

POLITECNICO DI MILANO

Master of Science in Automation and Control Engineering

Report of

Magnetic Levitation

Group components:

Antonello Lagalante (10938409)
Maria Laura D'Amato (10668205)
Giorgio Maria Ercoli (10678474)
Paolo Fiocca (10692271)

Contents

1	Introduction	1
1.1	System Description	1
1.2	Project Structure	2
2	System Model	3
2.1	Non-linear Mathematical Model	3
2.1.1	Electrical Subsystem	3
2.1.2	Mechanical Subsystem	3
2.1.3	Complete Non-linear Mathematical Model	4
2.2	Parameters Identification	5
2.2.1	Relationship PWM - Voltage	5
2.2.2	Inductance characteristics $L(x)$	6
2.2.3	Magnetic force constant K_{em}	7
2.2.4	Other parameters	8
2.3	System Validation	9
2.4	SISO Single Magnet Linearized System	11
3	Noise Characterization	15
3.1	Properties of the noise	15
3.2	Noise filtering	18
4	Control Schemes	21
4.1	PID	21
4.1.1	Single Loop	21
4.1.2	Cascade Control	24
4.2	Pole-Placement Control	27
4.3	Linear Quadratic Regulator (LQR)	31
5	State Observers	34
5.1	Luenberger's Observer	34
5.2	Extended Kalman Filter (EKF)	36
6	Results Analysis	39
6.1	Step Response	39
6.2	Disturbance Rejection	40
A	MATLAB Codes	42
A.1	RL Identification	42
A.2	Extended Kalman Filter (EKF)	43

Chapter 1

Introduction

1.1 System Description

Magnetic levitation, commonly referred to as maglev, is a technology that uses magnetic fields to lift objects without direct contact with a solid surface. This form of levitation offers numerous advantages, including frictionless motion, reduced maintenance costs, and enhanced speed capabilities.

One of the most notable applications of magnetic levitation is in maglev trains, which offer a revolutionary approach to high-speed rail transport. These trains can achieve remarkable speeds, well above those of traditional rail systems, due to the elimination of friction between the train and the tracks. Beyond transportation, magnetic levitation has significant implications in other areas such as manufacturing, where it is used for contactless handling of materials, and in scientific research, where it enables precision positioning of instruments and components.

Magnetic levitation relies on the principles of electromagnetism, in particular our system uses the electromagnetic suspension (EMS) method to lift the object. EMS systems use electromagnets to generate a magnetic field that attracts the object towards the electromagnet, maintaining a stable levitation state through control feedback.

Our laboratory system (Inteco Magnetic Levitation) is a system aimed to keep a ferromagnetic ball levitate at a desired height. It works applying voltage to two solenoids, which generate a magnetic field that makes the ball soaring.

The system is fully integrated with MATLAB/Simulink and operates in the real-time, loading the controller in a RT-DAC/USB2, a multifunction analog and digital I/O board dedicated to real-time data acquisition and control.

Hardware:

- Two electromagnets
- Ferromagnetic sphere
- Position sensor
- Current sensor
- Power interface
- RTDAC/USB2 I/O board

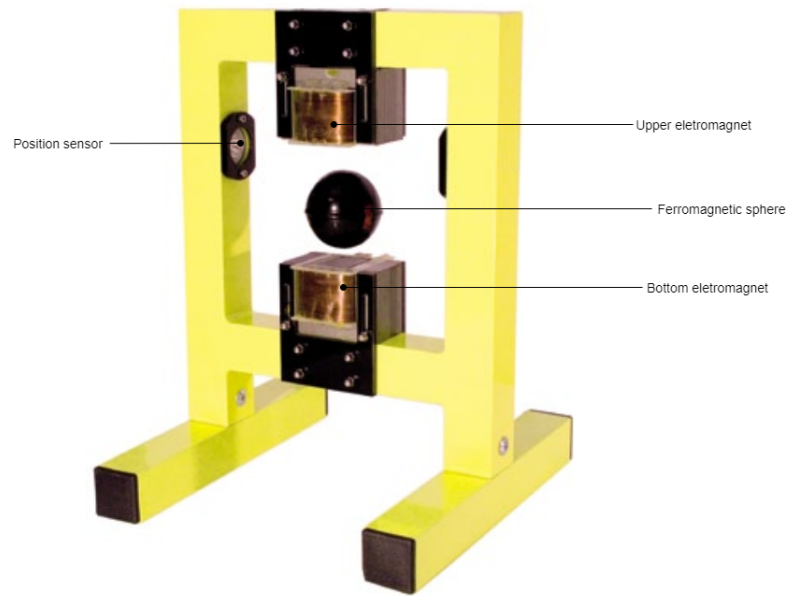


Figure 1.1: Inteco Maglev.

1.2 Project Structure

The path followed in the project development is schematized in Figure 1.2:

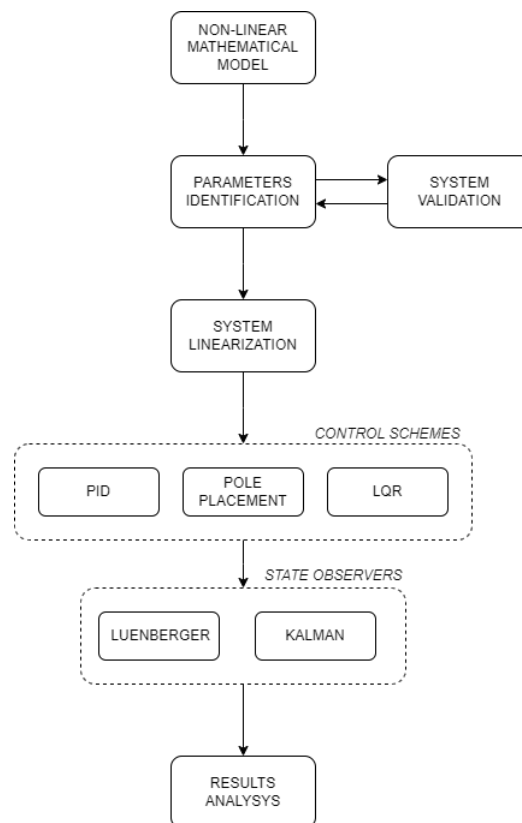


Figure 1.2: Path followed in the project development.

Chapter 2

System Model

2.1 Non-linear Mathematical Model

The mathematical model of the magnetic levitation system includes two parts: mechanical model and electrical model.

2.1.1 Electrical Subsystem

For the modelling of the electrical part of the levitator we can apply the Kirchhoff's second law to the electrical circuit of the system, so it follows that:

$$v(t) = Ri(t) + L(x) \frac{di(t)}{dt} \quad (2.1)$$

where R is the total resistance of the circuit, $L(x)$ is its inductance and v is the instantaneous voltage across the magnet coil which controls the excitation current i . Since the two magnets are at a sufficiently large distance, we can consider the mutual inductance between the two coils to be zero.

The inductance $L(x)$ of the magnets is not constant, but varies as a non-linear function of the ball position x . In the literature there are numerous expressions with which we can mathematically model the inductance as a function of x . A typical approximation is to assume that this inductance is inversely proportional to the ball's position x :

$$L(x) = L_1 + \frac{L_0 x_0}{x} \quad (2.2)$$

2.1.2 Mechanical Subsystem

The magnetic energy of the system is a function of the coil current i and the ball position x :

$$W(i, x) = \frac{1}{2} L(x) i^2 \quad (2.3)$$

The magnetic force generated by a single magnet and acting on the sphere is given by:

$$F_{em} = -\frac{\partial W}{\partial x} = -\frac{1}{2} \frac{dL(x)}{dx} i^2 = -\frac{L_0 x_0}{2} \left(\frac{i^2}{x^2} \right) = -K_{em} \left(\frac{i^2}{x^2} \right) \quad (2.4)$$

2.1.3 Complete Non-linear Mathematical Model

Since we are determining a control-oriented model, we can make a change of notation and consider the following states for our system:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ i_{sup} \\ i_{inf} \end{bmatrix} \quad (2.5)$$

Where \mathbf{x} and \mathbf{v} are the position and velocity of the ball and i_{sup} and i_{inf} are the currents of the two (upper and lower) magnets. The forces acting on the ball can be schematised as in the following figure:

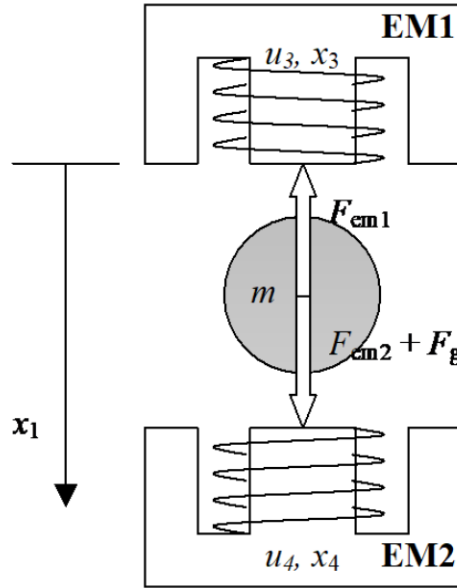


Figure 2.1: Scheme of the Magnetic Levitation System.

The physical equilibrium happens when:

$$F_{em1} = F_{em2} + F_g \quad (2.6)$$

Where F_{em1} is the electromagnetic force due to upper magnet, F_{em2} the electromagnetic force due to bottom magnet and F_g the gravitational force due to sphere weight. Using Newton's second law, we now obtain the differential equation governing the ball position as:

$$m \frac{d^2 x}{dt^2} = mg - K_{sup} \left(\frac{i_{sup}}{x} \right)^2 + K_{inf} \left(\frac{i_{inf}}{x_d - x} \right)^2 \quad (2.7)$$

With laboratory instrumentation, we cannot directly act on the current, but the Pulse Width Modulation (PWM) technique is used as a control signal. For this reason, equation 2.1 must be slightly modified to take into account the linear relationship between duty cycle $u(t)$ and voltage $v(t)$:

$$\mathbf{v}(t) = \mathbf{k}u(t) \Rightarrow \mathbf{k}u(t) = \mathbf{R}i(t) + \mathbf{L}(x)\frac{di(t)}{dt} \quad (2.8)$$

If we combine the equations (2.8), (2.7) and considering the states defined in (2.5), we can write the following non-linear model:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g - \frac{K_{sup}}{m} \left(\frac{x_3}{x_1} \right)^2 + \frac{K_{inf}}{m} \left(\frac{x_4}{x_d - x_1} \right)^2 \\ \dot{x}_3 = \frac{1}{L_{sup}(x_1)} (k u_3 - R_{sup} x_3) \\ \dot{x}_4 = \frac{1}{L_{inf}(x_d - x_1)} (k u_4 - R_{inf} x_4) \end{cases} \quad (2.9)$$

Depending on how we manage the two duty cycle of the upper and lower magnet we can obtain two different types of system:

Upper Magnet	Bottom Magnet	Type	Order
$u_3(t) \in [0, 1]$	$u_4(t) = 0$	SISO	3rd order
$u_3(t) \in [0, 1]$	$u_4(t) \in [0, 1]$	MISO	4rd order

Table 2.1: Different types of system that could be obtain with various combination of the control action.

For reasons of time and complexity in the design of the various types of controllers, during the course of the project we will focus only on the first type of system shown in Table (2.1) using in some cases also the bottom magnet as a disturbance.

2.2 Parameters Identification

2.2.1 Relationship PWM - Voltage

The first parameter identified is the relationship between duty cycle and voltage. To do this, we can supply the system with different values of the duty cycle and measure the voltage on the coil terminals with a multimeter. In this way we'll obtain the linear characteristic shown in Figure (2.2).

We can fit the experimental points obtained from the measurements either with a generic line ($\mathbf{y} = \mathbf{m}\mathbf{x} + \mathbf{q}$) or with one passing through the origin ($\mathbf{y} = \mathbf{m}\mathbf{x}$). In the following, the latter expression has been chosen for the sake of simplification in the subsequent calculations.

We can repeat the exact same procedure to determine the same relationship for the lower magnet, however the determined experimental points are exactly the same, which is why we can deduce that the duty cycle - voltage relationship is the same for both magnets.

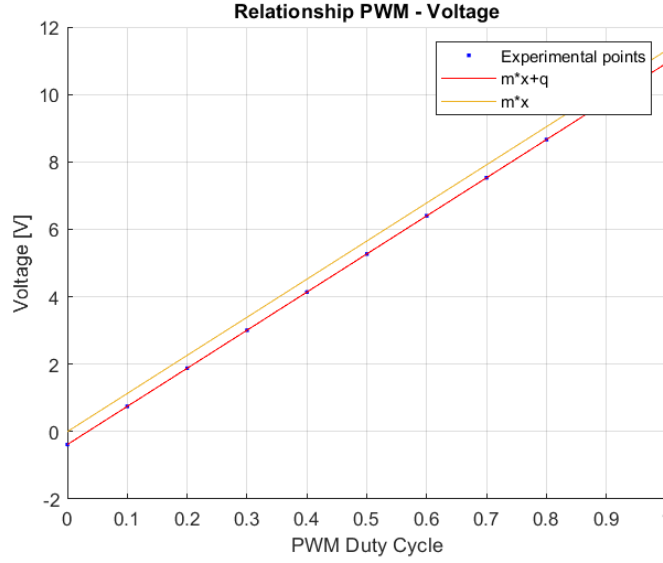


Figure 2.2: Relationship PWM - Voltage.

2.2.2 Inductance characteristics $L(x)$

To identify the trend of inductance L as a function of ball position x we can analyze the step response of the system's internal RL circuit performing the following experiment:

1. Screw the sphere on the support so that it cannot be moved.
2. Feed the system with a step signal.
3. Use the step response to identify the time-constant and the inductance in the current position.

By repeating the experiment first with the upper magnet and then with the lower magnet for various ball positions and maintaining the same value of the duty cycle ($u(t) = 1$), it is possible to determine several experimental points. With the help of the script (A.1) we can find and fit the obtained experimental points with an hyperbolic function with expression (2.2) and the inductance characteristics reported in Figure (2.3) can be obtained.

