



**ENSTA  
BRETAGNE**



# Internship : Magnetic Levitation System and Ellipsoid State Estimator

Carl von Ossietzky  
**Universität  
Oldenburg**

2<sup>nd</sup> year internship

**September 30, 2022**

## **Abstract**

Magnetic levitation is a growing technology in the industry, as it allows for swift motions without friction caused by the support. The goal of this presentation is to give a brief overview of the system and the mathematical model used, as well as a study on stabilization and feedback control laws for this system. The final objective of this study is to find a stable and robust control law design for a variety of different desired outputs, first with standard methods, and then with the use of interval- or ellipsoid-based state estimators, and making a comparison between the performance of the different methods.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Aim of this Work . . . . .	2
1.3	Structure of this paper . . . . .	2
<b>2</b>	<b>The System and Model</b>	<b>3</b>
2.1	System Overview . . . . .	3
2.2	Sensors . . . . .	3
2.2.1	Position sensor . . . . .	3
2.2.2	Current sensor . . . . .	4
2.3	Equations . . . . .	5
<b>3</b>	<b>Simulation</b>	<b>6</b>
3.1	First Method : Documentation Simulation . . . . .	6
3.1.1	Parameters identification using the simulation . . . . .	6
3.1.2	Parameters identification using the test bench . . . . .	7
3.2	Second Method : Step by Step identification . . . . .	8
3.2.1	First step : Current equation . . . . .	8
3.2.2	Second Step : Electromagnetic force and Newton's second law . . . . .	9
<b>4</b>	<b>Observers and Filters</b>	<b>11</b>
4.1	Observers . . . . .	11
4.1.1	Standard Observer . . . . .	11
4.1.2	Kalman Filter/Observer . . . . .	11
4.2	Filters . . . . .	12
4.3	Results . . . . .	13
4.4	Control Laws . . . . .	13
<b>5</b>	<b>Ellipsoids</b>	<b>16</b>
5.1	Ellipsoids State Estimation . . . . .	16
5.2	Algorithms . . . . .	16
5.3	C++ implementation . . . . .	18
<b>6</b>	<b>Conclusion</b>	<b>22</b>
	<b>Appendices</b>	<b>25</b>

# Chapter 1

## Introduction

### 1.1 Motivation

Magnetic levitation (MagLev) or magnetic suspension, is a method by which an object is suspended with no support other than magnetic fields. Magnetic force is used to counteract the effects of the gravitational force and any other forces. The two primary issues involved in magnetic levitation are lifting forces, providing an upward force sufficient to counteract gravity, and stability, ensuring that the system does not spontaneously slide or flip into a configuration where the lift is neutralized. Magnetic levitation have very high expectations in industry, as it allows for movements without friction, for example a MagLev train for a unidirectional translation movement, or a magnetic bearing for unidirectional rotation. Due to Earnshaw's theorem, static stability in magnetic levitation is impossible in most common use cases. This implies the necessity to use dynamic stability using various ways, and most of the time we use electromagnets and micro-controllers as it allows for more precision. For MagLev systems, the output performance in comparison to standard systems, allows for a higher precision in positioning, dynamic tuning during the movement for better stability, especially against vibrations, and finally they can support very high velocity outputs. As of now, MagLev technology is very restrictive and not very present in the industry, as it require heavy infrastructures for high end performances, therefore being costly for the benefits it provides.

### 1.2 Aim of this Work

We will test a one electromagnet system stability against various commands and find the best control law for a robust and precise system. In this paper, we are going to use different command models and algorithms, as well as filters to improve stability and robustness for our Magnetic bench. Another aim of this paper, is a first attempt to use a real time implementation of an ellipsoid based approach state estimation, from interval methods. The end goal of this paper is to give a comparative study of different ways to implement a MagLev command laws, and to find the one with the best results.

### 1.3 Structure of this paper

This paper will be divided in three main parts. The first part will be the study of the given system, which includes various test of the test bench captors and reactions to inputs, as well as an in-depth study of the chosen mathematical model and the different consequences it could have on the stability and precision. The second part of this paper would be about the study of standard command laws and filters effects on the performances of our system. The final part will then be about the real time implementation of a C++ algorithm for an ellipsoid based approach of the system state, and the corresponding control law.

# Chapter 2

## The System and Model

### 2.1 System Overview

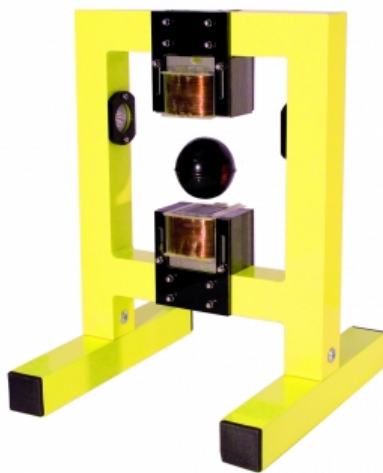


Figure 2.1: Inteco MagLev bench

We are using the Inteco MagLev bench in this paper, shown in figure 2.1. The state equations of this system are described as :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{F_{em}}{2m} + g \\ \dot{x}_3 = \frac{1}{f_{ip}}(k_i \cdot u + c_i - x_3) \end{cases} \quad (2.1)$$

With  $F_{em} = \frac{F_{emP1}}{F_{emP2}} \cdot x_3^2 \cdot e^{-\frac{x_1}{F_{emP2}}}$  and  $f_{ip} = \frac{f_{ip1}}{f_{ip2}} \cdot e^{-\frac{x_1}{f_{ip2}}}$ , where  $x_1$  is the position of our ball,  $x_2$  is the velocity of the ball and  $x_3$  is the current inside the coil. This is the system as described in the documentation, and the model we will be using for reference in the following report. The system comes with a made PID controller, a few calibrations procedures and a simulation, and from this model, we can start by looking at a few particularities.

## 2.2 Sensors

### 2.2.1 Position sensor

The sensor for the ball position is a light sensor, who is sensitive to outside light, so here is a graph showing the evolution of the position measured for a fixed ball and a varying ambient light. From figure 2.2, for a given luminosity, the position is accurate with a  $\pm 0.02$  mm error. The exterior light can add a

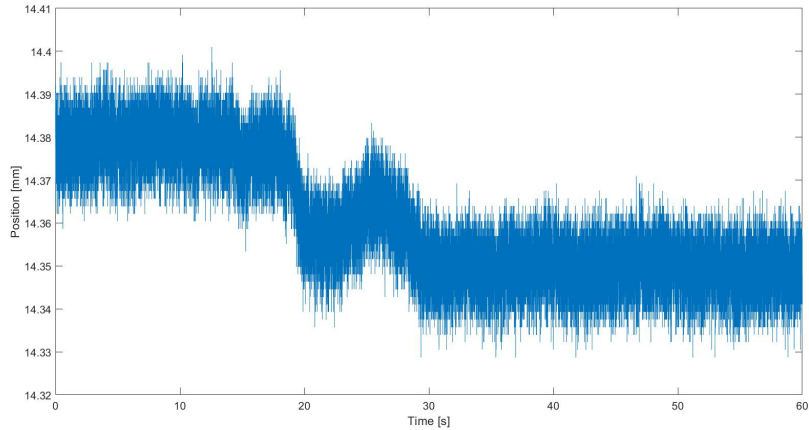


Figure 2.2: Extensive position sensor test

$\pm 0.1$  mm offset to the actual measure, which might need to be taken into account during the calibration of the sensor, or during extensive tests on the precision.

### 2.2.2 Current sensor

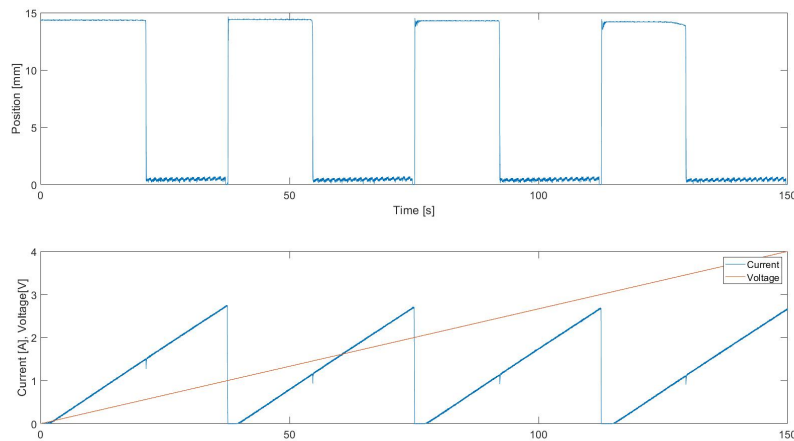


Figure 2.3: Extensive current reaction test (1)

From a ramp input, we can also study the current response, as the input  $u$  in volt is converted into PWM. For an input between 0 and 5 V, we can see that the PWM amplitude is about 1 V, and that under a certain voltage, the current remains null (Figure 2.3). We can also see the influence of the ball movements, as it makes small variations visible on the graph. From the current response, we can estimate values for the differential equation parameters (later in the report).

