LINMA2671 - Advanced Control and Applications

Biomedical/Applied Mathematics Engineering

Lab#1 : Control of a ball in magnetic levitation - UCLouvain [EN]

Authors:

Afonso Araújo afonso.soares@student.uclouvain.be
Colin Cloos colin.cloos@student.uclouvain.be

2023/2024 – 1$^{\text{st}}$ Semester

Contents

1 Introduction 2

2 Part I : PID control 2

2.1 PD Control . . . 2

2.2 PID Control with Switched Integral Action . . . 3

2.2.1 High Frequency Sine . . . 3

2.2.2 Limits of the Sensor . . . 4

2.2.3 Manual Tuning . . . 5

3 Part II : Advanced controller 7

3.1 Iterative Feedback Tuning (IFT) . . . 7

3.2 Other Approaches . . . 10

3.2.1 Model Predictive Control (MPC) . . . 10

3.2.2 Robust Control . . . 10

4 Conclusion 11

A Model Analysis 12

A.1 Results . . . 13

1 Introduction

The goal of the laboratory sessions was to give the students a better understanding of the intrincacies of PD and PID controllers.

In the first half of the project, the students were tasked with manual tuning of PD and PID controllers alike, in order to achieve the best response for several types of setpoint functions.

In the second half of the project, the students were tasked with implementing any advanced controller they envisioned, as long as properly justified. In the end the group wound up implementing an IFT PID controller.

2 Part I : PID control

Firstly, the group analysed the given benchmark files to assess them and understand the best way to improve each controller.

For both controllers the group proceeded with manual tuning, since the Ziegler-Nichols method proved inneffective for an unstable open loop system.

Moreover, when testing with the PID controller, the group also tested the limits of the hardware, identifying the limits of the sensor and the working highest frequency.

2.1 PD Control

The gain values as present in the *Maglev_Control_Basic.slx* were $K_p = 2.5$ and $K_d = 0.05$.

Through multiple iterations of trial and error, the group wound up with gains $K_p = 3.5$, $K_d = 0.05$ for a frequency of 1 Hz.

The side by side comparison of the benchmark PD controller vs the group's PD controller can be checked below, where the ball falls at 15 seconds on the left and at 20 seconds on the right,

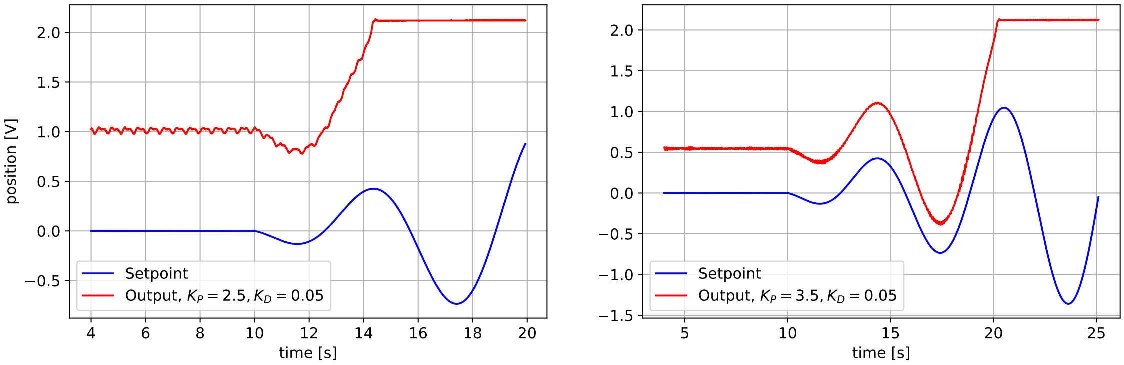

Figure 1: Output Response for - Benchmark PD (**left**); Group's PD (**right**)

The $r(t)$ function setpoint utilized for the above plots was $0.1t\sin(2\pi t)[V]$, for $t > 10[s]$. Hence the setpoint amplitude exceeds $1[V]$, as proposed in the laboratory notice.