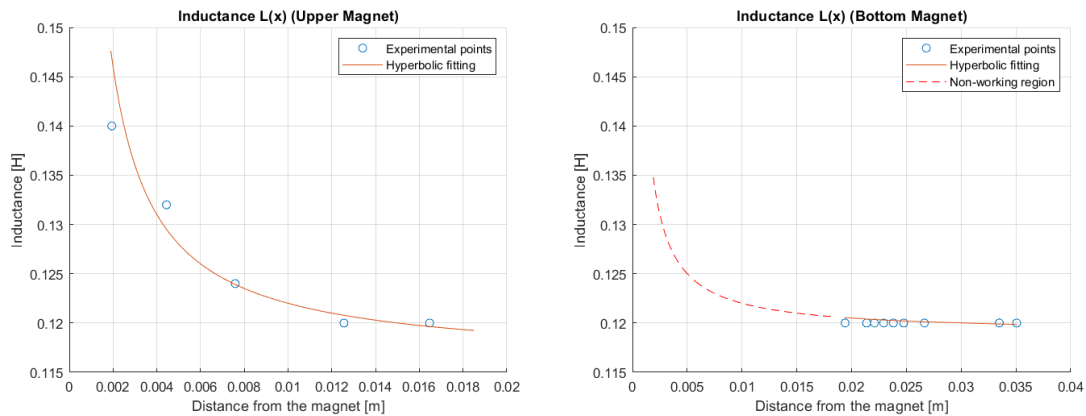


Figure 2.3: Inductance trend of the upper and bottom magnets.

For completeness, in the right figure referred to the bottom magnet, has been reported also the estimated trend of the inductance in the positions in which the ball cannot physically be due to the presence of the base (dashed line). From this graph we can see that the sphere has very little influence on the inductance of the lower magnet, so this means that the settling time of the current flowing in the lower magnet remains almost constant in all the positions assumed by the sphere.

2.2.3 Magnetic force constant K_{em}

To determine the value of the magnetic force constant K_{em} in (2.4), another experiment is needed:

1. Put the sphere on the support.
2. Supply the system with a slow increasing ramp.
3. Save the current and position data at the point slightly before the jump occurs (in this point the magnetic force is equal to the gravity force).

Repeating multiple time the experiment and changing every time the position of the ball, we can determine the value of K_{sup} for each position with the following relationship:

$$mg = K_{sup} \frac{i_{sup}^2}{x^2} \Rightarrow K_{sup} = \frac{mgx^2}{i_{sup}^2} \quad (2.10)$$

Where the values of position and current can be measured through the sensors. For the lower magnet, a similar experiment can be performed exploiting the force balance on the ball, in particular:

$$K_{inf} = \left(K_{sup} \frac{i_{sup}^2}{x^2} - mg \right) \frac{(x_d - x)^2}{i_{inf}^2} \quad (2.11)$$

The Figure (2.4) shows the trends of the magnetic constant of the two magnets, where of course the same consideration applies as for the inductance over the ball operating region limited by the plastic base.

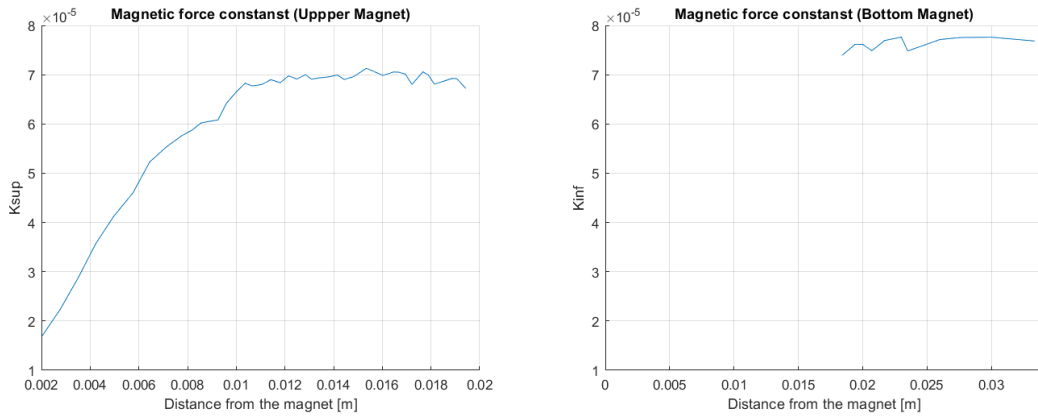


Figure 2.4: Magnetic force constant trend of the upper and bottom magnets.

At this point we can conclude that, although the two magnets are almost identical in terms of electrical and mechanical parameters, the limitation imposed by the plastic base on which we can place the ball as starting position make that the two magnets always work in different positions, and therefore the two models cannot be considered the same.

2.2.4 Other parameters

Finally, all the remaining parameters of the system are shown in Table 2.2, which also shows the remaining mechanical and electrical parameters of the system, obtained using the laboratory instrumentation such as an oscilloscope and digital multimeter.

Parameter	Value	Unit
Ball Mass	0.06157	[Kg]
Ball Diamater	0.062	[m]
EMs Distance	0.098	[m]
R_{sup}, R_{inf}	4	[Ω]
k	11	[V]
x_d	0.036	[m]
L_{sup}, L_{inf}	Function of x	[H]
K_{sup}, K_{inf}	Function of x	[Kg $\frac{m^3}{s^2 A^2}$]

Table 2.2: System parameters.

The initial condition for each state of the system is reported in the table below:

State	Initial Condition	Unit
Position	Variable	[m]
Velocity	0	[m/s]
EM1 Current	0	[A]
EM2 Current	0	[A]

Table 2.3: States initial conditions.

2.3 System Validation

In order to validate the model defined in (2.9), to determine the correctness of the parameters determined in the previous section and to test future controllers, we developed on Simulink the block diagram of our model. We also tried to model the drag force acting on the sphere when it is in motion in order to test whether it significantly affects the system. The expression used to model this phenomenon is as follows:

$$F_D = \frac{1}{2} \rho v^2 C_D A \quad (2.12)$$

where F_D is the drag force, ρ is the density of the air, v is the speed of the object relative to the air, C_D is the drag coefficient, and A is the cross-sectional area of the object (in this case, a sphere). However, it has been noted that it effects on the system's response is negligible, so in order to reduce the complexity of the mathematical model, it has been neglected.

To make the model easier to read, each part has been divided into subsystems. In the following, the block diagram implementation of the electrical and mechanical part is shown:

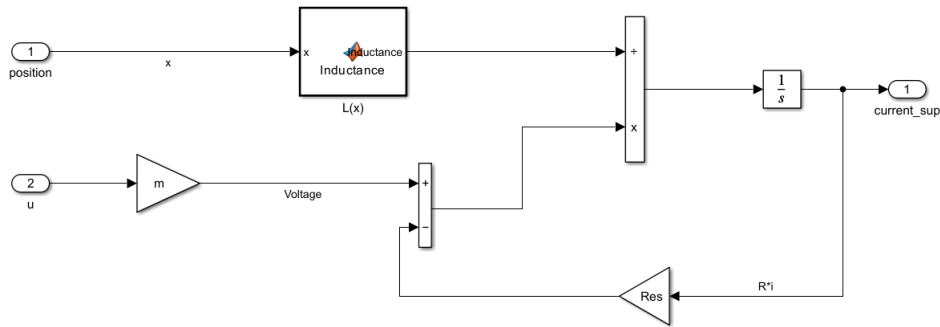


Figure 2.5: Block scheme of the electrical subsystem.

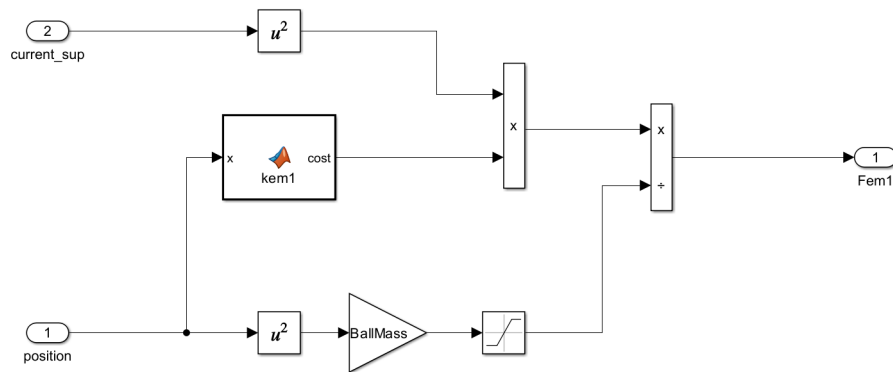


Figure 2.6: Block scheme of the mechanical subsystem.

The complete implementation of the model of the system in simulink is reported in Figure (2.7).

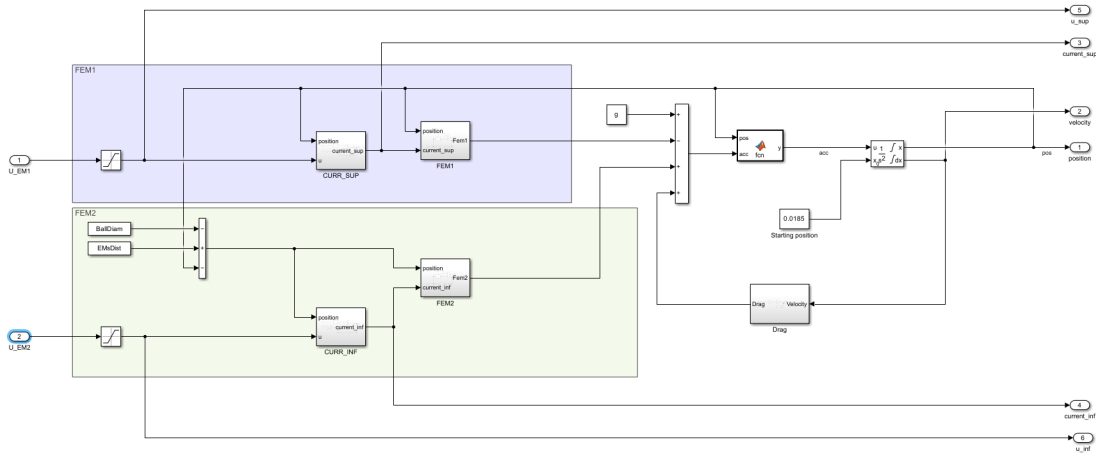


Figure 2.7: Scheme of the model employed in the identification process. The two inputs are the duty-cycles of the upper and bottom magnets saturated between $[0, 1]$.

To verify that the response of our model is equal to the one of the real system, it is possible to feed both with train of steps with increasing amplitude, verifying that the readings of the various sensors coincide with those of the simulation.

In order to have a better visualization of the trends, the results shown in Figure (2.8) have been zoomed in on the first control action large enough to make the sphere rise. It can be seen that the position, velocity and current time series of the simulated system coincide with the response of the real system to the same input signal, except for the noise caused by the sensors.

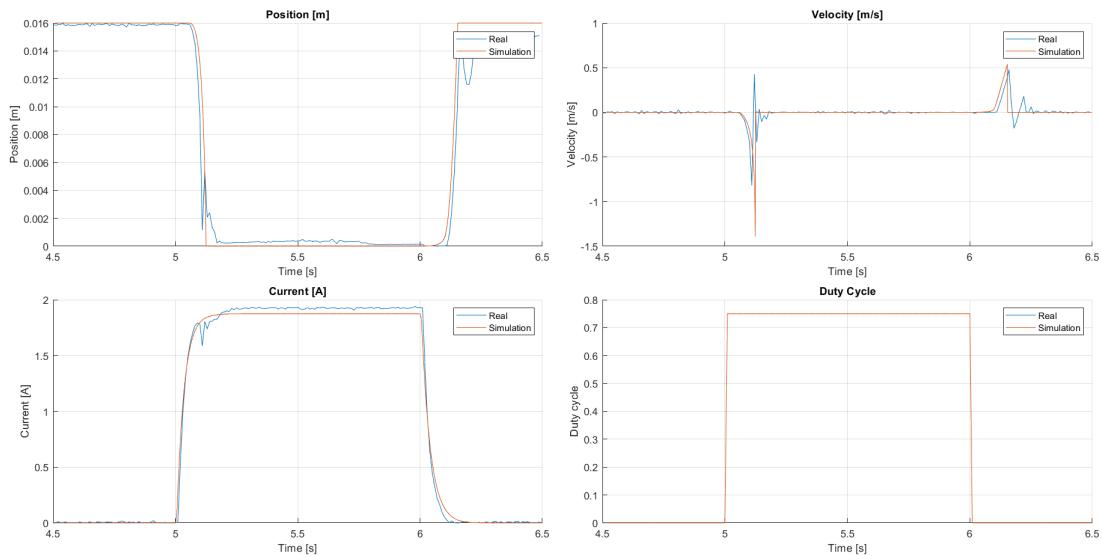


Figure 2.8: Results of the validation test with zoom on the first control action large enough to make the sphere rise.

2.4 SISO Single Magnet Linearized System

In this section the state-space representation and the transfer function of the linearized system is determined, considering only the upper magnet as a control action and the position as output (type 1 in Table 2.1).

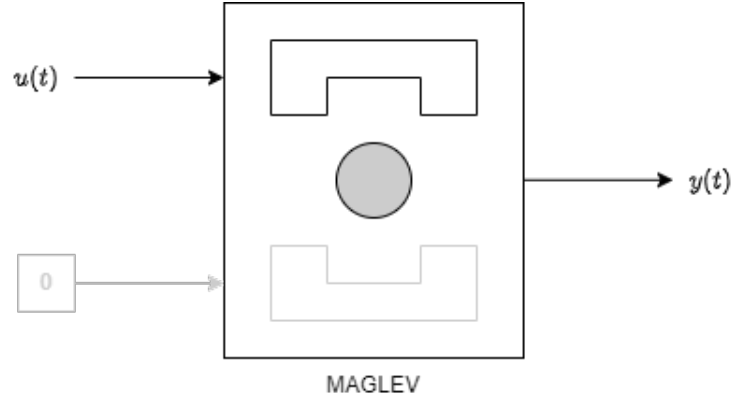


Figure 2.9: Scheme of the magnetic levitation system considering only the upper magnet.

Using (2.7) we can compute the steady-state electromagnet coil current i_{eq} that produces the desired steady-state constant ball position x_{eq} . Specifically, setting $\frac{d^2x}{dt^2} = 0$ yields:

$$i_{eq} = \sqrt{\frac{mg}{K_{em}}} x_{eq} \quad (2.13)$$

Which corresponds to a duty cycle:

$$u_{eq} = \frac{R}{k} i_{eq} \quad (2.14)$$

Now theoretically we can use (2.14) to regulate the ball position, however due to external disturbances and system parameters uncertainty a feedback controller is needed.