With a better view of the current curve, for a stable input  $u = 0.2V$ , we can see that the spikes are caused by the ball hitting the electromagnet and the ball leaving the electromagnet. From this test, we can conclude that as long as the ball do not touch the electromagnet, the current should behave according to the input  $u$ . This effect could also be the reason why the system is unstable when the ball hit the magnet. We can also have the measure of the noise in the current measure, which is around  $\pm 0.25A$  (Figure 2.4).

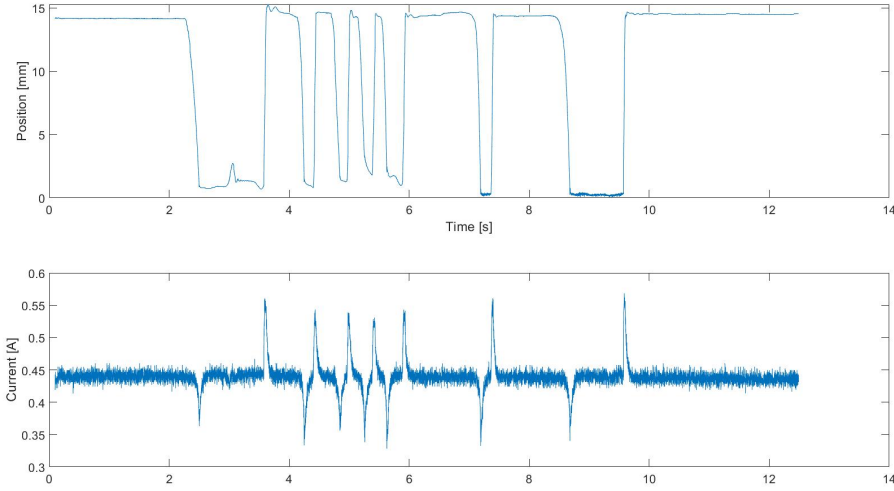


Figure 2.4: Extensive current reaction test (2)

## 2.3 Equations

From the state equations above, and the measure of the maximum current output, we can deduce some useful information by studying the limit cases of the system. According to the equations, for a stable ball at a constant distance from the electromagnet, we have :

$$\dot{x}_2 = -\frac{F_{em}}{2m} + g = 0 \iff F_{em} = 2mg \iff x_1 = k_2 \cdot \ln\left(\frac{2mg}{k_1 \cdot x_3}\right)$$

With  $k_1 = \frac{F_{emP1}}{F_{emP2}}$  and  $k_2 = \frac{-1}{F_{emP2}}$ , assuming  $x_3$  and  $x_1$  are constant. For  $x_{3max} \approx 2.74$  A, we obtain  $x_1 \leq 17$  mm, with the other parameters according to the constructor. From that, we now have a formula to find the limit of the electromagnet to lift the ball.

From the same limit case, we can also find the voltage input necessary for a certain desired position, and the "Equilibrium" voltage at  $t_\infty$  for a constant desired output is obtained by :

$$u_e = \frac{\sqrt{\frac{g \cdot m}{k_1 \cdot e^{k_2 \cdot x_1}} - c_i}}{k_i}, \text{ as an example, for } x_1 = 10 \text{ mm, we have } u = 0.38 \text{ V.}$$

We can use that approximation as a way to validate our state equations using data from experiences.

A quick overview of the electromagnetic force expression allows us to see the system can not have static stability. For a constant current, the electromagnetic force is still dependant of the ball position. If we suppose a stable position for a given current input, the electromagnetic force equation can be expressed as :  $F_{em} = \alpha \cdot e^{-\beta \cdot x_1}$  with  $\alpha$  and  $\beta$  constants. This expression is statically unstable for the position  $x_1$ . The stability of magnetic levitation using only one electromagnet can therefore only be achieved using dynamic stability.

# Chapter 3

## Simulation

The first and forward method to improve the stabilisation and precision, is to use a simulation to find optimal PID coefficients. For the optimisation, we use a custom error function, based on the mean square error between the simulation result and the desired output, and the `fminsearch` function of Matlab. This part was also inspired by the work of Peter Balko and Danica Rosinová [1].

### 3.1 First Method : Documentation Simulation

#### 3.1.1 Parameters identification using the simulation

By using the provided simulation with Simulink and the provided internal parameters from the documentation, we can therefore run optimisations on the PID controller to find the best possible combination for a fast and stable response. The first objective is to make an accurate simulation, and by using the same optimization method, we can make an estimation of the internal parameters of the test bench. For the

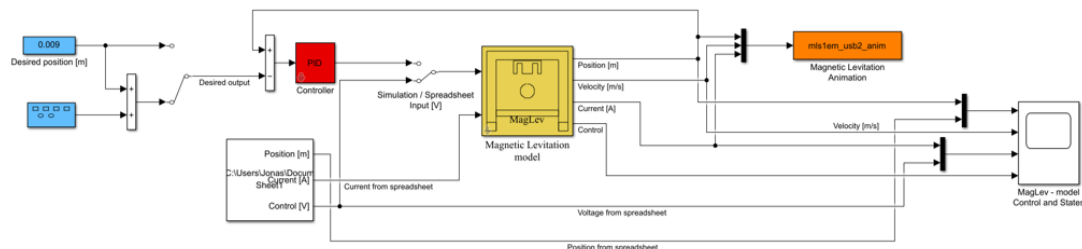


Figure 3.1: Maglev Simulation model from documentation

identification of the internal parameters, we used two different methods. The first method is to use the same generated desired output, which correspond to the blue blocks in Figure 3.1. By using the same PID coefficients and sample time, as well as the adequate initial conditions, we can compare the result of our simulation with the results measured on the test bench. The second method, is to use the voltage and current data from a test run and directly using either of them as the input in our simulation.

From those results, we can see that the coefficients found by the simulation, displayed in the table 3.1, are correct in respect of the parameters from the documentation. In addition, the PID coefficients values are in range of what was expected from the previous coefficients in the simulation, but they are far from the ones used for the test they were optimized on. The simulation results are shown in Figure 3.2 and 3.3.



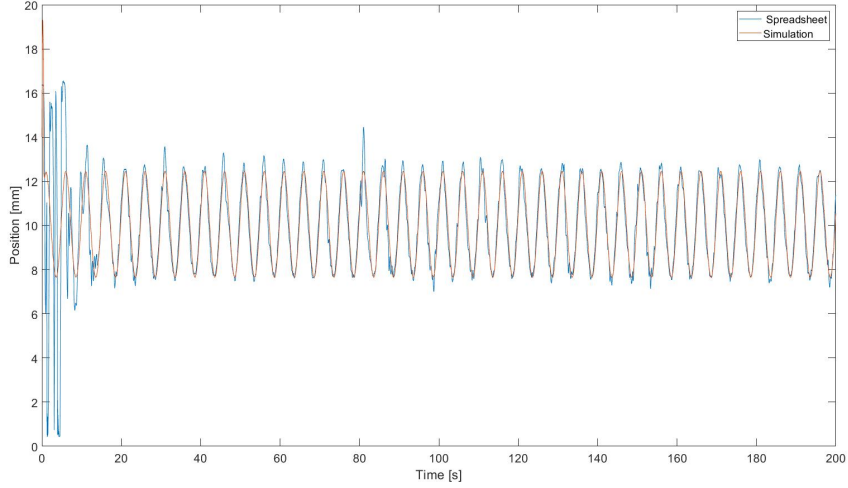


Figure 3.2: Internal parameters optimization result

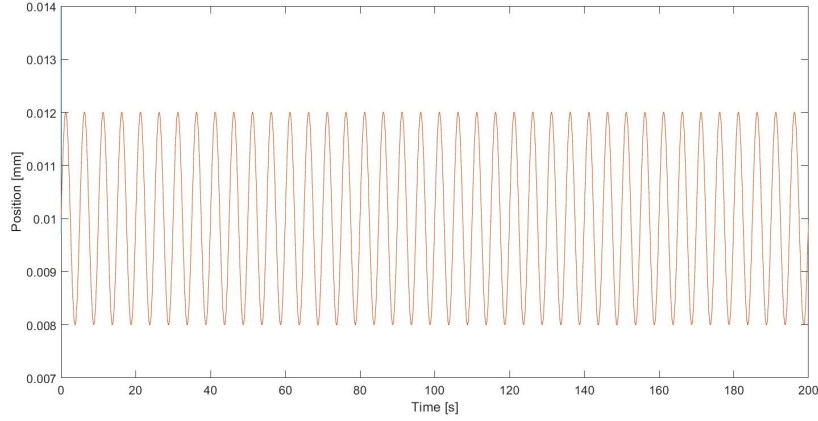


Figure 3.3: PID optimization result

Unfortunately, as shown in Figure 3.4, the system is not stable and the coefficients are not working for the test bench. The ball can not stabilize around the desired output (10 mm), and even worse, the ball is bumping against the electromagnet. This could be explained by a variety of differences between the test bench and the simulation model, or by the noise of the sensors, but we will go more in detail in the next section. From these results, we can however question the simulation model from the documentation that does not seem to be accurate.