While the static error can already be gauged from the above plots, the below plots show that indeed, the group's PD controller takes the error closer to 0 than the benchmark controller,

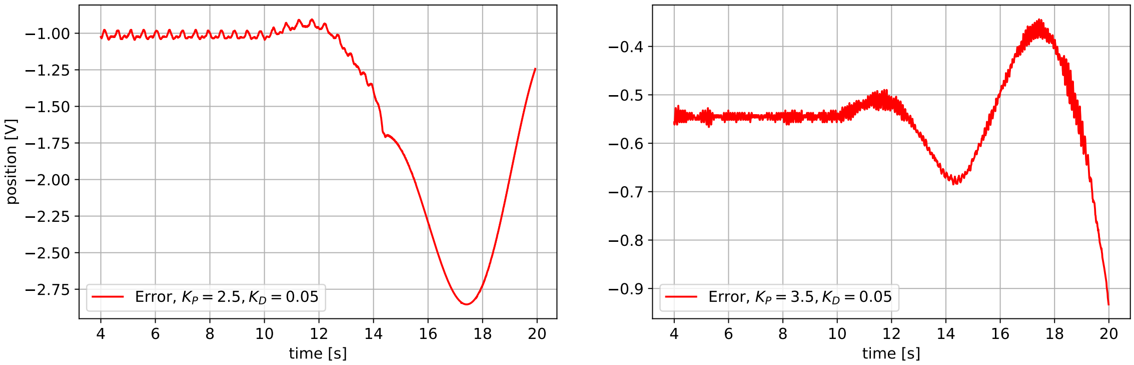

Figure 2: Error Evolution for - Benchmark PD (**left**); Group's PD (**right**)

However, this controller is still not perfect. Hence the need for an integrating section, with a **PID controller**.

2.2 PID Control with Switched Integral Action

2.2.1 High Frequency Sine

The sine setpoint with frequency $f = 23\,\text{Hz}$ corresponds to the limit frequency at which the closed-loop system becomes unstable. In fact, Figure [3] shows that at this critical frequency, the system is marginally stable, since the "Maglev Output" is very volatile for small changes in the error.

Due to time constraints, the group was not able to collect the data points and plot them through a Python script, hence the lesser quality of the *MATLAB* plots.

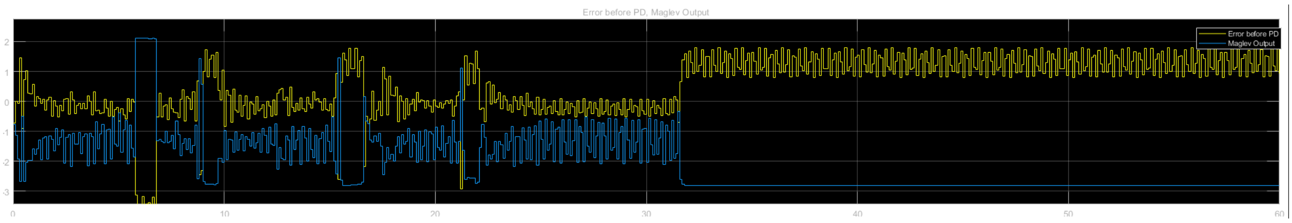

Figure 3: Marginally Stable Response for the Maglev

When testing the high frequency limit of the Maglev the setpoint function used was $r(t) = -1.5 + 0.5sin(2\pi 23t)$ with a sampling time of $0.01[s]$.

2.2.2 Limits of the Sensor

In order to determine the scope of the sensor that measures the position of the metallic ball, the group used two ramp signals: one with a negative slope of -0.1 and the other with a positive slope of 0.1.

Figure [4] shows that at times $t = 24\,\mathrm{s}$ and $t = 5\,\mathrm{s}$, the balls stopped floating for the negative slope ramp, and the positive slope ramp, respectively.

By reading the corresponding positions of the metallic balls, we identify the scope of the sensor to be: $[-2.5, 1.8][V]$.

Due to the small range of the scope, the integral action is required to cancel the static error and prevent the controller from becoming unstable.