To linearize the system we can define the following set of shifted variables:

$$\hat{x}(t) = x(t) - x_{eq} \quad (2.15)$$

$$\hat{i}(t) = i(t) - i_{eq} \quad (2.16)$$

$$\hat{u}(t) = u(t) - u_{eq} \quad (2.17)$$

We can rewrite the dynamic equation (2.7) as:

$$m \frac{d^2 \hat{x}}{dt^2} = mg - K_{em} \left(\frac{\hat{i}(t) + i_{eq}}{\hat{x}(t) + x_{eq}} \right)^2 \quad (2.18)$$

Now we can linearize (2.18) around $(\hat{x} = 0, \hat{i} = 0)$:

$$\ddot{\hat{x}}_{lin} = \frac{d}{d\hat{x}} \left(g - \frac{K_{em}}{m} \frac{(\hat{i} + i_{eq})^2}{(\hat{x} + x_{eq})^2} \right) \bigg|_{\hat{x}=0, \hat{i}=0} \hat{x} + \frac{d}{d\hat{i}} \left(g - \frac{K_{em}}{m} \frac{(\hat{i} + i_{eq})^2}{(\hat{x} + x_{eq})^2} \right) \bigg|_{\hat{x}=0, \hat{i}=0} \hat{i} \quad (2.19)$$

or, equivalently:

$$\ddot{\hat{x}}_{lin} = \left(\frac{2K_{em}}{m} \frac{i_{eq}^2}{x_{eq}^3} \right) \hat{x} - \left(\frac{2K_{em}}{m} \frac{i_{eq}}{x_{eq}^2} \right) \hat{i} \quad (2.20)$$

Reusing the shifted variables, we can rewrite the dynamic equation 2.8 as:

$$k(\hat{u} + u_{eq}) = R(\hat{i} + i_{eq}) + L(x) \frac{d\hat{i}}{dt} \quad (2.21)$$

Now we can linearize it around $(\hat{x} = 0, \hat{i} = 0)$:

$$\begin{aligned} \hat{i}_{lin} = & \frac{d}{d\hat{x}} \left(\frac{1}{L_1 + \frac{L_0 x_0}{\hat{x} + x_{eq}}} (k(\hat{u} + u_{eq}) - R(\hat{i} + i_{eq})) \right) \bigg|_{\hat{x}=0, \hat{i}=0, \hat{u}=0} \hat{x} + \\ & + \frac{d}{d\hat{i}} \left(\frac{1}{L_1 + \frac{L_0 x_0}{\hat{x} + x_{eq}}} (k(\hat{u} + u_{eq}) - R(\hat{i} + i_{eq})) \right) \bigg|_{\hat{x}=0, \hat{i}=0, \hat{u}=0} \hat{i} + \\ & + \frac{d}{d\hat{u}} \left(\frac{1}{L_1 + \frac{L_0 x_0}{\hat{x} + x_{eq}}} (k(\hat{u} + u_{eq}) - R(\hat{i} + i_{eq})) \right) \bigg|_{\hat{x}=0, \hat{i}=0, \hat{u}=0} \hat{u} \end{aligned} \quad (2.22)$$

or, equivalently:

$$\hat{i}_{lin} = \left(-\frac{R}{L(x_{eq})} \right) \hat{i} + \left(\frac{k}{L(x_{eq})} \right) \hat{u} \quad (2.23)$$

Continuous-time system. Now, using (2.20) and (2.23), we can write the continuous-time state-space representation of our system:

$$\begin{cases} \dot{x}(t) = A(x) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (2.24)$$

Where the system matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2K_{em}}{m} \frac{i_{eq}^2}{x_{eq}^3} & 0 & -\frac{2K_{em}}{m} \frac{i_{eq}}{x_{eq}^2} \\ 0 & 0 & -\frac{R}{L(x_{eq})} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \frac{k}{L(x_{eq})} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad (2.25)$$

The state-space representation depends on the equilibrium position in which we linearize the system. If we linearize the system around the equilibrium position $\mathbf{x}_{eq} = 0.01$ m, the state-state representation is:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1962 & 0 & -20.6 \\ 0 & 0 & -32.8 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 92.62 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad (2.26)$$

The relative transfer function is:

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \frac{-1907}{s^3 + 32.77s^2 - 1962s - 64300} \quad (2.27)$$

The determined transfer function has two poles in $s = \pm 44.3$ related to the mechanical part of the system and one in $s = -32.8$ related to the electrical part. Obviously, the determined system is open-loop unstable since it has a pole in the right half-plane.

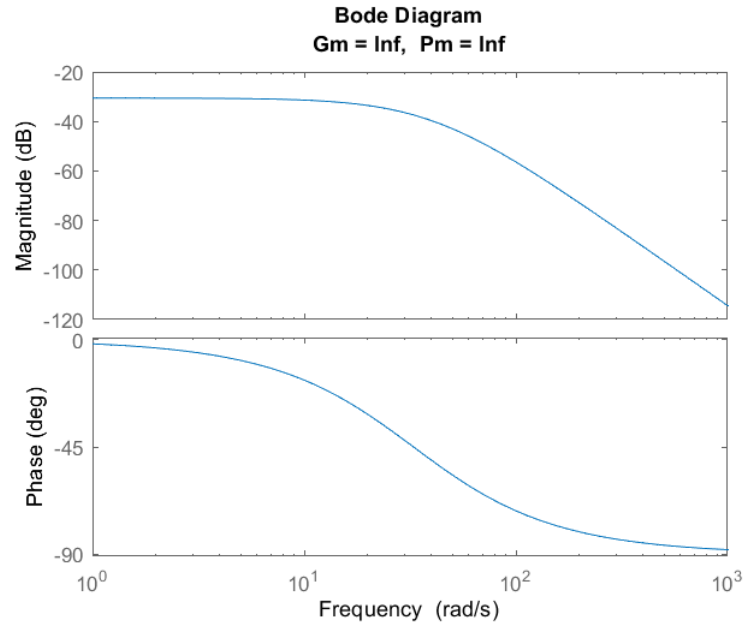


Figure 2.10: Bode diagram of $\mathbf{G}(s)$.

Discrete-time system. Since on laboratory instrumentation all the controllers must necessarily be implemented in discrete time, it is necessary to discretize the determined system in order to design the various control schemes.

To discretize the system the Forward Euler Method was used, therefore:

$$\dot{\mathbf{x}} \approx \frac{1}{T_s}(\mathbf{x}(k+1) - \mathbf{x}(k)) \quad (2.28)$$

where $T_s = 0.001$ s is the sampling time and coincide with the lower limit of the instrumentation. Then, in discrete time the system becomes:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}u(k) \\ \mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) \end{cases} \quad (2.29)$$

Where:

$$\mathbf{F} = \mathbf{I} + \mathbf{A}T_s \quad \mathbf{G} = \mathbf{B}T_s \quad \mathbf{H} = \mathbf{C} \quad (2.30)$$

The matrices of the state-space representation of the discrete system are:

$$\mathbf{F} = \begin{bmatrix} 1 & 0.001 & 0 \\ 1.9630 & 1 & -0.0203 \\ 0 & 0 & 0.9678 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ 0.0911 \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad (2.31)$$

The relative transfer function is:

$$\mathbf{G}(s) = \mathbf{H}(\mathbf{z}\mathbf{I} - \mathbf{F})^{-1}\mathbf{G} = \frac{-(1.4 \cdot 10^{-6})z^2 - (1.468 \cdot 10^{-8})z - (4.616 \cdot 10^{-7})}{z^3 - 2.97z^2 + 2.937z - 0.9678} \quad (2.32)$$

The determined transfer function has two poles in $z = 0.9567$ and $z = 1.0453$ related to the mechanical part of the system and one in $z = 0.9678$ related to the electrical part. As we expected, due to the pole in $z = 1.0453$, the system is open-loop unstable.

Chapter 3

Noise Characterization

3.1 Properties of the noise

Estimating and characterizing the noise is important because it affects the performance and the stability of our system. Open-loop measurements can provide valuable insight on the noise of the system itself but we preferred to use the closed loop measurements with the PID controller because it simulates the conditions in which the system is actively controlled, including disturbances, control actions, and nonlinearities, all of which can contribute to the overall noise in the system. However it may mask some of the noise inherent in the system due to the corrective actions of the controller but, comparing it with open-loop measurements, it was seen that this effect is negligible.

To visualise the distribution of the data, we start with a histogram of the relative probability. To get a comprehensive view of the noise characteristics, we visualise the frequency of occurrence of different position values using a histogram, where each bar represents a bin and its height represents the probability of data points falling within that bin, depending on the chosen normalisation. We start with the relative probability, i.e. the number of elements in each bin relative to the total number of elements in the input data, which can be at most 1.

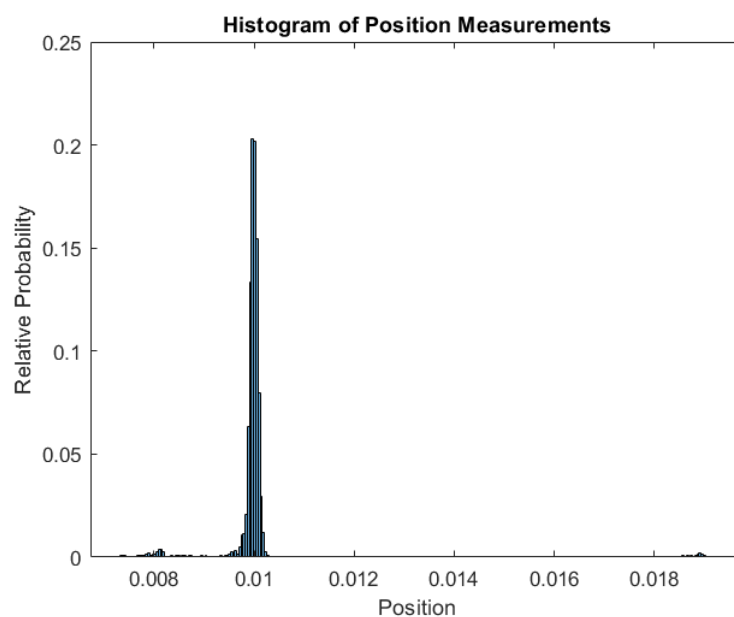


Figure 3.1: Histogram of the steady-state position measurements.

We can observe that the position distribution is very similar to a Gaussian because there is a symmetric bell curve. Visualizing the boxplot we can also see that in the measurements there are some outliers:

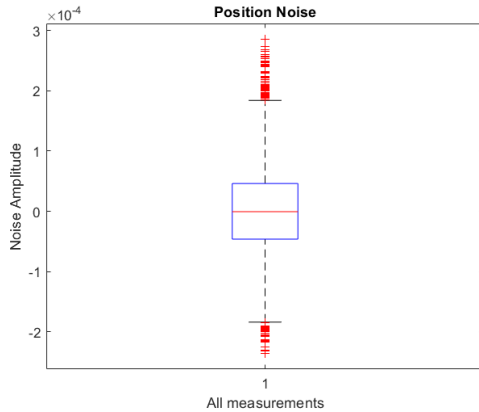


Figure 3.2: Boxplot of the steady-state position measurements.

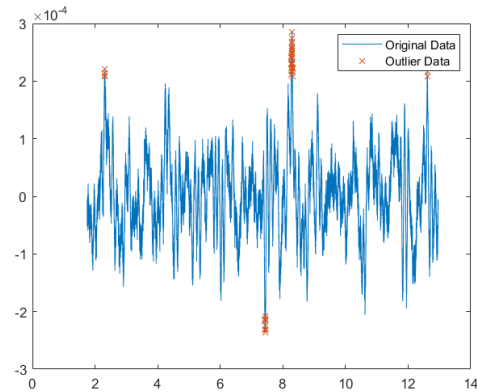


Figure 3.3: Time-series of the steady-state position measurements.

The middle box represents the interquartile range (IQR), i.e. the difference between the third quartile (Q3) and the first quartile (Q1), and contains the middle 50 per cent of the data. The red line inside the box represents the median of the data. The lines extending from the edges of the box show the extent of the data that are not considered outliers and extend up to 1.5 times the IQR both above Q3 and below Q1. These represent the range of data that are not considered outliers. Red dots outside the whiskers are considered outliers. These are data that are more than 1.5 times the IQR above Q3 or below Q1. Outliers are represented as red crosses on the graph and there are 110 of them.

Before going any further, we decide to remove them, because we assumed that data follow a normal distribution, and outliers can significantly distort the mean and standard deviation, which are key measures in noise estimation. In the following there are the boxplots of the data with and without outliers:

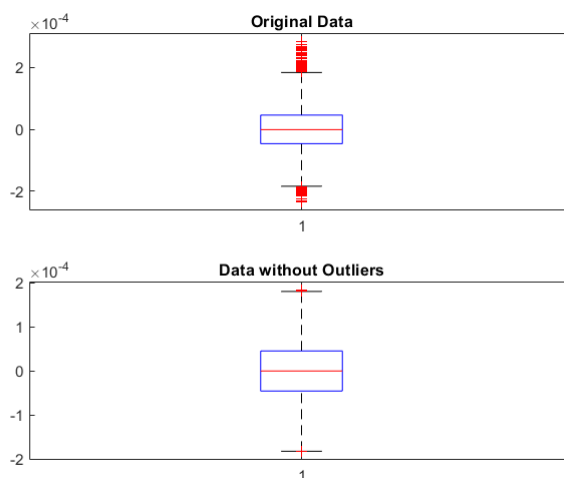


Figure 3.4: Boxplots of the position data with and without outliers.

We also repeated the same procedure for velocity and current measurements.

Now we can model the noise using stochastic processes, i.e. calculating the statistical properties of the noise, such as its mean, variance, standard deviation and covariance between the 3 states, position, velocity and speed, to obtain some information about their relationships.

