our system are given by the eigenvalues of the matrix that is multiplying the state vector x. Hence, in equation 2.64 the matrix which will give us the eigenvalues is $(A - B \cdot K)$. To obtain the matrix K, poles must be selected and imposed in the equation 2.66.

$$|I \cdot s - (A - B \cdot K)| = poles = 0 \tag{2.66}$$

To use the pole placement method it is necessary to know the matrix K and all the state variables of the vector x which are multiplying the matrix K ($w = u = -K \cdot x$). The problem of using the pole placement method comes when we want to know all values of x. The robot incorporates a gyroscope and encoders which give the angular velocity and the position of the robot, in other words, these devices give us x_2 and x_3 but not x_1 and x_4; therefore, a state observer to get these values is required. A **state observer** is a device or a computer program which is used to calculate an approximate value of state variables by using the output values of the state space representation(y vector) and the input w [10].

Observer design

The following image displays the system and the observer. In the case studied in this thesis, it is impossible to directly measure x_1 and x_4; therefore, a state observer is necessary to obtain these values that will be used by the vector K.

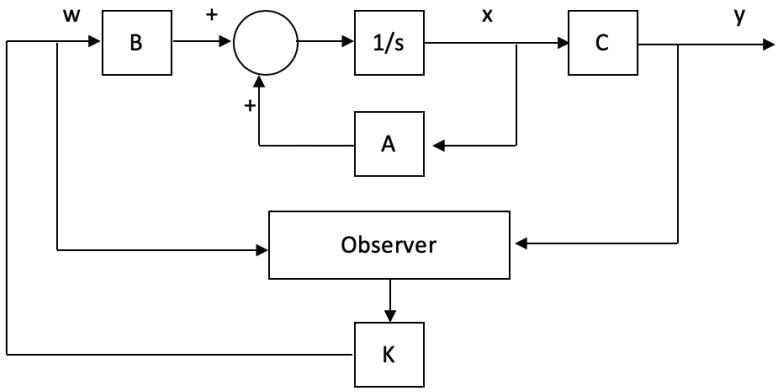


Figure 2.21: Observer model

First of all, there are two kinds of observers, full-order observers and minimum-order observers. Full-order observers give an approximation of all the state variables; however, to design a full-order observer it is necessary that the system will be observable which in our case it is not. On the other hand, there is another kind of observer called the minimum-order observers, these kind of observers are used to obtain an approximation of just the state variables that can not be measured. In our case, a minimum-order observer will be used due to the fact that the system is not observable. In the following explanation of the minimum-order state observer, the full-order state observer is going to be explained first to give a gentle introduction to the reader.

Full-order observer

All observers have an error that is represented as the expression 2.69, where $\hat{x}$ is the observed state space vector.

$$\dot{x} = A \cdot x + B \cdot w \tag{2.67}$$

$$y = C \cdot x \tag{2.68}$$

$$e = x - \hat{x} \tag{2.69}$$

The expression 2.70 is the modification of the equation 2.67 which includes the therm $K_e \cdot C \cdot e$. This term at the end is used to correct the approximations of the state variables, in which K_e is a vector [10]. The equation 2.70 is the state space model in which the observer will be based to estimate $\hat{x}$.

$$\dot{\hat{x}} = A \cdot \hat{x} + B \cdot w + K_e \cdot (y - C \cdot \hat{x}) \tag{2.70}$$

$$\dot{\hat{x}} = A \cdot \hat{x} + B \cdot w + K_e \cdot (C \cdot x - C \cdot \hat{x}) \tag{2.71}$$

$$\dot{\hat{x}} = A \cdot \hat{x} + B \cdot w + K_e \cdot C \cdot e \tag{2.72}$$

Now our target is to obtain an observer which will approximate the state variables with few errors. Moreover, errors have to decrease faster than the stabilization of the system, because if not, the errors in the estimated state variables will cause instability when the system will try to reach the balance. In other words, the error dynamics of the observer must be studied to select the appropriate K_e which will decrease the steady-state time and the steady-errors. This study will be executed through expression 2.73.

$$\dot{e} = \dot{x} - \dot{\hat{x}} \tag{2.73}$$

The term $\dot{x}$ is given by the state space model (Equation 2.67) while $\dot{\hat{x}}$ is provided by equation 2.70. Therefore, equation 2.70 can be subtracted from equation 2.67 which will give expression 2.74 [10].

$$\dot{e} = \dot{x} - \dot{\hat{x}} = (A - K_e \cdot C) \cdot e \tag{2.74}$$

The error dynamics' equation is structurally similar to the state variables dynamics' equation (2.67), except for the lack of the $B \cdot w$ term in the error dynamic's equation. Thus, due to the similar mathematical structure, we could be able to obtain the eigenvalues of the error if the K_e vector had values. As K_e is not determined, the desired poles can be selected to obtain the vector K_e which will impose the desired error behavior, the solution of the expression 2.75 give us the vector K_e.

$$|I \cdot s - (A - K_e \cdot C)| = (s + p_1) \cdot (s + p_2) \cdot ... \cdot (s + p_n) = 0 \tag{2.75}$$

The next image displays the block representation of a full-order state observer, where the terms $\tilde{x}$ and $\tilde{y}$ correspond to the terms $\hat{x}$ and $C \cdot \hat{x}$.

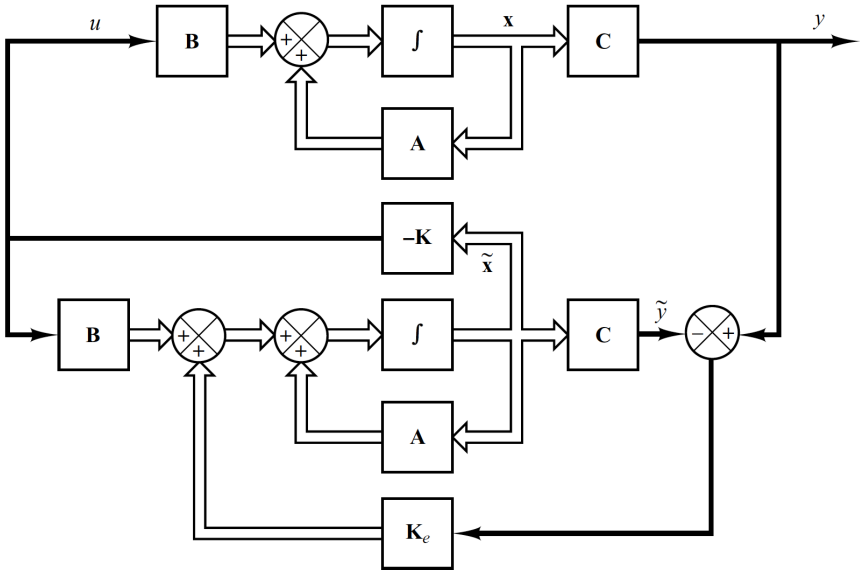


Figure 2.22: Full-order state observer
[10]

Minimum-order observer

As in the full-order observer, we will look at the state space representation to obtain the minimum-order observer. Nevertheless, expression 2.67 will be modified into expression 2.76. Where x_a is a vector containing the known terms x_2 and x_3 and x_b is the vector containing x_1 and x_4.

$$\begin{bmatrix} \dot{x_a} \\ \dot{x_b} \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} B_a \\ B_b \end{bmatrix} \cdot w \tag{2.76}$$

$$\dot{x_a} = A_{aa} \cdot x_a + A_{ab} \cdot x_b + B_a \cdot w \tag{2.77}$$

$$\dot{x_b} = A_{ba} \cdot x_a + A_{bb} \cdot x_b + B_b \cdot w \tag{2.78}$$

$$\dot{x_a} - A_{aa} \cdot x_a - B_a \cdot w = A_{ab} \cdot x_b \tag{2.79}$$

The objective of using an observer is to obtain the unknown state variables. In the full-order observer, there was the whole vector x as an unknown whereas in the minimum-order observer x_b is the unknown vector. Hence, the output of our minimum-order observer will be $A_{ab} \cdot x_b$ due to the similarity to $C \cdot x$.

$$\dot{x_a} - A_{aa} \cdot x_a - B_a \cdot w = y \tag{2.80}$$

To find the expression of the minimum-order observer, it is going to be taken into consideration expression 2.70 of the full-order observer. Terms y, $\hat{x}$ and $\hat{x}$ will be substituted by expression 2.79 and by terms $\dot{x_b}$ and x_b. Furthermore, expressions 2.78 and 2.67 are also quite similar, this similarity can be used to associate the term $A_{ba} \cdot x_a + A_{bb} \cdot x_b$ to $A \cdot x$ and the matrix B to B_b.

$$\dot{x_b} - A_{bb} \cdot x_b - B_b \cdot w = A_{ba} \cdot x_a \tag{2.81}$$

$$\dot{\hat{x}_b} = A_{ba} \cdot x_a + A_{bb} \cdot \hat{x_b} + B_b \cdot w + Ke \cdot (\dot{x_a} - A_{aa} \cdot x_a - B_a \cdot w - A_{ab} \cdot \hat{x_b}) \tag{2.82}$$

As stated before, the aim is to have an approximation of the unknown state variables with the minimum error. To study this error it is compulsory to study the error dynamics which is represented as the next equation (2.83).

$$\dot{e} = \dot{x_b} - \dot{\hat{x}_b} \tag{2.83}$$

$$e = x_b - \hat{x_b} \tag{2.84}$$

Substituting the term $A_{ba} \cdot x_a$ of the equation 2.81 into 2.82, it is going to be find the expression 2.86 which will give the error's dynamics (Equation 2.87).

$$\dot{x_b} - A_{bb} \cdot x_b - B_b \cdot w = A_{ba} \cdot x_a \tag{2.85}$$

$$\begin{aligned} \dot{\hat{x}_b} = \dot{x_b} - A_{bb} \cdot x_b - B_b \cdot w + A_{bb} \cdot \hat{x_b} + B_b \cdot w + Ke \cdot (A_{aa} \cdot x_a \\ + A_{ab} \cdot x_b + B_a \cdot w - A_{aa} \cdot x_a - B_a \cdot w - A_{ab} \cdot \hat{x_b}) \end{aligned} \tag{2.86}$$

$$\dot{e} = (A_{bb} - K_e \cdot A_{ab}) \cdot e \tag{2.87}$$

As it has been mentioned before, the poles of the error dynamics can be imposed due to the unconstrained values of the matrix K_e. Thus, the desired behavior of the error dynamics can be imposed.

However, there is one condition, the error dynamics behavior of the minimum-order observer must be faster than the behavior of the system, because in the case that the system would be faster than the error dynamics, the system would catch a bad approximation of the x_b that would affect the input to the system $w = -K \cdot x$.

