



POLITECNICO
MILANO 1863

Maglev Control

Ahmed El Hanafi - 10580177

Francesco Serdoz - 10597444

Luca Fiorendi – 10608342

Tancredi Terranova - 10624590

Instructor:

Prof. Hamid Reza Karimi

Automation and Control Laboratory

A.Y. 2022-2023

Contents

1 System Modelling and Analysis	4
1.1 Physical Model and State Space Representation	4
1.2 System Linearization	7
1.3 Linearized System Analysis	9
2 Sensors Characterization	14
2.1 Current Sensor	14
2.2 Position Sensor	15
2.3 Noise Characterization	17
3 Parameters Identification	20
3.1 R_{tot} and L_c Identification	20
3.1.1 R_{tot} Identification with Steady State Measurements Analysis	20
3.1.2 L_c Identification with Transient Measurements Analysis	23
3.1.3 Temperature Effect on R_{tot} Identification	26
3.1.4 R_{tot} and L_c Identification with Real Model Matching	27
3.1.5 R_{tot} and L_c Identification with System Identification Toolbox	29
3.1.6 R_{tot} and L_c Validation with an Extended Kalman Filter	30
3.1.7 R_{tot} and L_c Identification as a Solution of a Minimization Problem	32
3.2 Km identification	34
4 Controller design of Magnetic Levitation System (MAGLEV)	37
4.1 Cascade control	37
4.1.1 Internal Current loop synthesis	38
4.1.2 External Position loop synthesis	44
4.2 Variable Structure Control (VSC)/ Sliding Mode Control	61
4.3 Feedback Linearization	63
4.4 Extended Kalman Filter	68
5 Conclusions	72

Introduction

In this first introductory section, the motivations behind the choice of the project, the importance of magnetic levitation and its practical applications, a detailed description of the Quanser MAGLEV system, as well as a summary of the expected results, will be discussed in detail.

The project to levitate the metal sphere using the Quanser MAGLEV system has many motivations. Firstly, our project aims to demonstrate the potential of Maglev technology and control applications. Magnetic levitation is indeed a technology that can be used in many applications, including public transportation, logistics, manufacturing, and scientific research. MEGLEV technology offers numerous advantages over traditional transportation systems, including higher efficiency, lower maintenance costs, lower emissions, and higher speeds. Secondly, the project aims to improve our knowledge in the field of control of an electrical and mechanical system. Finally, our project helped us to develop a deeper understanding of the physical principles of magnetic levitation and its non-linear behavior.

The MAGLEV system provided by Quanser is an electromagnetic suspension system that acts on a solid steel ball with a diameter of 2.54 cm. It consists mainly of an electromagnet in the upper part of the device, which can lift the steel sphere from its base and hold it in free space. Two system variables are measured directly on the MAGLEV device and are available for feedback. These are the coil current and the distance of the sphere from the electromagnetic surface. A more detailed description can be found in the Magnetic Levitation User Manual recalled in the reference[1].

Due to the inherent non-linearities in the system, we encountered many challenges that we needed to overcome during the control design. These challenges, for instance, are the noise in data measurement and instability of the equilibrium point, the velocity measurement, parameter uncertainties and the computation of the derivative in the proposed controls.

Various control strategies are presented in the literature, and in general there are some experimental challenges to overcome, because of the fact that generally only some ball position and electric current are easy to measure. Some approaches assume Proportional-Integral-Derivative (PID) control, which requires precise knowledge of the parameters of the system, and state feedback using a linearized model around an equilibrium point, such as the work of Ahmad and Javaid[5], where an observer is used to reconstruct the velocity and compare it with nonlinear state feedback. Other work, such as that of Morales[4], proposes a Generalized-Proportional-Integral (GPI) controller together with an online parameter identification. However,

the electrical dynamics are neglected and simplified to an algebraic relationship.

Our project aims to achieve the stable levitation of the iron sphere in the magnetic field generated by the coils. It also aims to gain a detailed understanding of the control principles and techniques used to hold the sphere in position. The experiment is provided by Quanser and it is interfaced to an Arduino board called PoliArd, allowing to program it through the Matlab and Simulink environment.

1 System Modelling and Analysis

In this chapter, we first introduce the physical model proposed by Quanser [1] and, after the derivation of its non linear state space representation, we find the resultant linearized model. Furthermore, we analyse the obtained system and we discuss about its properties.

1.1 Physical Model and State Space Representation

In this section, the physical system proposed by Quanser [1] is presented. Figure 1 depicts a schematic of the MAGLEV plant that was used in this experiment. The system is composed by an electromagnetic circuit with a resistance R_c and an inductance L_c , a current sensor represented by the resistance R_s and a photo-sensitive position sensor mounted inside the ball's pedestal.

Moving towards a system modelling, it is then possible to identify two systems:

1. An electrical system, made by a steel core and a copper coil solenoid that generates the magnetic attraction force.
2. A mechanical system, represented by the iron ball's position, velocity and acceleration.

Notice that the magnetic steel core represents the center of the reference frame, with the positive X-axis directed downwards. As it will be further explained in section 2 this implies that, when the system is not powered, the iron sphere is at the maximum distance with respect to the origin of the reference system. Hence, we can then introduce the ball's travel parameter T_b as the overall distance minus two times the sphere's radius.

Note finally that, even if the ball can move and oscillate along X, Y and Z direction, in our system we can only control the vertical direction.

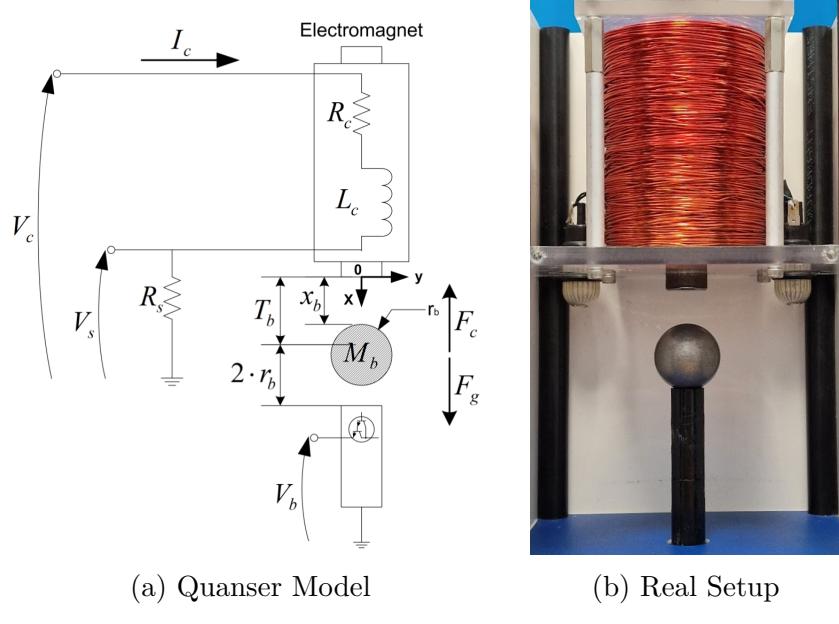


Figure 1: Overview of the system

Starting from the electrical model, with a KVL analysis of the RL circuit we can obtain:

$$V_c(t) = (R_c(t) + R_s(t))I_c(t) + L_c(t) \left(\frac{d}{dt} I_c(t) \right) \quad (1.1)$$

In order to find the model equation of the mechanical system, it is necessary to perform a balance of forces on the ball's center of gravity. The gravitational and magnetic force are respectively:

$$F_g = M_b g \quad (1.2)$$

$$F_c = -\frac{1}{2} \frac{K_m I_c^2}{x_b^2} \quad (1.3)$$

where M_b is the mass of the iron ball, g is the gravity acceleration, K_m is the electromagnetic force constant, I_c is the current flowing into the coil and x_b is the ball's position measured from the reference system.

From the balance of forces we obtain:

$$F_{tot} = M_b a = F_c + F_g = -\frac{1}{2} \frac{K_m I_c^2}{x_b^2} + M_b g \quad (1.4)$$

where a is the vertical ball's acceleration.

Starting from the equations of the physical system, we can now refer to a state space representation, considering the current i_c , the ball's position x_b and its velocity \dot{x}_b as the three states:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} i_c \\ x_b \\ \dot{x}_b \end{bmatrix} \quad (1.5)$$

Furthermore, we can consider the voltage v_c as the system's measurable input, and the current i_c and the ball's position x_b as the two outputs (provided by the two available sensors):

$$\mathbf{u} = [v_c] \quad \mathbf{y} = \begin{bmatrix} i_c \\ x_b \end{bmatrix} \quad (1.6)$$

Finally, it's possible to introduce the state space system representing the physical model's dynamics:

$$\begin{cases} \dot{x}_1 = -\frac{R_{tot}}{L_c} x_1 + \frac{1}{L_c} u \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -\frac{K_m}{2M_b} \frac{x_1^2}{x_2^2} + g \\ y_1 = x_1 \\ y_2 = x_2 \end{cases} \quad (1.7)$$

where R_{tot} is defined as the sum of the coil's resistance R_c and the sensor's one R_s . We can then conclude that the model is **SIMO**, 3^{rd} order, time invariant (with the strong hypothesis that R_{tot} does not vary with the temperature) and non linear.

1.2 System Linearization

Because of the system's non linear behaviour, we then proceeded by linearizing it in the neighbourhood of an equilibrium position.

Hence, setting the state derivatives equal to 0 and a constant input fort the system:

$$\dot{x}_1 = 0 \quad \dot{x}_2 = 0 \quad \dot{x}_3 = 0 \quad u = \bar{u} \quad (1.8)$$

and considering:

$$\begin{cases} 0 = -\frac{R_{tot}}{L_c} \bar{x}_1 + \frac{1}{L_c} \bar{u} \\ 0 = \bar{x}_3 \\ 0 = -\frac{K_m}{2M_b} \frac{\bar{x}_1^2}{\bar{x}_2^2} + g \end{cases} \quad (1.9)$$

we were able to find the states in the following equilibrium conditions:

$$\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{R_{tot}} \bar{u} \\ \sqrt{\frac{K_m}{2gM_b}} \bar{x}_1 \\ 0 \end{bmatrix} \quad (1.10)$$

Notice that, since we want to stabilize the ball in a equilibrium position $\bar{x}_b = \bar{x}_2$, it's convenient to re-write all the equilibria function of the desired set point:

$$\bar{u} = R_{tot} \bar{x}_1 \quad \bar{x}_1 = \sqrt{\frac{2gM_b}{K_m}} \bar{x}_2 \quad (1.11)$$

Finally, we obtained the linearized model:

$$\begin{cases} \delta \dot{x}_1 = -\frac{R_{tot}}{L_c} \delta x_1 + \frac{1}{L_c} \delta u \\ \delta \dot{x}_2 = \delta x_3 \\ \delta \dot{x}_3 = -\frac{K_m}{M_b} \frac{\bar{x}_1}{\bar{x}_2^2} \delta x_1 + \frac{K_m}{M_b} \frac{\bar{x}_1^2}{\bar{x}_2^3} \delta x_2 \\ \delta y_1 = \delta x_1 \\ \delta y_2 = \delta x_2 \end{cases} \quad (1.12)$$

from which we can define the state space matrices of the linearized system:

$$A_{lin} = \begin{bmatrix} -\frac{R_{tot}}{L_c} & 0 & 0 \\ 0 & 0 & 1 \\ -\frac{K_m}{M_b} \frac{\bar{x}_1}{\bar{x}_2^2} & \frac{K_m}{M_b} \frac{\bar{x}_1^2}{\bar{x}_2^3} & 0 \end{bmatrix} \quad B_{lin} = \begin{bmatrix} -\frac{1}{L_c} \\ 0 \\ 0 \end{bmatrix} \quad (1.13)$$

$$C_{lin} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad D_{lin} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

1.3 Linearized System Analysis

Considering the state space matrices introduced in section 1.2, we can analyse the system's properties.

Looking at equation (1.14), we can first conclude that the system is fully observable, i.e. that all of its states can be known from the two outputs of the system, and thus fully controllable.

$$Obs = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{R_{tot}}{L_c} & 0 & 0 \\ 0 & 0 & 1 \\ (\frac{R_{tot}}{L_c})^2 & 0 & 0 \\ -\frac{K_m}{M_b} \frac{\bar{x}_1}{\bar{x}_2^2} & \frac{K_m}{M_b} \frac{\bar{x}_1^2}{\bar{x}_2^3} & 0 \end{bmatrix} \quad (1.14)$$

$$Contr = \begin{bmatrix} \frac{1}{L_c} & -\frac{R_{tot}}{L_c^2} & \frac{R_{tot}^2}{L_c^3} \\ 0 & 0 & -\frac{\bar{x}_1 K_m}{L_c M_b \bar{x}_2^2} \\ 0 & -\frac{\bar{x}_1 K_m}{L_c M_b \bar{x}_2^2} & \frac{\bar{x}_1 K_m R_{tot}}{L_c^2 M_b \bar{x}_2^2} \end{bmatrix}$$

In fact, setting $\bar{x}_1 \neq 0$ and $\bar{x}_2 \neq 0$, we can conclude that $rank(Obs) = rank(Contr) = 3$, that make the two matrices full rank.

We can now apply the Laplace transformation $C(sI - A)^{-1}B + D$ to the linearized system and obtain the two following transfer functions:

$$G_{v_c \rightarrow i_c} = \frac{\frac{1}{L_c}}{s + \frac{1}{L_c} R_{tot}} \quad (1.15)$$

$$G_{v_c \rightarrow x_b} = \frac{K_m \bar{x}_1 \bar{x}_2}{(-M_b \bar{x}_2^3 s^2 + K_m \bar{x}_1^2)(L_c s + R_{tot})} \quad (1.16)$$

We can now observe that it's possible to consider the output of the electrical system i_c as the input of the mechanical one. Hence, the relation between the input-output dynamics is described by a second order transfer function, reported as follows:

$$G_{i_c \rightarrow x_b} = \frac{-\frac{K_m \bar{x}_1}{M_b \bar{x}_2^2}}{s^2 - \frac{K_m \bar{x}_1^2}{M_b \bar{x}_2^3}} \quad (1.17)$$

In the following figure, it is possible to have a better understanding of the block scheme representing the three transfer functions introduced.

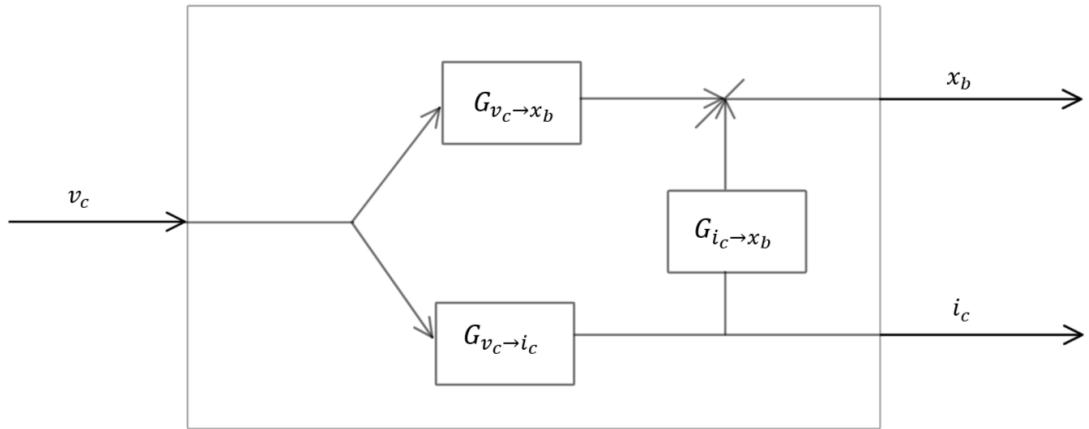


Figure 2: System's block scheme

Thus, we can analyse the system as two separated ones, characterized by the two transfer functions reported in (1.15) and (1.17). This separation will then be exploited in section 4 in order to develop a cascade control scheme, relying on the fact that the inner loop (that controls the current behaviour) has a faster dynamics with respect to the outer one, that is responsible of the control of the ball's position. In table 1 are reported the nominal parameters provided by Quanser, that are now used to conduct a preliminary analysis of the transfer function introduced.

Note that, despite we identified different parameters (see section 3), these considerations are valid also in our real system.

Quantity	Value	Unit
R_c	10	Ω
R_s	1	Ω
R_{tot}	11	Ω
L_c	412.5	mH
K_m	6.5308×10^{-5}	Nm ² /A ²

Table 1: Nominal system's parameters

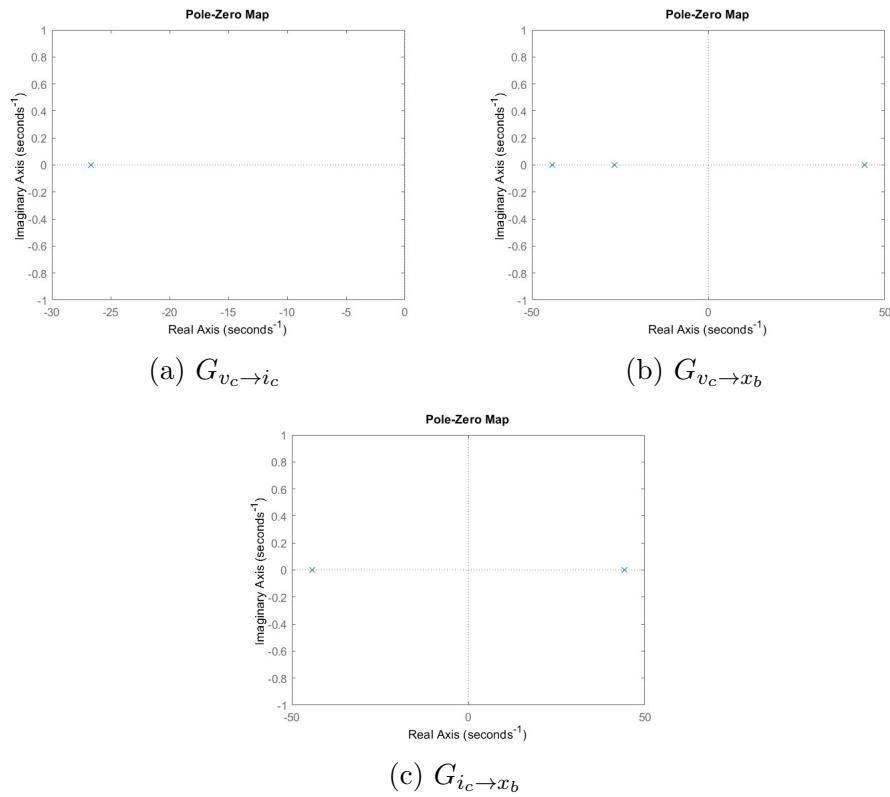


Figure 3: Poles of the three transfer functions of the system

First of all, it is important to discuss the stability of each of the three systems previously introduced. In order to perform this analysis, it's convenient to use the MATLAB function *pzmap*, that allows to plot all the zeros and poles of the considered transfer function. Note that (1.15), (1.16) and (1.17) are strictly proper transfer functions with no zeros at the numerator.