The standard deviations are:

$$\sigma_{pos} = 0.0146 \quad \sigma_{vel} = 6.9 \cdot 10^{-5} \quad \sigma_{curr} = 0.0216 \quad (3.1)$$

While the covariance matrix is:

$$\rho = \begin{bmatrix} 4.8 \cdot 10^{-9} & 2.3 \cdot 10^{-7} & 7.4 \cdot 10^{-7} \\ 2.3 \cdot 10^{-7} & 4.7 \cdot 10^{-4} & 1.4 \cdot 10^{-6} \\ 7.4 \cdot 10^{-7} & 1.4 \cdot 10^{-6} & 2.1 \cdot 10^{-4} \end{bmatrix} \quad (3.2)$$

To validate our idea, we can also plot the histogram of the probability density function of the position noise with a Gaussian function, using the previously estimated parameters, and the result is visible in Figure 3.5.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (3.3)$$

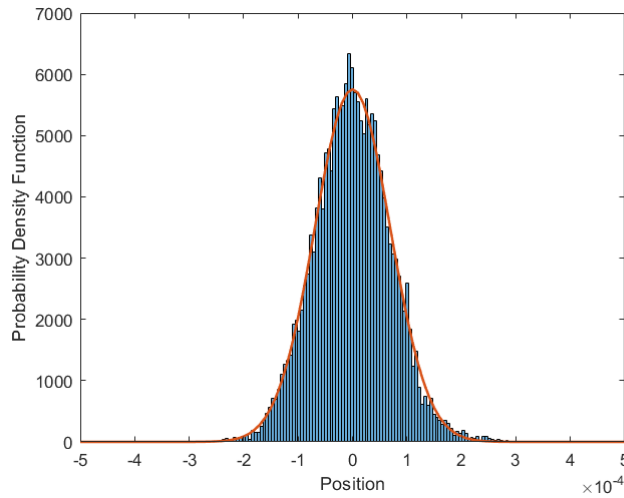


Figure 3.5: Analysis of the measurements shows that the distribution is very close to a normal distribution.

We can now continue to characterise the noise by performing a frequency domain analysis using the Fast Fourier Transform (FFT) to identify the frequency components of the noise. We have used MATLAB to calculate the one-sided magnitude spectrum of a noise signal using the Fast Fourier Transform (FFT) algorithm. The frequency spectrum plot can provide valuable insight into the frequency components present in the noise signal.

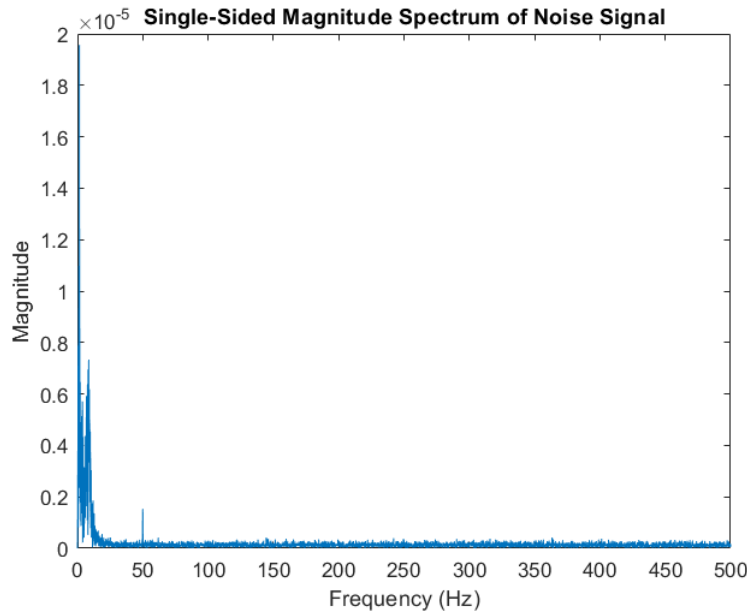


Figure 3.6: Single-sided magnitude spectrum of the noise signal on the position.

The x-axis represents the frequency (Hz) from 0 Hz to 500 Hz, which is the Nyquist frequency (half the sampling frequency, $F_s = 1000$ Hz). The y-axis represents the magnitude of the frequency components in the noise signal.

We can see a significant concentration of energy in the low frequency range, particularly from 0 Hz to about 20 Hz. This indicates the presence of low frequency mechanical vibration, environmental factors, or inherent system dynamics. We can also identify a peak at around 50 Hz, which is likely to be electrical noise or interference from the power line frequency.

If the magnetic levitation system is sensitive to low-frequency vibrations, these components could cause fluctuations in the position of the ball; the interference at 50 Hz could affect the accuracy of the sensors and the stability of the control system.

The magnitude of the noise decreases significantly at higher frequencies (> 100 Hz), indicating that the system is less affected by high frequency noise. This suggests that the system is inherently less sensitive to high frequency noise, or that the PID is already designed to mitigate such noise.

All these experiment were also done in order to estimate the variances of the sensors and to have enough information to build the variance matrices for the Kalman filter.

3.2 Noise filtering

With the data obtained in the previous section, we can design a filter to be applied on the measurements signals in order to filter out their high-frequency components and improve the steady-state performance of the control system.

We tried to model different low pass filters using different approaches:

- Discretisation of a first-order continuous low-pass filter:

$$H(s) = \frac{1}{sT + 1} \quad (3.4)$$

- IIR filter:

$$H(z) = z \frac{(1 - \alpha)}{(z - \alpha)} \quad (3.5)$$

where $\alpha = e^{-(\omega_c T_s)}$ and T_s is the sampling time. ω_c can be at most 10% of the sampling rate and it's discretized with ZOH

- Exponentially weighted moving average filter (EWMA): same expression of IIR but $\alpha = f_c/f_s$ and f_c is the cutoff frequency.

After some consideration, we have chosen a cut-off frequency of 60 Hz. This effectively reduces the noise at 50 Hz while allowing the desired signal frequencies to pass through with minimal distortion.

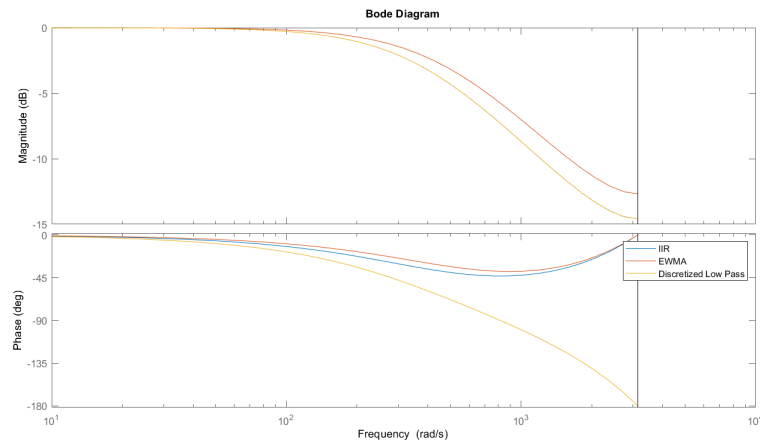


Figure 3.7: Comparison between the 3 types of filters.

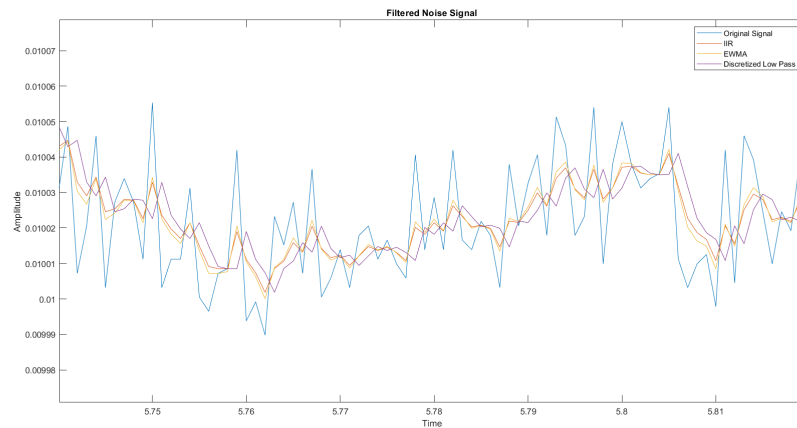


Figure 3.8: Comparing the filtered signal, we can better observe the phase shift.

We chose to use an EWMA filter because, although its magnitude is very similar to the others, the phase characteristics introduced by the zero at the origin made it the most suitable choice, providing the desired frequency response while maintaining control over the phase behaviour, in fact as excessive phase lag could lead to instability or oscillation.

The final EWMA filter transfer function with cut-off frequency of 60 Hz is:

$$H(z) = \frac{0.3142z}{z - 0.6858} \quad (3.6)$$

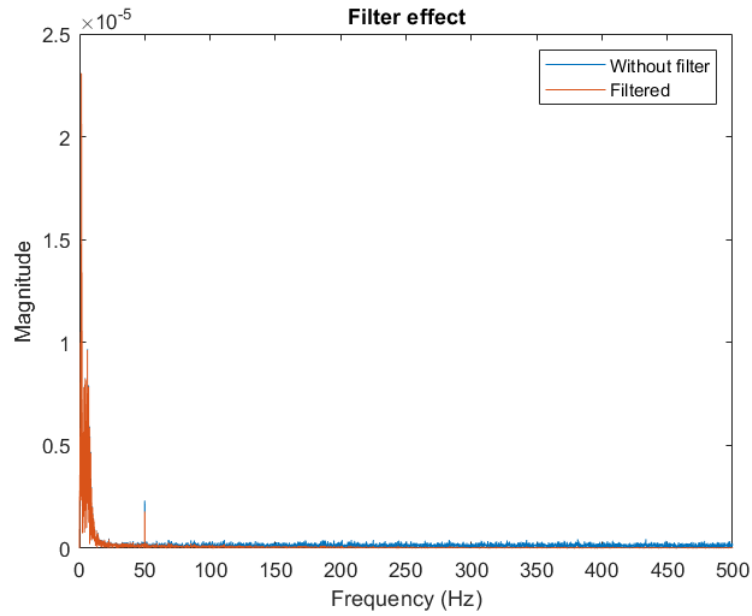


Figure 3.9: Filter effect.

Chapter 4

Control Schemes

4.1 PID

4.1.1 Single Loop

We used a PID as the first controller to stabilize the sphere at a desired constant height. The controller was designed on the continuous-time linearized system and then discretized to implement it on the real laboratory system.

The system considered is the SISO system with the bottom magnet turned off, whose transfer function is the one determined in the previous chapter and reported below:

$$G(s) = \frac{-1907}{(s + 32.8)(s + 44.3)(s - 44.3)} \quad (4.1)$$

The predetermined specifications that the closed-loop system must meet are as follows:

- Settling time less than 1.5 s
- Overshoot less than 30%
- Zero steady-state error

We have seen that the open loop is unstable due to the pole in the right half of the s-plane.

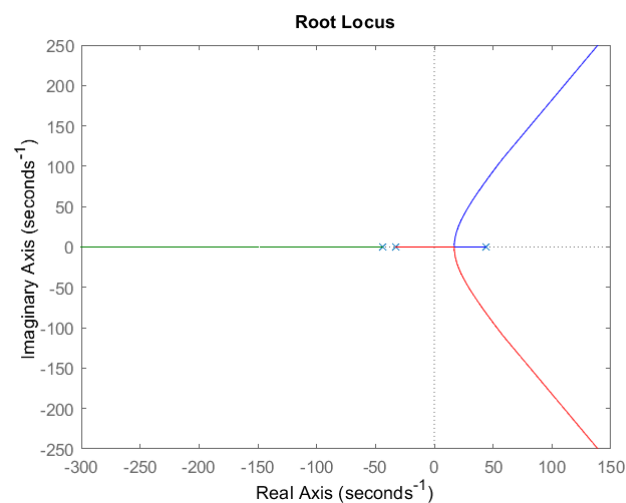


Figure 4.1: Root-locus and Bode plot of the open-loop system.

Adding a simple proportional controller will not lead to a stable closed loop system for any value of the gain as may be seen from the root locus diagram of Fig. 4.1, so it will not stabilize the system. Adding an integral action to our controller leads to the same result as for the P controller, so the simpler controller found that stabilizes the system is a PD, but to remove the steady-state error, we decide to add also integral action.

$$R(s) = K_P + \frac{K_I}{s} + K_D s = \mu \frac{(1 + sT_1)(1 + sT_2)}{s} \quad (4.2)$$

Where:

$$K_P = \mu(T_1 + T_2) \quad K_I = \mu \quad K_D = \mu T_1 T_2 \quad (4.3)$$

To design the PID controller, we can chose the zeros of $R(s)$ in order to modify the shape of the root locus and have a prescribed position of the closed-loop poles to meet the specifications.

After several attempts, the following parameters were determined:

$$\begin{cases} \mu = 63.6 \\ T_1 = 1/2 \\ T_2 = 1/20 \end{cases} \Rightarrow \begin{cases} K_P = 34.98 \\ K_I = 63.6 \\ K_D = 1.59 \end{cases} \quad (4.4)$$

Since the duty cycle is confined in the range $[0, 1]$ and there is an integral action, we have incorporated an anti-wind-up strategy with the clamping method (Figure 4.2). We introduced a saturation block to the control action and when it saturates, the integral action is set to 0 using a switch.

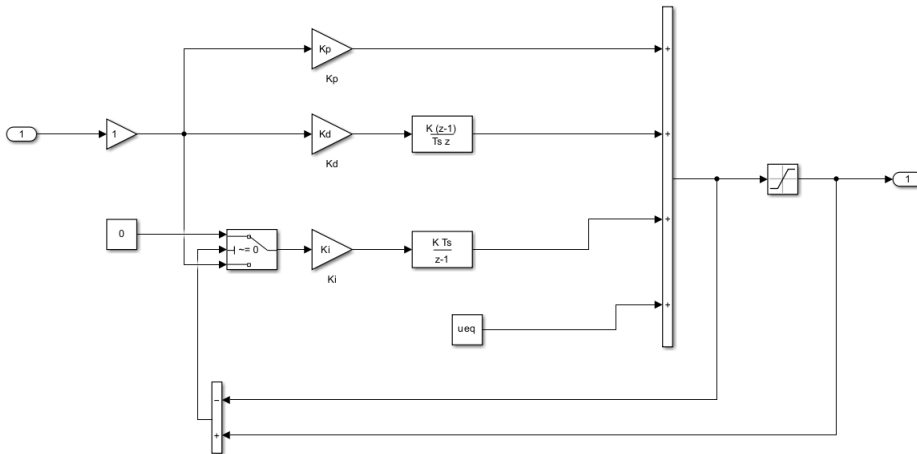


Figure 4.2: Manual implementation of the discrete-time PID controller on the laboratory system.

The system response is shown in Figure 4.3. The system is able to maintain the sphere at the desired height with zero steady-state error and within the prescribed settling time, however the overshoot is too big ($\approx 62\%$) for our desired performances. Such high overshoot values

could let the ball touching the top magnet during the transient, so it is preferable to reduce it.