To represent the model of a minimum-order observer, equation 2.86 is going to be rewritten into the following set of expression.

$$\dot{\hat{x}_b} = (A_{bb} - K_e \cdot A_{ab}) \cdot \hat{x_b} + A_{ba} \cdot x_a + B_b \cdot w + K_e \cdot (\dot{x_a} - A_{aa} \cdot x_a - B_a \cdot w) \tag{2.88}$$

$$\dot{\hat{x}_b} = (A_{bb} - K_e \cdot A_{ab}) \cdot \hat{x_b} + (A_{ba} - K_e \cdot A_{aa}) \cdot x_a + (B_b - K_e \cdot B_a) \cdot w + K_e \cdot \dot{x_a} \tag{2.89}$$

$$\dot{\hat{x}_b} = \hat{A} \cdot \hat{x_b} + \hat{B} \cdot w + \hat{E} \cdot x_a + K_e \cdot \dot{x_a} \tag{2.90}$$

Picture 2.23 present the block representation of a minimum-order state observer using Simulink. To make the representation more understandable, a description of the names used in figure 2.23 are presented.

xbhat= $\hat{x_b}$

Ahat=$\hat{A}$

Bhat=$\hat{B}$

Ehat=$\hat{E}$

As it has been said, the pole placement method is based on the modification of the poles of the system by using the matrix K. Therefore, knowing the desired poles of the system is needed to build the matrix K, and here is when the big problem comes. In this bachelor thesis, it is desired that the robot will act fast enough and with good precision to place the bar at the desired position. Hence, the response of the system must be stable, must have a small steady state error and a small settling time. This is traduced into poles that are positioned at the left real axis of the complex plane as well as complex conjugate poles which must have small complex values apart from having the real part at the left real axis. Although it is easy to select any poles in simulation and obtain the desired behavior, in experimentation, it is not that easy due to the limits of the system which restrict the poles selection.

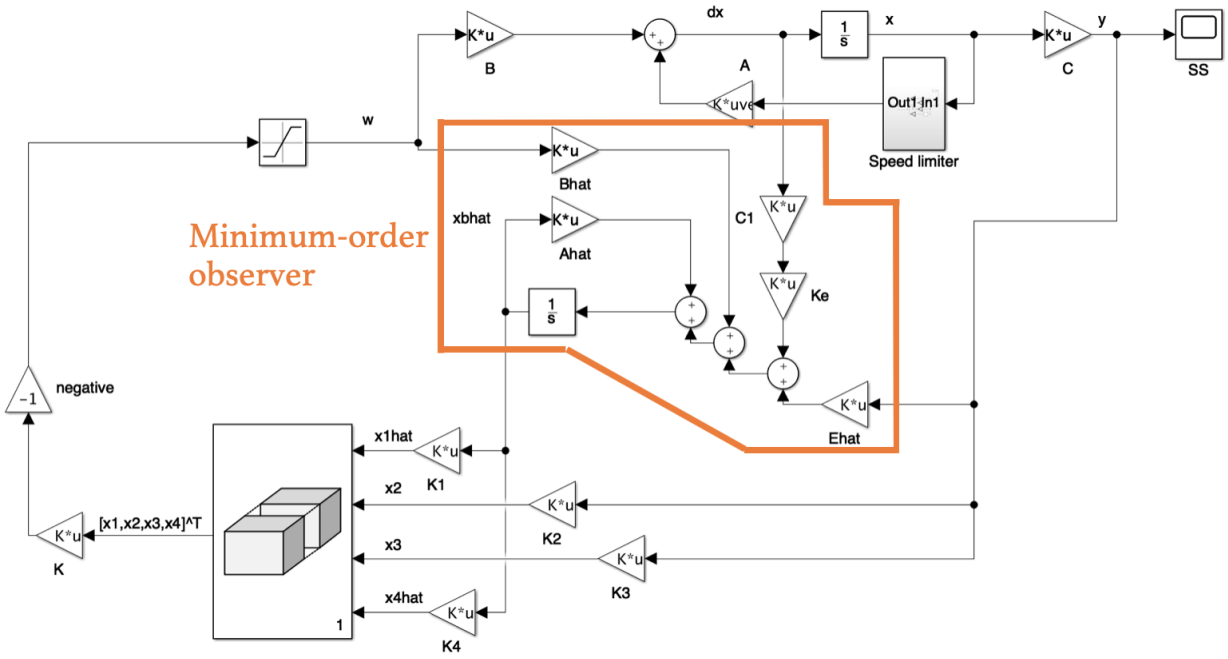


Figure 2.23: Minimum-order state observer

2.5.2 Linear quadratic regulator

The linear quadratic regulator control method commonly known as LQR, is a control method based on the optimization of a function named "cost function"(equation 2.91) through the variation of two matrixes, Q and R, which are determined by the user [12] [13].

$$J = \int_0^\infty (x^T \cdot Q \cdot x + w^T \cdot R \cdot w)dt \tag{2.91}$$

The LQR controller is quite similar to the pole placement method, in other words, the aim of this method is to obtain the K matrix that will change the behavior of the model to the desired performance. However, the difference is the way in which the matrix K is obtained.

$$K = R^{-1} \cdot B^* \cdot P \tag{2.92}$$

$$w(t) = -K \cdot x(t) = -R^{-1} \cdot B^* \cdot P \cdot x(t) \tag{2.93}$$

In the LQR method, K is defined as the expression 2.92, where the B matrix of the expression 2.92 is the B matrix of our state space model. As R is determined by the user, the only unknown matrix that is necessary to find is P, which will be found by solving the Riccati equation(equation 2.94) [10], where the A matrix corresponds to the matrix A in the state space model .

$$A^* \cdot P + P \cdot A - P \cdot B \cdot R^{-1} \cdot B^* \cdot P + Q = 0 \tag{2.94}$$

The problem in the LQR method is to select the proper Q and R matrixes to find the matrix K which will approximate the performance of the robot to the desired one. Matrixes Q and R must be positive definite or positive-semidefinite matrixes in order to reach the minimum value(optimal value) of the function J. The positive definite matrix is defined as a diagonal matrix like the one below with the particular characteristic that all the diagonal terms are bigger than 0. On the other hand, the positive-semidefinite matrix is a diagonal matrix like the one below but with some diagonal terms equal to 0.

$$M = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_r \end{bmatrix} \tag{2.95}$$

To select Q and R, the terms of these two matrixes will take the values of the identity matrix's terms, after this step, a simulation will be committed to know the effect of the matrix Q and R over the behavior of the robot. Consequently, several terms of Q and R will be changed taking into consideration the simulation results [14] and the wanted behavior.

To modify the matrixes Q and R correctly, it is necessary to understand the effect of these matrixes over the state variables and the input w. The Q matrix affects the state variables x while the matrix R affects the input w, this can be seen in equation 2.94. Therefore, as the aim is minimizing the function J, if the values of matrix Q are increased, the state variables will decrease to make the function J minimum. On the other hand, if the matrix Q is decreased, the state variables will increase to reach that minimum, the same concept is applied to the $(w^T \cdot R \cdot w)$ term [15].

In order to fully understand how to modify matrixes Q and R, an example will be presented. Taking the state space representation of the used robot, the state vector has 4 state variables, x_1 is the angle, x_2 is the velocity of the angle, x_3 is the position of the robot and x_4 is the velocity of the robot. Hence, taking the expression 2.94 the matrix Q has to be a 4x4 matrix, where the term [1][1](first row first column), will modify the value of the x_1 term, the term [2][2] of the matrix Q will affect x_2, term [3][3] will adjust x_3 and [4][4] change the performance of x_4. Therefore, if the value corresponding to the term [1][1] of the matrix Q is increased, the values of the angle (x_1) will be restricted, in other words, peaks will be decreased. Regarding the angular velocity (x_2), if term [2][2] of matrix Q is increased the

angular velocity would be limited; thus the velocity of the angle would be slower than if the term [2][2] would have been decreased.

MATLAB has a command named $lqr(A, B, Q, R)$ which is really helpful in regard with solving the equations 2.94, 2.92 and 2.91. Hence, with MATLAB the only thing that the designer has to worry about is selecting the correct Q and R matrixes [10].

Kalman filter

As an optimal control method has been used to design the controller, an optimal observer will be introduced. The Kalman filter is a type of observer which takes into account noises that affect our system to achieve a good approximation of the state variables.

In real life, our systems are subjected to noises which affect the behavior of systems. Taking into consideration the state space model(equation 2.25 and 2.26), these equations are modified by noises in the following way.

$$\dot{x} = A \cdot x + B \cdot w + ns_x \tag{2.96}$$

$$y = C \cdot x + D \cdot w + ns_y \tag{2.97}$$

To facilitate the comprehension of the Kalman filter a discretization of the equations 2.96 and 2.97 is going to be done.

$$x[k+1] = A \cdot x[k] + B \cdot w[k] + ns_x[k] \tag{2.98}$$

$$y[k] = C \cdot x[k] + D \cdot w[k] + ns_y[k] \tag{2.99}$$

The noises presented in equations 2.98 and 2.99 are disturbances which follow a normal distribution with a 0 mean value and σ^2 as a variance. Knowing that noises behave as a normal distribution, the main objective of the Kalman filter is to reduce the variance of these noises to make the state variables approximations more accurate in comparison with the full-order and the minimum-order state observers. This reduction of noises is achieved through the modification of the Kalman gain K [16], [17].

In order to estimate the state variables, it is necessary to use an observer with a little modification in this case. The K of the expressions below is the Kalman filter gain which is found through optimization processes [16].

$$\hat{x}[k+1] = A \cdot \hat{x}[k] + B \cdot w[k] + K[k] \cdot (y[k] - C \cdot \hat{x}[k]) \tag{2.100}$$

To make the mathematical expressions more simple and more similar to our case, the w term is going to be deprecated due to the null value of the D matrix in the expression 2.99.

$$\hat{x}[k+1] = A \cdot \hat{x}[k] + B \cdot w[k] + K[k] \cdot (C \cdot x[k] + ns_y[k] - C \cdot \hat{x}[k]) \tag{2.101}$$

In the observer case, the robot is not able to obtain all the state variables by only using a controller; hence, an observer is needed to approximate the state variables missing. Thanks to the Kalman filter, the controller can estimate those state variables with more precision due to the contemplation of noise effects in the model.