Starting from $G_{v_c \rightarrow i_c}$, it's possible to notice that it is a first order transfer function with an asymptotically stable pole in R_{tot}/L_c , that nominally is equal to 26.7 rad/s. Considering $G_{i_c \rightarrow x_b}$, it is a second order transfer function with two real poles in $\pm K_m \bar{x}_1^2 / M_b \bar{x}_2^3$, nominally in ± 44.3 rad/s. Hence, it is possible to conclude that the mechanical system is not asymptotically stable.

Finally, $G_{i_c \rightarrow x_b}$ is a third order system with the three poles of the electrical and mechanical subsystems. Thus, this system could be consider as the overall input output one, and so it is rather important in order to develop control strategies such as LQG, pole placement etc.

2 Sensors Characterization

In this chapter, we first introduce the two available sensors analysing their properties and explaining the preliminary steps we had to perform in order to use their reading properly. Finally, section 2.3 is devoted to the characterization of the two sensors' noises.

2.1 Current Sensor

As previously introduced in chapter 1, the system is equipped with a current sensor that is responsible of the voltage drop $v_s = R_s i_c$ on the R_s resistance. Even if the experiment is provided by Quanser, it is interfaced with the Arduino board PoliArd. Hence, the original configuration is no more present, and it wasn't possible to calibrate the sensor's offset and gain.

In order to check for a possible sensor's gain, we opened the Maglev's box and we directly measured the current through a digital multimeter with the help of the laboratory assistant.

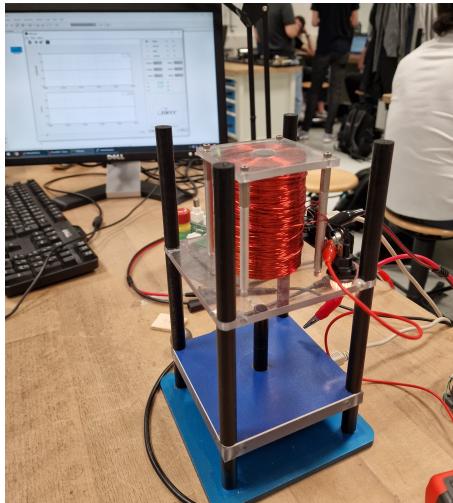


Figure 4: Multimeter setup

Furthermore, comparing the sensor's readings provided by the interface Poliscope with the ones of the multimeter, we concluded that the current sensor has approximately a gain of about 20%. In fact, from table 2 it is possible to notice how, at different voltage levels, the actual current value is higher than the one reported in the interface provided.

In sight of this constant gain, in every control scheme we had to multiply the current sensor measurement, i.e. the feedback, by 1.2 in order to have the right current error.

Voltage [V]	$I_{poliscope}$ [A]	$I_{multimeter}$ [A]
6	0.35	0.45
12	0.75	0.90
18	1.10	1.37
24	1.50	1.82

Table 2: Poliscope Vs Multimeter measurements

2.2 Position Sensor

A key element for closed loop systems design is the position sensor. In fact, its readings allow us to compare the actual ball's position with the desired one.

The implemented sensor is an optical light-dependent resistor one and it works such that, with the variation of the light intensity, it changes its physical properties (i.e. its electrical resistance).

An important drawback of these kind of devices, is the low repeatability in different light conditions. Hence, we had to deal with this problem by trying to maintain the same conditions for each experiment.

As previously introduced, a photo-conductive light sensor does not produce electricity but it modifies its resistance when subject to light. Thus, this device returns voltage measurements in levels, that are directly linked with the resistance variation. It is important to notice that, since our model is fully characterized in meters, an important step is to perform the conversion from levels to this reference unit.

In order to find this function, we placed several wooden pieces of a known thickness between the top of the ball and the steel core (see figure 5). Thus, powering the system with the maximum voltage (i.e. 24, V), we were able to keep the ball at different known positions and to collect several measurements.



Figure 5: Conversion curve identification procedure

We then made use of “Curve Fitting” toolbox in MATLAB in order to fit different polynomial functions in the collected points. Notice that, since the reference frame of the system is placed at the bottom of the steel core (with the x-axis pointing downwards), we assigned 0 mm to the highest value in levels.

Knowing that the curve characteristic of a photo-resistor has a negative exponential behaviour, we first considered an even curve of high order. However we realized that, as soon as the degree of the polynomial increases, the curve over-fits the data increasing the conversion error. Hence, we fitted a 2nd order polynomial such that it minimizes the distance with all the data set points. In fact, with the characteristic reported in Eq. 2.1, we were able to obtain a small SSE (i.e. the sum of squares due to error) and it means that the following fit will be more useful for prediction.

$$f(x) = -3.571 \times 10^{-5}x^2 + 0.035x + 3.162 \quad (2.1)$$

In figure 6 it is possible to appreciate the behaviour of the considered curve over the experimental data, with y indicating the conversion in mm and z indicating the real output of the sensor in levels.

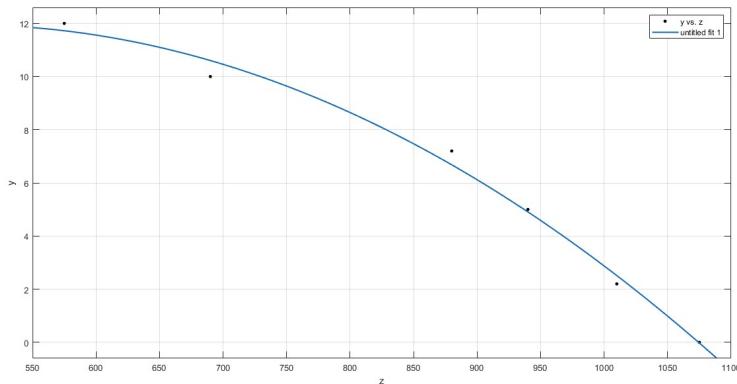


Figure 6: Polynomial second order curve

It is possible to appreciate that, as soon as the ball’s distance from the pedestal (i.e. from the sensor) increases, the resistance decreases leading to high values of voltages. Furthermore it is also important to notice that, as soon as the ball reaches the steel core, the sensor’s quality gets worse and it leads to a higher slope of the final part of the curve.

2.3 Noise Characterization

The sensors introduced in 2.1 and 2.2 are affected by noise. Performing a good characterization of this quantity it's crucial for accomplish the goal of this project. In fact, noisy measurements can affect the robustness and the stability of the designed controls.

Furthermore, by knowing the entity of this quantity, it is possible to implement filters (e.g. low pass filter or Kalman filter) to partially or completely eliminate it.

First of all, it's straightforward to find the mean values and the variances of the two noises by removing the DC component of the signals and analysing the remaining part with the MATLAB functions “mean” and “var”. In the following table, are reported these two parameters for each signal, where the position noise is analyzed in millimeters.

	Current Sensor	Position Sensor
Mean μ	2.9e-16 ≈ 0	1.4346e-15 ≈ 0
Variance σ^2	0.0016	0.027

Table 3: Mean values and variances of the two sensors

It is clear to notice that the two signals are characterized by zero mean and variance greater than zero. This means that we are dealing with Gaussian distributed noises, where:

$$v_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \quad v_x \sim \mathcal{N}(\mu_x, \sigma_x^2) \quad (2.2)$$

In figure 7 is reported the theoretical PDF of the Gaussian current noise.

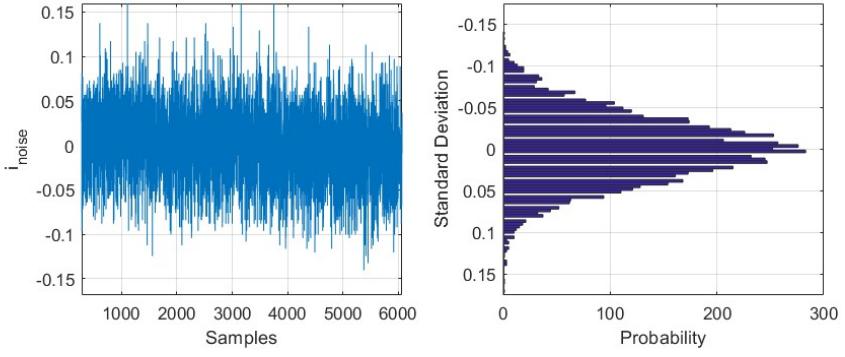


Figure 7: Theoretical Probability Density Function of gaussian current noise

Furthermore, as previously introduced, in order to reduce the amplitude of the noises, it is important to move towards the implementation of a filter.

Notice that, since in many control strategies (such as pole placement) a Kalman Filter (KF) is needed, it is important to study the entity of their spectra. In fact, these kind of filters work only provided that the input signals are affected by White Gaussian noise.

It is then enough to plot the spectra of the two noises to check their "whiteness", from which we can conclude that they are flat. In fact, this is a sufficient condition for a signal's noise to be white.

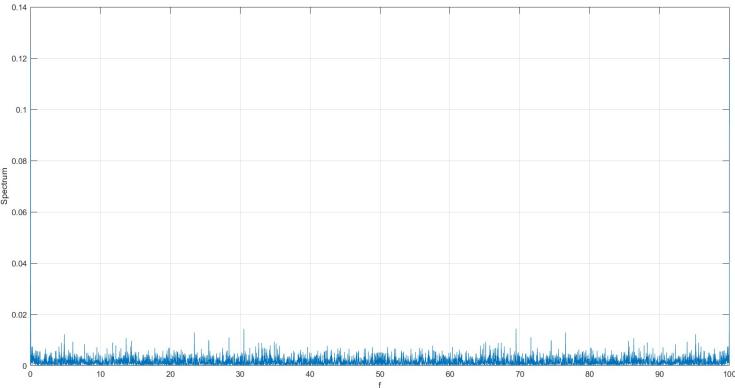


Figure 8: Current noise spectrum

From this analysis, it is then possible to conclude that the current and position sensors are affected by White Gaussian noises, and that it's possible to implement a Kalman filter in order to filter these quantities and to perform a state estimation.

3 Parameters Identification

In this chapter we propose several different identification procedures to characterize the model's parameters, such as the coil's resistance and inductance, and the electromagnetic force constant. In particular we start introducing simple procedures, such as formulas reversing, to finally analyze more advanced and complex methods.

3.1 R_{tot} and L_c Identification

In the following paragraphs, different methods for the identification of the Resistance and Inductance are presented.

3.1.1 R_{tot} Identification with Steady State Measurements Analysis

For the identification of the resistance R_{tot} we analyzed the electrical system in different steady state conditions. We then performed several measurements by selecting different constant voltages as input. The resistance computation was obtained considering the Ohm's law as follows:

$$V_c = R_{tot}I_c \quad (3.1)$$

The different results, such as the average measured current and the computed resistance, are represented in Table 4.

Dimensions	Data
Input V [V]	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24.
Measured I [A]	0.0913, 0.2557, 0.4224, 0.5994, 0.7778, 0.9400, 1.1102, 1.2689, 1.4762, 1.6247, 1.8113, 1.9898.
Resistance R_{tot} [Ω]	21.8907, 15.6447, 14.2061, 13.3454, 12.8568, 12.7659, 12.6104, 12.6101, 12.1928, 12.3098, 12.1457, 12.0610.

Table 4: Steady state experiments

Notice that, as previously introduced, we considered a steady state condition where the contribution of the inductance voltage drop is zero. Furthermore, is

straightforward to observe that the value of R_{tot} decreases with higher voltage values. This is due to un-modelled phenomena at low voltages that cannot be neglected. Bearing it in mind, we considered as outliers all the measurements before 10 V, discarding them from the resistance identification.

Notice also that, in these experiments, the gain of the current sensor was taken into account and, computing the average of the remaining resistance's values, we obtained:

$$R_{tot} = 12.4441\Omega$$

We also performed other experiments at steady state, applying a rising tension with a stair function, setting a step of 1 V every 2 seconds. We then measured the current values with Poliscope at 1000 Hz, processing only the data in the steady state intervals.

Thus, the resistance can be simply computed applying Ohm's law as reported in 3.1.

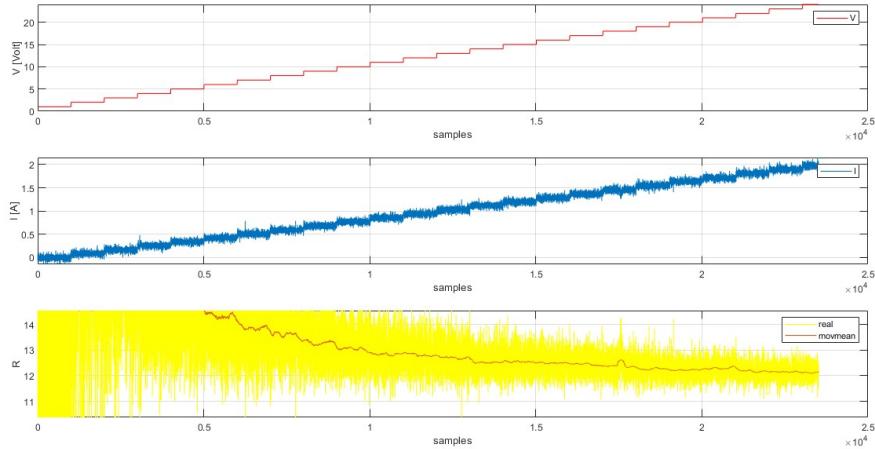


Figure 9: R second experiment

Computing the average of the remaining medium-high voltages measurements, we obtained $R_{tot} = 12.4308, \Omega$, similar to the one previously found.

As a third experiment, we applied a voltage stair function with a step of 2 V every 2 seconds. Like we did in the previous experiment, we measured the current at 1000Hz selecting the steady state intervals to compute R_{tot} .

Notice that, even if we selected a sampling frequency $f_s=1000$ Hz on Poliscope, from a close analysis of the collected data we observe that the real f_s is 734 Hz. In fact, since we applied a tension that varies every 2 seconds we expected 2000

samples in that interval, obtaining instead 1468 samples. This error between the selected f_s and real one happened in several experiments and unfortunately it is not deterministic. Thus, a pre-analysis on the collected data was always performed to find the real sample frequency, necessary for the computation of the inductance L_c (see sub-section 3.1.2). Once again, we underline that the visualized data are a subset of the experiment considering only the ones in steady state condition.

The results of this experiment are presented in Fig.10.

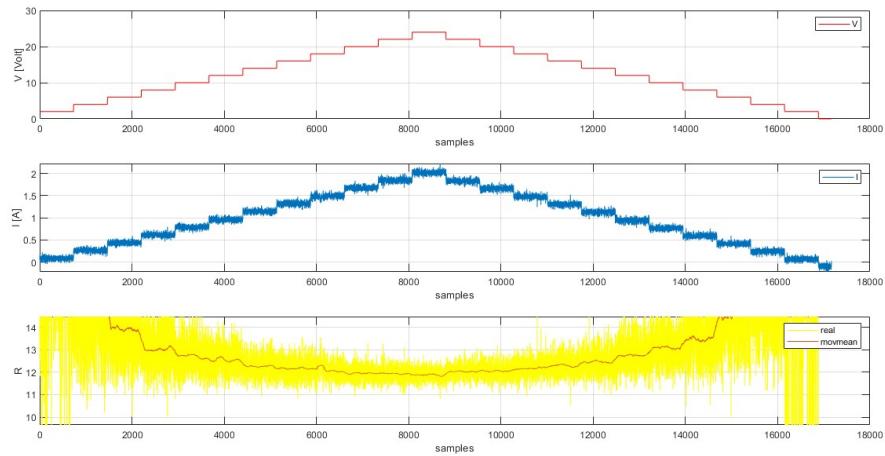


Figure 10: R third experiment

As for the first experiment we consider the data for medium-high tensions and, computing an average we obtain $R_{tot} = 12.2895\Omega$.

3.1.2 L_c Identification with Transient Measurements Analysis

Regarding the identification of the inductance L_c , we performed several experiments with a step input tension and, measuring the transient of the current, we were able to obtain the rise time with the “stepinfo” function in MATLAB.

First of all, we computed the mean value of the current before the step and we offsetted the selected data with this average. This pre-processing of the data is necessary to use the “stepinfo” function on MATLAB, which requires the transient to start from a 0 value. We also applied a moving average to smoothen the measurements in order to obtain a more accurate transient analysis.