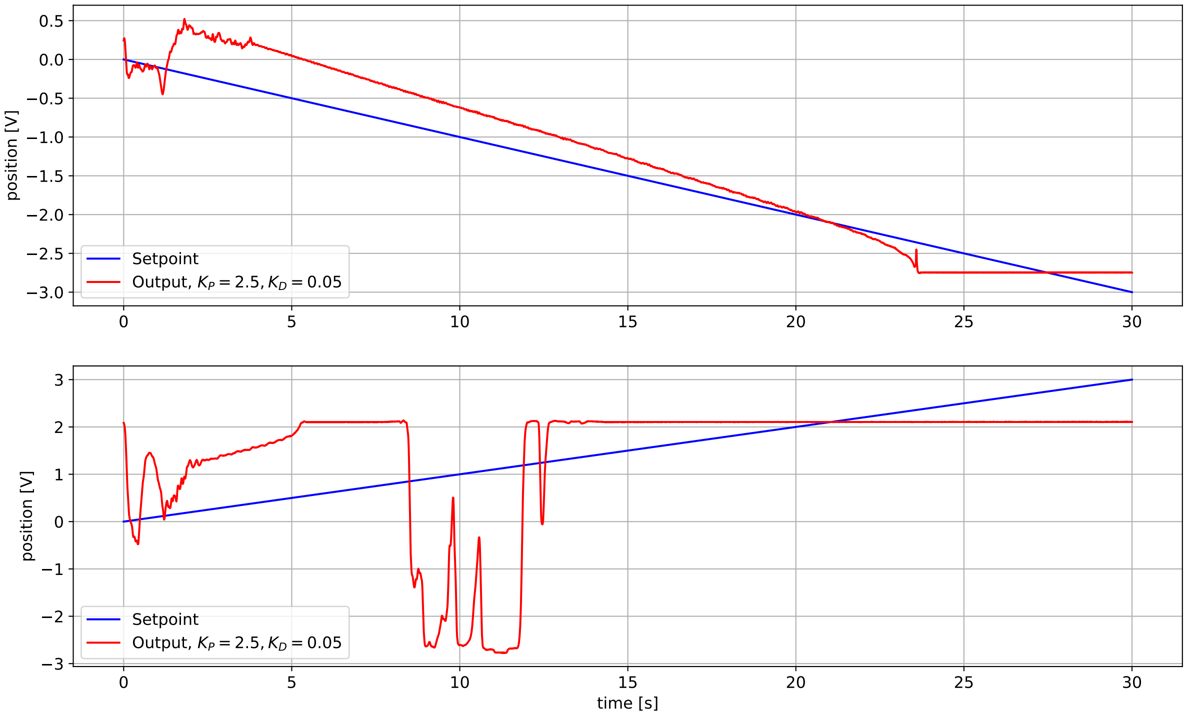

Figure 4: Setpoint $r(t) = \pm 0.1t$. Sampling time : $0.01s$.

2.2.3 Manual Tuning

After defining the hardware limits, the group proceeded with the trial-and-error method to obtain both controller gains.

The presented Figure [5] shows the comparison between the benchmarked PID and the group's PID for a setpoint of $r(t) = 0.1t\sin(2\pi t)$. Indeed the group's PID performs better with gains $K_p = 3.5$, $K_D = 0.05$ and $K_I = 10$.

Moreover, this is proved with Figure [6] that shows that the group's PID reaches a null error faster that the benchmarked PID, and the group's PD, as showcased before (Figure [2]).