Following the steps done in the full-order state observer, a null error is wanted between the state variables and the state variables approximated.

$$e[k+1] = x[k+1] - \hat{x}[k+1] \tag{2.102}$$

$$e[k+1] = A \cdot (x[k] - \hat{x}[k]) + ns_x[k] - K[k] \cdot ns_y[k] - K[k] \cdot C \cdot (x[k] - \hat{x}[k]) \tag{2.103}$$

$$e[k+1] = A \cdot e[k] + ns_x[k] - K[k] \cdot ns_y[k] - K[k] \cdot C \cdot e[k] \tag{2.104}$$

$$e[k+1] = (A - K[k] \cdot C) \cdot e[k] + ns[k] \tag{2.105}$$

As it was mentioned, the Kalman filter achieves good results of state variables due to the diminution of the noise variance. This reduction in the noise variance can be achieved through the following process.

$$V(k+1) = E[e[k+1] \cdot e[k+1]^T] \tag{2.106}$$

$$V(k+1) = E[((A - K[k] \cdot C) \cdot e[k] + ns[k])(e[k]^T \cdot (A - K[k] \cdot C)^T + ns[k]^T)] \tag{2.107}$$

$$V(k+1) = E[((A - K[k] \cdot C) \cdot e[k] + ns[k])(e[k]^T \cdot (A - K[k] \cdot C)^T + ns[k]^T)] \tag{2.108}$$

$$\begin{aligned} V(k+1) = (A - K[k] \cdot C) \cdot E[e[k] \cdot e[k]^T] \cdot (A - K[k] \cdot C)^T + (A - K[k] \cdot C) \\ \cdot E[e[k] \cdot n[k]^T] + E[n[k] \cdot e[k]^T] \cdot (A - K[k] \cdot C)^T + E[n[k] \cdot n[k]^T] \end{aligned} \tag{2.109}$$

As it can be seen in equations 2.96 and 2.96, the derived state vector depends on ns_x but the state vector does not depend on any noise. Hence, the term $E[e[k] \cdot n[k]^T]$ and $E[n[k] \cdot e[k]^T]$ are 0. Moreover, the term $E[e[k] \cdot e[k]^T]$ can be rewritten as $V[k]$.

$$V(k+1) = (A - K[k] \cdot C) \cdot V[k] \cdot (A - K[k] \cdot C)^T + E[n[k] \cdot n[k]^T] \tag{2.110}$$

$$V(k+1) = (A - K[k] \cdot C) \cdot V[k] \cdot (A - K[k] \cdot C)^T + E[(ns_x - K[k] \cdot ns_y) \cdot (ns_x - K[k] \cdot ns_y)^T] \tag{2.111}$$

Terms ns_x and ns_y are independent of each other. Thus terms $-ns_x \cdot (K[k] \cdot ns_y)^T$ and $K[k] \cdot ns_y \cdot ns_x$ are 0.

$$E[(ns_x - K[k] \cdot ns_y) \cdot (ns_x - K[k] \cdot ns_y)^T] = E[ns_x \cdot ns_x^T + K[k] \cdot ns_y \cdot ns_y \cdot K[k]^T] \tag{2.112}$$

$$E[ns_x \cdot ns_x^T] = Ns_x \tag{2.113}$$

$$E[ns_y \cdot ns_y^T] = Ns_y \tag{2.114}$$

$$V(k+1) = (A - K[k] \cdot C) \cdot V[k] \cdot (A - K[k] \cdot C)^T + Ns_x + K[k] \cdot Nsy \cdot K[k]^T \tag{2.115}$$

As it is possible to vary K[k] to change the error variance, expression 2.115 is going to be derived with respect to K[k] to find the minimum variance [16].

$$\frac{\partial Q}{\partial k} = 2(A - K[k] \cdot C) \cdot V[k] \cdot (-C^T) + 2 \cdot K[k] \cdot Ns_y = 0 \tag{2.116}$$

$$K[k] = A \cdot V[k] \cdot C^T \cdot (Ns_y + C \cdot V[k] \cdot C^T)^{-1} \tag{2.117}$$

The expression 2.117 depends on V[k] which is a parameter that is not specified; hence, to obtain $K[k]$ it is necessary to define an iteration process. First, an initial value $V[0]$ will be specified, with this value, $K[0]$ can be found and finally, these values of $K[0]$ and $V[0]$ will be used to obtain $V[k+1]$. If this process is repeated, $K[k]$ will converge, resulting in the obtaining of the K vector for a predictive type Kalman filter [16].

Thanks to MATLAB, the iteration process is not required to find K. Instead, a block of Simulink named Kalman filter will be used to facilitate the representation [18] [19].

2.6 MATLAB simulations

2.6.1 LQR controller

First, it is necessary to define matrixes Q and R in order to use the MATLAB command $lqr(A, B, Q, R)$, which will gives the K vector(Kalman gain). As Q in equation 2.91 is multiplying the state vector, the dimensions of Q must be n x n where n is the number of state variables of our system, which in our case is 4. In regard to R, this matrix is multiplying the input w; therefore, R has the dimension of 1x1. As a common practice to design the LQR controller, it is usual to assign the matrixes Q and R the values of the identity matrix and vary the values observing the behaviour of the system through simulations.

$$Q = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \tag{2.118}$$

$$R = [1] \tag{2.119}$$

Using the command $lqr(A, B, Q, R)$, MATLAB gives the following K, which will be named K_{lqr} in order to differentiate this vector from the K in the pole placement method.

$$K_{lqr} = [1524, 379.9, -1.000, -512.1] \tag{2.120}$$

As the block structure of the LQR method is similar to the pole placement method, it is possible to obtain the poles of the system to see which is the system's behavior. These poles indicates that the system is stable with little oscillations, this behavior can be observed in figure 2.26 for the initial conditions 2.123.

$$|I \cdot s - (A - B \cdot K_{lqr})| = (s + p_1) \cdot (s + p_2) \cdot ... \cdot (s + p_n) = 0 \tag{2.121}$$

$$\begin{aligned} |I \cdot s - (A - B \cdot K_{lqr})| &= (s + 0.0039) \cdot (s + 18.3) \\ &\cdot (s + (4.01 + 0.0128i)) \cdot (s + (4.01 - 0.0128i)) \end{aligned} \tag{2.122}$$

In regard to the model used to perform the simulations, a non linear model of the robot is going to be used; furthermore, as it is desired to control the position and the angle of the robot, x_1 and x_3 state variables are going to be observed during simulations to see how they behave.

NOTE: all the models displayed in Simulink include 2 saturation blocks, one for the maximum velocity of the motor and another for the input to the motors of the EV3. The saturation block for the motor, saturates the input between 0.513m/s [20] and -0.513m/s, regarding the saturation block for the input of motors, limits are between 100 and -100.

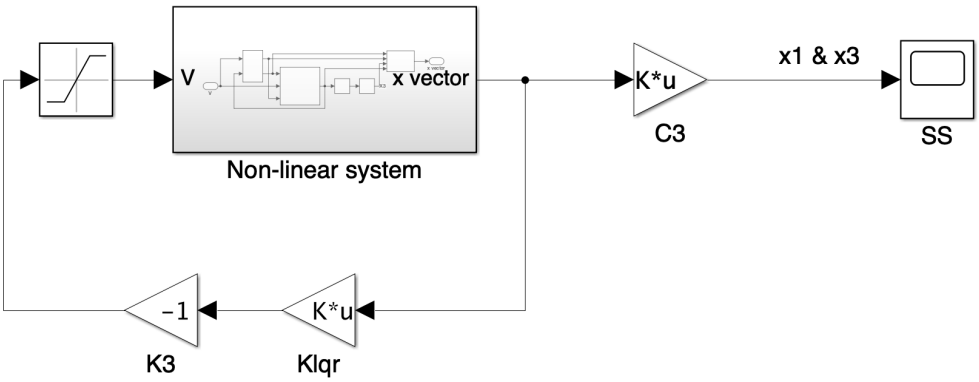


Figure 2.24: LQR

$$\begin{aligned} \theta &= -0.114rad \\ \dot{\theta} &= -0.06rad/s \\ x &= 0.005m \\ \dot{x} &= 0.05m/s \end{aligned} \tag{2.123}$$

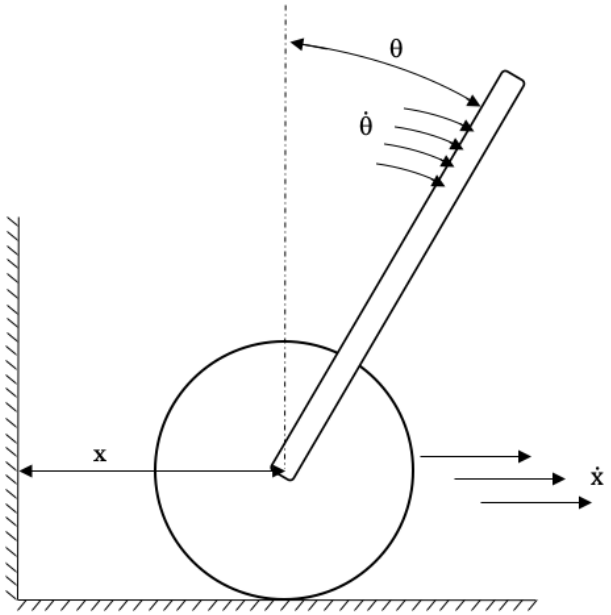


Figure 2.25: Representation of the initial conditions 2.123

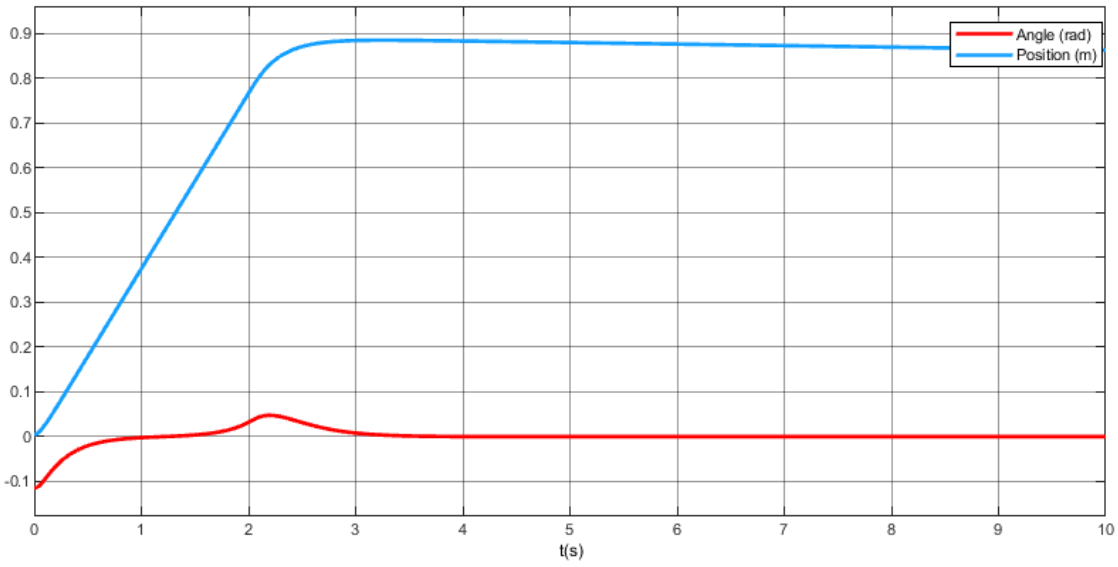


Figure 2.26: $x1$ and $x3$ in the first LQR simulation

The aim of this thesis is to design a control method which will produce a fast response to disturbances and will maintain a two wheel robot in balance. Hence, the velocity of the angle x_2 and the velocity of the position x_4 must be fast to generate a fast response. Therefore, to obtain a fast angle's velocity and a fast position's velocity, the system must be capable of driving as much voltage(w) as possible to motors in the minimum time, this is translated into the following, the value of the R matrix must be as low as possible to produce high values of w. Furthermore, to design a system with a fast response, terms [2][2] and [4][4] which affects to the velocity of the angle and the velocity of the position respectively, must be also as low as possible to reach the 0 value as rapid as possible.