Applying all this considerations, we were able to measure the rise time of the current step. It is important to notice that this quantity is measured in samples. Hence, we needed to divide the stepinfo.risetime by the sampling frequency f_s . For this reason, as previously explained, it is very important to check the real f_s achieved by Poliscope.

$$i_c(t) = \begin{cases} i_0 & t < t_{on} \\ i_0 + \frac{V_c}{R_{tot}}(1 - e^{-\frac{R_{tot}}{L_c}(t-t_{on})}) & t_{on} \leq t \leq t_{off} \\ i_0 + \frac{V_c}{R_{tot}}e^{-\frac{R_{tot}}{L_c}(t-t_{off})} & t > t_{off} \end{cases} \quad (3.2)$$

$$\begin{aligned} t_{90\%} &= -\tau \ln(0.1) & t_{10\%} &= -\tau \ln(0.9) \\ t_{rise} &= \frac{\text{stepinfo.risetime}}{f_s} = t_{90\%} - t_{10\%} & &= \tau \ln(9) \end{aligned} \quad (3.3)$$

Finally, starting from the rise time definition (3.3) and considering the RL circuit equations (3.2), we were able to compute L_c as follows:

$$\tau = \frac{L_c}{R_{tot}} \quad L_c = \frac{t_{rise}}{\ln(9)} R_{tot} \quad (3.4)$$

From an initial experiment with a step input of 24 V, we obtained the data represented in Fig. 11

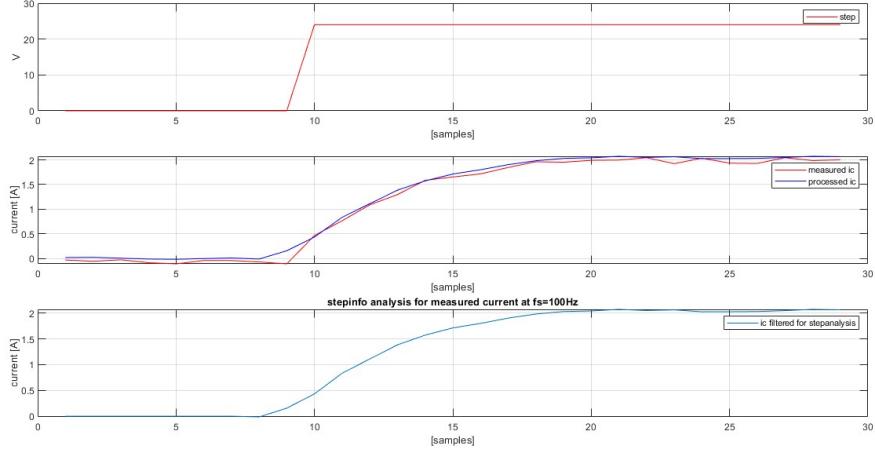


Figure 11: L analysis from 24V step input

From the second experiment, previously described in Fig.9, we obtained the following data:

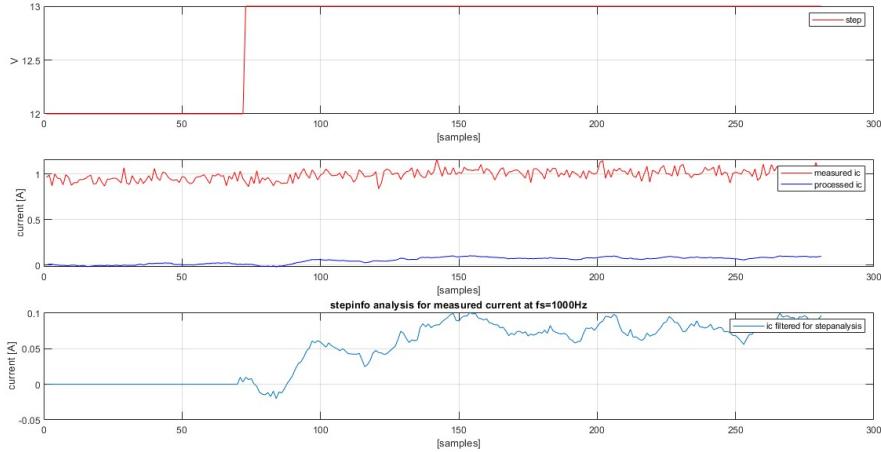


Figure 12: L analysis from 1V step input

Instead, from the third experiment described in Fig.10, we obtained the data represented in Fig.13.

The numerical results are summarized in table 5.

Notice that the identification of the inductance L_c using the “stepinfo” function is not the best method, because the transient analysis was done on filtered measurements to reduce the noise on the current sensor. Furthermore, for little steps

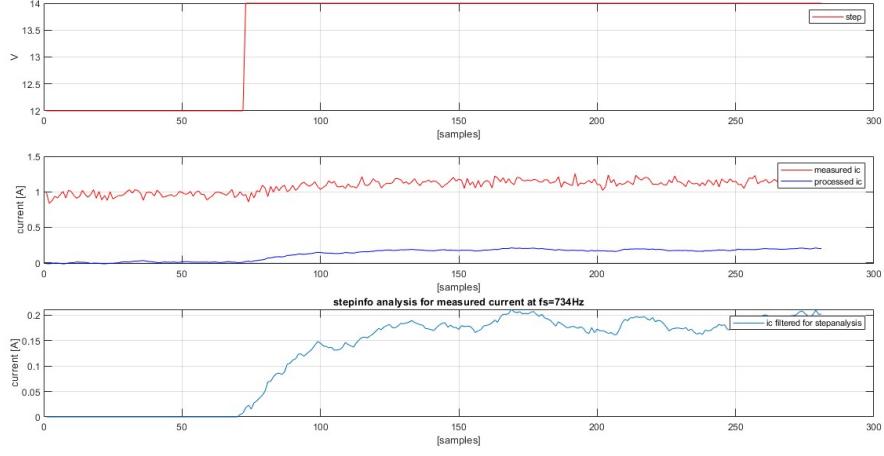


Figure 13: L analysis from 2V step input

like in Fig12, the transient is hard to notice, thus the filtering of the curve leads to a modification of the real transient.

We can also observe from table 5 that, for smaller step analysis, we have greater variance in the computed inductance.

Data (# transient analysed)	Average L_c	Variance L_c	Standard deviation σ
24V step data (7)	0.459508571	0.002393457	0.048922975
1V step data (23)	0.335213333	0.028674457	0.16933534
2V step data (23)	0.381403333	0.009592741	0.097942539

Table 5: L values from transient analysis

3.1.3 Temperature Effect on R_{tot} Identification

The identification procedure previously introduced, is really temperature dependent. In fact, due to joule losses, the temperature of the coil increases and it results in a variation of the resistance R_{tot} , as shown in the following equation:

$$R_{tot}(T, t) = \frac{V_c}{i_c(T(t))} = R_0[1 + \alpha(T(t) - T_0)] \quad (3.5)$$

Notice that the model introduced in 3.5, relates the resistance R_{tot} with the temperature variation of the coil over time. Hence, we can introduce R_0 as the resistance at a reference temperature T_0 (e.g. 20°) and α as the temperature coefficient of a copper coil.

Consider for instance an increase of temperature in the system. In this case, since the resistance is directly proportional to this parameter, it increases as well. Hence, if we try to obtain the same voltage of the cold system, we need to supply an higher current I_c ($T \uparrow \Rightarrow R_{tot} \uparrow \Rightarrow I_c \downarrow$). This means that, imposing the fixed tension and applying the Ohm's law (3.1) in different temperature conditions, it leads to difference resistance values. We can observe this phenomena in Fig.14, that represents two current measurements at different temperatures.

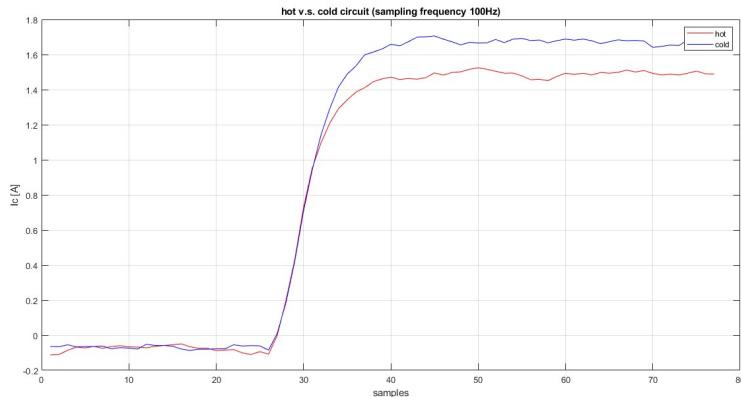


Figure 14: Hot v.s. Cold circuit

Notice that this problem is of utmost importance not only for the identification procedure, but also for the control system design. In fact, in model-based control strategies, the lower is the uncertainty on the model's parameters, the more accurate is the control. In fact, notice that the difference of R_{tot} value from cold and hot condition is about 10%, and it increases over the usage time. However, introducing a new state in the system's model representing the temperature's dynamics is not a

good approach, since we loose the observability property. Hence, in order to address this problem and to be as accurate as possible, we analyzed the system's parameter before every control run.

3.1.4 R_{tot} and L_c Identification with Real Model Matching

The parameters R_{tot} and L_c correspond to the real model parameters. It means that, if our estimation is accurate enough, the output of the ideal model should match the real one, given from the current sensor. Thus, the main idea of this estimation procedure is to change these two parameters until the perfect models' output matching.

In order to implement it, we fed the two systems with the same input v_c and we varied the value of the resistance R_{tot} and inductance L_c with the two available potentiometers. The procedure can be resumed in Fig. 15.

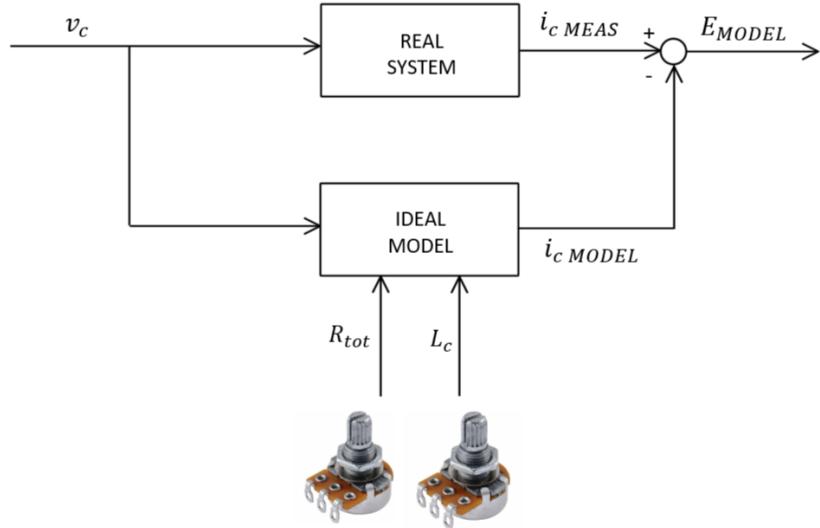


Figure 15: Real model matching scheme with varying R_{tot} and L_c

Notice that the optimal matching, i.e. the optimal parameters identification, is obtained when $E_{model}=0$. Notice however that, since $i_{c,meas}$ is affected by noise, in our case we were not able to precisely reach zero matching error.

In the following figure it is possible to appreciate the matching procedure in cold circuit condition:

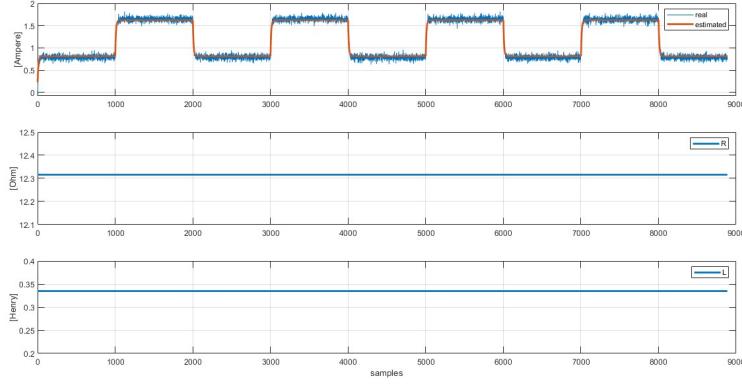


Figure 16: Real model matching in cold circuit condition

We fed as input a stair function that allows to capture both the steady state and the transient dynamics. In fact, in order to tune R_{tot} , we had to look at the perfect matching in the constant voltage period eliminating any kind of offsets. Instead, in order to correctly identify L_c , it is important to ensure to have the same transient duration τ . Indeed, a higher value of inductance leads to a faster response and vice versa. Hence, we were able to identify $R_{tot} = 12.4, \Omega$ and $L_c = 0.34, H$, that are consistent with the previous results.

This method can be used for resolving the previously introduced temperature effect, with a initial real model matching we control if the electrical parameters changed due to temperature. Nonetheless we always tried to keep the Maglev in cold conditions by limiting its duty cycle.

3.1.5 R_{tot} and L_c Identification with System Identification Toolbox

Another possible method to identify R_{tot} and L_c parameters is by using the system identification toolbox provided by MATLAB. Feeding this toolbox with an input controlled tension and a related output measured current, we were able to perform a time series analysis and to model the dynamic relationship between input and output.

Comparing the transfer function that represents an RL circuit (reported in eq. 1.15) with the estimated one, we were able to compute the resistance and inductance parameters. Hence, feeding the data from the second experiment reported in Fig.10, we obtained the following estimated transfer function:

$$G_{v_c \rightarrow i_c} = \frac{\frac{1}{L_c}}{s + \frac{1}{L_c} R_{tot}} = \frac{3.17}{s + 38.71} \quad (3.6)$$

Notice that this estimation has a “fitting” percentage of 89.17% with respect to the measured data, so it faithfully represents the electric dynamics. The identified R_{tot} and L_c parameters’ values, are reported in table 6. From now on we will consider this identified parameters in our control, in cold conditions.

R_{tot}	12.2114 Ω
L_c	0.3155 H
τ	0.0258 s

Table 6: Parameter identification with systemID

3.1.6 R_{tot} and L_c Validation with an Extended Kalman Filter

In order to perform a validation of the obtained parameters, we implemented an Extended Kalman Filter. In particular, we considered the two parameters R_{tot} and L_c as two additional fictitious states of the electrical system.

Hence, we obtained the following extended state vector:

$$x_e = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} i_c \\ L_c \\ R_{tot} \end{bmatrix} \quad (3.7)$$

Notice that an Extended Kalman Filter is needed because, adding these two additional states, the electrical system becomes non linear. Hence, we proceeded by considering the extended, third order and non linear, state space system in continuous time:

$$\begin{cases} \dot{x}_1 = -\frac{x_3}{x_2}x_1 + \frac{1}{x_2}u \\ \dot{x}_2 = 0 \\ \dot{x}_3 = 0 \\ y = x_1 \end{cases} \quad (3.8)$$

Notice that in order to implement correctly this kind of observer, we needed first to obtain a discretized state space system, exploiting the “Backward Euler” method. The state space system in discrete time is reported in 3.9.

$$\begin{cases} x_1(k+1) = x_1(k) \left(1 - \Delta t \frac{x_3(k)}{x_2(k)}\right) + \frac{\Delta t}{x_2(k)} u(k) + v_{x1}(k) \\ x_2(k+1) = x_2(k) + v_{x2}(k) \\ x_3(k+1) = x_3(k) + v_{x3}(k) \\ y(k) = x_1(k) + v_y(k) \end{cases} \quad (3.9)$$

Notice that we introduced state and output discrete noises with the following properties:

$$v_x = WGN(0, Q) \quad v_y = WGN(0, R) \quad (3.10)$$

where Q is the process noise covariance 3x3 diagonal matrix and R is the measurement noise covariance matrix, in this case just 1x1 and equal to the variance of the

current sensor reported in table 3.

Once we fully defined the system and its noises' properties, we fed the EKF Simulink block with steady state and transient measurements. In this way, by properly tuning the matrix Q, we found the following results:

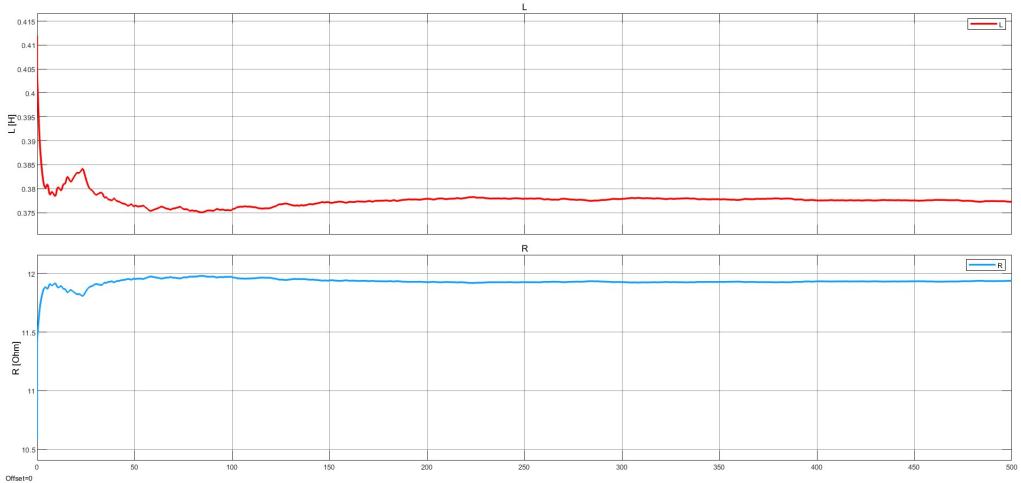


Figure 17: EKF R_{tot} and L_c estimation

It is then straightforward to notice that, setting the initial state vector x_0 to the nominal values of R_{tot} and L_c (reported in table 1), the two parameters converge to their real estimation. Although this method confirms the values previously introduced in sub-sections 3.1.1 and 3.1.2, we preferred to classify it as a validation procedure. In fact, since the values oscillate around their constant one, we weren't able to extract a dependable estimation, preferring thus to rely on other more precise methods.

Notice also that, in order to perform this analysis, we exploited the built-in EKF Simulink block. However, since it is not present in the MATLAB version compatible with the PoliArd environment, we faced the challenge to build this estimator from scratch. This problem is addressed and explained in the section 4.4, as it represents an important step for the control system design affected by measurements noise.

3.1.7 R_{tot} and L_c Identification as a Solution of a Minimization Problem

Another important identification procedure we adopted is the solution of a minimization problem. In particular, we considered a data-set composed by one thousand samples, each data point contains information of the applied input (voltage) and information of the measured output (i_{meas}). The data-set has been split in two parts selecting the first 70% of data to train the model, and using the remaining 30% to validate it.

The goal of this optimization problem is to minimize the following objective function:

$$J(\theta) = \sum_{i=1}^n (\tilde{y}_i - y_{th}(\theta))^2 \quad (3.11)$$

where i_{th} is the theoretical current of an inductor charging phase of an RL circuit. The following current characteristic is reported in equation (3.12).

$$i_{th}(t) = \begin{cases} i_0 & t < t_{on} \\ i_0 + \frac{V_c}{R_{tot}}(1 - e^{-\frac{R_{tot}}{Lc}(t-t_{on})}) & t_{on} \leq t \leq t_{off} \\ i_0 + \frac{V_c}{R_{tot}}e^{-\frac{R_{tot}}{Lc}(t-t_{off})} & t > t_{off} \end{cases} \quad (3.12)$$

We can define t_{on} and t_{off} as the time instants in which the two transients of charge and discharge start. Notice that this is a single-input, single output strictly proper system which can be solved with an unconstrained optimization problem based on least squares criterion. This algorithm works such that, it computes the best value for the parameter that minimize the objective function previously introduced. This objective function can be rewritten collecting data and variables into vectors and using regressors, obtaining:

$$J(\theta) = (\tilde{Y} - \tilde{X}\theta)'(\tilde{Y} - \tilde{X}\theta)$$

The linear unconstrained least square optimization problem is then:

$$\min_{\theta} J(\theta) = \min_{\theta} (\tilde{Y} - \tilde{X}\theta)'(\tilde{Y} - \tilde{X}\theta)$$

The solution of this linear unconstrained least square is:

$$\theta^* = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{Y}$$

Finally, as reported in table 7, we were able to identify the resistance and inductance values that minimize the square difference between the real and the theoretical current functions.