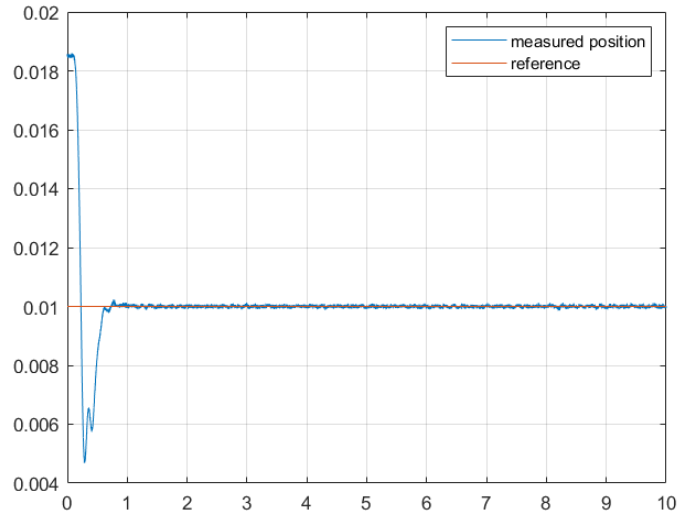


Figure 4.3: Response of the system with the first version of the PID.

For this reason we decided to increase the derivative action, moving one zero to the right, even if this will lead to an increase of the settling time. The new parameters are:

$$\begin{cases} \mu = 63.6 \\ T_1 = 1/1.5 \\ T_2 = 1/20 \end{cases} \Rightarrow \begin{cases} K_P = 45.6 \\ K_I = 63.6 \\ K_D = 2.2 \end{cases} \quad (4.5)$$

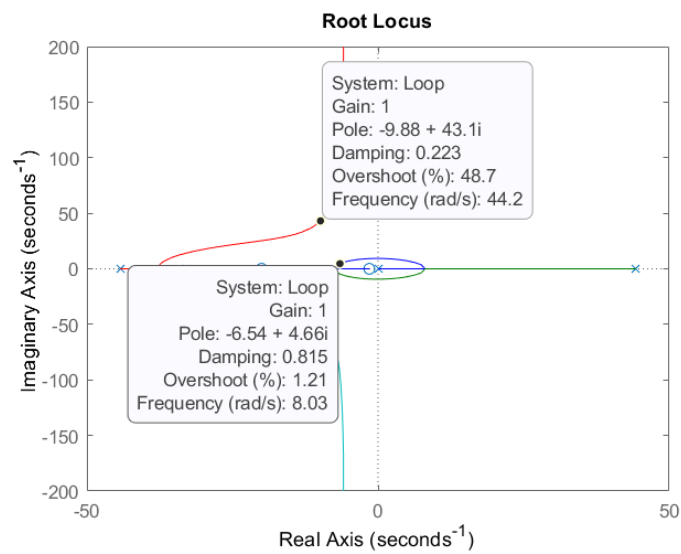


Figure 4.4: Root-locus of the loop transfer function with the final PID ($K_P = 45.6$, $K_I = 63.6$ and $K_D = 2.2$).

The new response is reported below:

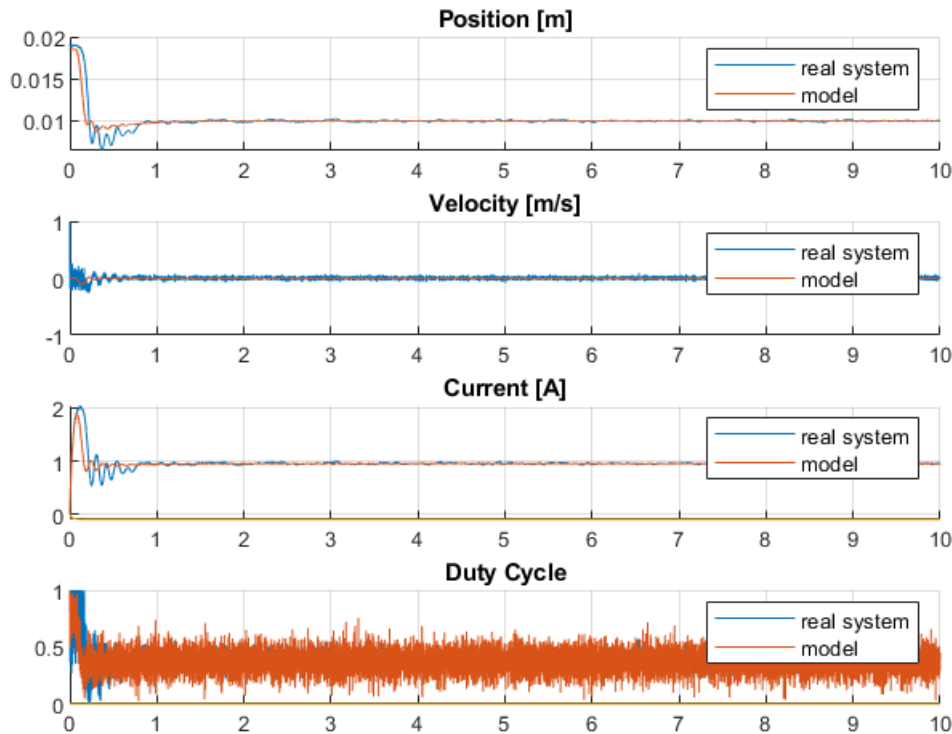


Figure 4.5: System response with the final version of the PID.

Through the real system data, it is possible to determine the response characteristics shown in Table 4.1, which are below the limit imposed by the specifications.

	1st version PID	Final PID
Rise Time	0.0727 s	0.0827 s
Settling Time	0.7893 s	1.406 s
Overshoot	62.50%	13.74%

Table 4.1: Performance comparison between the two version of the PIDs.

4.1.2 Cascade Control

Since our system can be seen as the series of two subsystems (electrical and mechanical) and since we also have a current sensor, we can implement a cascade PID scheme. Referring at the Figure (4.6), when a disturbance $d(t)$ acts on the output signal of the "inner" system (and on the input of the outer system), it is convenient to use a cascade scheme because the inner controller will attenuate this disturbance more effectively since we can design it with

a higher bandwidth. In our case this disturbance can be seen as any dynamics not captured by our model acting on the RL circuit.

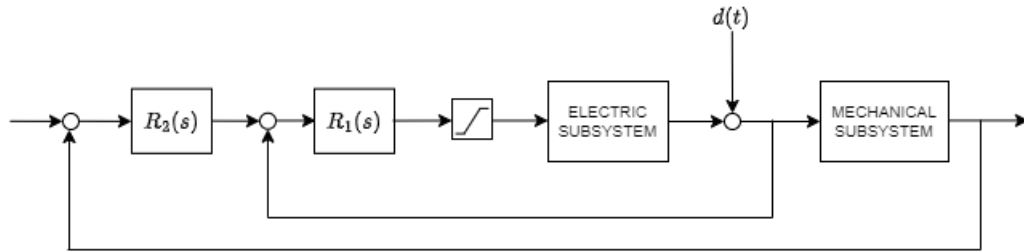


Figure 4.6: Cascade structure block scheme with the disturbance acting on the "inner" plant output highlighted.

In the design process, the inner-loop control system is designed first, and the inner-loop transfer function is obtained. The outer loop is then designed based on the inner loop, ignoring the relatively small time-constants resulting from the inner loop.

Inner loop. For the inner loop we decided to use a PI controller. The transfer function of the electrical part is:

$$G_e(s) = \frac{80.75}{s + 32.8} \quad (4.6)$$

If the bandwidth of the internal controller is significantly greater than the sampling rate, problems such as aliasing can occur. To avoid this problem, it's a good idea to choose a frequency that's half the Nyquist frequency, but as rule of thumb, it's advisable to keep a decade as the difference. Since the sampling time is 0.001 s, we start considering designing a PI 10 times slower, so the desired bandwidth will be 100 Hz \approx 628 rad/s.

$$PI(s) = \mu \frac{(1 + sT)}{s} \quad (4.7)$$

We can place the zero of the PI in $s = -32.8$, to cancel out the pole of the electric dynamic and we can impose the cutoff frequency using the resulting open loop transfer function. After some mathematical passages, the PI of the inner loop will be:

$$PI(s) = 7.75 \frac{s + 32.8}{s} \quad (4.8)$$

Which allows us to have a bandwidth of 99.55 Hz \approx 625 rad/s.

Outer loop. Now we can pass to the external loop. The maximum bandwidth in order to consider the inner loop ideal is 10Hz \approx 62.5 rad/s (1 decade separation). The transfer function of the mechanical part is:

$$G_m(s) = \frac{-21.08}{s^2 - 1962} = \frac{-21.08}{(s - 44.3)(s + 44.3)} \quad (4.9)$$

In this case as a controller we decided to use a PID. As we did for the inner loop, we chose the position of the two zeros so that: one cancel the mechanical dynamic ($s = -44.3$), and the other closer to the origin. After several mathematical steps, considering the limit on the bandwidth, we obtained the expression of the external PID:

$$PID(s) = -200 \frac{(s + 2)(s + 44.3)}{s} \quad (4.10)$$

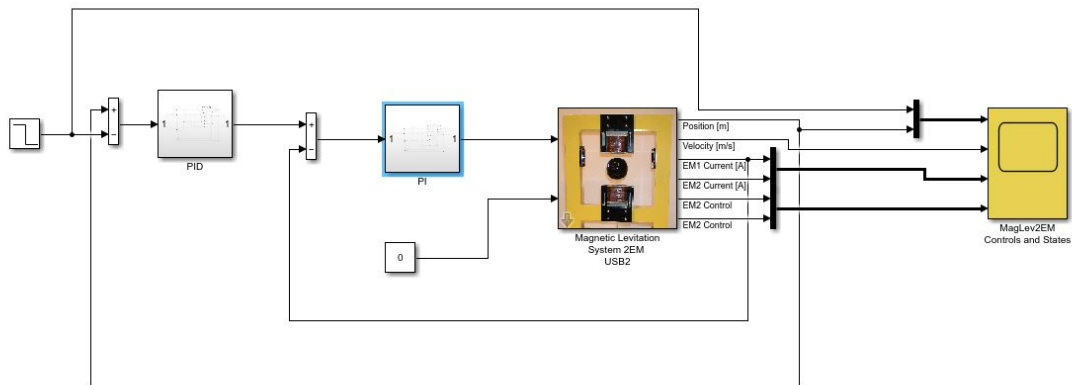


Figure 4.7: Cascade structure implementation on the laboratory system.

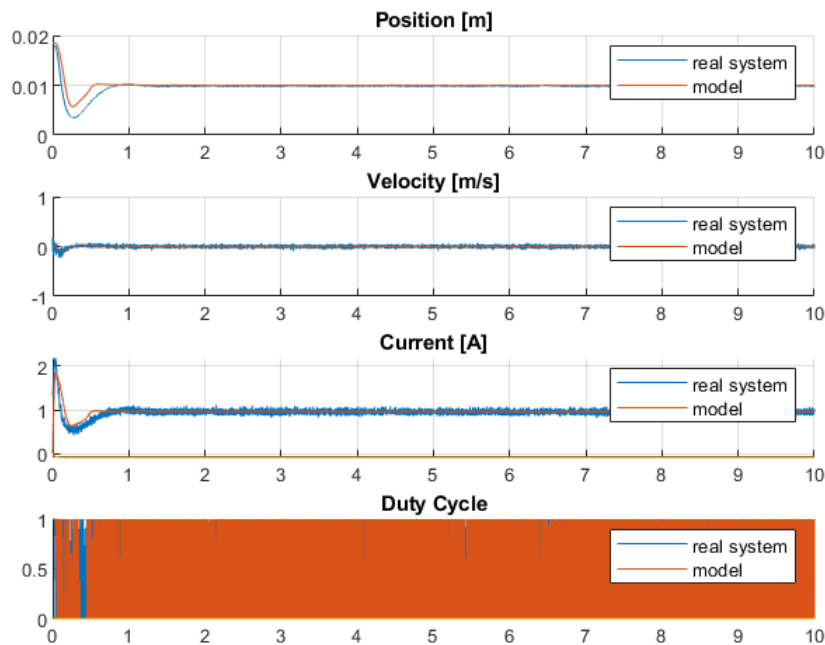


Figure 4.8: System response with cascade structure

We can observe that the settling time is smaller than the single PID, while the overshoot is increased but it still remains under the limits imposed by the specifications.

4.2 Pole-Placement Control

Since with the laboratory instrumentation we have real-time measurements of all the states (position, velocity and current), we can implement a state-feedback control scheme designed with Pole-Placement, without the use of state observers.

With this control architecture, considering our discrete-time linearized system (2.29), we can use the control law:

$$u(k) = -Kx(k) + \eta(k), \quad K \in R^{m,n}, \eta \in R^m \quad (4.11)$$

Where the feedback matrix K is designed to stabilize, such that the closed-loop system:

$$x(k+1) = (F - GK)x(k) + B\eta(k) \quad (4.12)$$

has prescribed eigenvalues (of the matrix $(F - GK)$). We can use the additional input $\eta(k)$ to design an external loop to improve the overall performances of our system. In our case, the external loop has been used to completely remove the steady-state error on the position (with an integrator) and to speed up the system response.

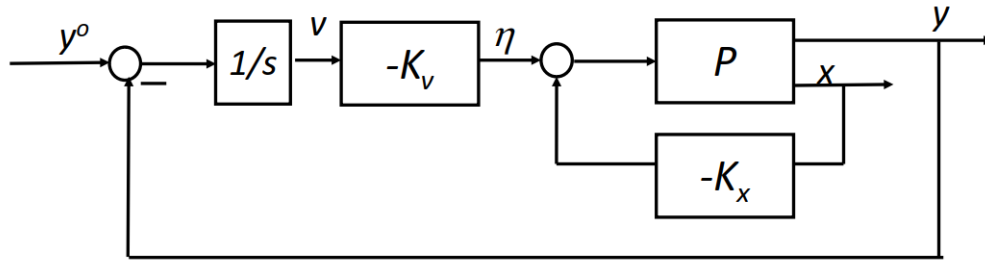


Figure 4.9: Pole-Placement Control Scheme.
Inner loop used for stabilization and outer loop for performances.