### 3.1.2 Parameters identification using the test bench

We can also try to identify the internal parameters using the state equations and data acquired from the test bench. From the equations, we have :  $F_{em} = 2mg = k_1 \cdot \bar{x}_{3,i}^2 \cdot e^{k_2 \cdot \bar{x}_{1,i}}$ .

We can then find an expression of  $F_{emP1}$  and  $F_{emP2}$  :  $F_{emP2} = \frac{\bar{x}_{1,i} - \bar{x}_{1,j}}{2 \ln(\frac{\bar{x}_{3,i}}{\bar{x}_{3,j}})}$  and

$$F_{emP1} = \frac{2mg}{\bar{x}_{3,j}^2 \cdot e^{k_2 \cdot \bar{x}_{1,i}}}$$

For this set of data, shown in tables 3.2 and 3.3 we get :  $F_{emP1} \approx 6.64 \cdot 10^{-2}$  H and  $F_{emP2} \approx 5.691 \cdot 10^{-3}$  m. The values for both parameters are in range from the documentation, the average value for  $F_{emP2}$  is very close from the one in the documentation, but not  $F_{emP1}$ . Using these results, we can

	$m$	$F_{emp1}$	$F_{emp2}$	$f_{ip1}$	$f_{ip2}$	$k_i$	$c_i$	$K_P$	$K_I$	$K_D$
$Z_{ini}$	0.0571	0.0175	0.0058	0.0001414	0.0046	2.516	0.0243	130	500	6
$Z_{opti}$	0.0571	0.0204	0.00745	0.0001574	0.005029	2.847	0.0194	1124.67	9989.1	17.27

Table 3.1: Coefficients comparison between documentation and optimization

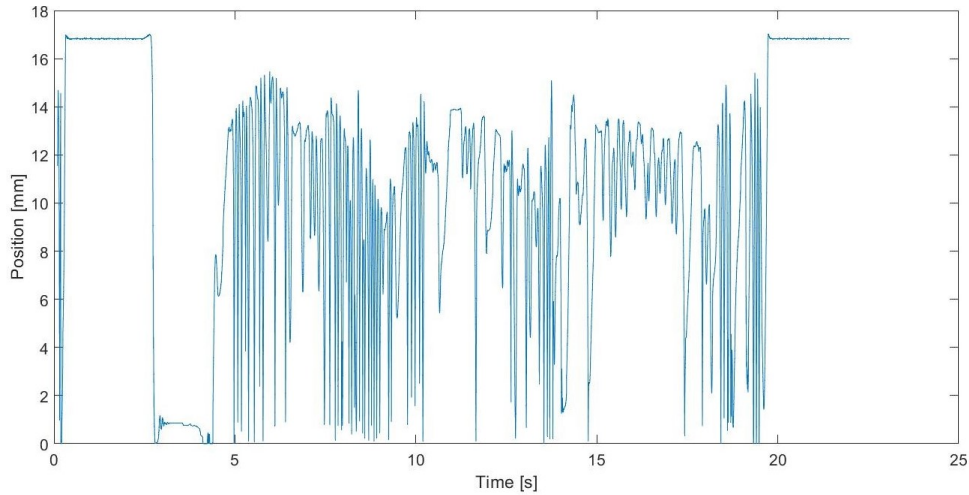


Figure 3.4: Run with the optimized PID coefficients

use them as a starting point to find the missing internal parameters. Unfortunately, the last optimization did not yield any positive result, as shown in Figure 3.5.

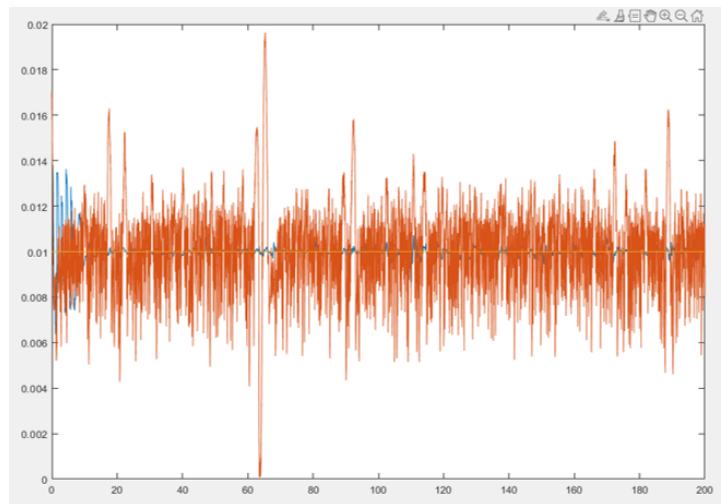


Figure 3.5: Optimization of the last parameters

## 3.2 Second Method : Step by Step identification

### 3.2.1 First step : Current equation

From the state equations 2.1 and data from experiments, we can start by fitting the current curve of the simulation on the measured one, as shown in Figure 3.6. The fitting was done on the four parameters,  $f_{ip1}$ ,  $f_{ip2}$ ,  $k_i$  and  $c_i$  with the cost function set as the mean square error between the simulated current

	Test n°1	Test n°2	Test n°3	Test n°4	Test n°5
$\bar{x}_{1,i}$ , [mm]	10	12	8	9	11
$\bar{x}_{3,i}$ , [A]	0.7476	0.8820	0.6045	0.6987	0.8306

Table 3.2: Test values for position and current

$F_{emP2_{i,j}}$ , $10^{-3}$ m	Test n°1	Test n°2	Test n°3	Test n°4	Test n°5
Test n°1	/	6	4.7	7.4	4.7
Test n°2	-	/	5.3	6.4	8.3
Test n°3	-	-	/	3.5	4.7
Test n°4	-	-	-	/	5.8

Table 3.3: Electromagnetic force parameters results

value of our model and the measured values. We can see very clearly that the results are very good, that the estimated current inside the coil is pretty accurate, and the uncertainties are reduced. The simulation model of the coil that we assumed in the beginning can therefore be validated. This adjusted current model can therefore be used for the next part of the simulation optimization process.

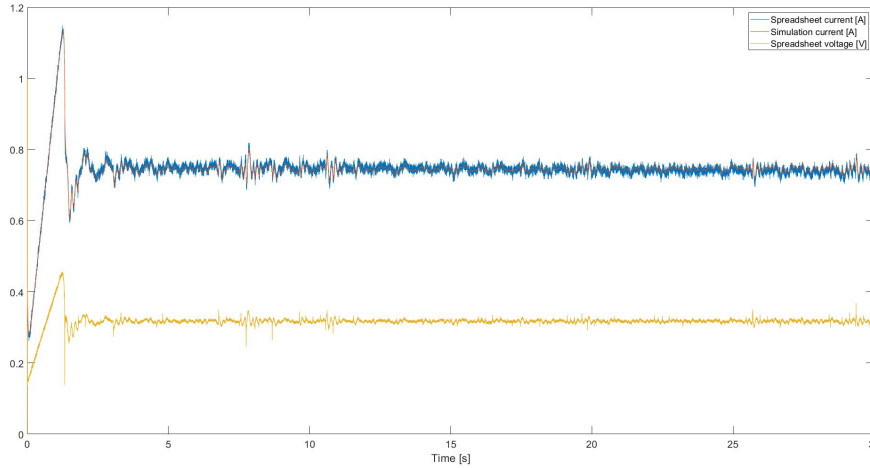


Figure 3.6: Optimization of the current state equation

### 3.2.2 Second Step : Electromagnetic force and Newton's second law

From the previous step, we managed to find accurate parameters for  $f_{ip1}$ ,  $f_{ip2}$ ,  $k_i$  and  $c_i$ . From this valid current simulation, we can add the remaining state equations, to estimate the actual Electromagnetic force and for the application of Newton's second law to find the acceleration, speed and position of the ball. The results are shown in Figure 3.7. The actual simulation part for the estimation of the ball trajectory is not working properly, as we have issues to find the real trajectory, and it could be because of three different things : The support of the frame that is not properly implemented in the simulation, as based on the captors, it is very hard to tell when the ball is resting on the support, the initial conditions of the speed and the position of the ball, that both have an impact on the trajectory, and lastly, the uncertainties in the measures from the sensor, that could make it hard for the optimization algorithm to find suitable parameters.