R_{tot}^*	12.3521 Ω
L_c^*	0.3432 H

Table 7: Parameter identification as result of a minimization problem

In the following figure it is possible to appreciate the measured and the theoretical current functions superimposed, highlighting once again the correctness of this procedure.

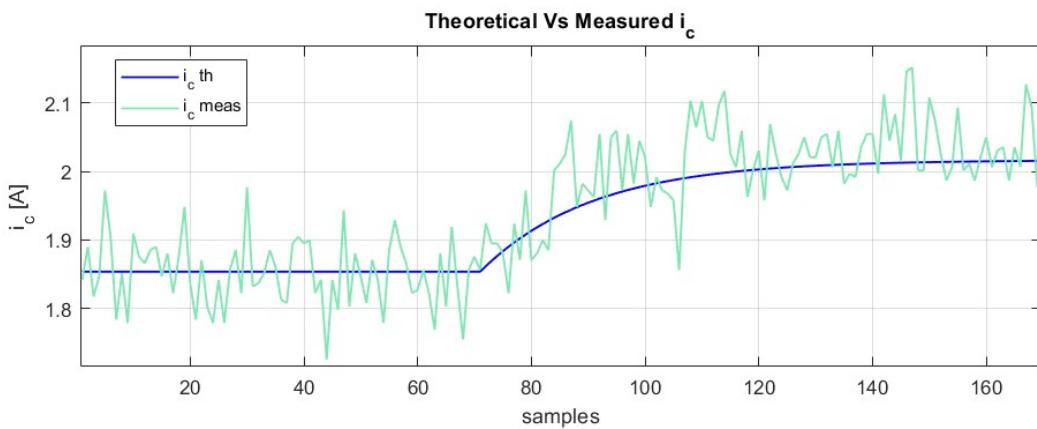


Figure 18: Theoretical Vs Measured i_c

3.2 Km identification

The parameter K_m is the electromagnetic force constant. We identified this quantity by reversing equation 1.4 as follows:

$$K_m = \frac{2x_b^2}{I_c^2} M_b(g - a) = \frac{2x_b^2}{I_c^2} M_b g \quad (3.13)$$

Notice that, since this parameter depends only on the magnetic force applied by the copper coil, it is possible to be estimated only when the iron ball is not lying on the pedestal. In fact, K_m identification, is the result of the balance of forces that includes only the magnetic and the gravitational one.

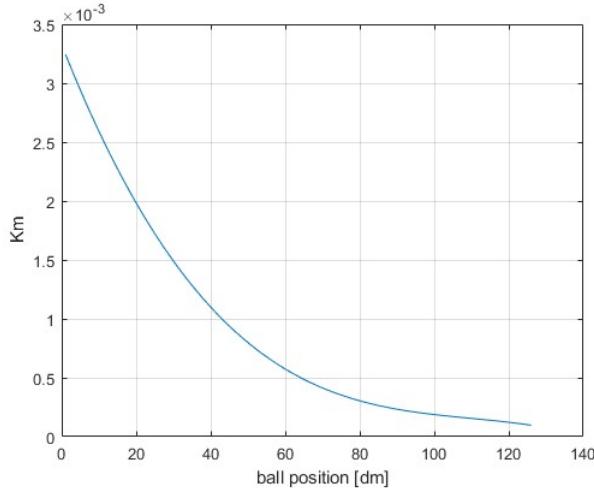
Notice however that, as reported in 3.13, the electromagnetic force constant is proportional to the vertical acceleration. This quantity can be considered zero when the ball is stabilized in the equilibrium position but, in order to get this result, we need a stable control.

To solve this problem, we executed different experiments in different ball's known positions and, by slowly increasing the input voltage, we measured the current in the instant where the ball starts rising. In fact, in this point, the ball loses contact from the pedestal being subject only to its weight and to the magnetic force. Furthermore, it is also possible to impose zero acceleration, simplifying thus the problem without the need of a state observer. Thus, with the help of little pieces of wood with known width we reproduced the experiment from different initial positions, obtaining the following data:

Initial position	Measured current	K_m
12 [mm]	1.464 A	0.9726×10^{-4} [Nm ² /A ²]
10.5 [mm]	1.116 A	1.6738×10^{-4} [Nm ² /A ²]
7.5 [mm]	0.78 A	3.4264×10^{-4} [Nm ² /A ²]
5.5 [mm]	0.564 A	6.5534×10^{-4} [Nm ² /A ²]

Table 8: K_m data

We can visualize the data fitted in the following graph:



(a) K_m values at different position levels



(b) K_m experiment setup

Figure 19: K_m Experimental Identification

We can observe that, even if the value of K_m is provided as constant (see table 1), it changes in function of the position of the ball. Notice however that, for the positions close to the pedestal, that are of interest for our controls, K_m is more or less constant with average value of $K_m = 1.3232 \times 10^{-4} [Nm^2/A^2]$.

4 Controller design of Magnetic Levitation System (MAGLEV)

Implementing control techniques is essential to guarantee steady and accurate management of both current and position in the MAGLEV system. This section will concentrate on a different series of control strategies used to manage both elements, assessing the specific problems that each presents as well as the solutions that have been successful in getting the best system performance. The key goal is to create precise and responsive regulation, which will allow the MAGLEV system to retain a desired position and quickly adapt to changing conditions. We shall examine the applied control techniques, outlining their attributes and anticipated effectiveness.

4.1 Cascade control

The precision and stability of a control system are attempted to be increased using the advanced control approach known as Cascade Control. It employs a large number of control loops connected in series that work together to create a cascading structure. The core idea is that whereas the secondary control loop regulates a secondary process variable (in our case the position of the ball) that directly impacts the primary variable, the primary control loop regulates the primary process variable (in our case the current). Since we have the model of the physical system and we have performed the parameter identification, we are now ready to split the model the previously discussed two parts: one taking care of the current part and one taking care of the position.

Following the cascade control approach we need to perform three sequential tasks:

1. Create the inner loop controller: in our case we design a simple Proportional Integral (PI) controller, to give the desired performance to the stable inner current loop;
2. Create the outer loop controller: in our case we proposed several types of controller managing the stability and the performance of the unstable outer position loop (e.g. PID, PIV, ...);
3. Optimize the controllers parameter: the primary and secondary controllers should be cascaded, this means that the output of the secondary controller serves as the primary controller's reference input. A tuning of the control gains is crucial to guarantee the desired performance in terms of overshoot and settling time.

4.1.1 Internal Current loop synthesis

In this section the internal control of the current loop is treated in detail. We can start our analysis considering a general closed-loop system, which is represented in the block diagram of Fig.20

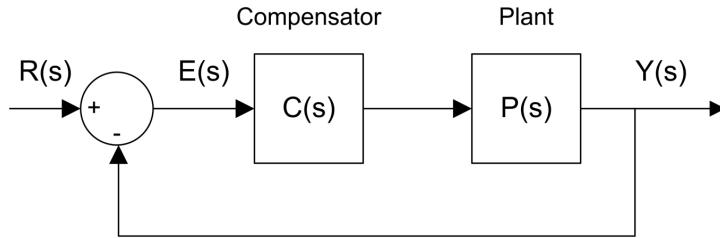


Figure 20: System controlled in feedback

$C(s)$ is the transfer function of the controller and $P(s)$ is transfer function of the considered plant model. The reference signal $R(s)$ must be tracked by the measured output $Y(s)$, and the tracking must adhere to certain predetermined criteria in terms of overshoot, settling time and robust to disturbance.

The electromagnet coil current is controlled by the commanded coil voltage in the inner closed-loop system. As compensator we designed a Proportional-plus-Integral (PI), correctly tuned to meet the desired design performance requirements. We used information coming from the response to a 0-to-1 A square wave coil current setpoint. Before introducing the computed gains of the PI controller and the procedure that we used to obtain these results, we can recall the current loop transfer function.

$$G_c = \frac{0.0668}{0.03084s + 1} \quad (4.1)$$

As mentioned, in order to tune the gains of the PI controller, we have imposed two different specifications: the maximum percent overshoot and the Maximum peak time, together with the null steady-state error. Let's analyse how these requirements can be expressed in a mathematical formulation.

In a second-order system, the amount of overshoot depends solely on the damping ratio parameter and it can be expressed using the equation

$$PO\% = 100 \times e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} \quad (4.2)$$

represents the percentage overshoot, which indicates the maximum deviation of the response from the desired set-point, while ζ is the damping ratio of the system.

The peak time depends on both the damping ratio and natural frequency of the system and it can be derived as

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (4.3)$$

The term t_p refers to the amount of time it takes for a system's response to reach its peak or greatest value, ω_n is the natural frequency of the system and ζ is the damping ratio of the system.

We can notice that the damping ratio affects the shape of the response while the natural frequency affects the speed of the response.

Solving a set of these requirements and choosing the natural frequency and the damping ratio as unknowns, we have found the expression of these containing our requirements:

$$\xi = \frac{\left| \log \left(\frac{PO_c}{100} \right) \right|}{\sqrt{\pi^2 + \log^2 \left(\frac{PO_c}{100} \right)}} \quad (4.4)$$

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \xi^2}} \quad (4.5)$$

Starting from these results, we computed the gains of the Proportional-Integral controller as:

$$Kp_c = \frac{2 \cdot \xi \cdot \omega_n \cdot \tau_c - 1}{K_c} \quad (4.6)$$

$$Ki_c = \frac{\omega_n^2 \cdot \tau_c}{K_c} \quad (4.7)$$

Where τ_c is the time constant and K_c is the gain of the first order current transfer function.

As numerical requirements we set:

1. Maximum percent overshoot of 1.5%, i.e., $PO\% \leq 1.5\%$
2. No steady-state error, i.e., $e_{ss} = 0$
3. Maximum peak time of 0.05 seconds, i.e., $t_p \leq 0.05$ s

In order to obtain these requirements, we need a second-order system with the following parameters:

$$\omega_n = 104.8945$$

$$\xi_c = 0.8007$$

this means values for controller gains of:

$$Kp_c = 62.5902$$

$$Ki_c = 5080$$

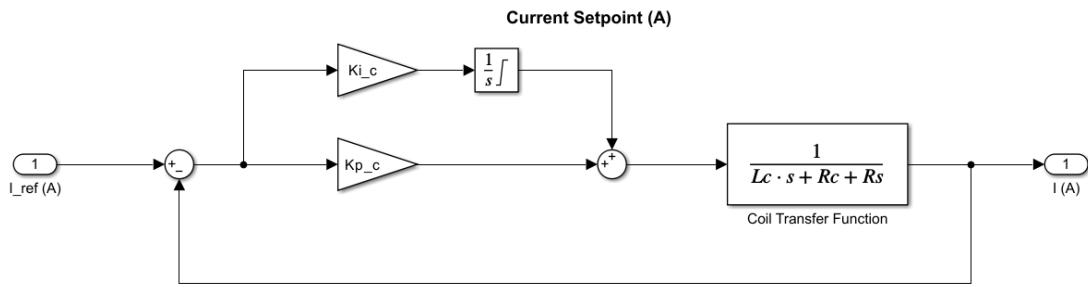


Figure 21: Inner current loop with a PI-controller

The closed loop system has the following transfer function:

$$T_c = \frac{4.181s + 339.3}{0.03084s^2 + 5.181s + 339.3} \quad (4.8)$$

Analysing the response to a step of current in Fig.22, we can notice the confirm the desired behaviour given by our specifications:

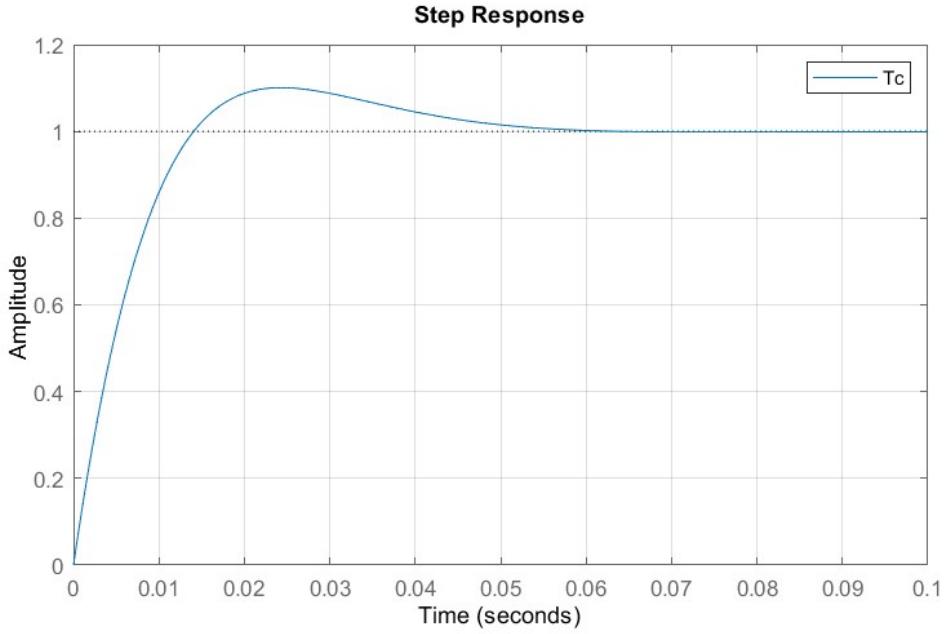


Figure 22: Step response of the closed-loop current system

As can be seen in Fig.21, we implemented inside the PI-controller an anti-windup strategy. This technique is used to reduce the undesired effects of integrator windup, which can happen when an integral controller is utilized in a negative feedback loop like in this case.

When the controller's output value exceeds the limits of the actuator or the controlled process, an integrator windup occurs. For instance, the PI (Proportional-Integral) controller in this situation regulates a system's current based on the voltage input. If the system does not intervene, the integral action of the controller can eventually lead to a continual increase in the output value as the control error accumulates over time.

Anti-windup is used to stop this occurrence. There are various methods that can be used to implement this anti-windup strategy. The typical technique applied in this case is to restrict the controller's integral action when the actuator or controller output hits the maximum or minimum limitations. The integral action is "clamped" or saturated in this instance to a maximum or minimum permitted value.

This prevents integrator windup, ensuring that the integral action of the controller remains within acceptable bounds and that a stable and controlled response of the system is obtained.

The overall controlled closed loop system can be described by the Bode plots in Fig.23 .

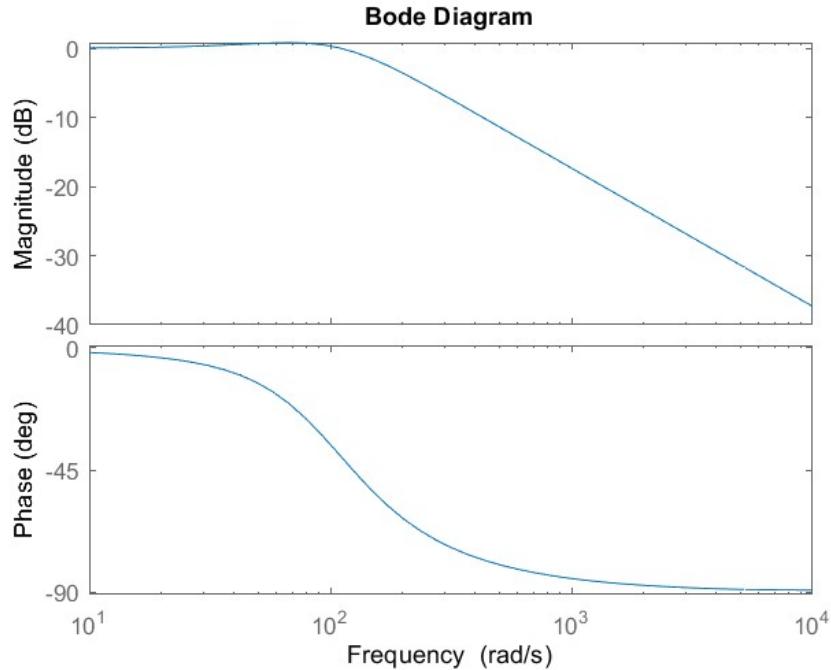


Figure 23: Bode plots of the closed-loop current system

Analyzing Bode plots, it can be shown that, as expected, the natural frequency ω_n has the value 104.8945 that we indirectly imposed.

To sum up, the designed regulator for the internal current loop is a PI-controller in the form:

$$u(t) = K_p + K_i \cdot \frac{1}{s}$$

with $K_p = 62.5902$, $K_i = 5080$ and saturation current $i_{sat} = 2.0A$

An alternative method used to tune the parameters of the PI was simply plugging in the desired specifications in PIDtune and specifying the desired phase margin 90 deg , and bandwidth, 500 Hz , that we tried to increase as much as possible baring in mind that the board running the code works at most at a frequency of 1000 Hz . While the derivative part wasn't necessary since the obtained response is smooth enough (since there is no oscillations), the integral part is necessary since the system manifests a big steady-state error.

Using the PID block on Simulink we toggled the anti-windup option to apply it, that's due to the 24 V limit that we have on the voltage input. The technique used for the anti-windup is clamping.

Finally, a Feed Forward is used to speed up the response even further, where we simply multiply the Resistance to the reference.

The computed gains are the following: $K_p = 240.43$, $K_i = 2104$ with a saturation voltage of $\pm 24 \text{ V}$.

The step response is reported below.