The state-space representation of the enlarged system (original system + integrator) is:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} F & 0 \\ -H & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} G \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ I \end{bmatrix} y^o \quad (4.13)$$

So our new state and input matrices are:

$$\bar{F} = \begin{bmatrix} F & 0 \\ -H & 0 \end{bmatrix} \quad \bar{G} = \begin{bmatrix} G \\ 0 \end{bmatrix} \quad (4.14)$$

And the control law is:

$$u(k) = -K \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} = - \begin{bmatrix} K_x & K_v \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \quad (4.15)$$

Before determining the values of the gains K_x and K_v , it is necessary to verify that the pair (\bar{F}, \bar{G}) is controllable (necessary and sufficient condition for the solution of the pole

placement problem). By calculating the controllability matrix with the help of the Matlab function `ctrb(F, G)`, we can see that it is full rank, so this means that the enlarged system is fully controllable from the input $u(t)$ and the applicability condition of the control scheme is met.

The choice of the position of the poles with this control scheme is never trivial. In this case, their position was chosen imposing that the closed loop system approximately behaves as a first order system, with an unique dominant pole and all the remaining closed-loop poles "much faster" than it. In addition, by mimicking a first order system we can also remove possible overshoot from the step-response.

$$G(s) = \frac{\mu}{1 + sT} \quad \Rightarrow \quad y(t) = \mu(1 - e^{-t/T}) \quad (4.16)$$

In particular, to obtain a desired settling time of 1 second, a pole is placed in -4 and the remaining three in -48 , -48.01 and -48.02 (12 times faster in order to make them negligible). However, since the controller needs to be implemented in discrete-time, it is necessary to determine their relative position in the Z-plane, as reported in the Table 4.2:

Continuous-time	Discrete-time
-4	0.9960
-48	0.9531
-48.01	0.9530
-48.02	0.9529

Table 4.2: Desired closed-loop poles.

At this point we can find the gain values using the Ackermann's formula through the Matlab function `place.m`.

$$K = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} M_r^{-1} P(A) \quad (4.17)$$

Where M_r is the reachability matrix and $P(A)$ can be determined using the desired poles position:

$$P(A) = (s + p_1)(s + p_2)(s + p_3)(s + p_4) \quad (4.18)$$

The resulting gains are reported in Table 4.3. With these values, our closed loop system will behave like the following first-order system:

$$\tilde{G}(s) = \frac{4}{s + 4} \quad (4.19)$$

First the controller was tested on our Simulink model in order to verify that the response of the system has the desired characteristic and to fine-tune the gains if necessary. The

K_{x_1}	K_{x_2}	K_{x_3}	K_v
-217.8232	-4.8046	1.2076	220.5613

Table 4.3: Pole-Placement controller gains.

complete control scheme implemented on the laboratory instrumentation is shown in Figure 4.10, where the feedback signals on each state are those of the available sensors (position, velocity and current).

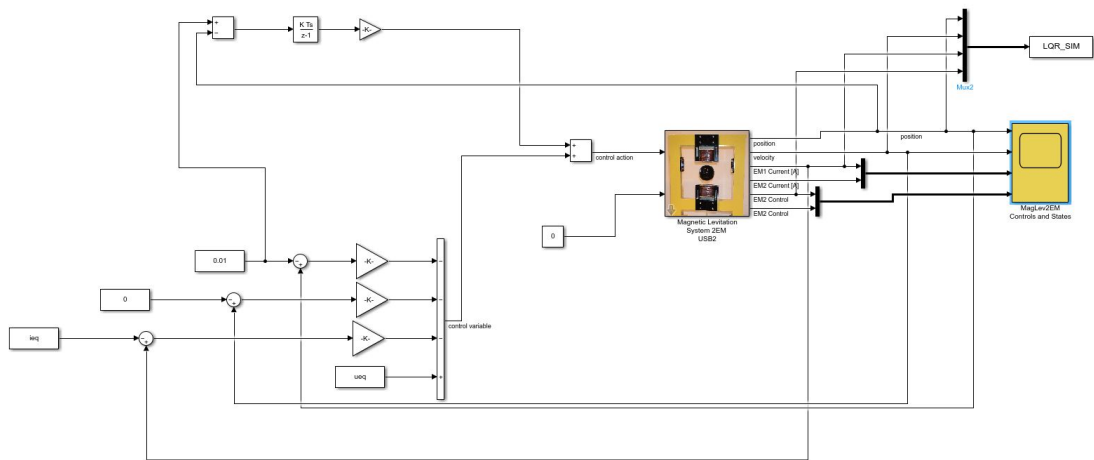


Figure 4.10: Implementation of the Pole-Placement control scheme on the laboratory instrumentation.

As can be seen from Figure 4.11, the step response of the closed-loop systems respects the desired settling time with a first-order system behaviour.

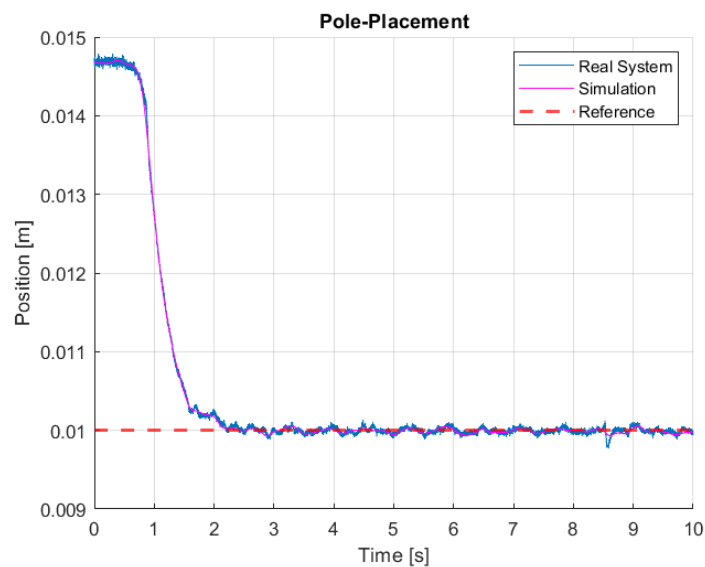


Figure 4.11: Step response pole-placement control.

The good feature of this scheme is that the system, mimicking a first-order one, has no overshoot so it is possible to use it in certain applications where we have tighter limits on the position that the sphere can assume.

The only problem with this control scheme is that, unlike the PID controllers seen in the previous section, it is less robust from the point of view of the starting position. This can simply be seen from the fact that if the starting position of the ball (first state variable) is the lowest available (≈ 0.0185 m), the pole-placement controller cannot stabilize the ball in the desired position, but requires a starting position closer to the one in which the system has been linearized ($x_{eq} = 0.01$ m). After several tests, it was determined that the maximum position from which the system behaves as designed is $x \approx 0.016$ m.

All the other signals acquired by the physical sensors on the system are shown in Figure 4.12.

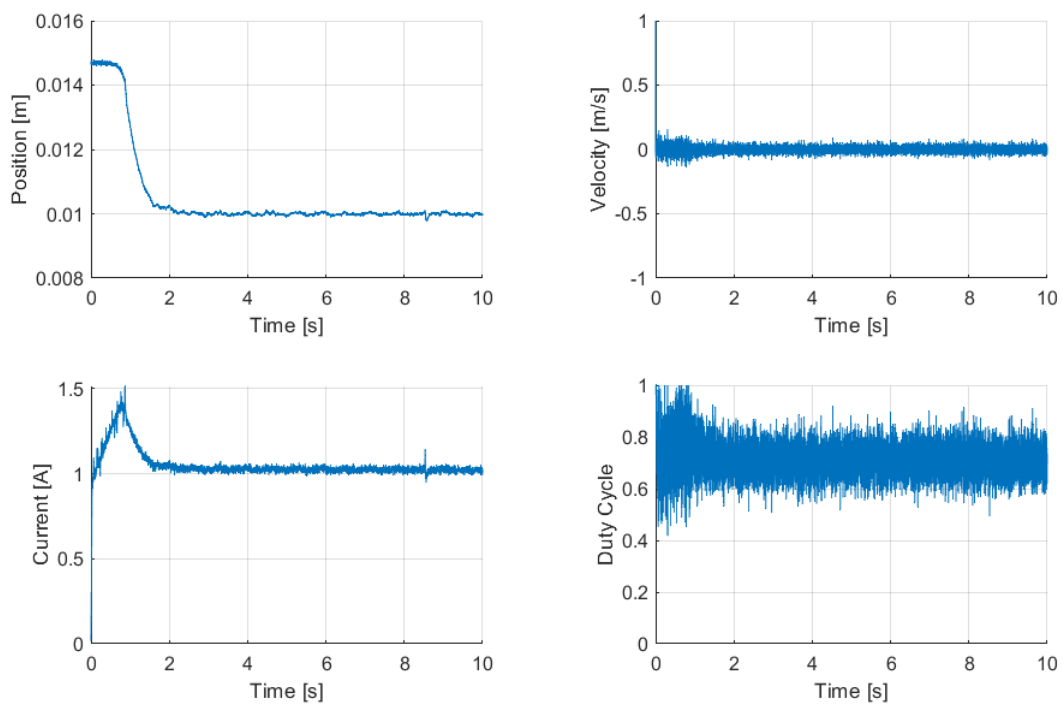


Figure 4.12: Results Pole-Placement Control.

4.3 Linear Quadratic Regulator (LQR)

The structure of the Pole-Placement controller and the Linear Quadratic Regulator are exactly the same since they are both full state feedback controller, so we can implement the results with the same scheme. The only difference between these two controls scheme is how we choose the gain matrix \mathbf{K} . Since the main problem of Pole-Placement is that is really hard to find a good position for the closed-loop poles, we decided to implement also this different type of state feedback controller since in this case we can choose the optimal value of \mathbf{K} penalizing closed-loop characteristics that are important to us like the position error and the control effort.

We can select a quadratic cost function J that takes into account the performances and the control effort:

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} + 2\mathbf{x}^T \mathbf{N} \mathbf{u}) dt \quad (4.20)$$

where \mathbf{Q} , \mathbf{R} , and \mathbf{N} are the weighting matrices. The matrix \mathbf{Q} penalizes the error on the state variables and the matrix \mathbf{R} penalizes the control effort. Larger values of \mathbf{R} minimize the control action and can be used to satisfy the saturation constraints of our actuator, at the expense of reducing the system's response speed. The matrix \mathbf{N} penalizes the cross product of \mathbf{x} and \mathbf{u} but in our case, to simplify the control design we'll leave it equal to zero.

Solving the LQR optimal problem 4.20 gives us an optimal state feedback control law:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) \quad (4.21)$$

where the matrix \mathbf{K} is defined by:

$$\mathbf{K} = (\mathbf{R} + \mathbf{G}^T \mathbf{P} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{P} \mathbf{F} \quad (4.22)$$

where \mathbf{P} is the solution of the discrete-time Algebraic Riccati Equation (ARE):

$$0 = \mathbf{F}^T \mathbf{P} \mathbf{F} + \mathbf{Q} - (\mathbf{F}^T \mathbf{P} \mathbf{G})(\mathbf{R} + \mathbf{G}^T \mathbf{P} \mathbf{G})^{-1} (\mathbf{G}^T \mathbf{P} \mathbf{F}) \quad (4.23)$$

Starting from the linear system 2.31 as a first step, we implemented a scheme with a single loop.

Before determining the feedback gains, it is necessary to check the applicability conditions of LQR with infinite horizon. In particular, by calculating through the aid of matlab the controllability and observability matrices we note that they are full rank, i.e. the pair (\mathbf{F}, \mathbf{G}) is controllable and the pair $(\mathbf{F}, \mathbf{C}_q)$ is observable, where $\mathbf{Q} = \mathbf{C}_q^T \mathbf{C}_q$.

The weight matrix \mathbf{Q} matrix been selected in order to have a quick convergence of the first state (position) to the desired position, while the matrix \mathbf{R} to respect the saturation constraint on the duty cycle since $\mathbf{u}(t) \in [0, 1]$. After some iterations the matrices \mathbf{Q} and \mathbf{R} were chosen as:

$$\mathbf{Q} = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad \mathbf{R} = 5.5 \quad (4.24)$$

At this point the control action was implemented on the laboratory system by slightly modifying the expression (4.21) since we want to regulate the system states to a non-zero set points:

$$\mathbf{u}(k) = \mathbf{u}_d - \mathbf{K}(\mathbf{x}(k) - \mathbf{x}_d) \quad (4.25)$$

Where \mathbf{u}_d and \mathbf{x}_d are the control action and the states values in the equilibrium position. Using the discrete-time linearized system matrices (2.31), we obtain the following solution \mathbf{P} of the ARE to which correspond the gains \mathbf{K} :

$$\mathbf{P} = 10^6 \begin{bmatrix} 7.9 & 0.18 & -0.02 \\ 0.2 & 0.004 & -0.0005 \\ -0.02 & -0.0005 & 0.0001 \end{bmatrix} \quad \mathbf{K} = \begin{bmatrix} -332.9 \\ -7.5 \\ 1.9 \end{bmatrix} \quad (4.26)$$

Response results to a constant reference are shown below.

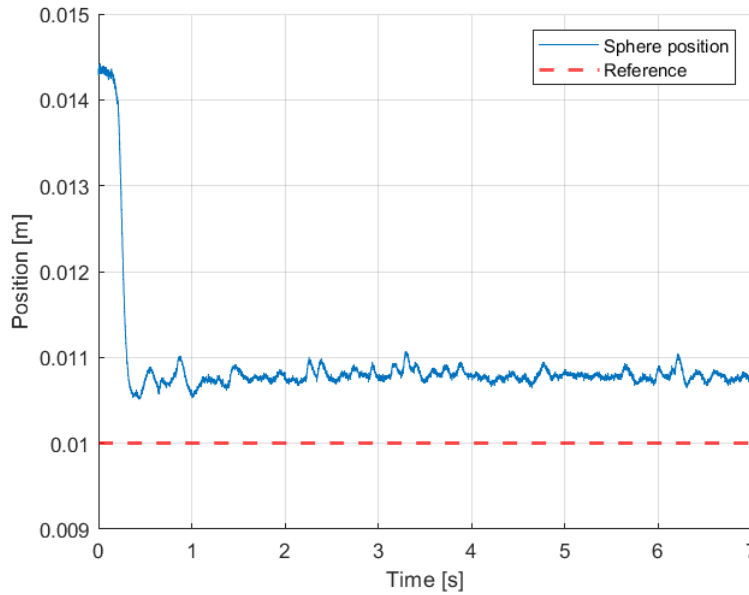


Figure 4.13: Ball position with single loop LQR control.

Although the ball manages to stabilize without major problems, the response of the system is not acceptable because there is a constant steady-state error of a few millimeters which does not allow us to reach the desired height. For this reason, similar to what was done with pole-placement, an external loop with an integrator was added to remove this error.