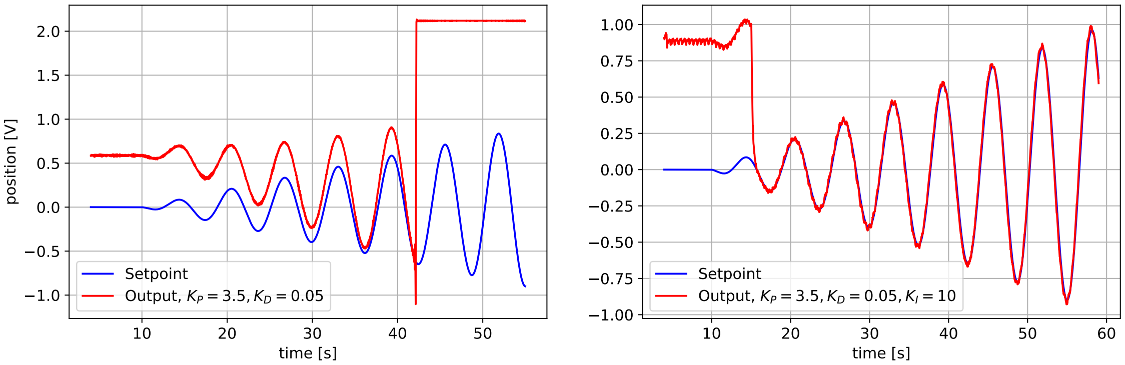

Figure 5: Output Response for - Benchmark PID (**left**); Group's PID (**right**)

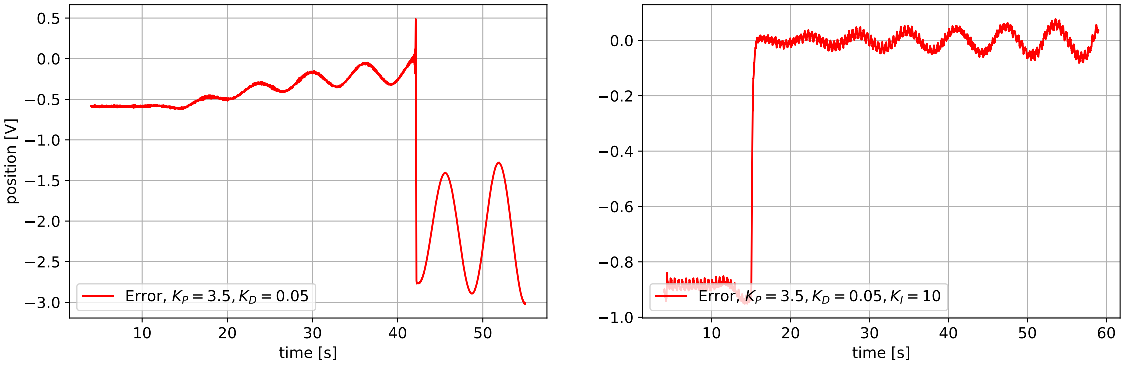

Figure 6: Error Evolution for - Benchmark PID (**left**); Group's PID (**right**)

Delving deeper into the robustness of the PID controller, it behaves well for an input triangle function of amplitude $[-2.1, 1.4][V]$,

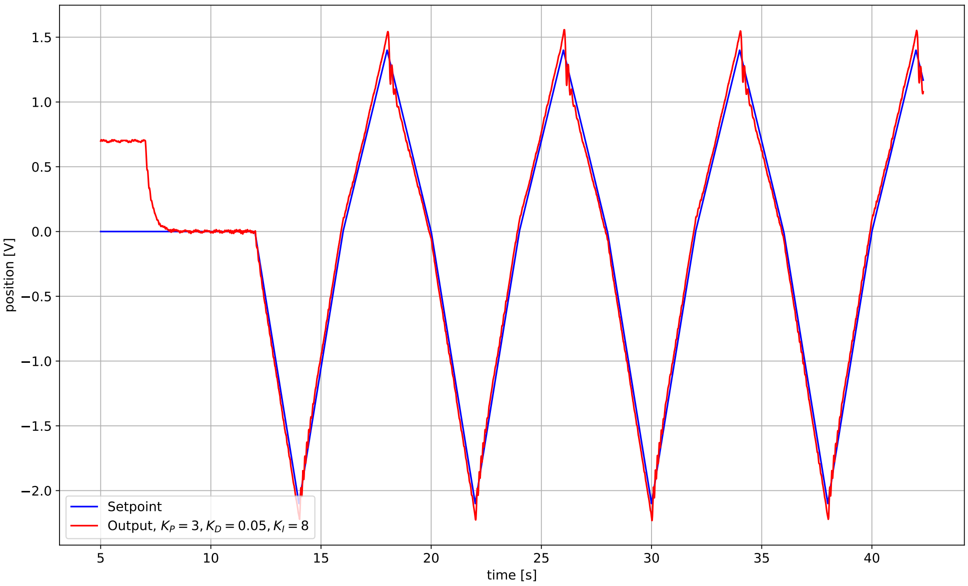

Figure 7: Setpoint triangle with ranging from $-2.1[V]$ to $1.4[V]$. Sampling time : $0.01s$.

and it behaves very poorly beyond that range.