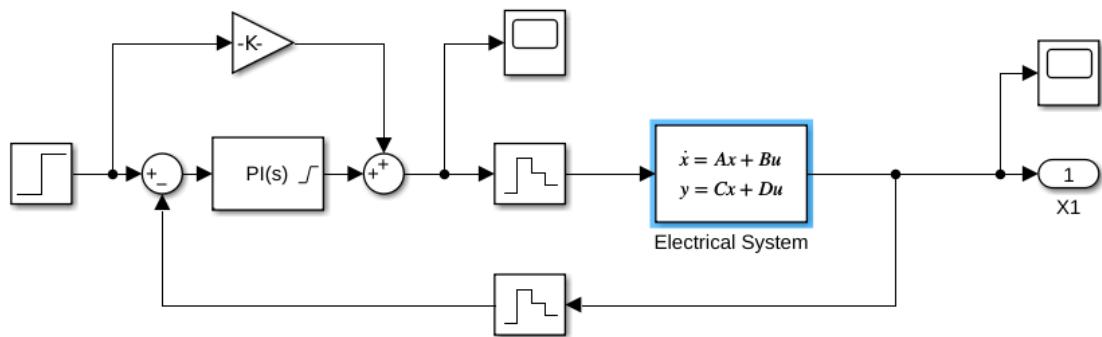


Figure 24: Electrical PI scheme

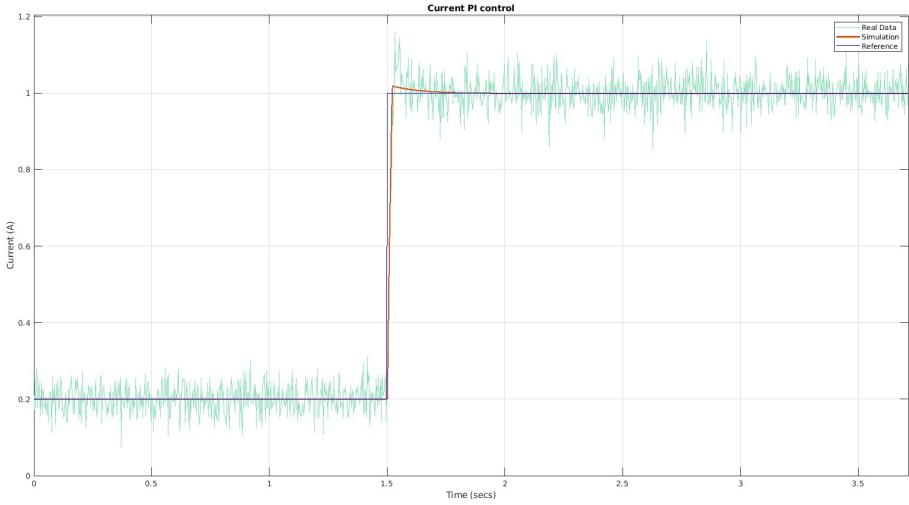


Figure 25: Electrical PI step response

4.1.2 External Position loop synthesis

Regulating and monitoring the ball's position when it is in midair is the second and final control problem that it has been solved. As mentioned, the problem regarding the stability and the performances of the mechanical (position and velocity) subsystem is solved using a proper selected controller for the outer loop of the cascade control. Different control structures has been tried. In this paragraph these architectures will be proposed, analysing the obtained results and critical issues.

Proportional-Integral-Velocity (PIV) controller

In this section, the steel ball position is controlled using the Proportional-plus-Integral-plus-Velocity (PIV) control and feed-forward (FF) action illustrated in Fig. 26 .

The idea is to design the ball position controller for the equilibrium position, so we exploited a fundamental property of the cascade controller: because the inner loop is much faster than the outer loop, we can consider the inner closed-loop system as an identity matrix of order n when we design a controller for the outer closed-loop system. So, an approach similar to the one used for the inner loop can be applied. The PIV+FF control is given by the following equation:

$$I_c(s) = (K_p + K_i \cdot \frac{1}{s})(X_{1ref}(s) - X_1(s)) - K_v \cdot s \cdot X_1(s) + K_{FF} \cdot X_{1ref}(s)$$

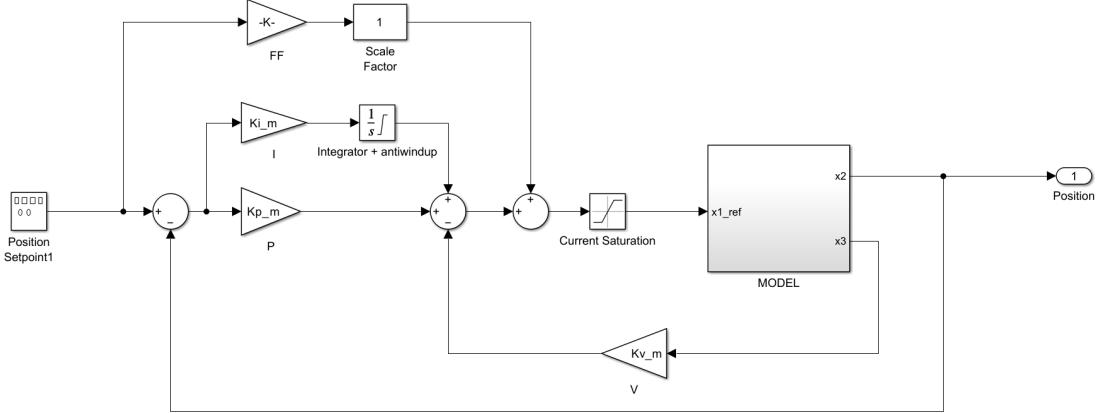


Figure 26: PIV + FF control scheme

Before introducing the computed gains of the PIV controller, we can recall the position-loop transfer function and how we computed it in this section. Defining as K_m the gain and the natural frequency of the mechanical transfer function, as:

$$K_{pos} = \frac{x_{2bar}}{x_{1bar}}$$

$$\omega_{nat,pos} = \sqrt{\frac{2 \cdot g}{x_{2bar}}}$$

The numerator and the denominator of the transfer function are:

$$G_{mec,num} = -K_{pos} \cdot \omega_{nat,pos}^2$$

$$G_{mec,den} = s^2 - \omega_{nat,pos}^2$$

and so the mechanical transfer function is:

$$G_x = \frac{-24.12}{s^2 - 2452}$$

In order to tune the gains of the PIV controller, we have imposed the same specifications we used for the design of the current PI with the first method: the maximum percent overshoot and the Maximum peak time, together with the null steady-state error. Additionally we set the position of the pole in a desired location. Let's analyse how these requirements can be expressed in a mathematical formulation.

$$\xi = \frac{\left| \log \left(\frac{PO_c}{100} \right) \right|}{\sqrt{\pi^2 + \log^2 \left(\frac{PO_c}{100} \right)^2}}$$

$$\omega_n = \frac{4}{\xi \cdot tspos}$$

This approach is similar to the one presented for the PI-current control, so the gains of the PIV controller are presented without any further description.

$$K_p = -\frac{(2 \cdot p_0 \cdot \omega_{nat,pos} \cdot x_{2bar} + \omega_{nat,pos}^2 \cdot x_{2bar} + 2 \cdot g) \cdot x_{1bar}}{2 \cdot x_{2bar} \cdot g}$$

$$K_v = -\frac{(p_0 + 2 \cdot \omega_{nat,pos}) \cdot x_{1bar}}{2 \cdot g}$$

$$K_i = -\frac{p_0 \cdot \omega_{nat,pos}^2 \cdot x_{1bar}}{2 \cdot g}$$

As numerical requirements we set:

1. Maximum percent overshoot of 5%, i.e., $PO\% \leq 5\%$
2. No steady-state error, i.e., $e_{ss} = 0$
3. Maximum peak time of 0.3 seconds, i.e., $t_p \leq 0.3$ s
4. Location of the pole between 30 and 50, i.e., $p_0 = 50$

In order to obtain these requirements, we need a second-order system with the following parameters:

$$\omega_n = 19.3207$$

$$\xi_c = 0.6901$$

this meas values for controller gains of:

$$Kp_m = -161.3590$$

$$Ki_m = -2.7636$$

$$Kv_m = -618.9770$$

The closed loop system considering the PIV-controller and the current closed-loop with the PI-controller, has the following transfer function:

$$T_{piv} = \frac{3892s + 1.493e04}{s^3 + 66.67s^2 + 1440s + 1.493e04}$$

The PIV control system is built to account for disturbances from the linearized operating point, the position and current equilibrium. Therefore the feed-forward action is required. In other words, the PIV controller makes up for dynamic disturbances while the feed-forward action corrects for the ball's gravitational bias. The gain of the feed-forward is:

$$K_{FF} = \frac{x1bar}{x2bar} = 101.6667$$

So the overall closed-loop transfer function, considering the PIV, the PI, the FF term and the entire system is:

$$T_m = \frac{2.323e05s^2 + 3.301e06s + 9.242e06}{161.4s^4 + 1.138e04s^3 + 2.736e05s^2 + 3.301e06s + 9.242e06}$$

Analysing the response to a step of current in Fig.27, we can notice behaviour given by our specifications and we can notice the asymptotically stability of our controlled system.

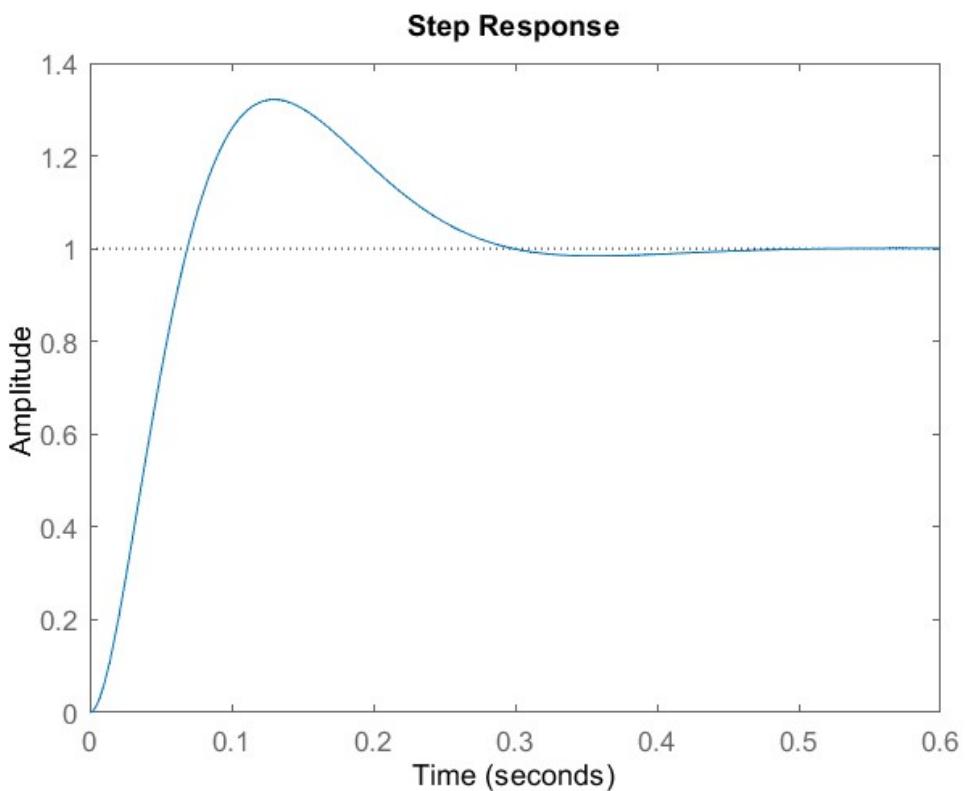


Figure 27: Step response of the overall closed-loop system

The overall controlled closed-loop system can be described by the Bode plots in Fig.28 .

Analyzing Bode plots, it can be shown that, as expected, the natural frequency ω_n has the value 19.3207 that we indirectly imposed and the system is asymptotically stable.

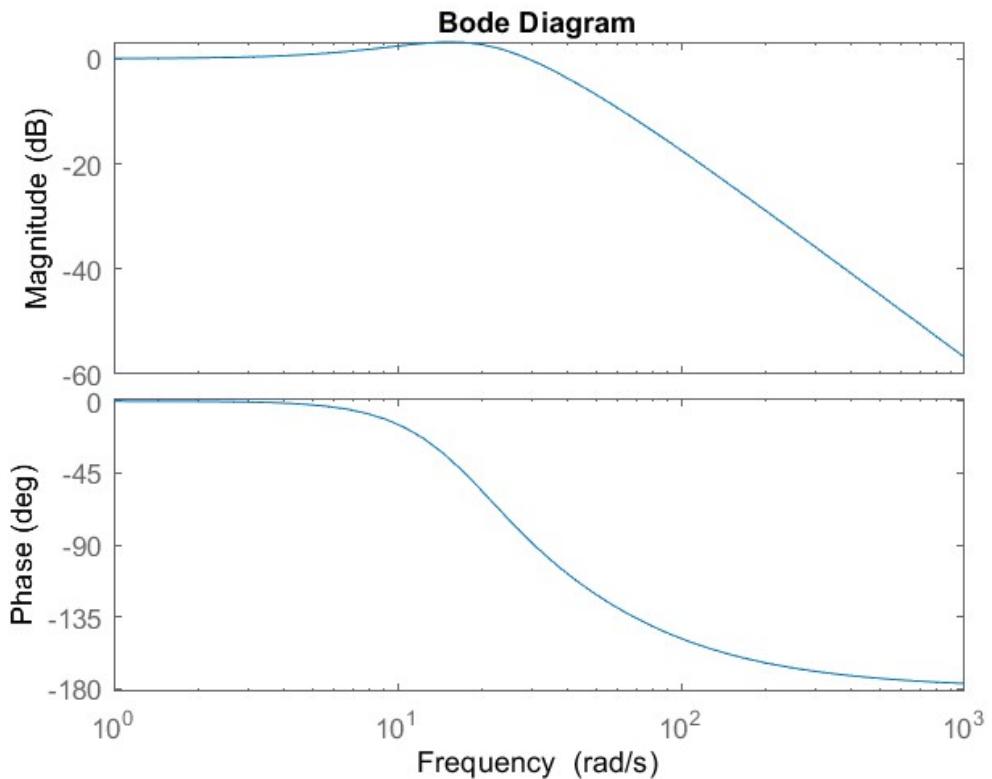


Figure 28: Bode plots of the closed-loop system controlled with PI and PIV

Before concluding the section dedicated to PIV, some simulations results are presented. Two different experiments are carried out.

1. keeping the equilibrium position i.e., $x_{bar} = 8mm$;

To perform this operation we used a constant reference position, clearly with the same value of the equilibrium position we have set.

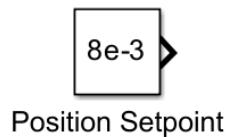


Figure 29: Constant position reference

The results of the PI+PIV+FF control are shown in the following plots.

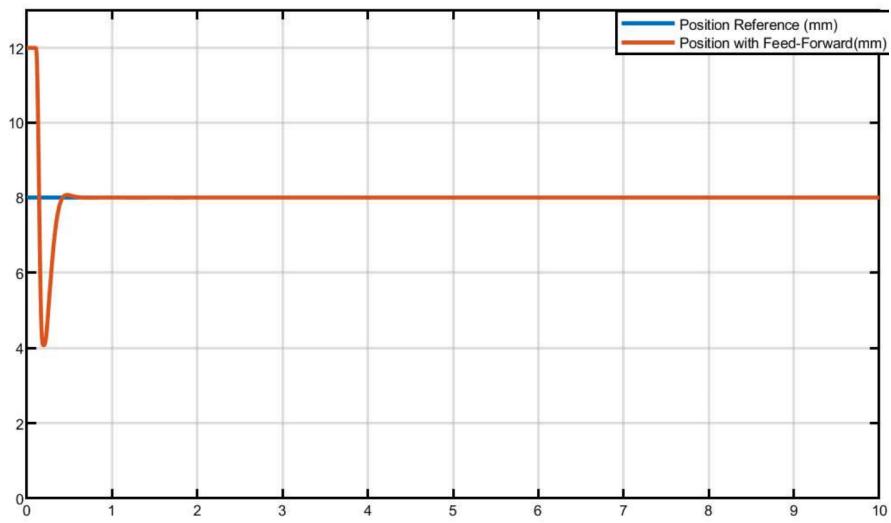


Figure 30: Position tracking results over time

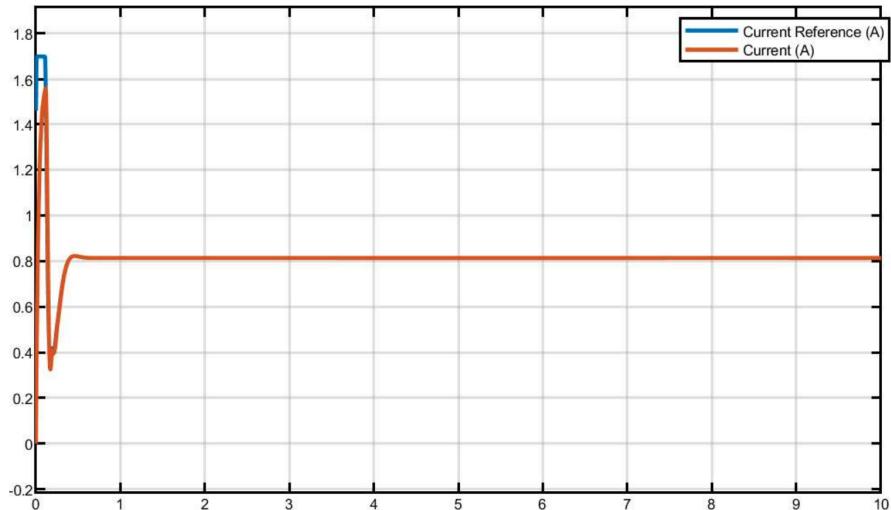


Figure 31: Current tracking results over time

In Fig. 30 we can notice the behaviour of the position when we want to keep the ball in a constant position. In Fig. 31 the behaviour of the needed current is shown and Fig. 32 treats the modulation of the input (V).

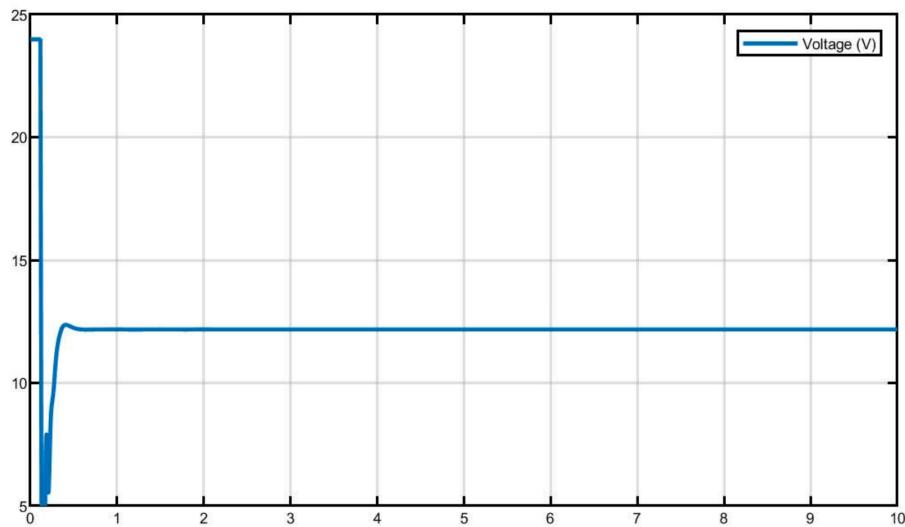


Figure 32: Input voltage over time

2. performing a square wave around the equilibrium position. At this point we want to check the ball position behavior in response to a desired ± 1 mm square wave position set-point from the ball equilibrium position in mid-air. In order to create the reference wave the scheme in Fig. 33 has been used.