Repeating the procedure already seen in which we consider the enlarged system, now the matrices of weights become 4x4. After several attempts they were chosen equal to:

$$\mathbf{Q} = \begin{bmatrix} 30 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1000 \end{bmatrix} \quad \mathbf{R} = 5.5 \quad (4.27)$$

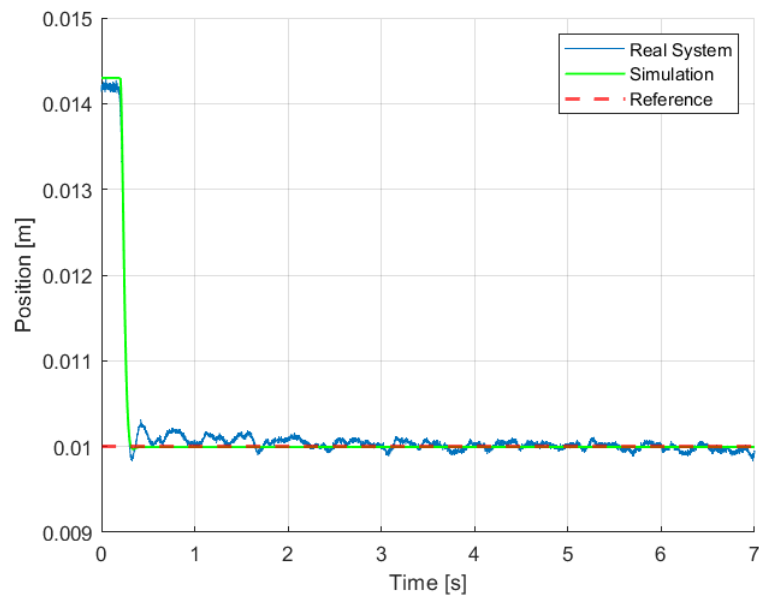


Figure 4.14: Ball position with LQR control (comparing with and without integrator).

We can see that through the use of the integrator the steady state error has been completely removed compared to the previous case.

Chapter 5

State Observers

5.1 Luenberger's Observer

The availability of the full state measurement is very important for:

- Control design (e.g. state feedback techniques)
- Monitoring (fault detection, predictive maintenance, etc.)
- Redundancy of the physical sensor

Now although our system is equipped with sensors through which we can measure position, velocity and current we thought that it can be appropriate to also develop a pair of state observers in order to be redundant in case of failures during the course and to reduce the noise that affects the measurements.

In the following, we'll assume that the system is not equipped with current and velocity sensors, and so we will use the position measurements to estimate the remaining states.

The first observer developed is Luenberger's observer, which can be modeled as:

$$\hat{\mathbf{x}}(k+1|k+1) = \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}u(k) + \mathbf{L}[y(k+1) - \mathbf{H}(\mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{G}u(k))] \quad (5.1)$$

Where \mathbf{L} is the gain of the observer (design parameter). The idea is to use pole assignment to select the observer gain \mathbf{L} in order to assign the eigenvalues of $(\mathbf{F} - \mathbf{L}\mathbf{H}\mathbf{F})$ in a prescribed position. If the eigenvalues of $(\mathbf{F} - \mathbf{L}\mathbf{H}\mathbf{F})$ are asymptotically stable, the state estimation error tends to zero since its dynamics is:

$$\tilde{\mathbf{e}}(k+1|k+1) = (\mathbf{F} - \mathbf{L}\mathbf{H}\mathbf{F})\mathbf{e}(k|k) \quad (5.2)$$

Since the observer is an algorithm, it does not suffer from actuator saturation, so we can choose the eigenvalues of the observer much faster than the eigenvalues of the system.

Since the goal is to use these observer with full state feedback-based controllers, we took the pole-placement controller developed in section (3.2) as a reference point. We can choose the eigenvalues of our observer 10 times faster than those given in (4.2), obtaining the following matrix of gains:

$$\mathbf{L} = \begin{bmatrix} 0.6 \\ 312.6 \\ -2253 \end{bmatrix} \quad (5.3)$$

At this point our observer can be seen as a dynamical system having the following state-space representation:

$$\hat{\mathbf{x}}(k+1|k+1) = (\mathbf{F} - \mathbf{LHF})\hat{\mathbf{x}}(k|k) + (\mathbf{G} - \mathbf{LHG})\mathbf{u}(k) + \mathbf{Ly}(k+1) \quad (5.4)$$

where $\mathbf{u}(k)$ and $\mathbf{y}(k+1)$ are our equivalent inputs and $\hat{\mathbf{x}}(k|k)$ our output. We can then implement the observer on the laboratory system using a Simulink state-space block with the equivalent matrices \mathbf{F} and \mathbf{G} given in equation (5.4),

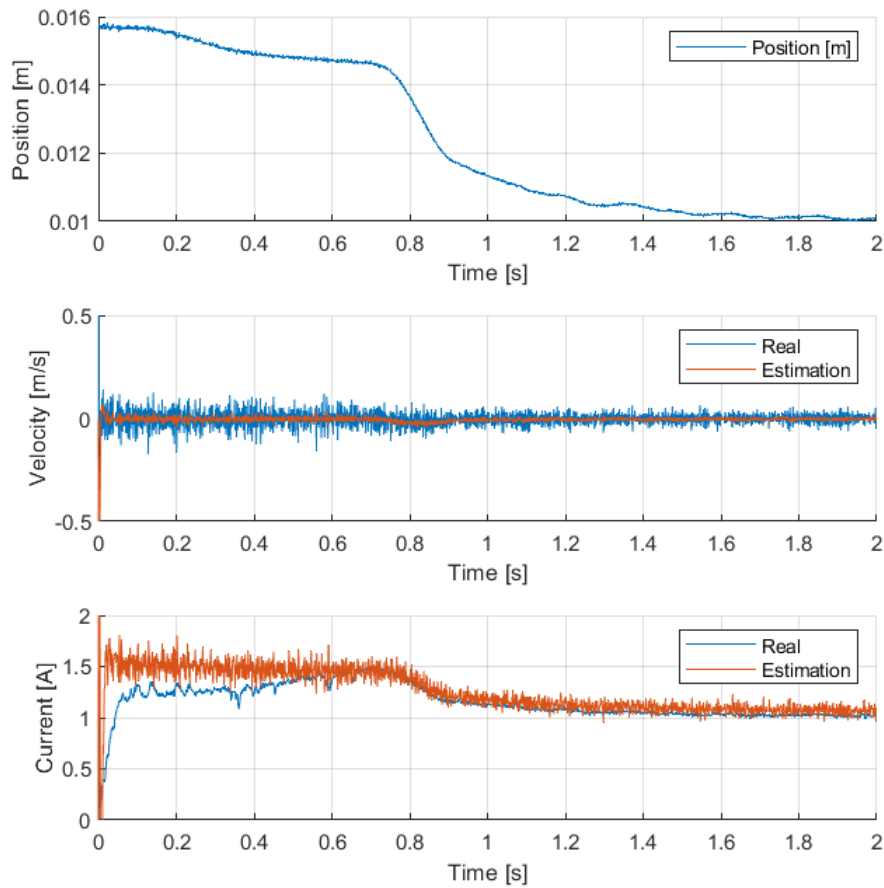


Figure 5.1: Comparison observer estimation and sensor data.

The estimated value of the velocity is accurate within all the duration of experiment even presenting less noise than the sensor. For the current, on the other hand, the estimated value when the sphere is at its equilibrium position is accurate, while the state observer makes an error up to 27 % in the transient. This can be explained by the fact that since the system is highly nonlinear, the transfer function referring to the linearized system at the

equilibrium position is not representative of the dynamics in other positions, even if they are very close to the equilibrium one.

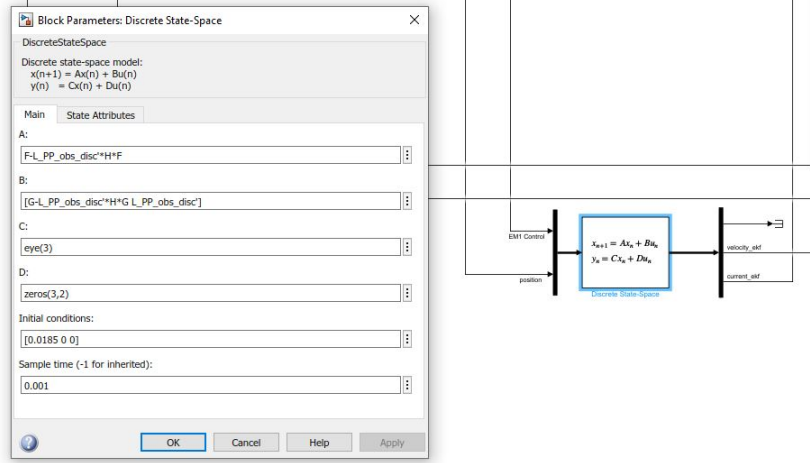


Figure 5.2: Luenberger's observer implementation on the laboratory bench.

5.2 Extended Kalman Filter (EKF)

The observer based on pole-placement seen in the previous section, being designed on the linearized system, manages to provide a good estimation of the velocity and current signals when the sphere is in the equilibrium position, but commits non-negligible errors in the transients which deteriorate the performance of the system if these estimations are used in feedback. For this reason, in case the measurements of current and velocity are not available, it is necessary to use a non-linear states observer like the Extended Kalman Filter (EKF).

Since the non-linear dynamics of our system can be represented as:

$$\begin{cases} \mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{u}(t)) + \mathbf{v}_1(t) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}_2(t) \end{cases} \quad (5.5)$$

We can consider the filter gain as a linear time-varying block $\mathbf{K}(t)$ computed making a local linear approximation of the non-linear system around the 1-step ahead state prediction at each time step. So at each time step our matrices $\mathbf{F}(t)$ and $\mathbf{H}(t)$ will be:

$$\mathbf{F}(t) = \left. \frac{\partial \mathbf{f}(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t+1|t)} \quad \mathbf{H}(t) = \left. \frac{\partial \mathbf{h}(\mathbf{x}(t))}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t+1|t)} \quad (5.6)$$

Once we have our matrices, in the same time step we can use the standard Difference Riccati Equation to find our gain $\mathbf{K}(t)$ and $\mathbf{P}(t+1)$ ready for the next time step.

The equations of the filter are;

$$\begin{cases} \hat{\mathbf{y}}(t+1|t) = \mathbf{H}\hat{\mathbf{y}}(t|t-1) \\ \mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t|t-1) \\ \hat{\mathbf{x}}(t+1|t) = \mathbf{F}\hat{\mathbf{x}}(t|t-1) + \mathbf{G}\mathbf{u}(t) + \mathbf{K}(t)\mathbf{e}(t) \\ \hat{\mathbf{x}}(t|t) = \mathbf{F}^{-1}\hat{\mathbf{x}}(t+1|t) \\ \mathbf{K}(t) = (\mathbf{F}\mathbf{P}(t)\mathbf{H}^T)(\mathbf{H}\mathbf{P}(t)\mathbf{H}^T + \mathbf{V}_2)^{-1} \\ \mathbf{P}(t+1) = (\mathbf{F}\mathbf{P}(t)\mathbf{F}^T + \mathbf{V}_1) - (\mathbf{F}\mathbf{P}(t)\mathbf{H}^T)(\mathbf{H}\mathbf{P}(t)\mathbf{H}^T + \mathbf{V}_2)^{-1}(\mathbf{F}\mathbf{P}(t)\mathbf{H}^T) \end{cases} \quad (5.7)$$

Where \mathbf{V}_1 and \mathbf{V}_2 are the covariance matrices of the model and measurement noises.

The setup is the SISO open loop model (bottom magnet turned off) fed by the duty cycle $\mathbf{y}(t)$, where only the ball position can be sensed. The plant is described by the state-space model:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2K_{em}}{m} \frac{i_{eq}^2}{x_{eq}^3} & 0 & -\frac{2K_{em}}{m} \frac{i_{eq}}{x_{eq}^2} \\ 0 & 0 & -\frac{R}{L(x_{eq})} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{k}{L(x_{eq})} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad (5.8)$$

Using the results obtained in the noise characterization and after several tries on the system, the following values of the noise covariance matrices were determined:

$$\mathbf{V}_1 = \begin{bmatrix} 9 \cdot 10^{-3} & 0 & 0 \\ 0 & 5 \cdot 10^{-4} & 0 \\ 0 & 0 & 10^{-3} \end{bmatrix} \quad \mathbf{V}_2 = 5 \cdot 10^{-9} \quad (5.9)$$

We then designed a Matlab function which, given the position measurement $\mathbf{x}(t)$ and the control action $\mathbf{u}(t)$, outputs the estimation of the states \mathbf{x}_2 and \mathbf{x}_3 (velocity and current). The Matlab code implementation based on 5.7 is listed in the Appendix A.2, while the block scheme implementation on the laboratory system is shown in Figure 5.3.

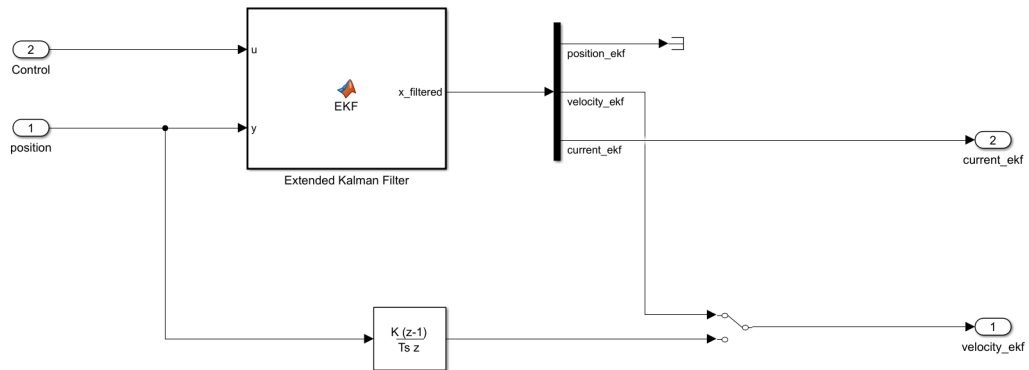


Figure 5.3: Extended Kalman Filter implementation on the laboratory bench.

To compare the accuracy of the estimations with respect to the measurements of the physical sensor, the filter has been tested on the pole-placement control scheme. The results are shown in Figure 5.4.