With respect to the term [3][3] of the Q matrix, to approximate the position to 0 the value of the term [3][3] must be increased to restrict the position to be 0. This can be observed doing a comparison between graphic 2.26, which has been done with matrixes 2.118 and 2.119, and graphic 2.27, which has been done with matrixes 2.119 and 2.124.

$$Q = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1000 & \\ & & & 1 \end{bmatrix} \tag{2.124}$$

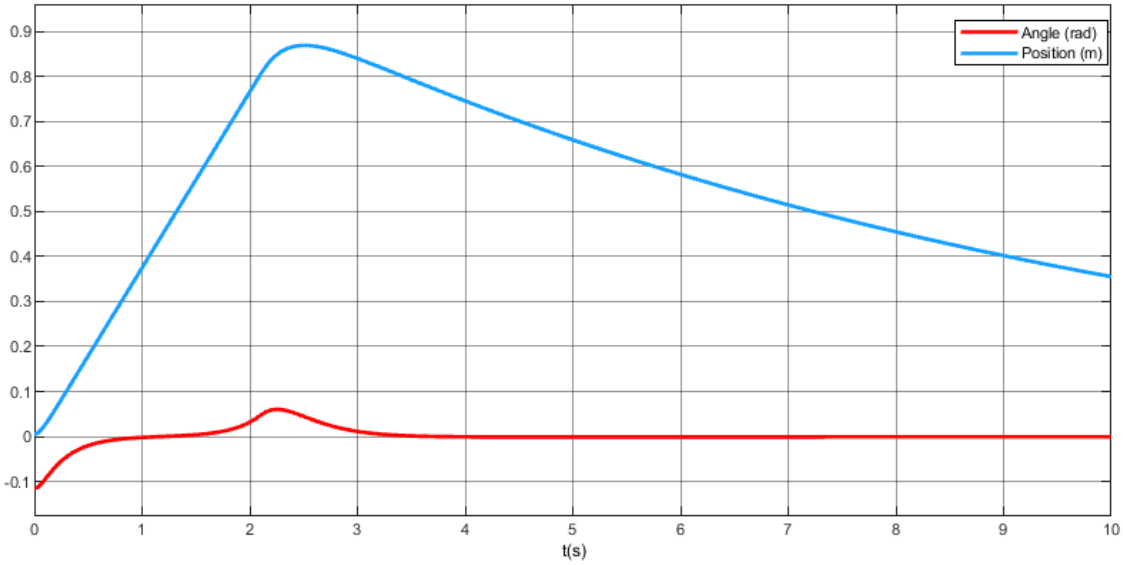


Figure 2.27: LQR simulation with matrixes 2.119 and 2.124

Regarding term [1][1], the value must be as high as possible to reduce the peak at approximately 2.3 s and to reach the o value as fast as possible.

After several simulations viewing the effect of each modification in matrixes Q and R over the performance of the robot, the selected Q and R matrixes are 2.125 and 2.126 which produce the output 2.28.

$$Q = \begin{bmatrix} 10 & & & \\ & 0.001 & & \\ & & 2100 & \\ & & & 0.08 \end{bmatrix} \tag{2.125}$$

$$R = [0.08] \tag{2.126}$$

Comparing images 2.28 and 2.26, it can be seen that the position has been driven to the 0 value faster than in figure 2.28. However, as a drawback of this increment in the velocity of the position to reach the 0 value, the angle peak at 2.2 s has increased and has been shifted at 2.5 s. Therefore, although the angle peak has experimented a little increment on its value , the response of the system has been improved in comparison with the initial behavior displayed in figure 2.26.

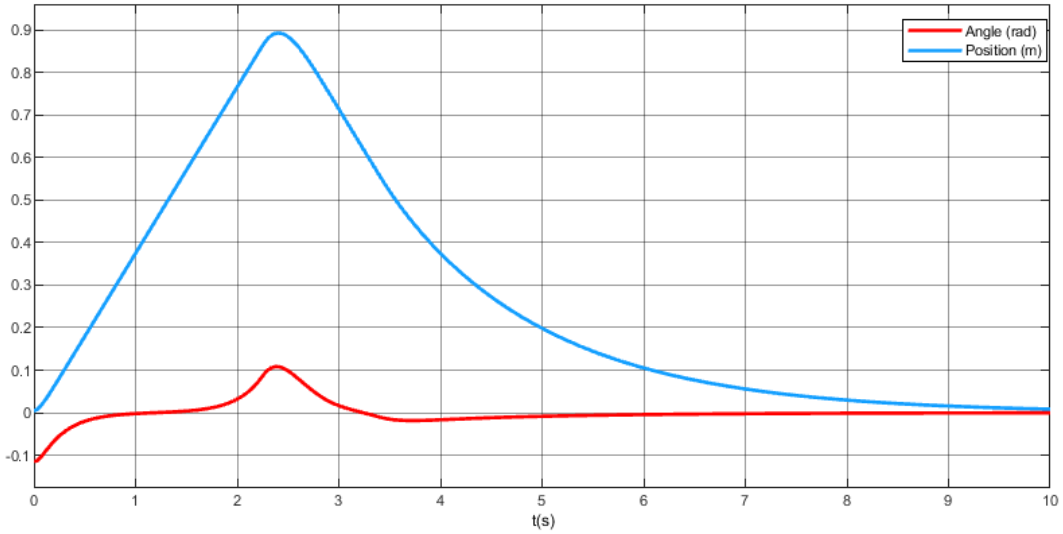


Figure 2.28: $x1$ and $x3$ in the LQR simulation

2.6.2 Pole placement controller

The pole placement method has the same block representation as the one presented in picture 2.24. However, instead of K_{lqr}, now it is a vector K found through the selection of the desired poles, which defines the behavior of the system.

As the model has 4 state variables, 4 poles must be selected. To make the system stable and with less oscillations, the 4 poles must be positioned at the negative real axis of the complex plane with no complex part. To start, 4 random poles close to the imaginary axes of the complex plane are going to be selected. This selection is done to ensure that the controller designed is possible, because poles with big negative values will increase the speed of the system which will turn out to rise the value of the input. And as it is known the value of the input is limited by a saturation block in the model; thus, the initial poles will have a low negative value and simulations will be performed to know if the system is possible, in the case that it will be possible, faster poles will be selected to make the system faster and after the selection simulations will be done to ensure that controller is feasible.

Initial poles:

$$(s + 0.15) \cdot (s + 0.1) \cdot (s + 0.2) \cdot (s + 0.19) \tag{2.127}$$

With these initial random poles close to the imaginary axes, the system behaves in the following way (Figure 2.29). Although the system should be stable due to the selected poles, it has to be taken into account that the simulated system include saturation blocks and a non linear model which makes the overall performance not linear, that is the reason why the behavior of the simulation is not stable.

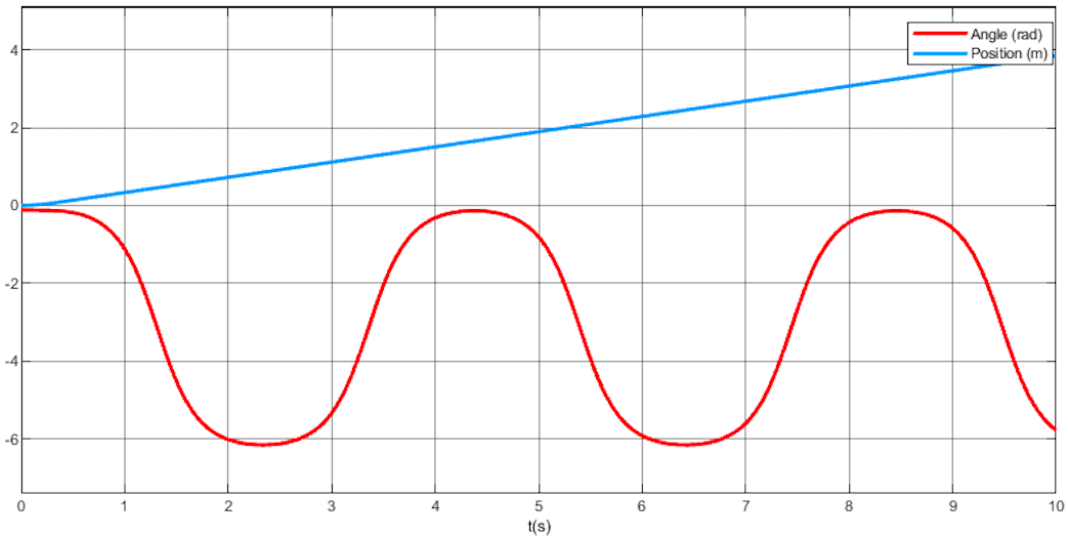


Figure 2.29: $x1$ and $x3$ in the pole placement simulation with poles 2.127

Although the initial poles have been proved to make the system unstable, 4 new random poles will be selected with bigger negative values. In a linear system without saturations, selecting bigger poles would not make any difference to make the system stable; however, in this case, a non linear system is treated and selecting bigger poles could cause the stabilization of the behavior.

Second simulation poles:

$$(s + 2.1) \cdot (s + 2.15) \cdot (s + 2.2) \cdot (s + 2.19) \tag{2.128}$$

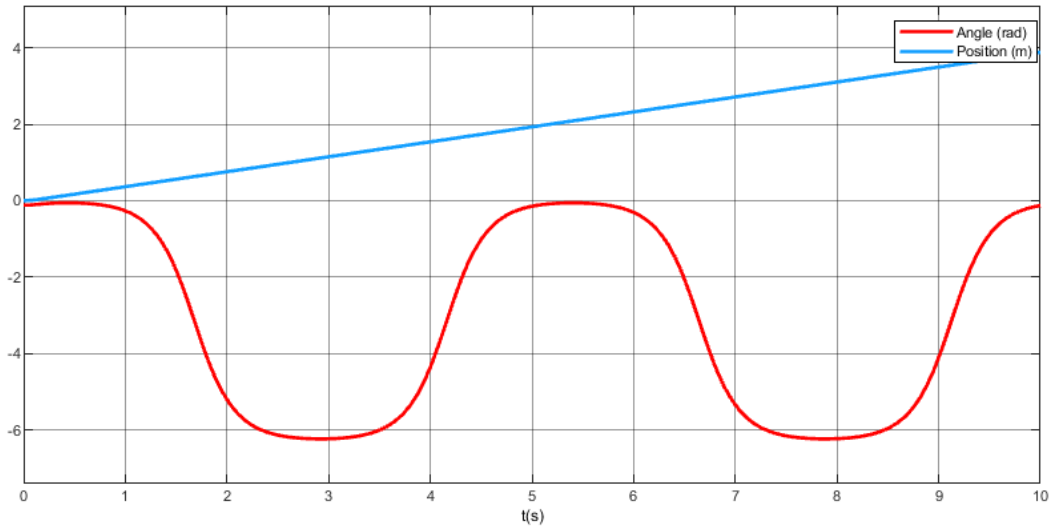


Figure 2.30: $x1$ and $x3$ in the pole placement simulation with poles 2.128

This second simulation also displays an unstable system. Therefore, 4 new random poles

with bigger negative values will be selected to see if the system makes any change in its behavior.