That is due to the fact that the sensor range is $[-2.5, 1.8][V]$ which means that the closer the ball is to that range, the greater the impact small errors created by the physical up and down bouncing motion of the ball have. When the ball comes down it comes with gravity, so it comes with greater force. This, in addition to the inertia of the ball, make it so that if the triangle amplitude was around [-2.4, 1.7] [V], which is very close to the sensor limit, the PID would struggle very hard to control the ball and pull it back up.

Hence our PID controller is robust when constrained to setpoint amplitudes of $[-2.1, 1.4][V]$ and it lacks performance when the amplitude goes beyond those values.

3 Part II : Advanced controller

3.1 Iterative Feedback Tuning (IFT)

In order to attempt to improve the PID controller given by the initial set of parameters:

$$\rho_0 := \{K_P = 2.5, K_D = 0.3, K_I = 1.5\}$$

the IFT method has been used. In this section, the method is presented based on the scientific paper [1]. Then, the obtained results for the *Maglev System* are described.

The goal of the method is to obtain an efficient PID controller that stabilizes the system rapidly, and with the smallest steady-state error for a certain setpoint signal $r(t)$.

The structure of the PID controller is the one described in the Figure 1 of [1], with :

$$\begin{aligned} C_r(\rho) &= -K_P - K_I T_s \frac{z}{z-1} \\ C_y(\rho) &= -K_p - K_I T_s \frac{z}{z-1} + \frac{K_D}{T_s}\frac{z-1}{z} \end{aligned}$$

T_s represents the sampling time and "The derivative action is calculated on y and not on the control error" as described in [1].

Description of the method

In the following description, for simplicity, the temporal signals are written without the explicit time dependency. For example, for a signal $f(t)$, it is represented by f or f_t. Additionally, the notation est$[f]$ is used for the estimator of f.

Below is the procedure followed to obtain the results:

1. For $i = 0, 1, 2, 3$, generate

$$\begin{cases} y^{(1)}(\rho_i) : r_i^{(1)} = r \\ y^{(2)}(\rho_i) : r_i^{(2)} = r_i^{(1)} - y^{(1)}(\rho_i) \\ y^{(3)}(\rho_i) : r_i^{(3)} = r \end{cases}$$

2. Compute

$$\begin{cases} \tilde{y}(\rho_i) := y^{(1)}(\rho_i) - r \\ \text{est}[\frac{\partial y}{\partial \rho}(\rho_i)] = \frac{1}{C_r(\rho(i)}[(\frac{\partial C_r}{\partial \rho}(\rho_i) - \frac{\partial C_y}{\partial \rho}(\rho_i))y^{(3)}(\rho_i) + \frac{\partial C_y}{\partial \rho}(\rho_i)y^{(2)}(\rho_i)] \end{cases}$$

3. Compute

$$\text{est}[\frac{\partial J}{\partial \rho}(\rho_i)] = \frac{1}{N}\sum_{t=1}^{N}(\tilde{y}_t(\rho_i)est[\frac{\partial y_t}{\partial \rho}(\rho_i)])$$

4. Finally apply one step of the estimated gradient method $\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \text{est}[\frac{\partial J}{\partial \rho}(\rho_i)]$

At step 1, the signal $y^{(3)}(\rho_i)$ is required to obtain an unbiased estimator of the gradient of the cost function J, as described in [1].

At step 4, the following is used:

$$R_i = \frac{1}{N}\sum_{t=1}^{N}\left(\text{est}\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]\text{est}\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]^T\right)$$

This provides a biased approximation of the Gauss-Newton descent direction.