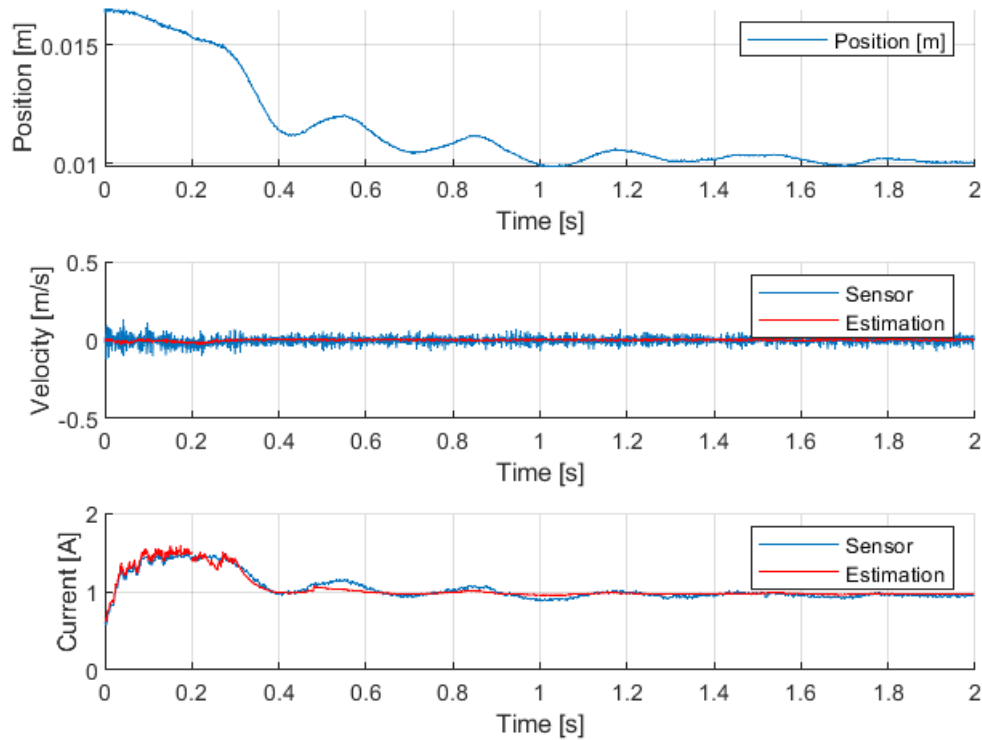


Figure 5.4: Extended Kalman Filter online estimation vs sensor measurements.

From the plot we can see that through the use of EKF it is possible to obtain accurate estimates of un-measurable states even at positions far from the equilibrium one. In addition, it is important to note how the output estimates from the observer are less noisy than those from physical sensors. In conclusion the results are satisfactory as the unknown states are estimated with good accuracy.

Chapter 6

Results Analysis

6.1 Step Response

To get a better view of the differences in the 4 control systems we can plot their step response on the same graph.

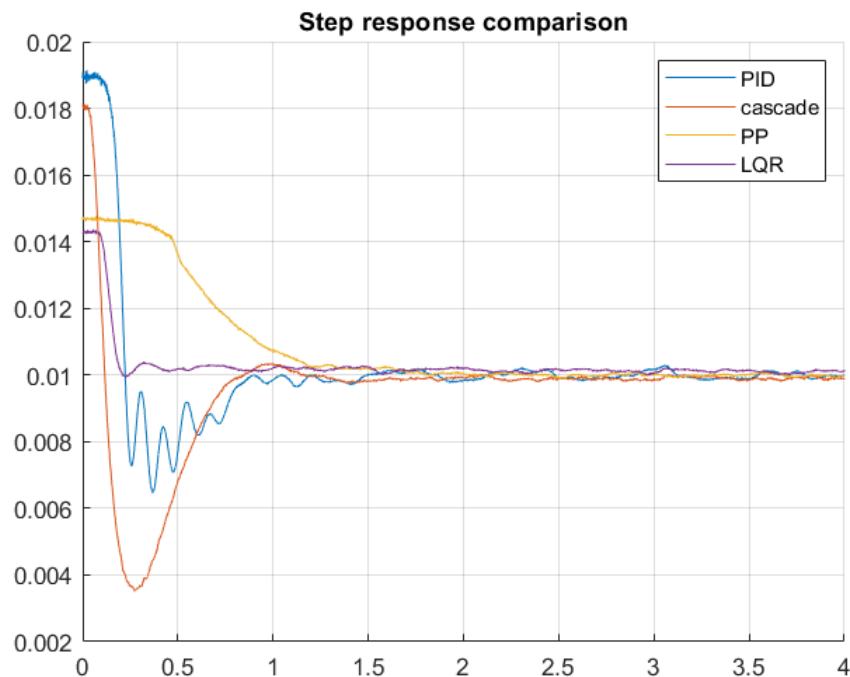


Figure 6.1: Step response comparison between the 4 control systems.

The static and dynamic performances of the 4 control systems developed are reported in the Table 6.1.

Through these data, it is possible to conclude that the control scheme that provides the best trade-off between settling time and overshoot is LQR. Its only disadvantage is robustness; in fact, it fails to stabilize the sphere from the lowest point allowed by the system.

	PID	Cascade	PP	LQR
Settling time	1.406 s	0.7380 s	1.126 s	0.182 s
Overshoot	13.74 %	65.92 %	0 %	2 %
Robustness	Entire stroke	Entire stroke	$[0 - 0.015]$ m	$[0 - 0.015]$ m
Steady-state err.	0	0	0	0

Table 6.1: Static and dynamic performances of the 4 control systems.

6.2 Disturbance Rejection

To compare the performance of the controllers in case of disturbances on the output we used the lower magnet, feeding it with steps with increasing amplitude every 5 seconds.

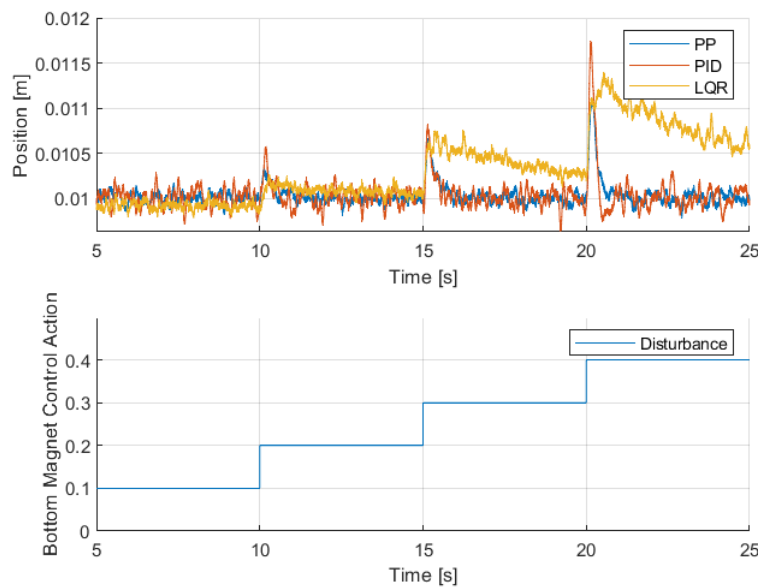
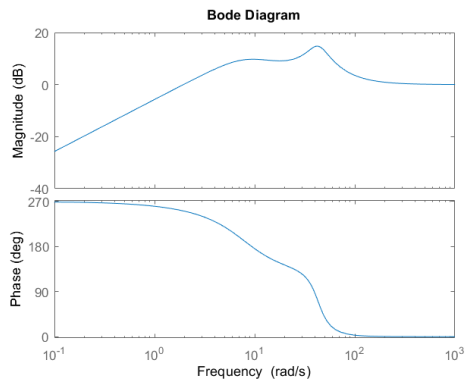
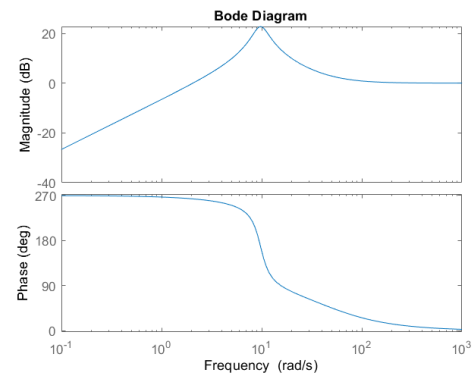


Figure 6.2: Comparison on the disturbance rejection performances of the controllers using the bottom magnet as disturbance.

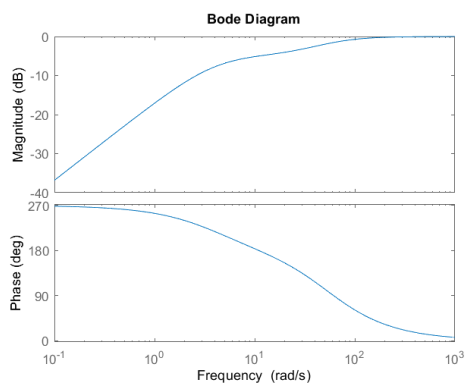
The 4 control systems behave in the same way, managing to reject the disturbance in an acceptable time bringing the sphere back to its equilibrium position. The only controller that fails to completely reject the disturbance is LQR, likely due to the fact that the control action was heavily penalized in the design phase to meet the limits on saturation. A possible way to solve this problem may be to design an observer to estimate constant non-measurable disturbances, so that we can instantaneously compensate it without waiting for the controller to react in feedback.



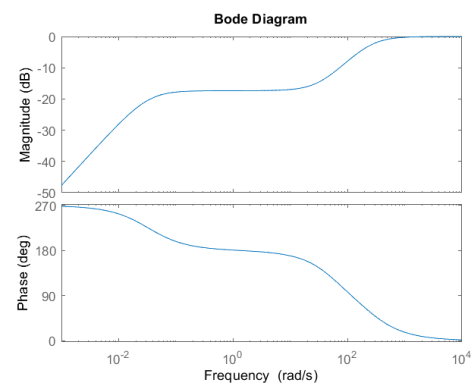
(a) PID.



(b) PID cascade.



(c) Pole-Placement



(d) LQR.

Figure 6.3: Comparison between the sensitivity functions of the 4 control systems.

Appendix A

MATLAB Codes

A.1 RL Identification

```

1 %% RL Identification
2
3 fileInductance = {'Induttanza/Induttanza1', 'Induttanza/
    Induttanza2', 'Induttanza/Induttanza3', 'Induttanza/
    Induttanza4', 'Induttanza/Induttanza5', 'Induttanza/
    Induttanza6', 'Induttanza/Induttanza7'};
4
5 Res = 4; % Resistance
6 V = 11; % Voltage for the experiment (1 duty cycle)
7
8 for i=1:7
9     experim = load(fileInductance{i});
10
11     position = mean(experim.MLS2EMExpData.signals(1).values);
12     curr_ss = mean(experim.MLS2EMExpData.signals(3).values
    (770:800));
13     curr_traj_fit = fit(experim.MLS2EMExpData.time(702:720),
    experim.MLS2EMExpData.signals(3).values(702:720)', '
    smoothingspline');
14     curr_traj_vect = curr_traj_fit(7.01:0.001:7.19);
15
16     for k=1:length(curr_traj_vect)
17         if curr_traj_vect(k) >= curr_ss*0.632
18             tau = k*0.001;
19             break
20         end
21     end
22
23     L(i) = tau*Res;
24     pos_L(i) = position;
25
26 end

```

A.2 Extended Kalman Filter (EKF)

```

1 function x_filtered = EKF(u,y)
2
3 persistent x_pred P Q R H dt BallMass g L pos_L kem_1 pos_1
   Res m
4
5 % First iteration (initialization)
6 if isempty(P)
7     g=9.81;                                % gravity acceleration [m/s^2]
8     BallMass=0.06157;                      % mass [kg]
9     dt=0.001;                              % Sample time [s]
10    m = 11;                                % Duty cycle - Voltage
11    Res = 4;                               % Resistance (Ohm)
12
13    L = [...];                             % Inductance samples
14    pos_L = [...];
15
16    kem_1 = [...];                         % Magnetic Constant samples
17    pos_1 = [...];
18
19    P = eye(3);
20    x_pred = [0.015; 0; 0];
21
22    Q = diag([9e-3 5e-4 e-3]);
23    R = 5e-9;
24
25    H = [1 0 0];
26 end
27
28 % Kem estimation
29 if y>pos_1(1)
30     K1f=kem_1(1);
31 elseif y>=pos_1(end) && y<=pos_1(1)
32     K1f=interp1(pos_1,kem_1,y);
33 else
34     K1f=kem_1(end);
35 end
36
37 % Inductance estimation
38 if y>pos_L(1)
39     Lx0=L(1);
40 elseif y>=pos_L(end) && y<=pos_L(1)
41     Lx0=interp1(pos_L,L,y);
42 else
43     Lx0=L(end);
44 end
45
46

```

```

47 % Linearization of the system around xhat(t+1)
48 xeq = x_pred(1);
49 ieq = sqrt((BallMass*g)/K1f)*xeq;
50
51 a21 = (2*K1f*ieq^2)/(BallMass*xeq^3);
52 a23 = -(2*K1f*ieq)/(BallMass*xeq^2);
53 a31 = 0;
54 a33 = -Res/Lx0;
55 b3 = m/Lx0;
56
57 A = [ 0      1      0;
58       a21    0      a23;
59       a31    0      a33];
60 B = [0; 0; b3];
61
62 % Discretization
63 F = eye(3) + A.*dt;
64 G = B.*dt;
65
66 K = (F*P*H')/(H*P*H'+R);
67
68 y_pred = H*x_pred;
69 x_pred = F*x_pred+K*(y-y_pred)+G*u;
70
71 P = (F*P*F'+Q) - (F*P*H')/(H*P*H'+R)*(F*P*H')';
72
73 x_filtered = F\x_pred;
74
75 end

```

Bibliography

- [1] Inteco: *Magnetic Levitation System 2EM (MLS2EM) User's manual*
- [2] Walter Barie & John Chiasson (1996): *Linear and nonlinear state-space controllers for magnetic levitation*, International Journal of Systems Science, 27:11, 1153-1163.
- [3] Lalo Magni & Riccardo Scattolini (2014): *Advanced and Multivariable Control*.
- [4] Sergio Matteo Savaresi (2023): *Model Identification and Data Analysis 2* (Lecture Notes).
- [5] Paolo Bolzern, Riccardo Scattolini & Nicola Schiavoni (2015): *Fondamenti di controlli automatici*.