The simulation model given by the documentation has a satisfying behavior given a set of inputs, however when compared to the test bench measurements, the outputs are not accurate. This implies that either the internal parameters of the system are wrong, even after optimizations and identifications,

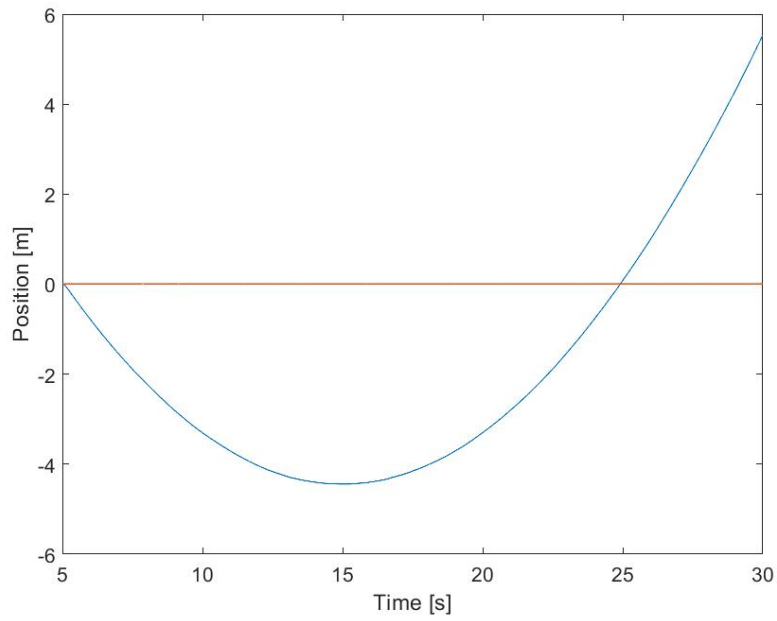


Figure 3.7: Optimization of the electromagnetic force parameters

that some perturbations were neglected when they do have quite an impact on the actual test bench, or that just the noise of the sensors are enough to completely break the algorithms we used to find the internal parameters. As the simulation can not be used to optimize the controller offline, we are going to use other methods to improve the performances.

# Chapter 4

## Observers and Filters

### 4.1 Observers

#### 4.1.1 Standard Observer

The first observer that we used was a very simple observer, based simply on integrators, as shown in Figure 4.1. The use of such observer was to obtain a better estimation for the speed and acceleration of the steel ball, and just as before we also used an optimization method for the three coefficients  $h_1$ ,  $h_2$  and  $h_3$ . Unfortunately, we could not use this observer efficiently as it was too slow to give accurate values on time.

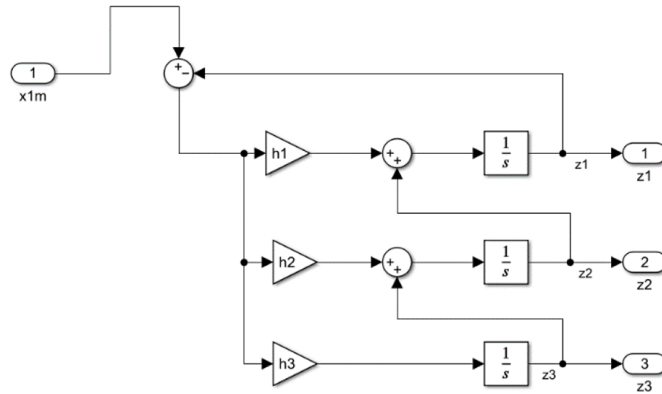


Figure 4.1: Standard Integrators Observer Model with Simulink

#### 4.1.2 Kalman Filter/Observer

To improve on that, we decided to implement a Kalman filter based on linear equations of our system :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{F_{em} - d_1}{2m} + g \\ \dot{x}_3 = \frac{1}{f_{ip}}(k_i \cdot (u - d_2) + c_i - x_3) \end{cases}$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{d}_1 \\ \dot{d}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{F_{em}}{2m} & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{f_i(\tilde{x}_1)} & 0 & \frac{k_1}{f_i(\tilde{x}_1)} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \tilde{x} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{k_1}{f_i(\tilde{x}_1)} & 0 & \frac{1}{f_i(\tilde{x}_1)} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{bmatrix} u \\ g \\ c_1 \end{bmatrix}$$

with  $\tilde{x}_1$  as the desired position, and with  $F_{em} = k_1 \cdot \hat{x}_3^2 \cdot e^{k_2 \tilde{x}_1}$  and  $f_{ip} = \frac{f_{ip1}}{f_{ip2}} \cdot e^{-\frac{\tilde{x}_1}{f_{ip2}}}$ . From this equations and a set of data, we could again try an optimization of the different internal parameters for the Kalman filter, to obtain the best fidelity possible. By doing that we get the two graphs shown in Figures 4.2 and 4.3, which represents a comparison between the data sheet values and our Kalman filter.

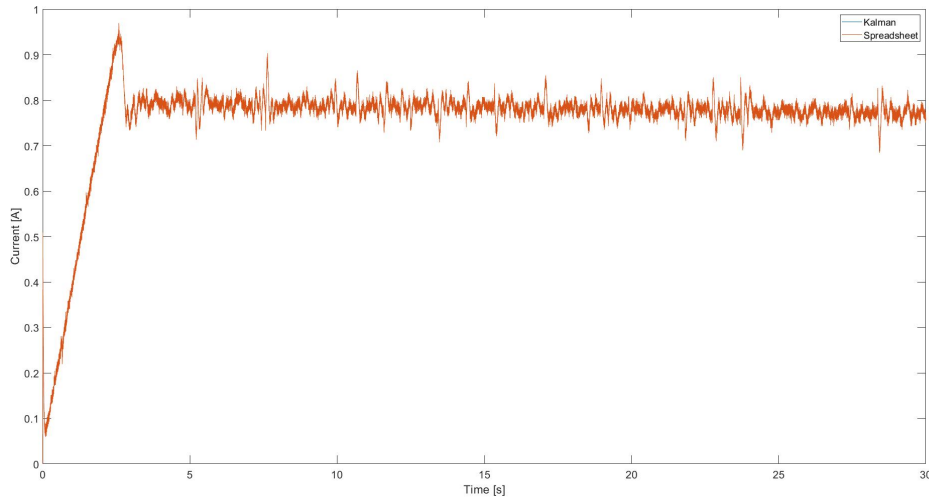


Figure 4.2: Current from the Kalman Filter

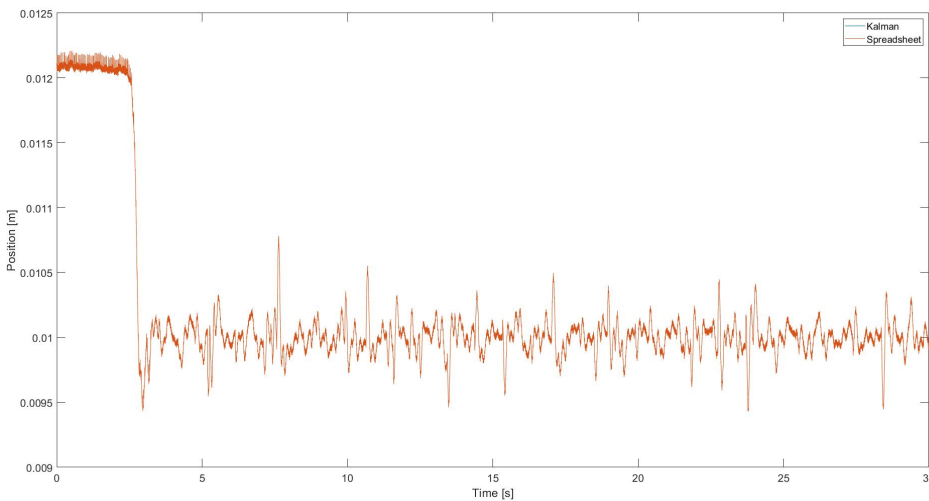


Figure 4.3: Position from the Kalman Filter

## 4.2 Filters

For the filters, we tried two kinds of filters, the first one is a low-pass filter and the second one is a mean filter. The low-pass filter was setup on the frequency to smooth out the sensors uncertainties. This filter worked well, but due to the computation time needed, it's utilisation in a numerical model was disregarded. For the mean filter, as in Figure 4.4, we kept the last five measurements, and the results are shown below.