Third simulation poles:

$$(s + 4.15) \cdot (s + 4.1) \cdot (s + 4.2) \cdot (s + 4.19) \tag{2.129}$$

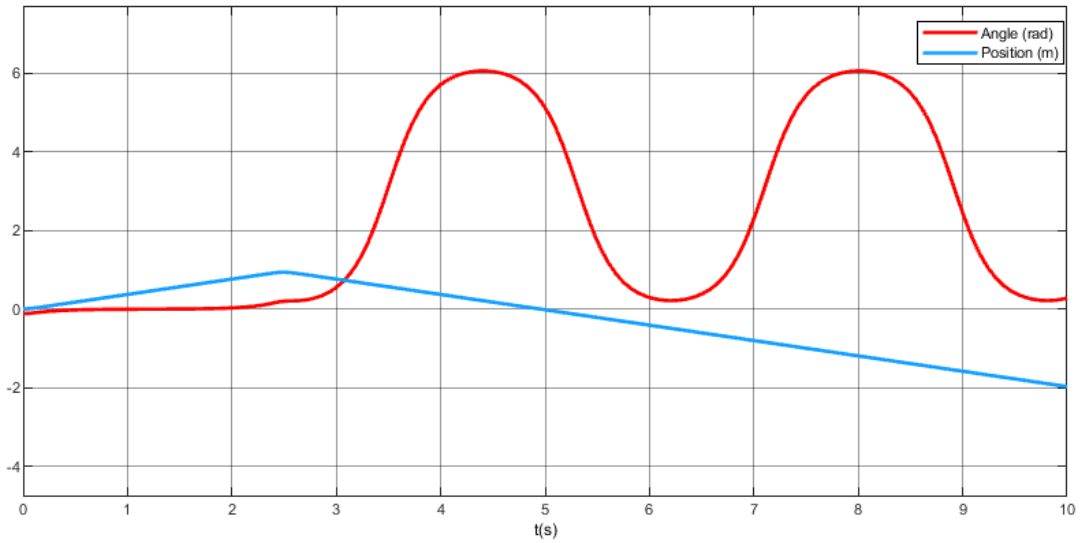


Figure 2.31: $x1$ and $x3$ in the pole placement simulation with poles 2.129

Examining figure 2.31, it can be seen that the position at approximately 2.5 s changes its direction to make the robot return to the position 0. Nevertheless, the controller is not able to make the system steady. After several simulations with bigger poles, it has not been possible to find the concrete poles which make the system stable. Thus, to develop a pole placement controller the poles of the system with the LQR controller will be obtained and changed to start from a reference which is known that produces a stable behavior.

To obtain the poles the eigenvalues of the matrix $(A - B \cdot K_{LQR})$ must be obtained.

$$poles = det(I \cdot s - (A - B \cdot k_{LQR})) \tag{2.130}$$

$$poles = det(I \cdot s - (\begin{bmatrix} 0 & 1 & 0 & 0 \\ 16.1 & 0 & 0 & -30.1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -18.3 \end{bmatrix} - \begin{bmatrix} 0 \\ 0.118 \\ 0 \\ 0.0716 \end{bmatrix} \cdot \begin{bmatrix} 1.76 \cdot 10^3 & 439 & -162 & -601 \end{bmatrix})) \tag{2.131}$$

$$poles = (s + 0.63) \cdot (s + 3.98) \cdot (s + 4.05) \cdot (s + 18.3) \tag{2.132}$$

To work with a pole placement controller, poles 2.132 will be used as if they were the initial poles imposed in the system. The aim of this thesis is to design a controller which will make the system reaching the balance fast and with the low errors as possible. Therefore, the slower pole will be changed to make the system faster. To start, the value of 1.63 will be changed into the value of 1, with this change the new poles are the following ones.

$$(s + 1.63) \cdot (s + 3.98) \cdot (s + 4.05) \cdot (s + 18.3) \tag{2.133}$$

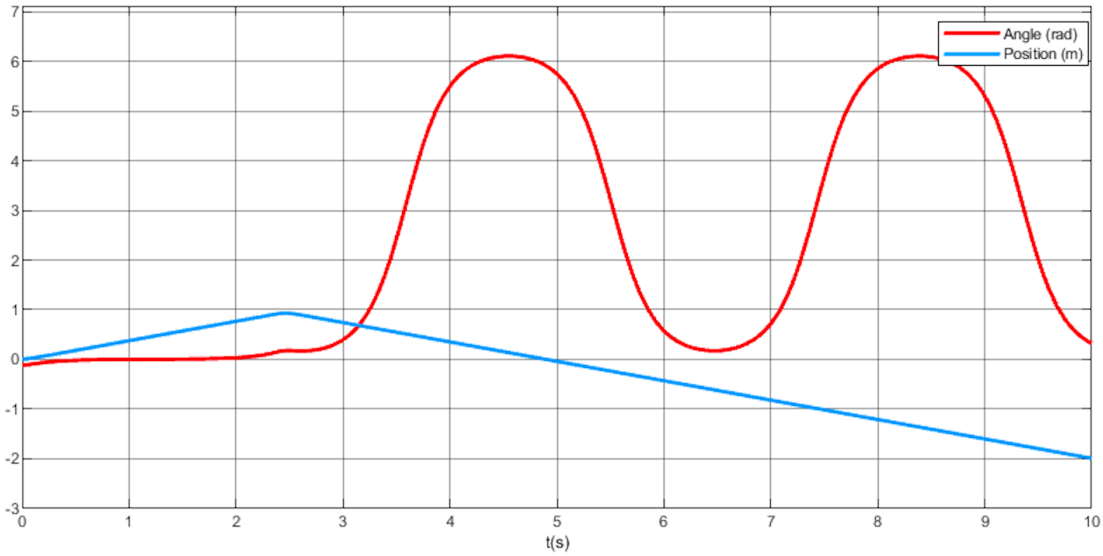


Figure 2.32: $x1$ and $x3$ in the pole placement simulation with poles 2.133

As changing the dominant pole has resulted into making the system unstable the second dominant pole will be increased and the dominant pole will be restored to its original value. The rise in the value of the second dominant pole will be a value big enough to produce a change and small enough to maintain the system stable.

$$(s + 0.63) \cdot (s + 4.2) \cdot (s + 4.05) \cdot (s + 18.3) \tag{2.134}$$

Also in figure 2.33 can be observed that the system is unstable due to the small increment that can be seen at almost 10 s. An alternative is to increase the less dominant pole or return the second dominant pole to its original value and increase the third dominant pole. It has been checked through simulations that the increase in the value of the third dominant pole until 4.5 makes the system unstable; thus, the only alternative is to modify the less dominant pole. This rise in the less dominant pole will increment the values of the vector K of the pole placement controller, due to this new K vector errors will be enlarged and as a corrective measure the system will act against these errors. The random value selected is

25 causing the new set of poles.

$$(s + 0.63) \cdot (s + 4.2) \cdot (s + 4.05) \cdot (s + 25.8) \tag{2.135}$$

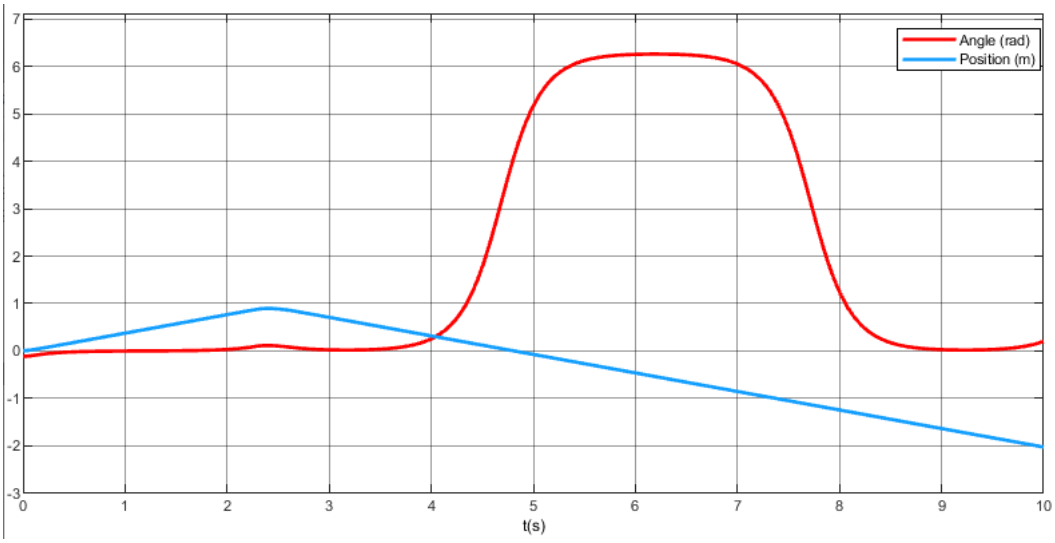


Figure 2.33: $x1$ and $x3$ in the pole placement simulation with poles 2.134

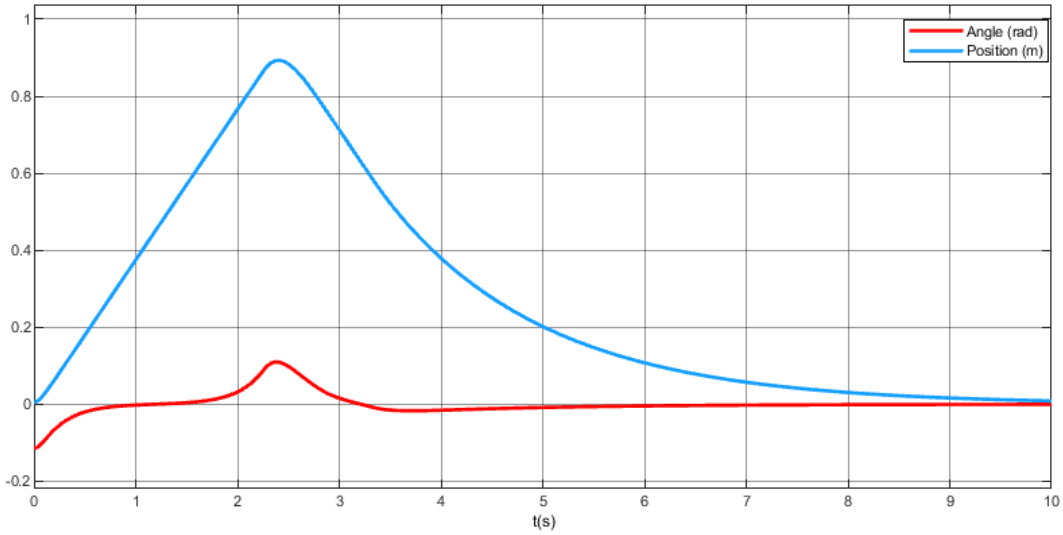


Figure 2.34: $x1$ and $x3$ in the pole placement simulation with poles 2.135

After obtaining a system that is stable with the poles 2.135, it has been simulated several set of poles with the less dominant pole increased, and the results where almost no noticeable. Thus, poles 2.135 are going to be selected as if they were the final poles of the pole placement controller.