Obtained results

The results of four iterations of the IFT method are presented in Figure [8]. y_3 and r_3 are not displayed since they are similar to y_1 and r_1.

The output signal y_1 for the first iteration exhibits small overshoots for the setpoint signal from the *Benchmark* Simulink file. This overshoot is amplified through iterations 2 and 3, but the results are still acceptable, as the ball was still hovering at the 3rd iteration. However, in iteration 4, the closed-loop system becomes unstable for r_2 setpoint, and the controlled output y_1 is significantly degraded for the square and sine signals.

Table 1 and Figure [8] illustrates that, contrary to expectations, the cost function increases and the controller worsens throughout the iterations. This unexpected outcome, suggests that the IFT method applied to the *Maglev System* is not efficient. One potential explanation is that the IFT method is inherently stochastic, while the controlled process exhibits minimal noise. Moreover, as seen in Figure [8], the provided PID gains ρ_0 are already reasonable, and the first iteration may not be problematic enough for the IFT method to work efficiently.

The step γ could have been chosen differently as well, as it could have been shorter/bigger than necessary. This means that if it was shorter than necessary, then the group needed more iterations to find a decrease in J. If it was bigger than necessary, then the algorithm skipped the decrease in J and was redirected to an increase in the cost function. The goal, after all, was to get closer to a global minimum of the convex cost function.

Iteration	Cost $J(\rho_i)$	ρ_i	γ_i
0	0.128	$\{2.5, 0.3, 1.5\}$	–
1	0.3195	$\{1.5935, 0.2824, 0.7847\}$	1
2	0.3720	$\{1.550, 0.2796, 0.7246\}$	0.1
3	0.3719	$\{1.4802, 0.722, 0.2493\}$	0.1
4	1.0498	$\{1.4802, 0.722, 0.2493\}$	–

Table 1: Obtained results for 4 iterations of the IFT method.