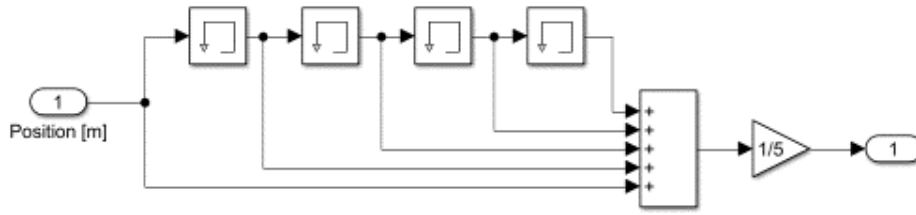


Figure 4.4: Mean Filter representation in Matlab Simulink

	Standard Deviation	MSE
PID only	0.24561	22247.17
Kalman Filter	0.22855	21603.80
Mean Filters	0.09835	20918.48
Kaman + Mean Filters	0.07640	19074.31

Table 4.1: Comparison of the filters

### 4.3 Results

Using all those tools at our disposition, we tried various configurations as shown in the Figures below. For all of the experiments, the PID values and the other various parameters were exactly the same.

For a better evaluation of the performance, we have the set of results shown in the table 4.1. The

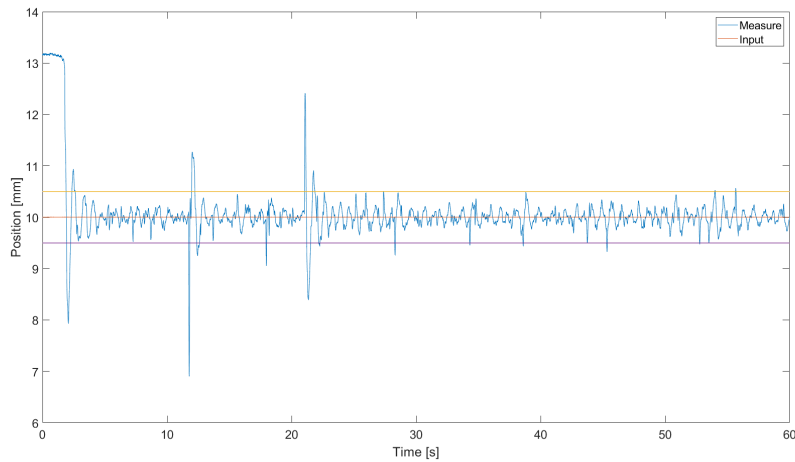


Figure 4.5: Standard PID

standard deviation is calculated over the stable period of the system, to get an idea of the stability. The Mean Squared Error (MSE) is calculated over the span of the entire test, for an idea on the precision and reactivity.

From table 4.1, we can clearly see that the mean filter greatly improves precision and stability over time, and that the Kalman also helps a little over time.

### 4.4 Control Laws

In addition to filters and observers, we also tried improved command laws using feedback linearization. The two command laws are written below, the first one being for the standard equations and the second

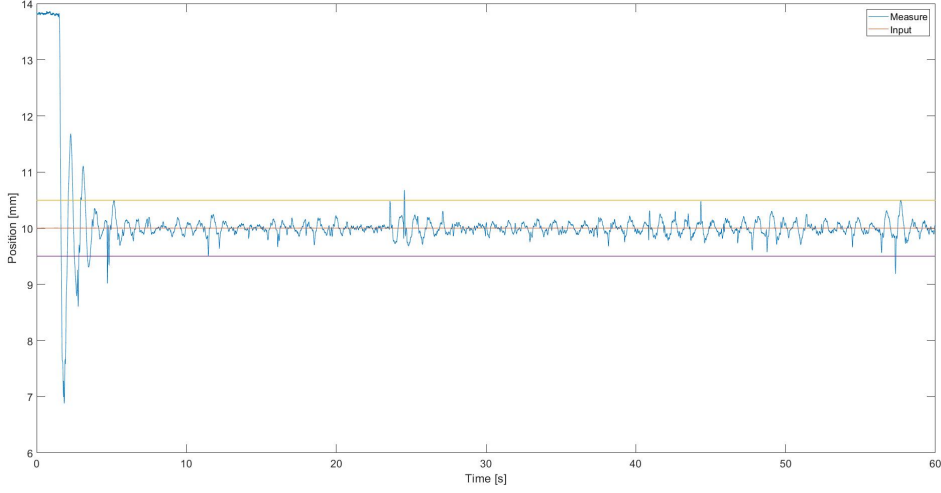


Figure 4.6: Standard PID and Kalman Filter

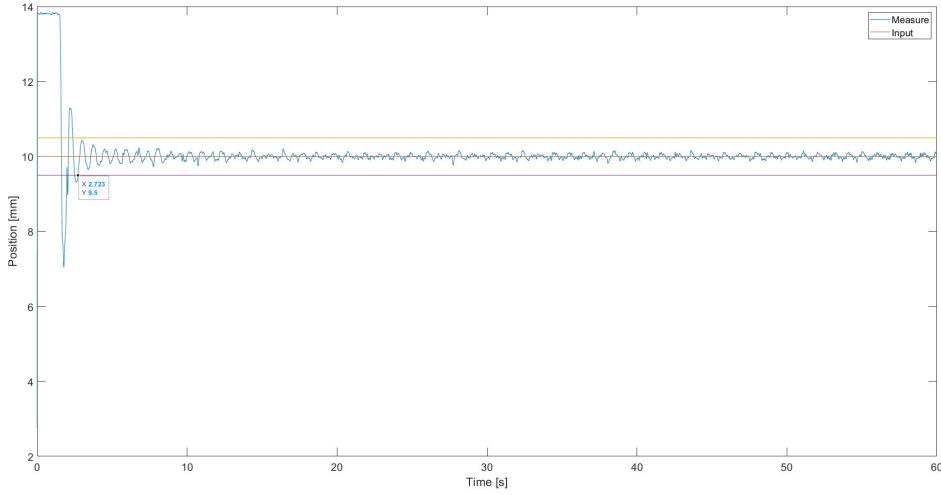


Figure 4.7: Standard PID and Mean Filter

one for the Kalman state equations from before.

$$u = \begin{cases} \frac{m \cdot f_i(x_1) \cdot e^{-k_2 \cdot x_1}}{k_1 \cdot x_3} \cdot v - \frac{f_i(x_1) \cdot k_2}{2} \cdot x_2 \cdot x_3 - \frac{1}{k_i} (c_i - x_3) \\ \frac{m \cdot f_i(\hat{x}_1) \cdot \hat{x}_3}{k_i (F_{emP1} \hat{x}_3 + \hat{d}_1)} \cdot v - \frac{f_i(\hat{x}_1)}{2 \cdot k_i \cdot F_{emP2}} \cdot \hat{x}_2 \cdot \hat{x}_3 - \frac{1}{k_i} (c_i - \hat{x}_3 + k_i \hat{d}_2) \end{cases}$$

The  $v$  function represent the PID implementation, were the  $K_I, K_P, K_D$  and  $K_{D2}$  coefficients can be found by using a pole placing method. Those coefficients can also be optimized using algorithms and numerical methods for better results.

$$v = K_I * \int_0^t (w - x_1) \cdot dt + K_P * (w - x_1) + K_D * (\dot{w} - x_2) + K_{D2} * (\ddot{w} - \dot{x}_2) + w^{(3)}$$

The comparison between the standard PID control method and our method can be seen in Figure 4.9. Unfortunately, this new control law could not be implemented in a satisfactory manner, as during real time experiments, the control law was way too slow to react at first, and then became completely chaotic.