2.6.3 Minimum-order observer

As the real robot has a gyroscope and one encoder in each motor, the device is only able to give the angular velocity of the body(x_2) and the position(x_3); hence, an observer has to be implemented to obtain x_1 and x_4, which will be used to generate the input w. This need for x_1 and x_4 is displayed in figure 2.24, where the full state vector is multiplied by the $Klqr$ vector to obtain w. It has to be specified that due to the lack of time, the minimum-order observer is going to be simulated with the pole placement controller while the LQR controller will incorporate the Kalman filter. However, due to the poles of the pole placement controller make the system behave as the poles in the LQR method the difference of the most suitable controller with the observer will depend mainly on the observer selected. In the following image it is displayed the minimum-order observer with the non-linear model.

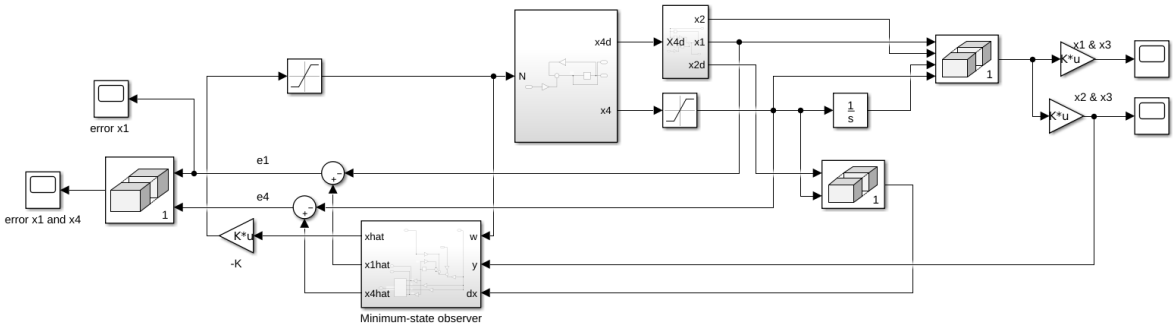


Figure 2.35: Non-linear model with the minimum-state observer

To design the minimum-order observer, it is necessary to decide 2 poles which will reduce the observation error. "As a general rule, however, the observer poles must be two to five times faster than the controller poles to make sure the observation error (estimation error) converges to zero quickly"[10]. Taking what it has been mentioned previously, several simulations are going to be performed using the poles of the controller multiplied by numbers comprehended in the interval [2,5] to obtain the poles of the observer. The first simulation poles have been selected multiplying the values of the second and the third dominant poles per 5. However, either the most dominant pole or the least dominant pole could have been selected to obtain the poles to simulate. Nevertheless, the behavior of the controller is mostly imposed by the most dominant pole; hence, using the least dominant pole would have given a wrong approximation of the observer pole.

Thanks to the MATLAB command $place(Abb, Aab, poles)$, the K gain of the minimum-state observer can be found for the following poles.

First simulation poles

$$(s + 19.92) \cdot (s + 20.25) \tag{2.136}$$

Figure 2.36 displays a system that is stable. Nevertheless, the angle increases until 6 radiant

which is equivalent to 344°, this rise is due to the bad approximation of variables x_1 and x_4. Moreover, this is impossible because the robot is only able to rotate an angle comprehended between the interval $\pm 90°$and reaching 90°degrees or $-90°$would mean that the robot would be on the floor, in other words, the minimum observer using poles 2.136 is not possible. Hence, another poles will be selected to see if the observer is able to estimate variables x_1 and x_4 with less errors.

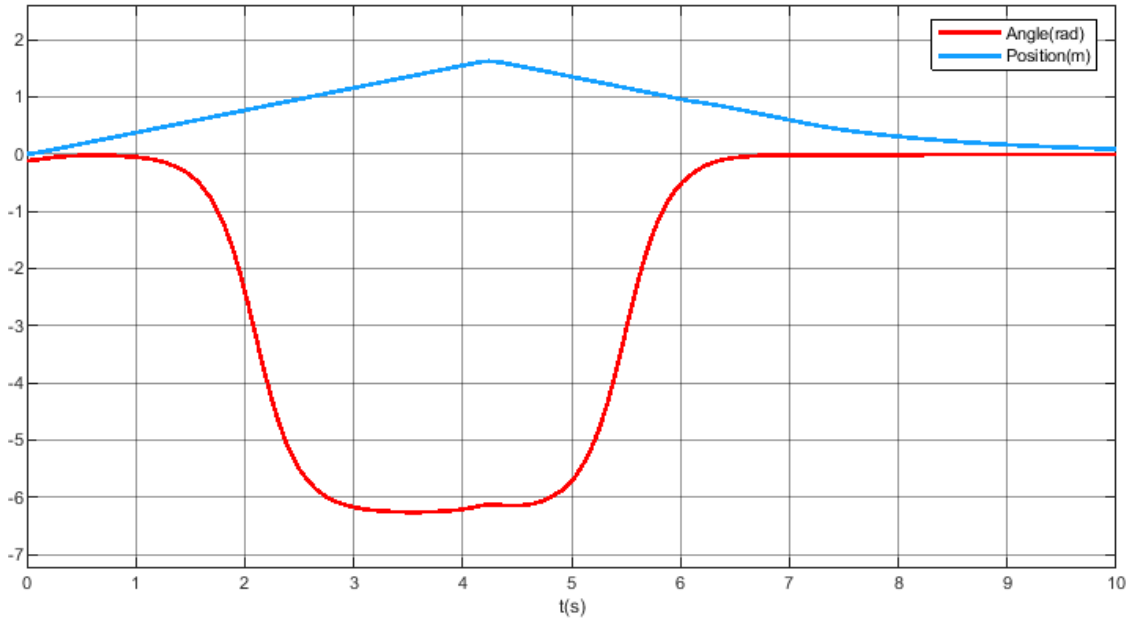


Figure 2.36: $x1$ and $x3$ in the pole placement simulation with poles 2.136

This increment in the angle can be provoked by the poor approximation of the variable x_1. Therefore, the second and the third dominant poles are going to be multiplied by 3 in order to decrease the velocity in which x_1 and x_4 are estimated and see it there is any change in the values of x_1 and x_3.

Second simulation poles

$$(s + 11.94) \cdot (s + 12.15) \tag{2.137}$$

After several simulations with poles whose values are 4 and 2 times the values of the second and the third dominant poles, it is been obtained behaviours that do not change too much with respect to figures 2.36 and 2.37. However, values bigger than 5 times the values of the second and the third dominant poles have not been tested.

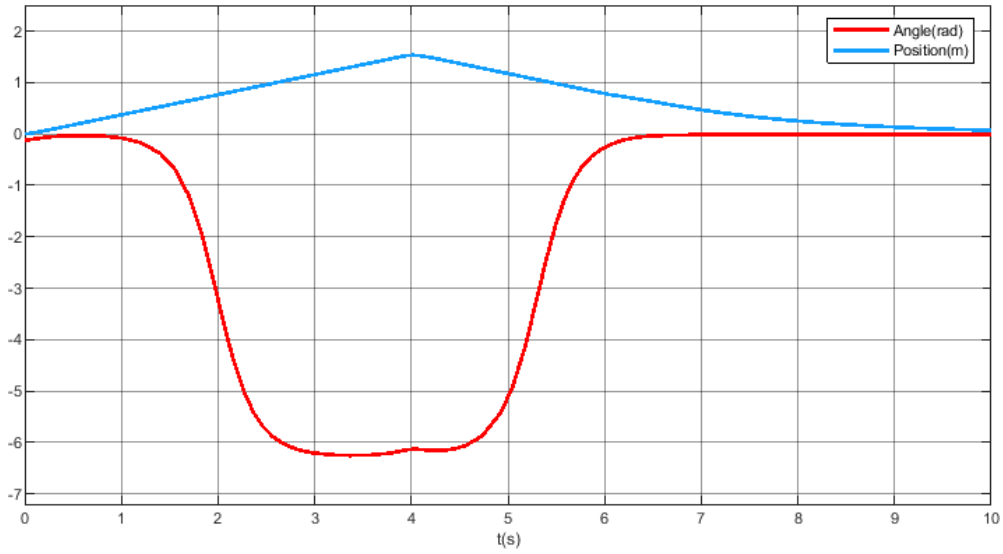


Figure 2.37: $x1$ and $x3$ in the pole placement simulation with poles 2.137

Third simulation poles

$$(s + 23.88) \cdot (s + 36.45) \tag{2.138}$$

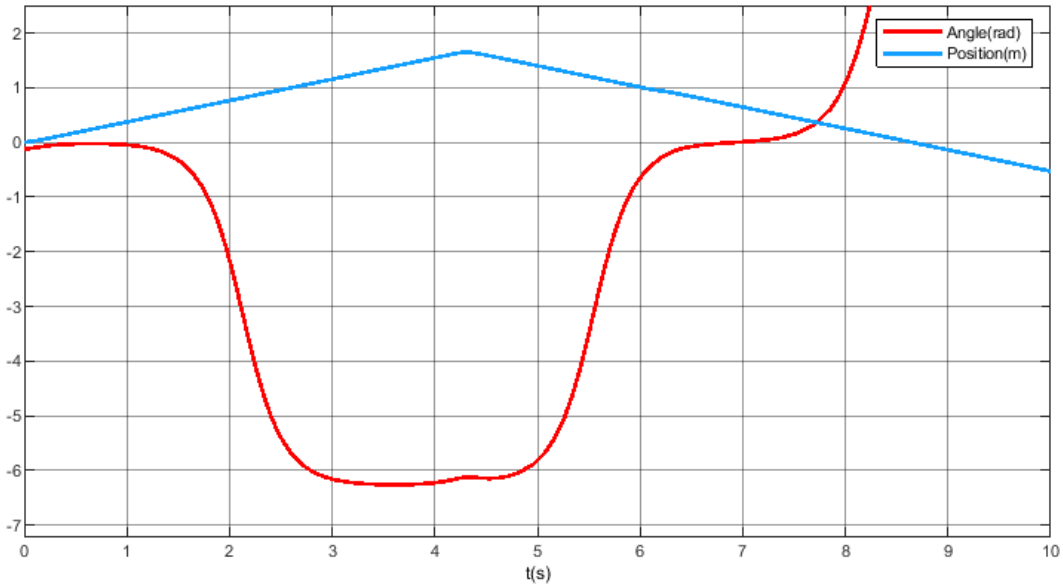


Figure 2.38: $x1$ and $x3$ in the pole placement simulation with poles 2.138

Taking into account the simulations performed, it is not possible to implement a minimum-order observer due to the unsteady behavior of the system or the high value of the angle during simulations.