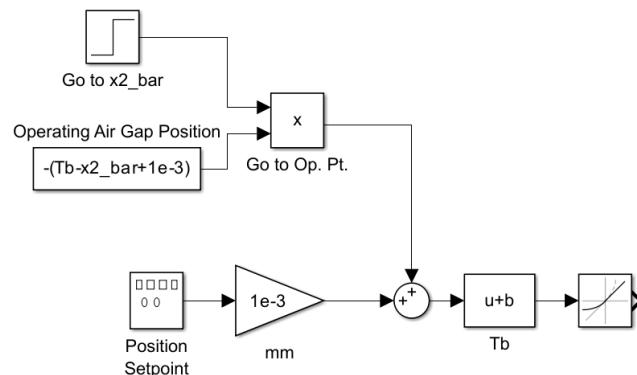


Figure 33: Constant position reference

The results of the PI+PIV+FF control are shown in the following plots.

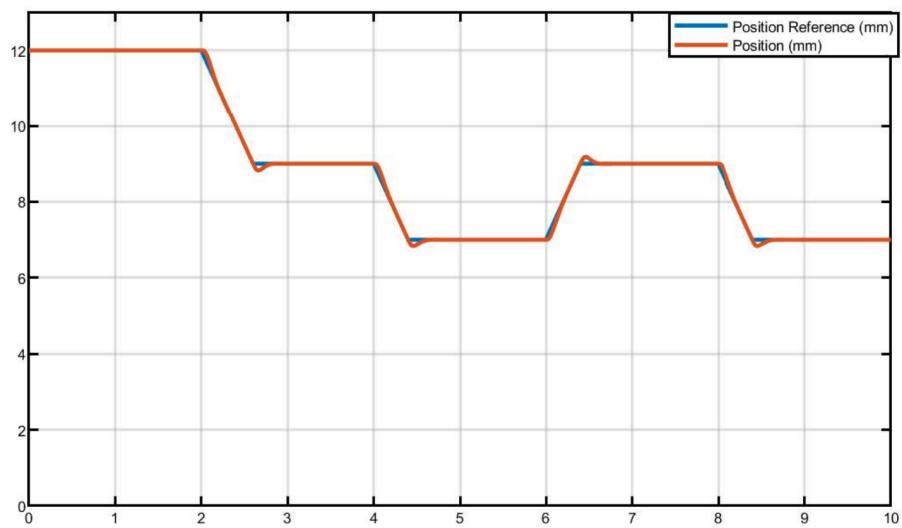


Figure 34: Position tracking results over time

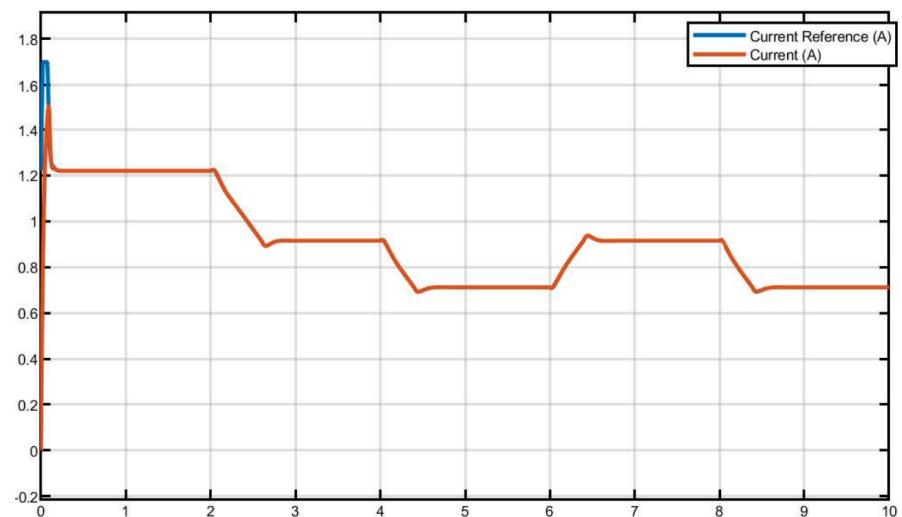


Figure 35: Current tracking results over time

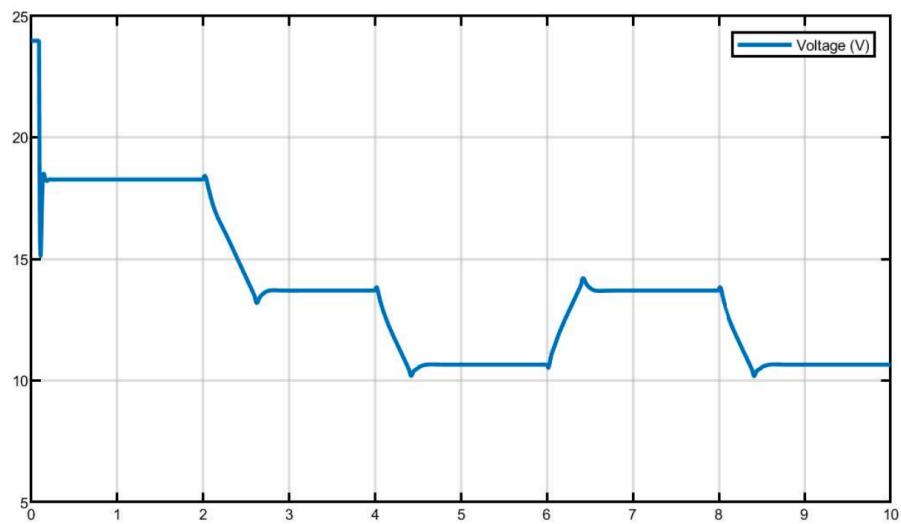


Figure 36: Input voltage over time

In Fig. 34 we can notice the behaviour of the position when we want the ball to follow the reference square wave In Fig. 35 the behaviour of the needed current is shown and Fig. 36 treats the modulation of the input (V).

Before concluding the section dedicated to the Proportional-Integral-Velocity controller and giving space to the Proportional-Integral-Derivative controller, the effect of the Feed-Forward term is interesting to consider. Using anticipated information, a feed-forward term in a control system can bring some benefits which include faster response times, lower control error rates, better system stability, and non-linearity and delay compensation. Some of these effects are shown in Fig. 36.

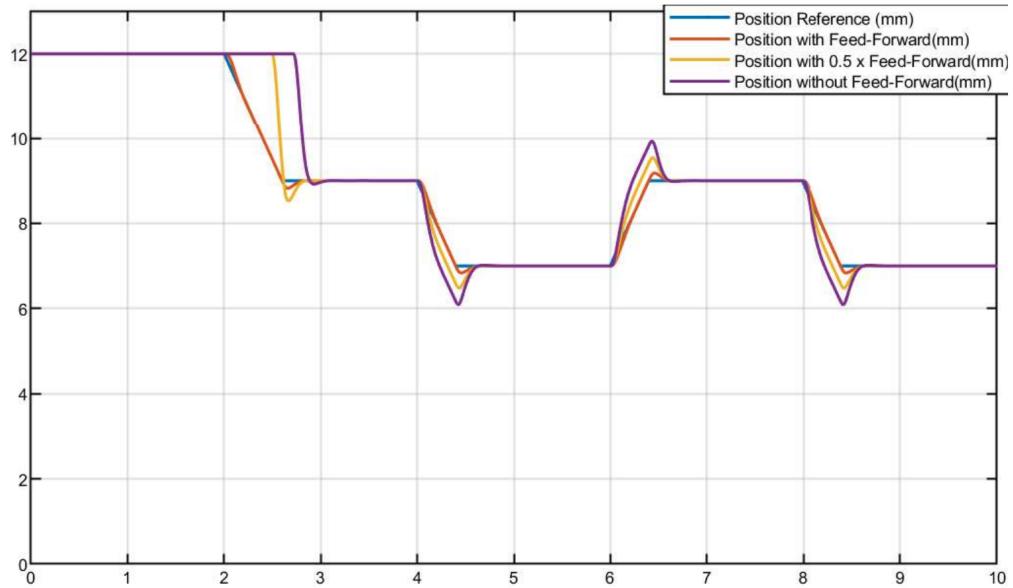


Figure 37: Effect of Feed-Forward(FF) term in a PI+PIV control

Proportional-Integral-Derivative (PID) controller

A simpler approach to a PID was considered as well, where we followed the same steps undergone to implement the electrical PI controller. Now a PID is needed instead since contrary to before we have an unstable pole in the system (as seen in 1.3), indeed if we plot the root locus of the mechanical system we might notice how it can't be brought to absolute stability. to solve this problem a pole zero

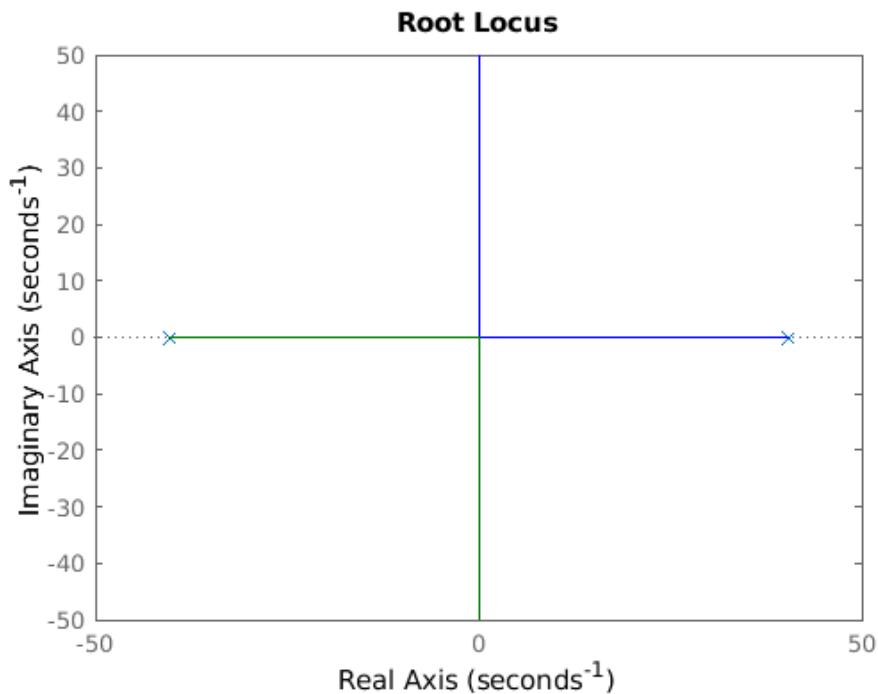


Figure 38: Root Locus of the mechanical system

cancellation is needed so to move the stable pole further left, or in other words make it a faster pole, so to obtain a stabilizable system using Root Locus, or as in this case the proportional part of the PID. A PID allows doing so since it has a zero in its formula (see 39). Alternatively a separate Transfer function could be used to cancel the stable pole and substitute it, then a PI control could be implemented. The integral part is needed because this system manifests a steady state error alternatively.

As for the electrical PI we implemented an anti-windup to avoid a demand of current that is too high using the camping method. A rate limited is added to the input to avoid a quick change in the reference, that would destabilize the system due to it being highly nonlinear, where the slope is between $\pm 10^{-3}$. The system was

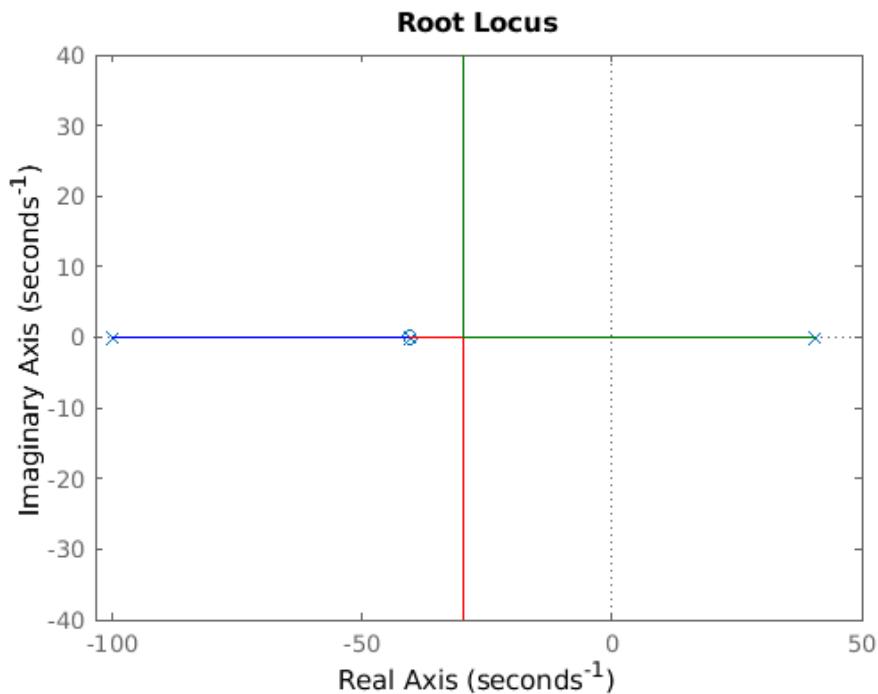


Figure 39: Root locus with the new pole

linearized in the ball position $x_2 = 12 \text{ mm}$. The computed gains are the following: $K_p = -301$, $K_i = -3270.8$, $K_d = -6.19$ with a saturation current of $\pm 2 \text{ A}$. Below the step response to 8 mm and the final control scheme are reported.

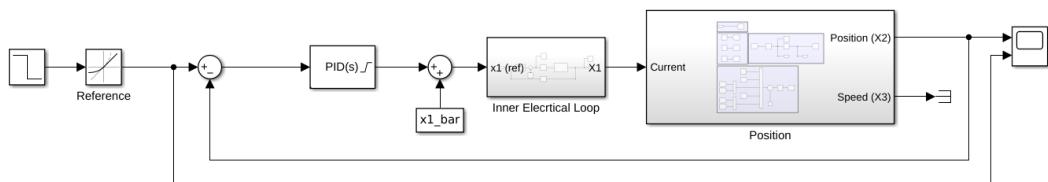


Figure 40: PID control scheme

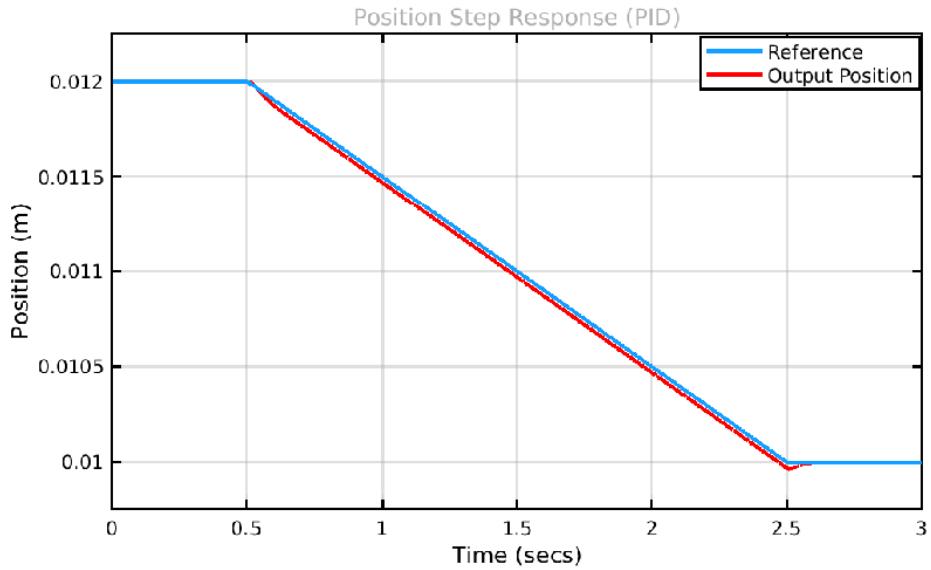


Figure 41: PID step response

State Feedback through Pole Placement

While with PID it is hard to stabilize the system due to the presence of the unstable pole, using Pole Placement this becomes much easier since it allows to change the position of the poles more freely, but there is a limit to how far the poles can be changed imposed by the saturation of the controller, so the control law can't be too aggressive.

To avoid using two different control strategies to close the mechanical loop so to follow the reference the system was enlarged with the error. This yielded such equations for the mechanical system:

$$A_{mech\ new} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{K_m \bar{x}_1^3}{M_b \bar{x}_2^3} & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (4.9)$$

$$B_{mech\ new} = \begin{bmatrix} -\frac{K_m \bar{x}_1}{M_b \bar{x}_2^2} \\ 0 \\ 0 \end{bmatrix} \quad (4.10)$$

$$C_{mech\ new} = [1 \ 0 \ 0] \quad (4.11)$$

$$D_{mech\ new} = 0 \quad (4.12)$$

To this system these poles where chosen:

$$poles = [-25 \ -20 \ -15],$$

and to do so such gains resulted for the feedback control law:

$$K_{pp} = [-173.7084 \ -3.7091 \ 463.6346].$$

The system was linearized around $x_2 = 12 * 10^{-3}$ to compute such values and a rate limiter was added as for the PID control. Due to the nature of this kind of control all the state values are needed in order to apply the feedback. To obtain them an observer is needed, either a Luemburg Observer or a Kalman Filter could be used (depending on how much disturbances concern our system). In the case of the Luemburg Observer the poles of the observer, that behaves as a system, must be set much higher than the process, so we had to follow a similar procedure as for the Pole Placement control but for the following virtual system that outputs all the states:

$$A_{obs} = A_{mech} - K_{obs}C_{mech} \quad (4.13)$$

$$B_{obs} = [B_{mech} \ -K_{obs}D_{mech} \ K_{obs}] \quad (4.14)$$

$$C_{obs} = I_{2 \times 2} \quad (4.15)$$

$$D_{obs} = 0_{2 \times 2} \quad (4.16)$$

The desired poles are the following:

$$poles_{observer} = [-550 \ -500],$$

hence the gains of the observer:

$$K_{obs} = \begin{bmatrix} 0.0105 \\ 2.7664 \end{bmatrix} \times 10^5$$

For the Kalman Filter instead the data obtained from 2.3 was used to solve the Riccati Equation and obtain the desired gain. Considering the covariance function:

$$V = \begin{bmatrix} \tilde{Q} & Z \\ Z^T & \tilde{R} \end{bmatrix}$$

\tilde{R} was obtained from the noise characterization while \tilde{Q} was tuned offline with an input/output dataset so to obtain an output that tracks accurately the one from the dataset, the value picked is: $\tilde{Q} = 1.3$. Finally, Z is a matrix of zeros of the proper dimension.