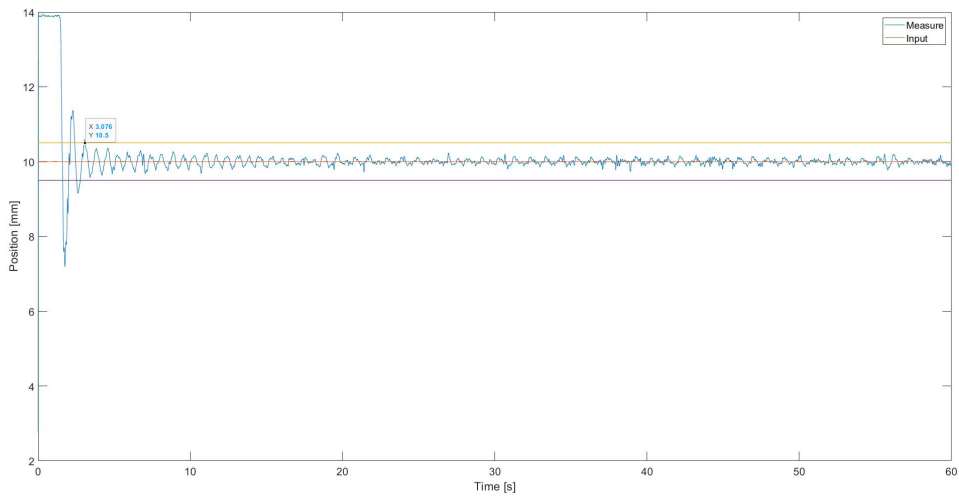


Figure 4.8: Standard PID, Kalman and Mean Filter

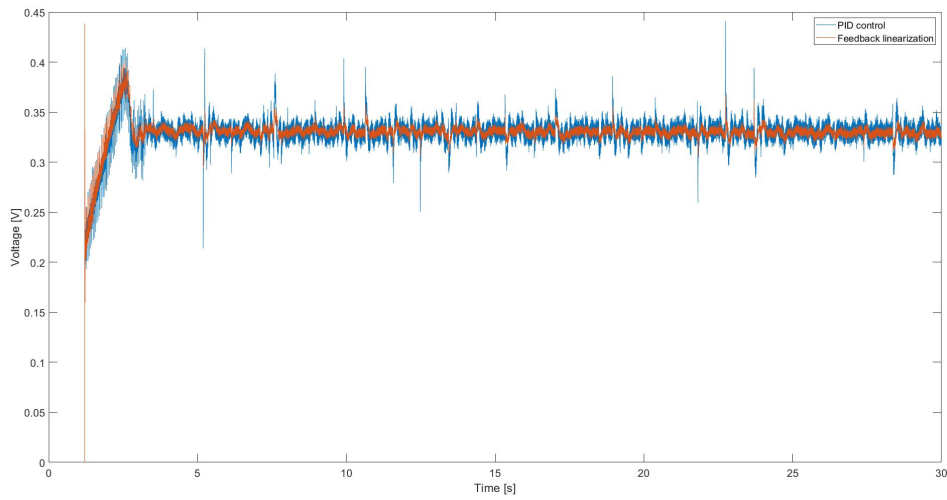


Figure 4.9: PID and Feedback linearization control laws

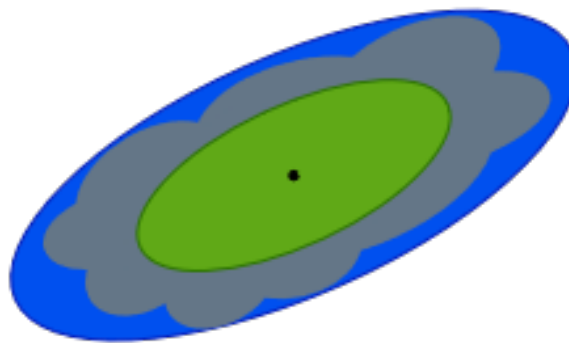
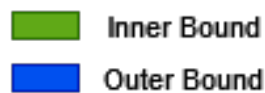
# Chapter 5

## Ellipsoids

The main goal of this work was to implement a real-time ellipsoid state estimator for the MagLev system, using the C++ language. Unfortunately, the interface between the C++ algorithm and the test bench is not implemented yet, but all the results will be shown below.

### 5.1 Ellipsoids State Estimation

As shown in Figure 5.1, a system state is represented by a center and two bounds, following the interval arithmetic norm. Making a state estimator using ellipsoids (Figure 5.2) only require to adapt the different algorithm to perform the different steps.



**Thick Ellipsoid State Enclosure (2D)**

Figure 5.1: Thick Ellipsoid representation

### 5.2 Algorithms

The first algorithm is the one used to predict the next state of our system, based on the actual state and the system equation. In our cases, we used the Euler discretised equations.

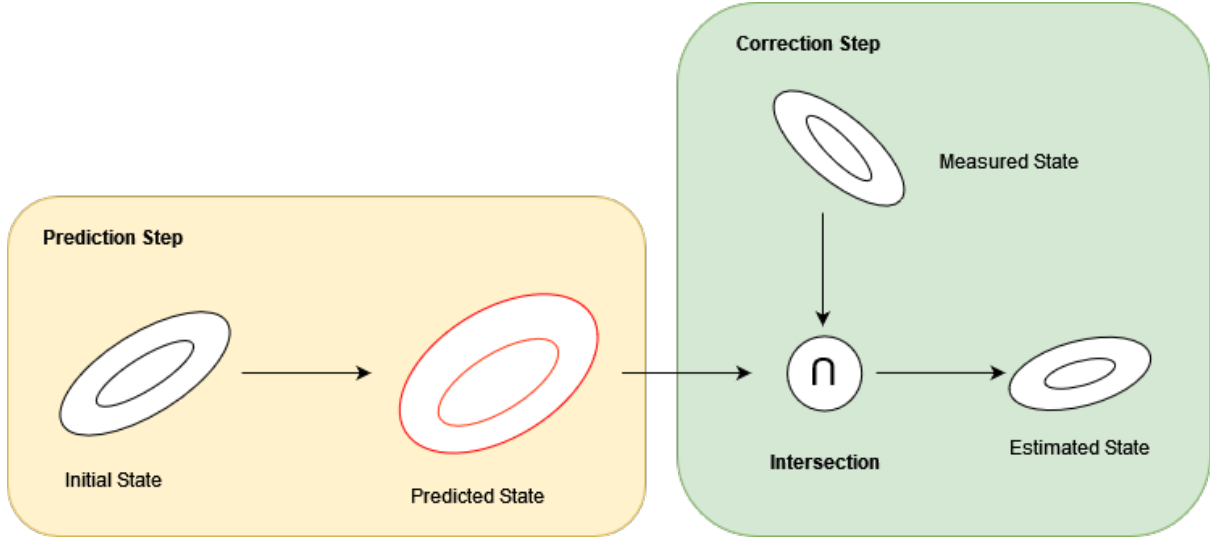


Figure 5.2: Ellipsoid state estimator

*Ellipsoid State Prediction algorithm [2]:*

*Inputs :*  $f(x), \mu_x, \Gamma_x$

*Outputs :*  $\mu_y, \Gamma_{y,O}, \Gamma_{y,I}$

- 1 :  $[x_i] = \mu_{x,i} + \|\Gamma_{x,i}\| \cdot [-1; 1], \quad i \in [1, \dots, n]$
- 2 :  $[\mathbf{x}] = [x_1] \times \dots \times [x_n]$
- 3 :  $\mathbf{A} = \frac{\delta f}{\delta x}(\mu_x)$
- 4 :  $\Gamma_y = \mathbf{A} \cdot \Gamma_x$
- 5 :  $[\mathbf{J}_f] = \begin{bmatrix} \delta f \\ \delta x \end{bmatrix}([x])$
- 6 :  $[\tilde{b}] = (\Gamma_y^{-1} \cdot [\mathbf{J}_f] \cdot \Gamma_x - I) \cdot [-1; 1]_{\times n}$
- 7 :  $\rho = \sup(\|\tilde{b}\|)$
- 8 :  $\mu_y = f(\mu_x)$
- 9 :  $\Gamma_{y,O} = (1 + \rho) \cdot \Gamma_y$
- 10 :  $\Gamma_{y,I} = (1 - \rho) \cdot \Gamma_y$

The second algorithm is for the correction step, and is responsible for the incorporation of the measurements in the overall state estimator.

*Ellipsoids Intersection Algorithm [3]:*

*Inputs :*  $Q_1, Q_2, \mu_1, \mu_2$

*Output :*  $\mu_y, Q_O, Q_I$

$$\begin{aligned}
1: & \quad L = Q_1 \cdot (Q_1 + Q_2) \\
2: & \quad \mu_y = \mu_1 + L \cdot (\mu_2 - \mu_1) \\
3: & \quad \Delta\mu_i = Q_i^{-1/2} \cdot (\mu_y - \mu_i), \quad \text{for } i \in [1, 2] \\
4: & \quad \xi_i = 1 - \|\Delta\mu_i\|, \quad \zeta_i = 1 + \|\Delta\mu_i\| \quad \text{for } i \in [1, 2] \\
5: & \quad Q_I = 2 \cdot (2 \cdot ((\xi_1^2 \cdot Q_1)^{-1} + (\xi_2^2 \cdot Q_2)^{-1}))^{-1} \\
6: & \quad Q_O = 4 \cdot (2 \cdot ((\zeta_1^2 \cdot Q_1)^{-1} + (\zeta_2^2 \cdot Q_2)^{-1}))^{-1}
\end{aligned}$$

### 5.3 C++ implementation

In this section, we are going to make a comparison between the original Matlab code implementation and the new C++ version. The table 5.1 shows the different run times comparison between the Matlab references and the C++ implementation :

As we can see, the C++ implementation is faster than the Matlab implementation, for the exact same

Execution time (magnitude) in s	Prediction	Correction	Prediction + Correction
Matlab	$1e^{-1}$	$1e^{-3}$	$1e^{-1}$
C++	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$

Table 5.1: Execution time comparison for the algorithms

inputs. For the MagLev system, that runs on a  $1e^{-3}s$  frequency, the Matlab implementation is not sufficient, however, the C++ implementation will be fast enough to perform all calculations without issues.