2.6.4 Kalman filter

The whole model which integrates the Kalman filter is displayed in figure 2.39, commonly known as Linear Quadratic Gaussian controller(LQG), which is constituted by a LQR controller and a Kalman filter.

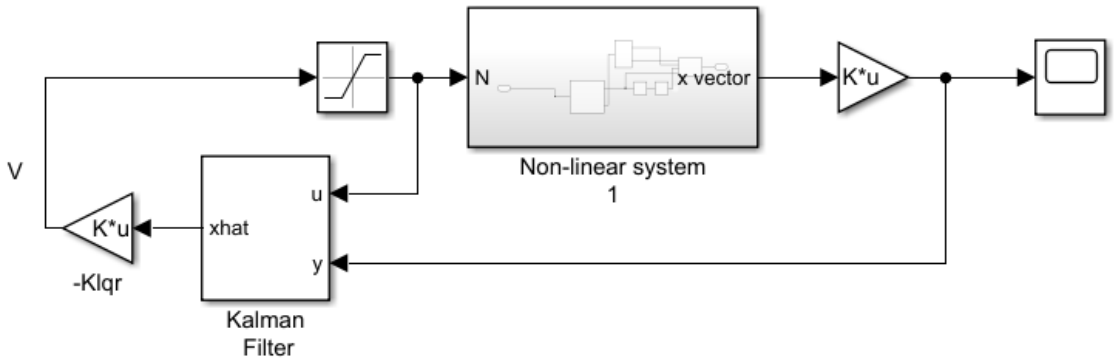


Figure 2.39: Non-linear model with Kalman filter and the LQR controller

In order to design the Kalman filter using the Kalman filter block in Simulink, it is necessary to find the covariance matrix of the sensor's noise Ns_y and the covariance matrix of the process' noise Ns_x. These covariances have to be determined through several proves, but first, it is necessary to know the dimensions of covariance matrixes Ns_y and Ns_x.

The state-space affected by noises is represented in the following way, where $ns_x[k]$ is a vector of 4x1 and the $ns_y[k]$ is a 2x1 vector.

$$x[k+1] = A \cdot x[k] + B \cdot w[k] + ns_x[k]$$
$$y[k] = C \cdot x[k] + D \cdot w[k] + ns_y[k]$$

Once the dimensions of $ns_x[k]$ and $ns_y[k]$ have been determined, the size of Ns_y and Ns_x can be found using operations 2.113 and 2.113, which gives the following dimensions using a discrete model during an interval of time. These matrixes are diagonal because the different noises are stochastic processes, which means that $ns_i[k_1]$ is completely independent from the measure $ns_i[k_2]$; hence, due to the independence, all terms that are not in the diagonal are 0 when doing the statistical hope.

$$COV(ns_x \cdot ns_x^T) = E\begin{bmatrix} ns_{x_1}^2 & & & \\ & ns_{x_2}^2 & & \\ & & ns_{x_3}^2 & \\ & & & ns_{x_4}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{x_1}^2 & & & \\ & \sigma_{x_2}^2 & & \\ & & \sigma_{x_3}^2 & \\ & & & \sigma_{x_4}^2 \end{bmatrix} = Ns_x \quad (2.139)$$

$$E\begin{bmatrix} ns_{y_1}^2 & \\ & ns_{y_2}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{y_1}^2 & \\ & \sigma_{y_2}^2 \end{bmatrix} = Ns_y \tag{2.140}$$

Sensors noise covariance

Although the noises of the different sensors are stochastic processes, the covariance of the sensor's noise(Ns_y) will be obtained through the following way. 2 experiments will be performed, in one experiment the gyroscope is going to be placed at the same position to measure only the noise. In the other experiment, the incremental encoder of a motors is going to measure the angle while the motor has a constant velocity; however, due to the encoder is an incremental encoder, the previous angle is going to be subtracted from the current angle to obtain only the variations in the angle caused by noises. With the experiments mentioned, data is going to be recorded, which will be a vector of measures for each experiment performed. Through using the vectors, the variances will be calculated and will be used to estimate the variance of each sensor($\sigma_{y_1}^2$ and $\sigma_{y_2}^2$) with the arithmetic mean(expression 2.141)

$$\sigma_{y_i}^2 = (\sum_{n=1}^{5} \sigma_n^2)/5 \tag{2.141}$$

$$\begin{aligned} \sigma_{y_1}^2 &= 0.1535 \\ \sigma_{y_2}^2 &= 1.803 \end{aligned} \tag{2.142}$$

$$Ns_y = \begin{bmatrix} 0.1535 & \\ & 1.803 \end{bmatrix} \tag{2.143}$$

Process noise covariance

On the other hand, the matrix Ns_x will also be found through experimentation. The experiment will consist in recording the data obtained form different sensors(gyroscope and encoders) when the robot is left at x_1=5°, x_2=0 rad/s, x_3=0m and x_4=0 m/s in open loop.

The first step to calculate Ns_x will consist on calculating the matrix A for the discrete mode, once determined the matrix A the term $A \cdot x[k] + B \cdot w[k]$ will be subtracted from $x[k+1]$ to obtain $ns_x[k]$ which will form a vector of 4 per i where i is the number of times that the state variables has been measured minus 1. In our scenario, no voltage will be applied to

motors; therefore, the term $B \cdot w[k]$ will be 0. To calculate the matrix A an assumption will be made, due to the small step time of the sensors to measure the angular velocity and the position, the trajectory between a measure taken at $k+1$ and a measure taken at k will be considered a non accelerated rectilinear trajectory. With this hypothesis the matrix A takes the following values where Δt is the step time of the sensors

$$A = \begin{bmatrix} 1 & \Delta t & & \\ & 1 & & \\ & & 1 & \Delta t \\ & & & 1 \end{bmatrix} \tag{2.144}$$

Being defined the matrix A it is possible to calculate the vector $ns_x[K]$ and using each row of this vector the variance $\sigma^2_{x_1}$, $\sigma^2_{x_2}$, $\sigma^2_{x_3}$ and $\sigma^2_{x_4}$ will be estimated.

$$\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \\ x_3[k+1] \\ x_4[k+1] \end{bmatrix} - \begin{bmatrix} 1 & \Delta t & & \\ & 1 & & \\ & & 1 & \Delta t \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \end{bmatrix} = \begin{bmatrix} ns_1[k] \\ ns_2[k] \\ ns_3[k] \\ ns_4[k] \end{bmatrix} \tag{2.145}$$

Taking into consideration the set of equations that form the set 2.145, and doing the experiment mentioned with the robot, the variances $\sigma^2_{x_1}$, $\sigma^2_{x_2}$, $\sigma^2_{x_3}$ and $\sigma^2_{x_4}$ are the following ones.

$$\begin{aligned} \sigma^2_{x_1} &= 0.1046 \\ \sigma^2_{x_2} &= 77.48 \\ \sigma^2_{x_3} &= 0 \\ \sigma^2_{x_4} &= 0 \end{aligned} \tag{2.146}$$

Although $\sigma^2_{x_3}$ and $\sigma^2_{x_4}$ are 0, it is impossible because of equation 2.7. If it solved for $\dddot{x}_w$(Equation 2.147) it can be seen that the acceleration of the bar has an influence over the acceleration of the wheel; thus, values $\sigma^2_{x_3}$ and $\sigma^2_{x_4} = 0$ can not be completely 0, to solve this a random variance has been selected for $\sigma^2_{x_3}$ and $\sigma^2_{x_4}$ taking into account that the variance values have to be smaller in comparison with variances $\sigma^2_{x_1}$ and $\sigma^2_{x_2}$. As $\sigma^2_{x_1}$ is the smallest between $\sigma^2_{x_1}$ and $\sigma^2_{x_2}$, the variance of $\sigma^2_{x_3}$ and $\sigma^2_{x_4}$ is 100 times smaller than $\sigma^2_{x_1}$.

$$\dddot{x}_w = \ddot{x}_b + l \cdot \sin(\theta) \cdot \dot{\theta}^2 - l \cdot \cos(\theta) \cdot \ddot{\theta} \tag{2.147}$$

$$Ns_x = \begin{bmatrix} 0.1046 & & & \\ & 77.48 & & \\ & & 0.001 & \\ & & & 0.001 \end{bmatrix} \tag{2.148}$$

Using matrix 2.148 and matrix 2.143 in the kalman filter block of Simulink. The following output(x_1 and x_3) for the system in figure 2.39 is presented.

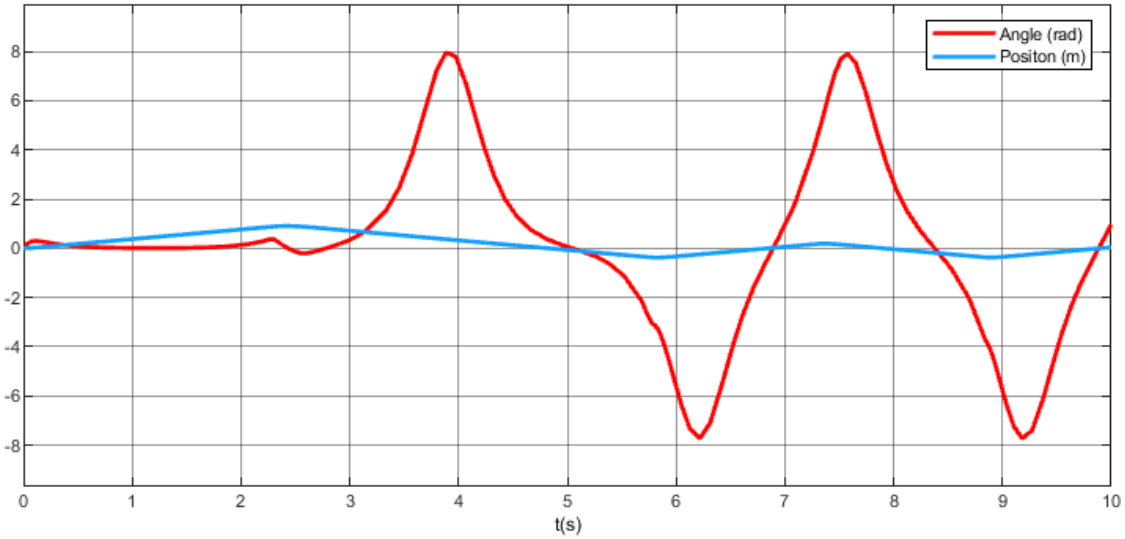


Figure 2.40: Kalman filter response

Analyzing figure 2.40 it can be seen that the angle θ increases until $|8|$ radiant which is completely unacceptable because in this situation the robot would be laying on the ground and this position the robot would not be able to return to the 0 position. This situation is similar to the situation explored in figure 2.36, where the robot also exceeded its angular position until reaching the $\pm 90°$.