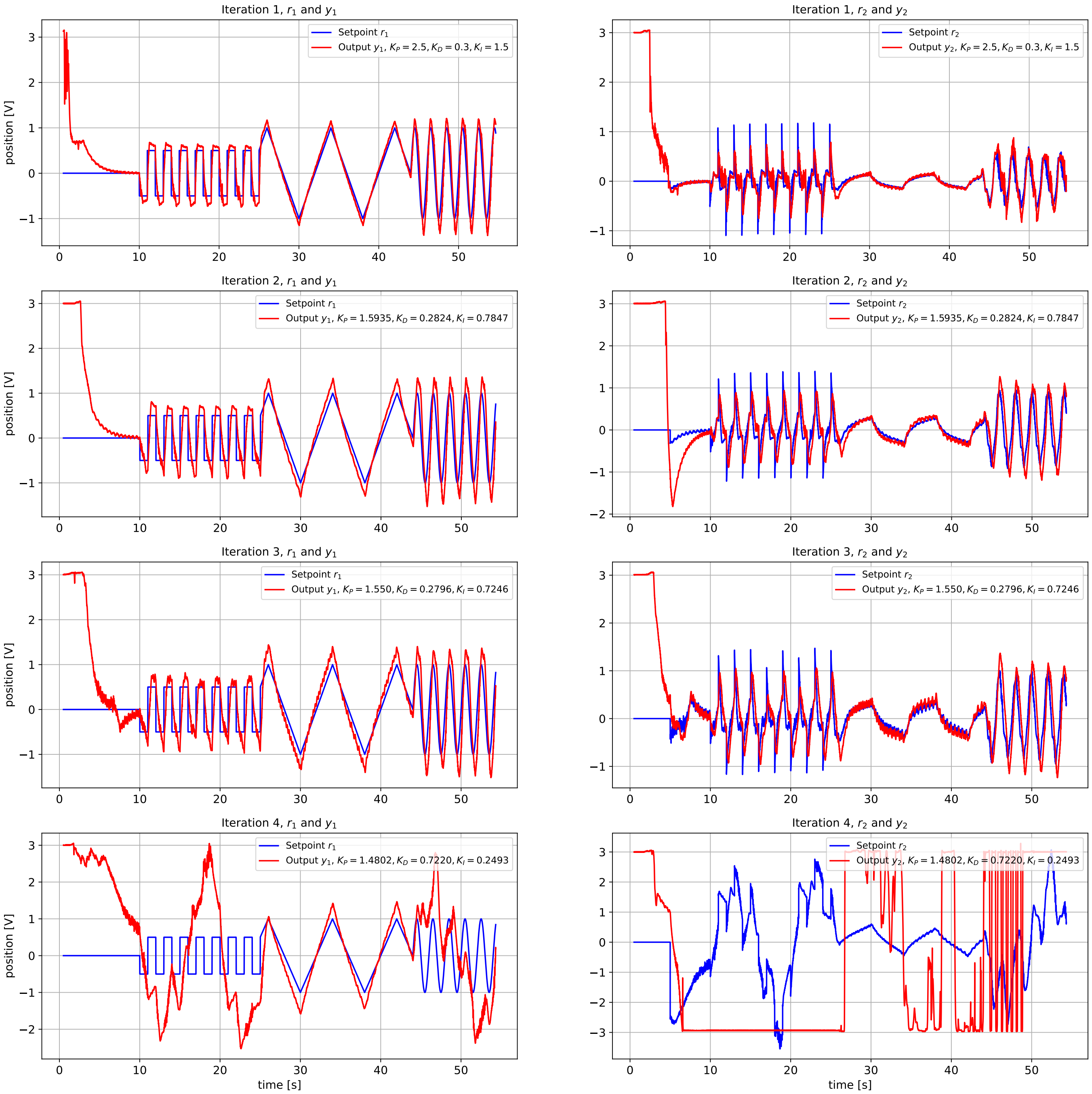

Figure 8: Obtained results for 4 iterations of the IFT method.

3.2 Other Approaches

3.2.1 Model Predictive Control (MPC)

MPC proves particularly advantageous for controlling the "Maglev System" using the model detailed in Appendix A. Given the inherent complexity and dynamism of the "Maglev System", it encounters uncertainties from various sources, including external disturbances, parameter variations, environmental changes, and the inherent simplifications present in the model. MPC excels in managing such uncertainties by systematically integrating a dynamic model with real-time measurements and optimizing control actions within a finite prediction horizon.

Due to the complexity of implementing such a controller, the group does not provide the implementation nor any results for it. However, an attempt has been made by using the guide provided in [2].

3.2.2 Robust Control

The group briefly considered utilizing Robust Control techniques as part of the "Advanced Controller" task.

Due to the nature of the system, the group did not have access to the true and real plant model. The group instead opted to use the linearized version of the model, A, where,

$$G(s) = \frac{-6.59}{s^2 - 35.34} \tag{1}$$

is the open loop linearized model of the plant.

The idea behind the Robust Control approach was to check the current gain and phase margins of the model and then discover which variation factor, i.e $1 + 1i$ or $3 + 4i$, could be added such that the model still worked for it. This variation factor is defined as,

$$\Delta_{\text{Factor}} = a + bi \tag{2}$$

where a affects the gain margin and b affects the phase margin.

The setup was then to apply the *margin* function from *MATLAB* to check for the current G_M and P_M and then apply the *diskmargin* to check whats the smallest disk D that exists in the complex plane, such that its radius gives us the upper and lower gain and phase margins for the system.

However since the linearized open-loop system has infinite gain and phase margins, this approach was not taken. The idea for this approach comes from [3].

4 Conclusion

With the first part of the report, the group developed a better intuition as to how the controller gains affect the response to an input sine, square and triangle functions. As expected the PID controller perfomed better than the PD controller. Moreover, the group also understood that every piece of hardware has its limitations, and it is important to know them before starting doing experiments with it.

With the second part of the report, the group tried 3 different approaches to the advanced controller, eventually progressing more with the IFT for a switched PID. Ideally, the IFT would allow the cost function to decrease every step, and for the controller gains to become better every iteration. However, that was not verified.