During the testing phase of the C++ implementation of the prediction algorithm, some minor differences appeared in rare cases, but as shown in Figure 5.3, those differences does not seem to affect the overall result, and it could be seen as a difference in the way both C++ and Matlab represent Ellipsoids matrices.

For the Ellipsoids intersection, as represented in Figure 5.4, we can obtain two bounds. In our case we will only keep the outer bound of the intersection as it is the more important one for asserting stability.

Depending on the predicted and measured state, and the different uncertainties values that we chose, the intersection of our Ellipsoids may vary. As shown in Figures 5.5 and 5.6, from the same state and measurements, the intersection could be nonexistent. This can lead to abnormal prediction and results, if the uncertainties are not selected properly.

Figure 5.7 is the representation of the overall estimation of the system for a given data set, using our C++ implementation of the ellipsoid state estimator. The right graph represent the position in respect to time, whereas the left graph represent the projected ellipsoids of the position in respect to speed. This graph is showing us that our algorithms are working, but that the selected uncertainties needs to be improved, as this estimator does not improve the position estimation enough from the measurements to be useful.

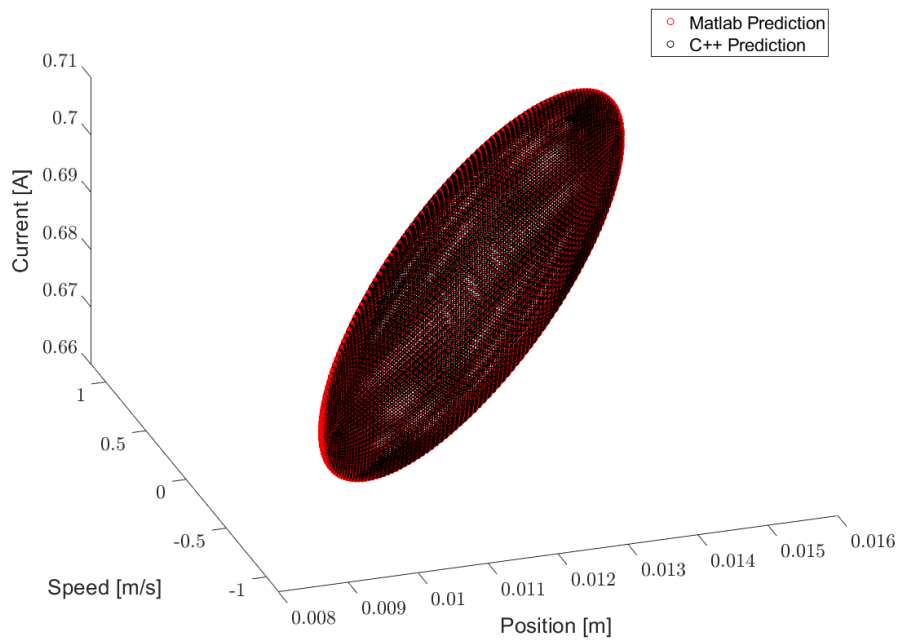


Figure 5.3: Ellipsoids comparison between Matlab and C++ implementations

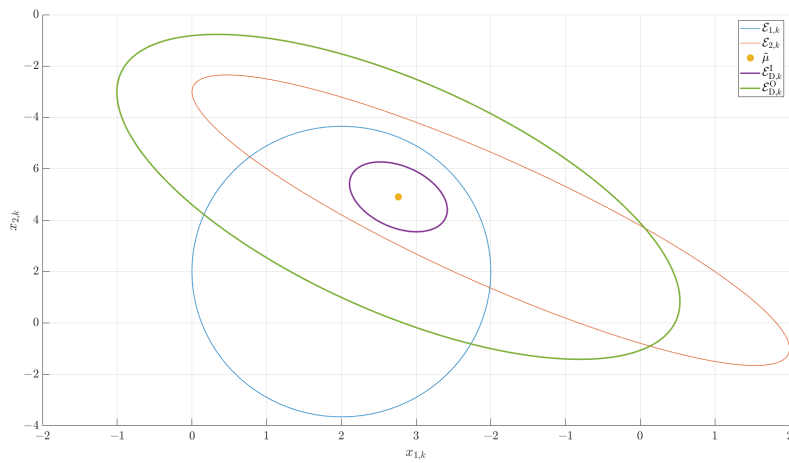


Figure 5.4: Ellipsoid Intersection Example

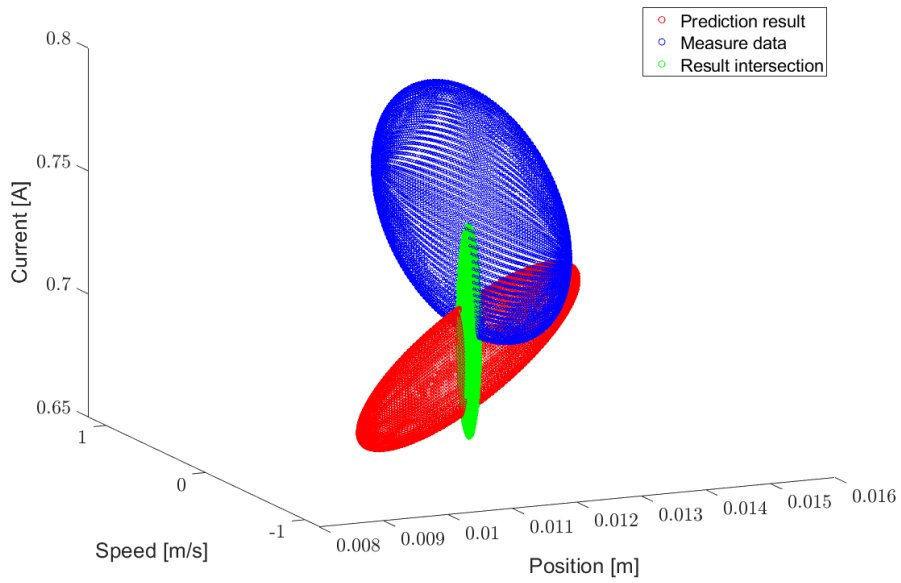


Figure 5.5: MagLev Estimation with low uncertainties

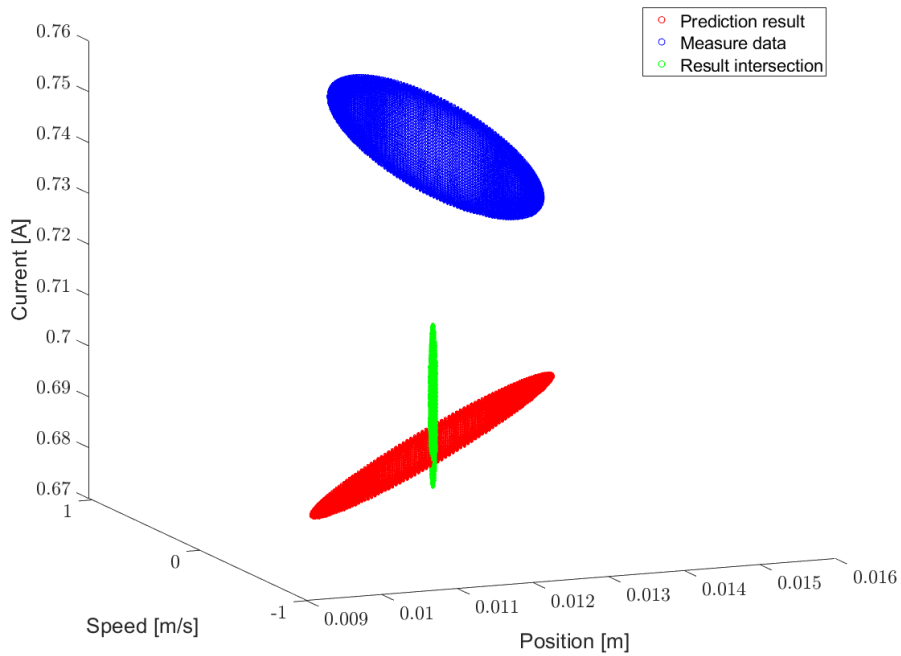


Figure 5.6: MagLev Estimation with high uncertainties

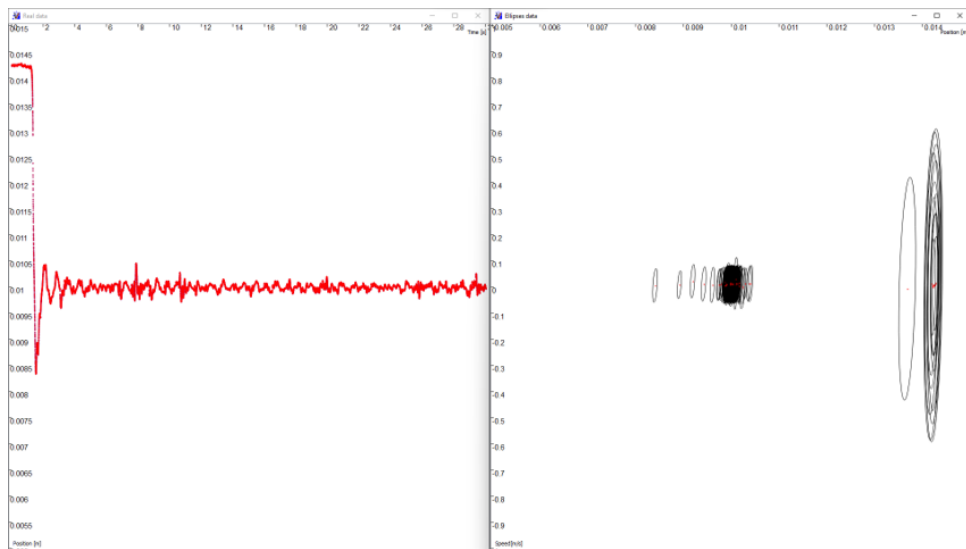


Figure 5.7: MagLev State Ellipsoid State Estimation

## Chapter 6

# Conclusion

Magnetic Levitation using a single electromagnet is a very unstable system and require a very robust control law for satisfying performances. As seen in the first parts of this report, most of the different strategies used did not bring useful results. The simulation used as a base for optimisation is working and behaving in accordance to what is observed, however the simulation and the test bench results do not match for the same parameters. Therefore, we can not yet rely on the simulation to improve the control laws. However, we did nonetheless manage to levitate a ball using a manually setup control law using a PID and some filters. For the ellipsoid state estimator and algorithms, their implementation in C++ did improve the execution time by a good enough margin for it to bit used for real-time applications. This work was merely a quick overview of the system and the few results can be used for further improvement. The future of this work would be to improve the simulation so that it could be used to find the best possible parameters for the controller model, improve the ellipsoid state estimator to gain in accuracy for the state estimation, as described in the C++ source file. The next step would then be to use the ellipsoid state estimator algorithm on a real experiment with the test bench, to test it's performance against a standard method.



# Bibliography

- [1] Peter Balko and Danica Rosinova. “Modeling of magnetic levitation system”. In: June 2017, pp. 252–257. DOI: [10.1109/PC.2017.7976222](https://doi.org/10.1109/PC.2017.7976222).
- [2] Andreas Rauh and Luc Jaulin. “A COMPUTATIONALLY INEXPENSIVE ALGORITHM FOR DETERMINING OUTER AND INNER ENCLOSURES OF NONLINEAR MAPPINGS OF ELLIPSOIDAL DOMAINS”. In: *International Journal of Applied Mathematics and Computer Science* 31.3 (Sept. 2021), pp. 399–415. DOI: [10.34768/amcs-2021-0027](https://doi.org/10.34768/amcs-2021-0027). URL: <https://hal-ensta-bretagne.archives-ouvertes.fr/hal-03344078>.
- [3] Andreas Rauh, Auguste Bourgois, and Luc Jaulin. “Union and Intersection Operators for Thick Ellipsoid State Enclosures: Application to Bounded-Error Discrete-Time State Observer Design”. In: *Algorithms* 14 (Mar. 2021), p. 88. DOI: [10.3390/a14030088](https://doi.org/10.3390/a14030088).

# List of Figures

2.1	Inteco MagLev bench . . . . .	3
2.2	Extensive position sensor test . . . . .	4
2.3	Extensive current reaction test (1) . . . . .	4
2.4	Extensive current reaction test (2) . . . . .	5
3.1	Maglev Simulation model from documentation . . . . .	6
3.2	Internal parameters optimization result . . . . .	7
3.3	PID optimization result . . . . .	7
3.4	Run with the optimized PID coefficients . . . . .	8
3.5	Optimization of the last parameters . . . . .	8
3.6	Optimization of the current state equation . . . . .	9