2.7 Decision

Due to the designed controllers are not stable, tests in the real robot have not been performed. On the other hand, it has been tested that it is less time consuming using the LQR method than the pole placement method owing to the fact that the LQR method finds the poles of the system with optimization methods. Regarding the designed observers, non of them were able to estimate the state variables x_1 and x_4 precisely enough, indeed, the estimation was extremely poor that the system became unstable in the Kalman filter and physically impossible to implement in the real robot in the case of the minimum order observer. Hence,

a decision to select the best controller can not be effected due to the impossibility to view the effects of they utilization on the real robot.

3 Future developments

Regarding the controllers, it has been obtained that it is not feasible to control the robot. Hence, one future development would be modifying the physical structure such as setting the EV3 block closer to the motors to decrement the inertia of the current robot, with modification over the inertia, motors will have to exert less torque to correct the position which in turns, it could cause a stable behavior of the robot.

Moreover, the robot has been designed using the EV3 microcontroller, something that it is not really necessary due to the huge number of inexpensive controllers in comparison with the price of the EV3. One future alternative is using the microcontrollers of STM32 or the famous Arduino boards in order to reduce the price of the project. Moreover, with the alternative microcontrollers mentioned, the user has the freedom to create more projects due to the tremendous number of devices that can be attached to Arduino boards or the STM32 microcontrollers. Examples of these devices would be: servos, GPS devices, cameras and LCD screens.

Another future development would be modeling the robot to be controlled by remote control instead of going forward and backward with no user control. Furthermore, a model to swivel the robot could be developed to add more capacities to the two-wheeled robot.

Finally, another possible development could be the modification of the model by using adaptative control methods or robust control methods with the aim of stabilizing the robot when different weights are put in the top of the structure, for instance, putting different glasses of water without the robot's balance loss.

4 Recommendations

In this bachelor thesis there are 6 fields which has been used in some way or another to achieve the aim purposed. Hence, it is recommended to have knowledge in the following fields.

Algebra

The state-state representation uses matrixes to render the robot's behavior. Moreover, the controllers used use matrixes to stabilize the robot. Therefore, a knowledge in algebra is needed to understand the mathematical operations.

Mechanics

Mechanics have been used to obtain the robot's equations using Newton's laws which are displayed in section 2.2 Robot physics. Hence, experience in forces' diagrams and the application of Newton's laws are required.

Programming

To design the controllers, the minimum order observers and the Kalman filter, MATLAB and Simulink are used alongside Python programing lenguage. MATLAB and Simulink have been used to plot the graphics, design the block diagrams of the controllers, the observer, and the Kalman filter. In regard to Python, the library Sympy has been used to calculate symbolic mathematical operations. Thus, MATLAB, Simulink and Python are highly recomended due to their extensive use in this bachelor thesis.

Control theory

Control theory is the main topic of this thesis. Hence, knowledge in stability, block diagrams, state-space representation, LQR controller, pole placement, the Kalman filter and observers are needed to establish the robot's desired behavior.

Electronics

Knowledge in electronics is needed in section 2.2 Robot physics, concretely, the Kirchhoff's laws are used to acquire the current of the motors in terms of the velocity of the wheel and the velocity of the bar.

Statistics

Concerning statistics, this field was used during the learning process of the Kalman filter and during the design, concretely, concepts about the normal distribution were highly used.

5 Budget

During the realization of the project, several services and materials were required. Thanks to the automatic department of the ESEIAAT it was not necessary to purchase the EV3; however, the cost of this device will be contemplated as if it was purchased by the author of this thesis. In addition, the hours spent with teachers will be taken into account as if these hours would have been technical advice. The price of the MATLAB student suit licence will also be included as it is the main virtual non free platform where the project has been developed..

Category	**Item**	**Quantity**	**Cost per hour (€/h)**	**Total cost (€)**
Software	MATLAB license	1		69.0
Material	EV3 educational pack	1		491
Services	Technical advice	35h	40	1400
Project	Project charter	**30h**	30	900
	Information search	**90h**	30	2700
	Write work	**200h**	30	6000
	MATLAB and Simulink	**40h**	30	1200
	Design experiments	**10h**	30	300
	Experimentation	**20h**	30	600
	Build script(Sympy)	**15h**	30	450
	Learn library Sympy	**10h**	30	300
	Learn Python	**5h**	30	150
TOTAL		**420h**		**14560**

Table 5.1: Budget table

6 Conclusions

To achieve the purpose of this thesis, building a two wheel robot fully balanced using control theory methods, several controllers were designed using a pole placement controller, a LQR, minimu-order observer and a Kalman filter. Although the model 2.50 was controlable and its use could have made the behavior of the controllers stable, a non linear system was elaborated to make the simulations as real as using the physical robot. However, thanks to Simulink it was possible to observe that the observers designed which incorporated the non linear model were not feasible due to the physical limitations in which the robot works or due to the non steady behaviour presented by the systems. Thus, due to this impossibility to use the designed systems in the robot, physical verifications of the controllers' performance have not been committed.

Nevertheless, it has been tested during the design of the controllers that using a LQR controller is faster in the sense that it is faster to find a system that is stable in comparison with the pole placement method, where the user has to try several set of poles until there is a range of values in which the system is stable. The same thing happens when the LQR is used, values of the matrix Q and R have to be found to make the system stable but with the imposition of the value of the identity matrix in matrixes Q and R it is faster to find the right values which make the system stable. In other cases, depending on the stability of the system it is better to directly impose the behavior by selecting directly the poles in the pole placement method rather than having an stable system and try values of matrixes Q and R. Therefore, the best method depends on the selection range of the poles which makes the system stable, if the range of poles which make the system stable is small the LQR method will be the appropriate since this method selects the optimal poles for each case. On the other hand, if the selection range of the poles which make the system stable is large, it is better to directly impose the poles which will make the system behave as it is desired. In this bachelor thesis the model had a small range to select the poles which made the system stable; thus, it has been faster designing a LQR controller rather than a pole placement controller.

In a similar way, the decision of the kind of observer depends on the approximation errors of the unknown state variables as well as the stability range of the system to select different poles. Hence, if the scope to select the poles which make the system stable is large, it is suitable to use a minimum order observer and select the desired poles to impose in the observer. Nevertheless, if the scope to select the poles which make the system stable is limited, the Kalman filter will save time to the user as this observer uses optimization methods to approximate the unknown state variables.

Bibliography

[1] Baptist. *Steam Engine*. 2014. URL: https://www.livescience.com/44186-who-invented-the-steam-engine.html (visited on 02/17/2019).

[2] *The why and how of landing rockets — SpaceX*. URL: https://www.spacex.com/news/2015/06/24/why-and-how-landing-rockets (visited on 03/22/2019).

[3] *How SpaceX Falcon Heavy undercuts its competition three-fold*. URL: https://www.teslarati.com/how-spacex-falcon-heavy-costs-undercuts-competition/ (visited on 03/22/2019).

[4] *Hoverboard Swagtron*. URL: http://www.stickpng.com/es/img/objetos/aeropatin/hoverboard-swagtron-autoequilibro (visited on 02/17/2019).

[5] *LEGO® MINDSTORMS® Education EV3 Core Set*. URL: https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-core-set-/5003400 (visited on 02/18/2019).

[6] *iBot 4000 - ROBOTS: Your Guide to the World of Robotics*. URL: https://robots.ieee.org/robots/ibot/ (visited on 02/27/2019).

[7] *Segway - Wikipedia, la enciclopedia libre*. URL: https://es.wikipedia.org/wiki/Segway (visited on 02/27/2019).

[8] *Segway - La historia de Segway*. URL: http://es-es.segway.com/the-story-of-segway (visited on 02/27/2019).

[9] *El Segway Pt, Rueda, Selfbalancing Scooter imagen png - imagen transparente descarga gratuita*. URL: https://www.freepng.es/png-6tc0ry/ (visited on 03/22/2019).

[10] Katsuhiko Ogata. *Modern Control Engineering*. Vol. 17. 2010, p. 912. ISBN: 9780136156734. DOI: 10.1109/TAC.1972.1100013. URL: http://www.pearsonhighered.com/educator/product/Modern-Control-Engineering/9780136156734.page.

[11] Brian Bonafilia et al. "Self-balancing two-wheeled robot". In: (2013).

[12] *ENGR487 Lecture17 Min/Max and Lagrange Mulitplier - YouTube*. URL: https://www.youtube.com/watch?v=%7B%5C_%7DIguxBjdBKc%7B%5C&%7Dlist=PLJzZfbLAMTelhoLd4hFW3mUIAWPvFgTLv%7B%5C&%7Dindex=6 (visited on 04/22/2019).

[13] *ENGR487 Lecture18 Linear Quadratic Optimal Control (Part I) - YouTube*. URL: https://www.youtube.com/watch?v=fV5JZ2yOujk%7B%5C&%7Dlist=PLJzZfbLAMTelhoLd4hFW3mUIAWPvFgTLv%7B%5C&%7Dindex=5 (visited on 04/22/2019).

[14] *ENGR487 Lecture19 Linear Quadratic Optimal Control (Part II) - YouTube*. URL: https://www.youtube.com/watch?v=NAk2cWANPK8%7B%5C&%7Dlist=PLJzZfbLAMTelhoLd4hFW3mUIAWPvFgTLv%7B%5C&%7Dindex=4 (visited on 04/22/2019).

[15] *LQR Method (Dr. Jake Abbott, University of Utah) - YouTube*. URL: https://www.youtube.com/watch?v=St5L-ekOKGA%7B%5C&%7Dt=211s (visited on 04/22/2019).

[16] *ENGR487 Lecture21 Kalman Filter - YouTube*. URL: https://www.youtube.com/watch?v=vOZYcSWGJBY%7B%5C&%7Dlist=PLJzZfbLAMTelhoLd4hFW3mUIAWPvFgTLv%7B%5C&%7Dindex=2 (visited on 04/28/2019).

[17] Draguna L. Lewis, Frank; L. Syrmos Vassilis; Vrabie. *Optimal Control*. ISBN: 9780470633496.

[18] *Kalman filter design, Kalman estimator - MATLAB kalman - MathWorks España*. URL: https://es.mathworks.com/help/control/ref/kalman.html (visited on 04/20/2019).

[19] *Understanding Kalman Filters, Part 6: How to use Kalman Filters in Simulink Video - MATLAB & Simulink*. URL: https://es.mathworks.com/videos/understanding-kalman-filters-part-6-how-to-use-kalman-filters-in-simulink--1508782499422.html (visited on 05/03/2019).

[20] *LEGO 9V Technic Motors compared characteristics*. URL: http://www.philohome.com/motors/motorcomp.htm (visited on 05/25/2019).

Appendix

B Declaration of honor

I declare that,

The work in this Master Thesis / Degree Thesis (*choose one*) is completely my own work, no part of this Master Thesis / Degree Thesis (*choose one*) is taken from other people's work without giving them credit, all references have been clearly cited,

I'm authorized to make use of the company's / research group (*choose one*) related information I'm providing in this document (*select when it applies*).

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by *The Universitat Politècnica de Catalunya - BarcelonaTECH*.

Sergio Prieto Molina
Student Name

10/6/2019
Date

Title of the Thesis: Study of different control methods applied to a self-balancing robot