The final scheme and step response are reported below:

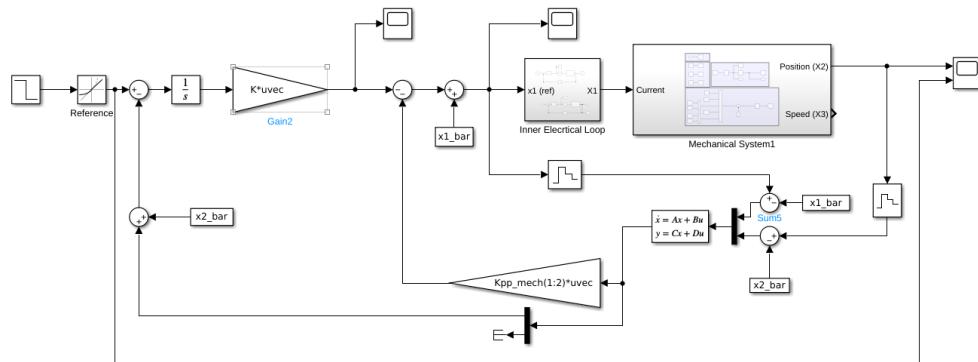


Figure 42: Pole Placement final control scheme

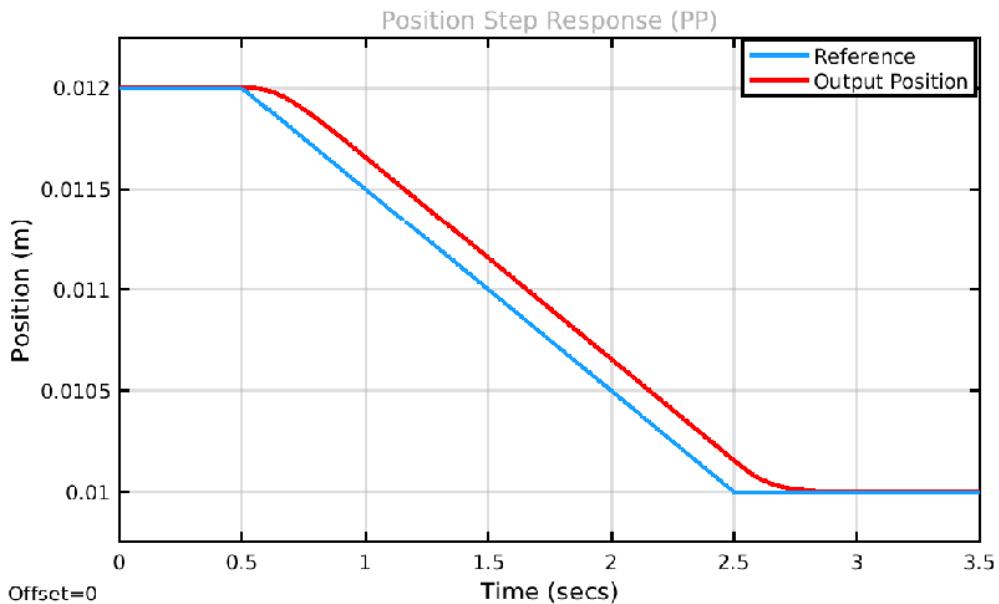


Figure 43: Pole Placement Step Response

State Feedback through LQG (Optimal Control)

LQ is a type of optimal control that takes the form of a state feedback control law, where the gains of the state feedback are obtained by solving the Riccati Equation. The final scheme is the exact same obtained from Pole Placement, but the approach is completely different. Indeed, given this quadratic cost function:

$$J(x_0, u(\cdot), 0) = \int_0^{T_s} (x^T(\tau) Q x(\tau) + u^T(\tau) R u(\tau)) d\tau + x^T(T) S x(T) \quad (4.17)$$

The aim is to tune the Q, R and S matrices in such way to obtain the desired performance from the system. By using the lqg function on Matlab it's possible to solve the Riccati equation and obtain the gains of the control. As for Pole Placement the system was expanded to account for the steady-state error, moreover an observer is needed for the same reasons of Pole Placement, making this control an LQG one by all means, since a Kalman Filter is used in it, this is not good news since it means that the gain and phase margins guaranteed from LQ_∞ are lost. After some iterations the chosen matrices values are the following:

$$Q = \begin{bmatrix} 50000 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 50000000 \end{bmatrix} \quad R = 1 \quad S = 0$$

As for other controls, a rate limiter was added to avoid a quick change of the reference.

Below the step response is reported. Below the step response is reported.

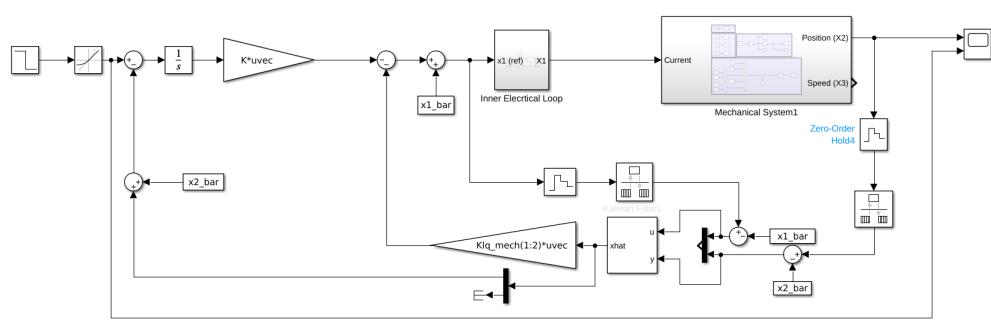


Figure 44: LQG final control scheme

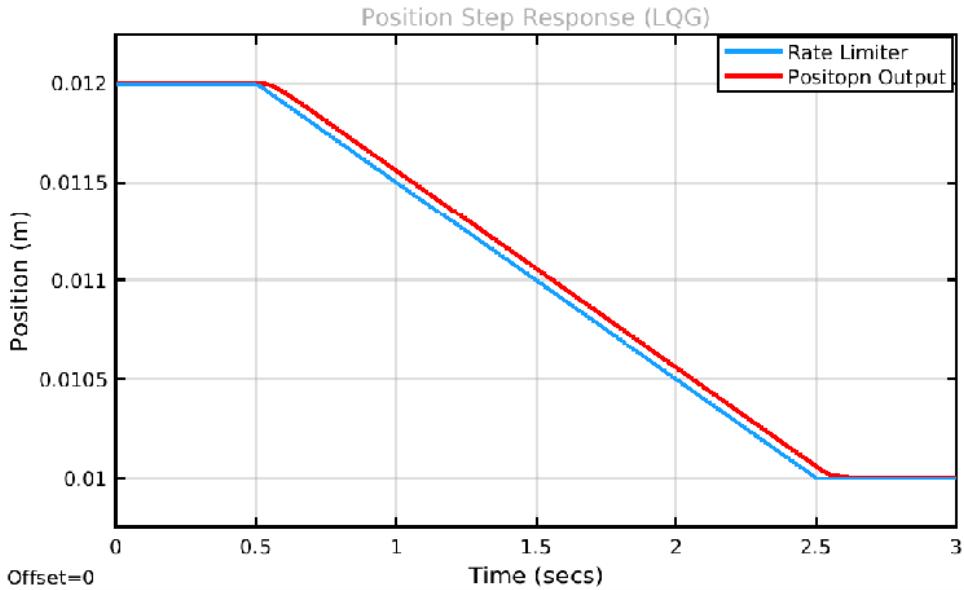


Figure 45: LQG step response

4.2 Variable Structure Control (VSC)/ Sliding Mode Control

VSC[2] is our first attempt of a nonlinear control strategy, and it is known for its robustness against external disturbances, and that makes it a good choice for our system since it's affected by many uncertainties in the model.

It's the first control strategy where the system is considered as a whole since it allows to easily limit the input values. The idea behind this control is the following: first the system must be written in the Controllable Canonical Form:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = [b_0 \ b_1 \ \cdots \ b_n]$$

$$D = 0$$

That is easy to extrapolate knowing the transfer function of the system 1.3, where we linearized around $x_2 = 12 \text{ mm}$, then we introduce a switching function: $s(x) = \beta_{n-1}x_1 + \beta_{n-2}x_2 + \cdots + \beta_1x_{n-1} + x_n - \bar{w}$, that in steady state is equivalent to expressing x_n w.r.t. the other variables: $x_n = \beta_{n-1}x_1 + \beta_{n-2}x_2 + \cdots + \beta_1x_{n-1} - \bar{w}$. This yields a system constrained on a surface, whose equation determines the new switching system, in our case we chose the following surface:

$$\chi(\lambda) = \lambda^{n-1} + \beta_1\lambda^{n-2} + \beta_2 + \cdots + \beta_{n-1} = \lambda^2 + 10\lambda + 25 \quad (4.18)$$

It has eigenvalues in $\lambda_1 = \lambda_2 = 5$. Now \bar{w} can be easily computed considering the system in steady state, since all the states will be null except the first one, so we get $\bar{w} = \frac{b_n}{\beta_{n-1}}$. So far we got this system:

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \beta_2x_1 - \beta_1x_2 + \frac{b_n}{\beta_{n-1}} \end{bmatrix} \quad (4.19)$$

Now we need to use the standard s dynamic equation, so we get the input expression:

$$\dot{s} = -q \operatorname{sgn}(s) - r g(s) \quad (4.20)$$

Where we set $r = 0$ for simplicity and $q = 6$, a value that we noticed avoided an input request higher than the $u = 24 \text{ V}$ we had.

Finally, after some computations, that won't be reported for brevity, we can express the input as:

$$u = -(\underline{\beta}^T A \underline{x} + q \operatorname{sgn}(\underline{\beta}^T \underline{x} - \bar{w} y_0)) \quad (4.21)$$

Manipulating this equation we finally get the control scheme reported below where we can notice that the $\operatorname{sgn}(\cdot)$ function was replaced with a $\tanh(\cdot)$ so to have a continuous function in the scheme, something that greatly speeds up the simulations since Simulink doesn't handle discontinuities that well, moreover in the real system it reduces chattering, which is a well known problem affecting sliding mode control laws.

Below the step response and control scheme are reported:

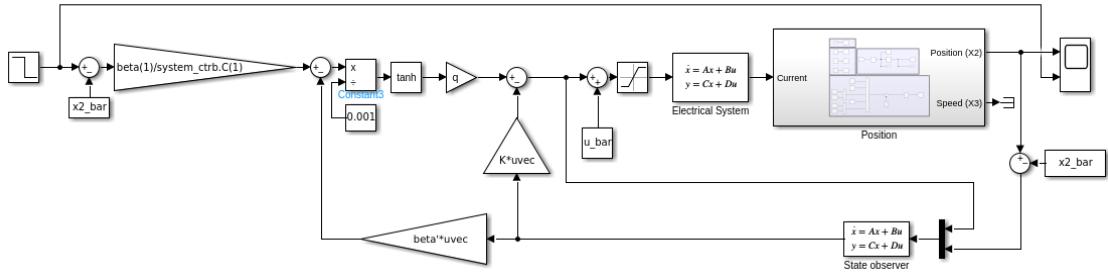


Figure 46: VSC control scheme

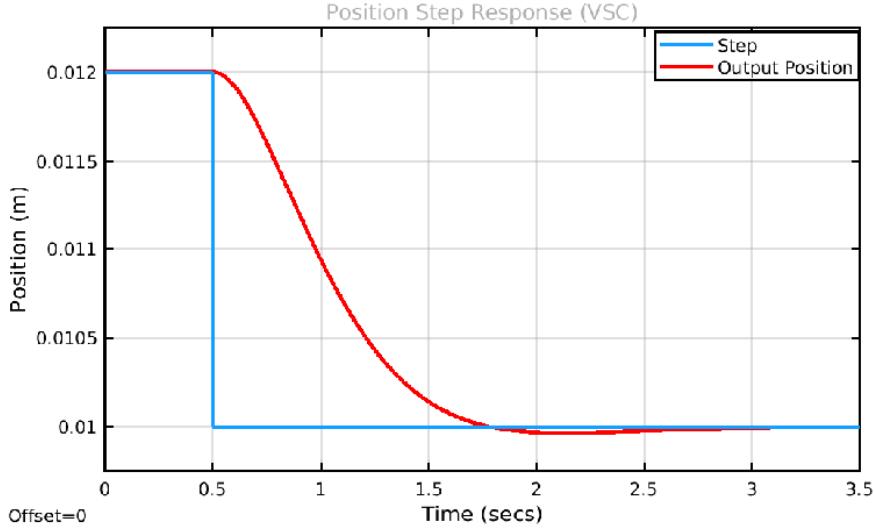


Figure 47: VSC step response

4.3 Feedback Linearization

Feedback Linearization is a normal form that allows the implementation of nonlinear control strategies starting from linear once. Generally speaking that equates in a better response of the system to the reference even far from the equilibrium where we've stabilized the system, in our case it also allows to remove the rate limiter that was added to the linear control strategies already treated.

To implement it first an equation called diffeomorphism was computed using the standard procedure of feedback linearization, that is needed both for the computation of the control law and to check whether it's fully feedback linearizable or not (the relative degree must equal the number of states). To do so we need to check if the relative degree is the same as the number of states. In our case that holds and indeed we got such equations

$$\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \ddot{\dot{y}} \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ -\frac{K_m x_1^2}{2M_b x_2^2} + g \\ \frac{K_m}{M_b} \left(\frac{x_1^2 x_3}{x_2^3} + \frac{x_1^2 R_{tot}}{x_2^2 L_c} - \frac{x_1 u}{x_2^2 L_c} \right) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ v \end{bmatrix} \quad (4.22)$$

We can see how the condition of full feedback linearization holds until $x_1 = 0$, but that's not a problem since we never get close to such value due to the fact that the ball is limited between 12 mm and 0 m.

Once this is done we can write the topologically equivalent system and hide the last state derivative equaling it to a fictitious input v

$$\begin{bmatrix} \dot{z}_1 \\ \ddot{z}_2 \\ \ddot{\dot{z}}_3 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ v \end{bmatrix} \quad (4.23)$$

From which we can extrapolate the equation of the final control law

$$u = -\frac{L_c x_2^3}{x_1} \left(v \frac{M_b}{K_m} - \frac{x_1^2 x_3}{x_2^3} - \frac{x_1^2 R_{tot}}{x_2^2 L_c} \right) \quad (4.24)$$

At this point it's all about controlling a system that is equivalent o to 3 integrators in series to get the input v , then through the equation we get the nonlinear control law.

Pole Placement in Feedback linearization

The first control strategy implemented is Pole Placement since it's the standard and easiest way to control the topologically conjugate system.

The system was expanded as in the linear control case (see state feedback through pole placement) and a similar procedure was followed, in this case the poles chosen are:

$$poles = [-45 - 40 - 35 - 30],$$

obtained with such gains:

$$K_{pp} = [0.20620.00840.0001 - 1.89] \times 10^6.$$

Still we needed an observer, both for the control strategy and for the nonlinear input computation, so a Luemburg observer was used. Since the nonlinear input computation needs the original system's states and the observer was designed for the new system though, ad diffeomorphism is needed, indeed the output of the observer goes through it and outputs the original system states, alternatively a second Luemburg observer could have been added changing its inputs, so it outputs the original system states.

This is the only control that requires the microcontroller and the sampling to work at 1000 Hz, while the other controls work at 500 Hz with no problem.

Below the final scheme and step response are reported.

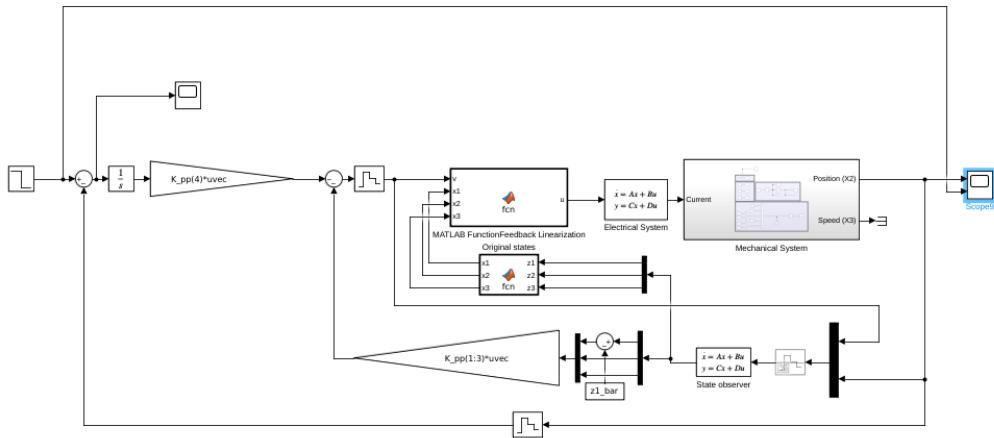


Figure 48: Feedback Linearization + Pole Placement control scheme

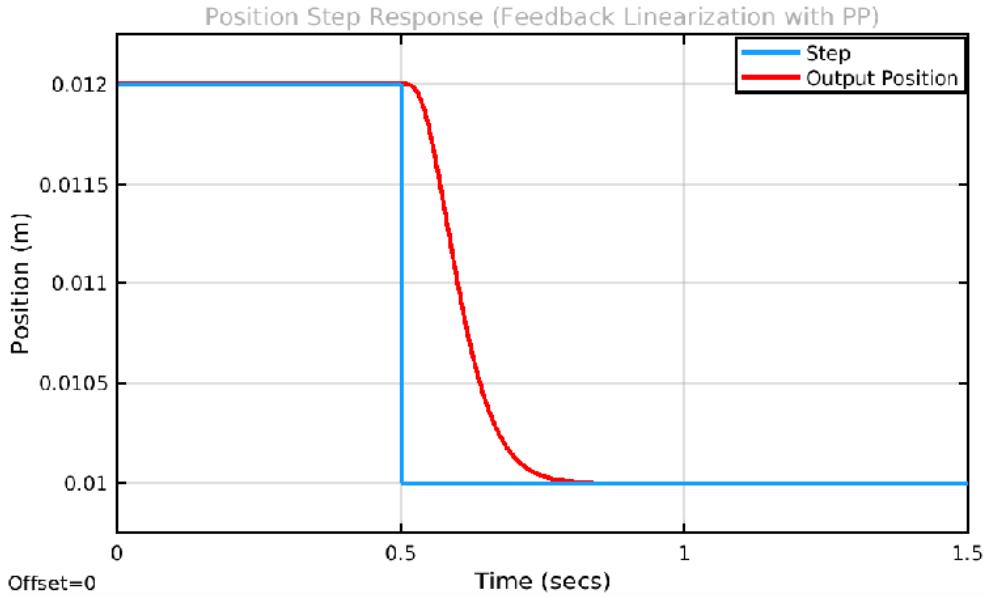


Figure 49: Feedback Linearization + Pole Placement step response

VSC in Feedback linearization

Applying VSC with Feedback linearization allows to obtain a control strategy that is both robust, as stated before about VSC above, and avoid linearizing. The exact same steps were taken as for the linear case except $q = 100$ and for the different system. Again a Luemburg observer is used exactly as in the pole Placement case treated above. Below the final scheme and step response are reported.