3.7	Optimization of the electromagnetic force parameters . . . . .	10
4.1	Standard Integrators Observer Model with Simulink . . . . .	11
4.2	Current from the Kalman Filter . . . . .	12
4.3	Position from the Kalman Filter . . . . .	12
4.4	Mean Filter representation in Matlab Simulink . . . . .	13
4.5	Standard PID . . . . .	13
4.6	Standard PID and Kalman Filter . . . . .	14
4.7	Standard PID and Mean Filter . . . . .	14
4.8	Standard PID, Kalman and Mean Filter . . . . .	15
4.9	PID and Feedback linearization control laws . . . . .	15
5.1	Thick Ellipsoid representation . . . . .	16
5.2	Ellipsoid state estimator . . . . .	17
5.3	Ellipsoids comparison between Matlab and C++ implementations . . . . .	19
5.4	Ellipsoid Intersection Example . . . . .	19
5.5	MagLev Estimation with low uncertainties . . . . .	20
5.6	MagLev Estimation with high uncertainties . . . . .	20
5.7	MagLev State Ellipsoid State Estimation . . . . .	21

# Appendices



## RAPPORT D'EVALUATION ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :  
At the end of the internship, please return this report via mail or email to:

ENSTA Bretagne – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE  
☎ 00.33 (0) 2.98.34.87.70 / [stages@ensta-bretagne.fr](mailto:stages@ensta-bretagne.fr)

### I - ORGANISME / HOST ORGANISATION

NOM / Name Carl von Ossietzky Universität Oldenburg

Adresse / Address Ammerländer Heerstraße 114-118, D-26111 Oldenburg, Allemagne

Tél / Phone (including country and area code) +49 441 798-4195

Nom du superviseur / Name of internship supervisor  
Prof. Dr. Andreas Rauh (professeur des universités)

Fonction / Function \_\_\_\_\_

Adresse e-mail / E-mail address andreas.rauh@uni-oldenburg.de

Nom du stagiaire accueilli / Name of intern

Jonas Soueidan

### II - EVALUATION / ASSESSMENT

Veillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**  
Please attribute a mark from **A (excellent)** to **F (very weak)**.

#### MISSION / TASK

❖ La mission de départ a-t-elle été remplie ? Ⓐ B C D E F  
Was the initial contract carried out to your satisfaction?

❖ Manquait-il au stagiaire des connaissances ?  oui/yes  non/no  
Was the intern lacking skills?

Si oui, lesquelles ? / If so, which skills? \_\_\_\_\_

#### ESPRIT D'EQUIPE / TEAM SPIRIT

❖ Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)

Ⓐ B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / If you wish to comment or make a suggestion, please do so here M. Soueidan a effectué ses travaux d'une manière extrêmement sérieuse et il était toujours disponible pour des discussions scientifiques avec tous les membres de l'équipe universitaire

## COMPORTEMENT AU TRAVAIL / BEHAVIOUR TOWARDS WORK

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?

*Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?*

(A) B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* Très ponctuel et engagé; Il a acquis des nouvelles connaissances

avec une manière indépendante, soit d'un point de vue méthodique, soit pratique

## INITIATIVE – AUTONOMIE / INITIATIVE – AUTONOMY

Le stagiaire s'est-il rapidement adapté à de nouvelles situations ?

(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

(A) B C D E F

*Did the intern adapt well to new situations?*

*(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)*

(A) B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* M. Soueidan a présenté les résultats de son stage dans une façon pédagogique

dans le cadre du séminaire de l'équipe de recherche universitaire. Il a en plus répondu clairement à toutes les questions posées et il a présenté des pistes claires à suivre dont les résultats de son stage peuvent être utilisés.

## CULTUREL – COMMUNICATION / CULTURAL – COMMUNICATION

Le stagiaire était-il ouvert, d'une manière générale, à la communication ?

*Was the intern open to listening and expressing himself/herself?*

(A) B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* C'était toujours un très grand plaisir de travailler avec M. Soueidan

## OPINION GLOBALE / OVERALL ASSESSMENT

❖ La valeur technique du stagiaire était :

*Please evaluate the technical skills of the intern:*

(A) B C D E F

## III - PARTENARIAT FUTUR / FUTURE PARTNERSHIP

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?

*Would you be willing to host another intern next year?*  oui/yes

non/no

Fait à Oldenburg, le 18/08/2022

In \_\_\_\_\_, on \_\_\_\_\_

Signature Entreprise \_\_\_\_\_

Company stamp \_\_\_\_\_

Signature stagiaire \_\_\_\_\_

Intern's signature \_\_\_\_\_



Fakultät II  
Informatik, Wirtschafts- und Rechtswissenschaften  
Department für Informatik  
Abt. Verteilte Regelung in vernetzten Systemen  
Prof. Dr. Andreas Rauh  
D-26111 Oldenburg

*Merci pour votre coopération*  
*We thank you very much for your cooperation*