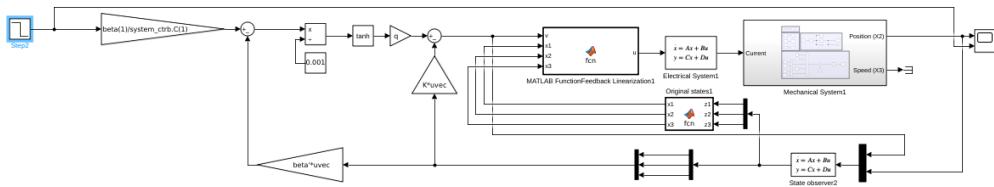


Figure 50: Feedback Linearization + VSC Scheme

The first attempt to implement a controller was by using a cascade control. PI controller for the electrical system internal loop and PID controller for the electro-mechanical system external loop.

The electrical model is already stable, but we used the PID tuner to tune the control to achieve an accurate and fast control. Considering the step response of

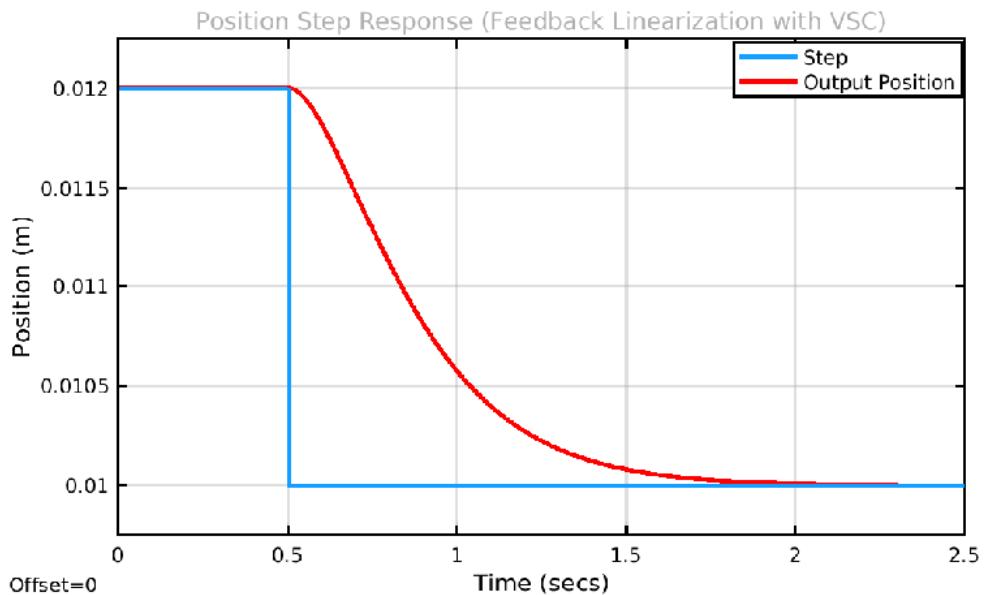


Figure 51: Feedback Linearization + VSC Step Response

the ideal system, without saturation, we designed the response to be fast and with overshoot ≤ 1.5 and selected the computed control parameters k_p & k_i .

With the computed controller, when we consider the complete model with also the voltage saturation we achieve the following step response:

4.4 Extended Kalman Filter

The Extended Kalman Filter (EKF)[3] implementation in the Maglev system is the main topic of this chapter. The Extended Kalman Filter is a popular estimating method for deriving precise state estimations of a dynamic system from observations impacted by noise. The EKF has many benefits when applied to the Maglev system, including the ability to estimate state variables precisely, which is essential for the system's best performance. As shown in this chapter, the EKF has been used to estimate two of the three states of our model: the position and the velocity. This chapter examines the effects of Extended Kalman Filter's use when applied to the Proportional-Integral (PI) control for the current and the Proportional-Integral-Velocity (PIV) controller for the position.

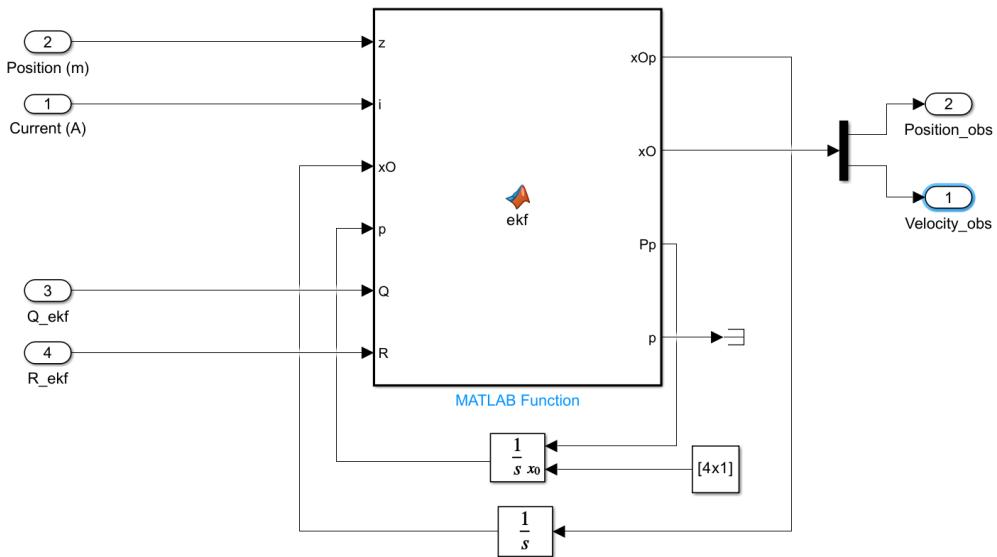


Figure 52: Extended Kalman Filter structure

The overall structure of the Proportional-Integral-Velocity controller using the estimation of the velocity for the Velocity and the estimated Position for the feedback is shown in Fig. 55.

After the identification of the model, we know how the state variables change over time and since the dynamics of the system are nonlinear, the state model is represented using nonlinear functions. The next step is to establish the observation model, which shows how the system state variables relate to the actual measurements provided by the current and position sensors. In particular the state equations are used to compute the system's future state during the prediction time using the system's current state and known inputs. This prediction takes into account the covariance of the prediction error, which indicates the uncertainty around the state

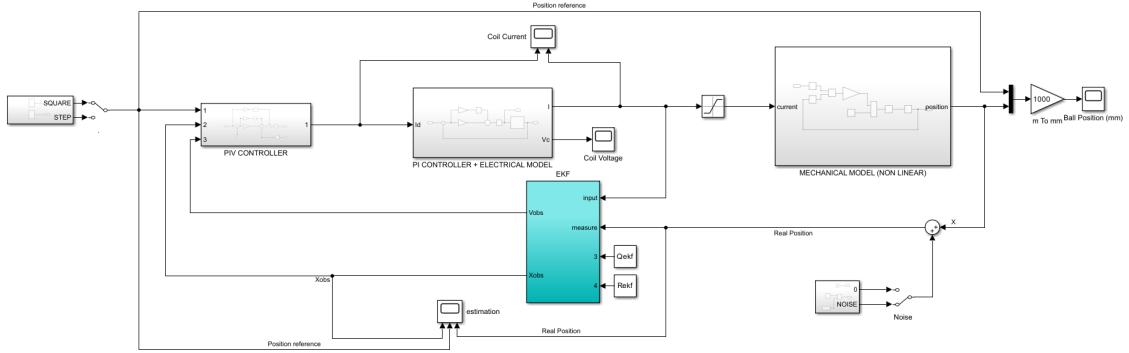


Figure 53: comparison between Estimated position and Real Position

estimation. The EKF integrates the actual measurements with the state predictions to produce a revised estimate of the system state. The accuracy of the corrected estimate is assessed at this point by computing the error covariance matrix. Then, the EKF updates the state estimations and their covariances, giving a more accurate picture of the real state of the Maglev system. As results, we can get a reliable state variable estimation even in the presence of measurement noise, like in our case.

To sum up, the basic idea is to minimize with respect to the gain $L(t)$ the covariance of the estimation error. The solution of this problem is:

$$L(t) = \tilde{P}(t)C^T\tilde{R}^{-1}$$

where $\tilde{P}(t)$ is the solution of the Riccati equation

$$\dot{\tilde{P}}(t) = A\tilde{P}(t) + \tilde{P}(t)A^T + \tilde{Q} - \tilde{P}(t)C^T - \tilde{P}(t)C^T\tilde{R}^{-1}C\tilde{P}(t)$$

with initial condition $\tilde{P}(0) = \tilde{P}_0$. In this particular case covariance matrices are set as:

$$Rekf = 0.027;$$

$$Qekf = 100.$$

Rekf represents the measurement noise, generally it is provided by sensors' manufacturers but here it has been computed in section 2.3.

Qekf is not provided, it can be considered as a measure of goodness for the model and as a design parameter to obtain faster or slower state.

Once the Jacobian of the dynamics of the states (position and velocity) has been computed, using the previously mentioned values for Rekf and Qekf, an estimation for the position and for the velocity has been found.

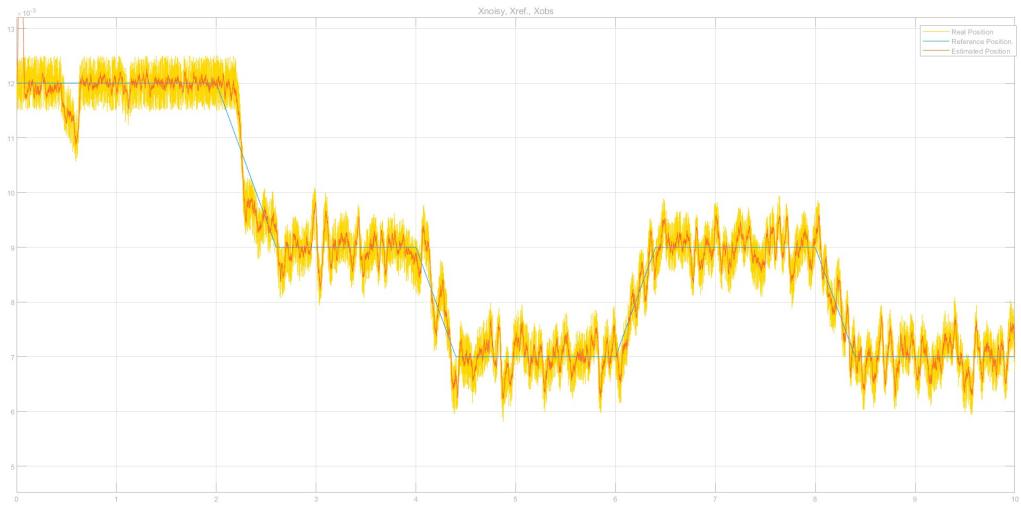


Figure 54: comparison between Estimated position and Real Position

As a results, an Extended Kalman Filter is applied to the previously discussed Proportional-Integral (PI) + Proportional-Integral-Velocity (PIV) + FeedForward (FF) with a rate-limited square wave around the equilibrium position as reference.

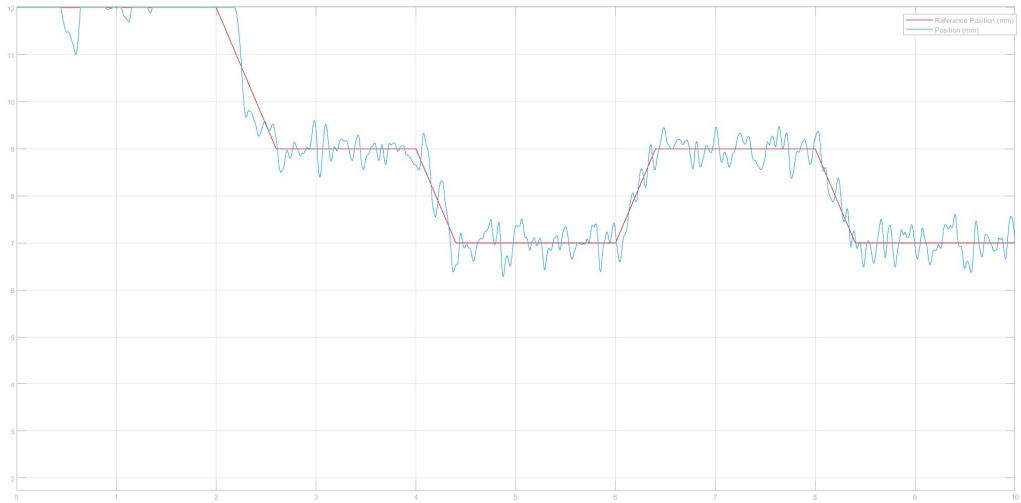


Figure 55: Proportional-Integral-Velocity controller using an EKF

As explained in the previous chapter, the position measure is highly noisy. For this reason and because of the need of a velocity estimation for the control, the use of an Extended Kalman Filter for the PIV control is crucial.

5 Conclusions

In this project, we first identified an accurate physical model representing the system's dynamics, deriving its space state representation and studying the properties of the resulting linearized model. Furthermore, we discussed the main characteristics of the two available sensors, highlighting their main critical problems among which the low position sensor's repeatability and the current sensor's gain. For the control implementation, discussed later, we decided to solve these non idealities by computing the position curve before running the algorithm in order to minimize the conversion error and considering the sensor gain in the current feedback loop. Notice that we also tried to cover the MAGLEV with a box in order to keep the same light conditions. Even though it successfully allowed us to work in a repeatable environment, we were not able to visualize the experiment and, because of the low convective heat transfer, it led to a further increase of temperature.

Furthermore, we performed a complete identification of the system's parameters, applying simple methods to more complex ones. Consequently, we introduced the coil's temperature variation problem and the relative resistance increase. As we previously discussed, we performed a Model Matching identification of the electrical parameters at every control's execution in order to be sure that the model was reliable enough for every temperature condition. Regarding the K_m , we estimated different values at different ball's heights that are consistent with the value provided by Quanser. In fact, in our control models, we considered the value of the electromagnetic constant at the correspondent equilibrium position. Notice that, despite we are confident with this identified parameter, we could have improved its estimation with a working control in all the travel range.

Having an accurate model with a reduced uncertainty as possible, we were able to design several control strategies. Unfortunately no matter the control strategy applied the system was never actually controlled, indeed we realized that the control was always too slow, as if there is something not keeping up with the system's speed. This could be attributed to the slow dynamics of the sensor, that makes it hard for the control law to react in time. We believe that there could also be something else not working at the expected frequency, making the control not reactive enough, so even if the board is supposed to work around 500 Hz and 1000 Hz and even if our controls are meant to work at around 500 Hz, it seems as if the board is still not keeping up. This is clearly visible with VSC, a control that has no integrator: when it's loaded into the board, what we can notice is that the ball gets thrown up, then dropped down, but the control is never fast enough to reduce the input whilst the ball is in midair.

References

- [1] Jacob Apkarian, Quanser Hervé Lacheray, and Quanser Michel Lévis. Student workbook magnetic levitation experiment for matlab /simulink users standardized for abet * evaluation criteria.
- [2] Dan Cho and Yoshifumi Kato. Sliding mode and classical content magnetic levitation system. *IEEE Control Systems*, 13:42–48, 1993.
- [3] Hamid Reza Karimi. Robotics mechatronics state observer and state estimation.
- [4] R. Morales and Hebertt Sira-Ramirez. Trajectory tracking for the magnetic ball levitation system via exact feedforward linearisation and gpi control. *International Journal of Control*, 83:1155–1166, 06 2010.
- [5] Robotics and Automation WSEAS International Conference on Signal Processing. *Recent advances in signal processing, robotics and automation : Proceedings of the 9th WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA'10) : University of Cambridge, UK, Februray 20-22, 2010*. WSEAS, 2009.

List of Figures

1	Overview of the system	5
2	System's block scheme	10
3	Poles of the three transfer functions of the system	11
4	Multimeter setup	14
5	Conversion curve identification procedure	15
6	Polynomial second order curve	16
7	Theoretical Probability Density Function of gaussian current noise	18
8	Current noise spectrum	18
9	R second experiment	21
10	R third experiment	22
11	L analysis from 24V step input	24
12	L analysis from 1V step input	24
13	L analysis from 2V step input	25
14	Hot v.s. Cold circuit	26
15	Real model matching scheme with varying R_{tot} and L_c	27
16	Real model matching in cold circuit condition	28
17	EKF R_{tot} and L_c estimation	31
18	Theoretical Vs Measured i_c	33
19	Km Experimental Identification	35
20	System controlled in feedback	38
21	Inner current loop with a PI-controller	40
22	Step response of the closed-loop current system	41
23	Bode plots of the closed-loop current system	42
24	Electrical PI scheme	43
25	Electrical PI step response	44
26	PIV + FF control scheme	45
27	Step response of the overall closed-loop system	48
28	Bode plots of the closed-loop system controlled with PI and PIV	49
29	Constant position reference	49
30	Position tracking results over time	50
31	Current tracking results over time	50
32	Input voltage over time	51
33	Constant position reference	51
34	Position tracking results over time	52
35	Current tracking results over time	52
36	Input voltage over time	53
37	Effect of Feed-Forward(FF) term in a PI+PIV control	54
38	Root Locus of the mechanical system	55
39	Root locus with the new pole	56

40	PID control scheme	56
41	PID step response	57
42	Pole Placement final control scheme	59
43	Pole Placement Step Response	59
44	LQG final control scheme	60
45	LQG step response	61
46	VSC control scheme	63
47	VSC step response	63
48	Feedback Linearization + Pole Placement control scheme	65
49	Feedback Linearization + Pole Placement step response	66
50	Feedback Linearization + VSC Scheme	66
51	Feedback Linearization + VSC Step Response	67
52	Extended Kalman Filter structure	68
53	comparison between Estimated position and Real Position	69
54	comparison between Estimated position and Real Position	70
55	Proportional-Integral-Velocity controller using an EKF	70

List of Tables

1	Nominal system's parameters	11
2	Poliscope Vs Multimeter measurements	15
3	Mean values and variances of the two sensors	17
4	Steady state experiments	20
5	L values from transient analysis	25
6	Parameter identification with systemID	29
7	Parameter identification as result of a minimization problem	33
8	K_m data	34