The group believes the experimental missfire could have come from different sources, either acting independently or not:

- The fact that the first batch of PID gains were already associated with a reasonably good controller, which meant the IFT had trouble improving it;
- The γ step size not being chosen correctly which meant every step that was being taken was too small/big to create any meaningful improvements;
- The fact that the IFT method is stochastic, which means that for little noise this process has a hard time performing well.

Most importantly, the group understood that even if there is no way to access the model of a plant it is always possible to control the closed loop system through manual tuning and model approximations.

A Model Analysis

The Maglev model given in the assignment statement is :

$$m\ddot{x} = mg - kk_1^2 \frac{u}{(x+a)^2}$$

where x is the ball position, u is the control voltage, m is the mass of the metalic ball ($m = 20.7[g]$) and g is the gravitational acceleration ($g = 9.81 m/s^2$). k, k_1 and a are parameters of the system that need to be identify.

The system is a SISO plant with u the input and x the output.

In order to write a state reepresentation of the system and then obtain it's linearised around an equilibrum open loop and closed loop transfer function, let's define the state vector $\mathbf{x} := \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$. The chosen input is the control voltage u, the model can then be rewriten as :

$$\dot{x} = \begin{bmatrix} x_2 \\ g - \frac{kk_1^2}{m} \frac{u}{(x_1+a)^2} \end{bmatrix}$$

The output $y = C\mathbf{x} = \begin{bmatrix} 1 & 0 \end{bmatrix} x_1$.
Linearizing around the equilibrum $\mathbf{x}^*$:

$$\dot{\tilde{\mathbf{x}}} = A\tilde{\mathbf{x}} + B\tilde{u} = \begin{bmatrix} 0 & 1 \\ 2\frac{kk_1^2}{m} \frac{u^*}{(x_1^*+a)^3} & 0 \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} 0 \\ -\frac{kk_1^2}{m} \frac{1}{(x_1^*+a)^2} \end{bmatrix} \tilde{u}$$

with $\tilde{\mathbf{x}} := \mathbf{x} - \mathbf{x}^*$ and $\tilde{u} := u - u^*$.

The open loop transfer function $G(s) = C(sI - A)^{-1}B = \frac{B_{21}}{S^2 - A_{21}}$. The linearized open loop system has two poles in $\pm A_{21}$ and thus is unstable.

In order to identify the parameters, the system is stabilized around a certain equilibrum by a PID controler with transfer function $PID(S) = K_p + K_d \frac{S}{1+\frac{K_p}{N}S} + \frac{K_i}{S}$ [4]. Note that in order to not have a static error, the integral action is required.

In order to identify the parameters kk_1^2 and a, the procedure is :

1. Choose two different setpoints r_1 and r_2 for $x_1 = x$.
2. Get numerical values for $\mathbf{x}_1^* = \begin{bmatrix} (x_1)_1^* \\ (x_2)_1^* \end{bmatrix}$, $\mathbf{x}_2^* = \begin{bmatrix} (x_1)_2^* \\ (x_2)_2^* \end{bmatrix}$ and u_1^*, u_2^* once the equilibrum is reached. Note that the closed system must be stable for those values.
3. Solve, by using *WolframAlpha* for example, the nonlinear system of 2 equations and 2 unknowns :

$$\begin{cases} g = \dfrac{kk_1^2}{m} \dfrac{u_1^*}{((x_1)_1^* + a)^2} \\ g = \dfrac{kk_1^2}{m} \dfrac{u_2^*}{((x_1)_2^* + a)^2} \end{cases}$$

.

A.1 Results

As seen in Figures [9] and [10], the chosen values for the setpoints are $r_1 = 0$ and $r_2 = -1$. The obtained states are $(x_1)_1^* = 0.0244, u_1^* = 1.49$ and $(x_1)_2^* = -0.97, u_2^* = 0.93$. Using the numerical values for m and g then solving the nonlinear system of equations by using *Wolfram Alpha*, the results are:

$$\begin{cases} kk_1^2 = 42.058 \\ a = 0.531 \end{cases}$$

or

$$\begin{cases} kk_1^2 = 3\ 057.01 \\ a = 4.712 \end{cases}$$

Therefore, the system of equation has two solutions.

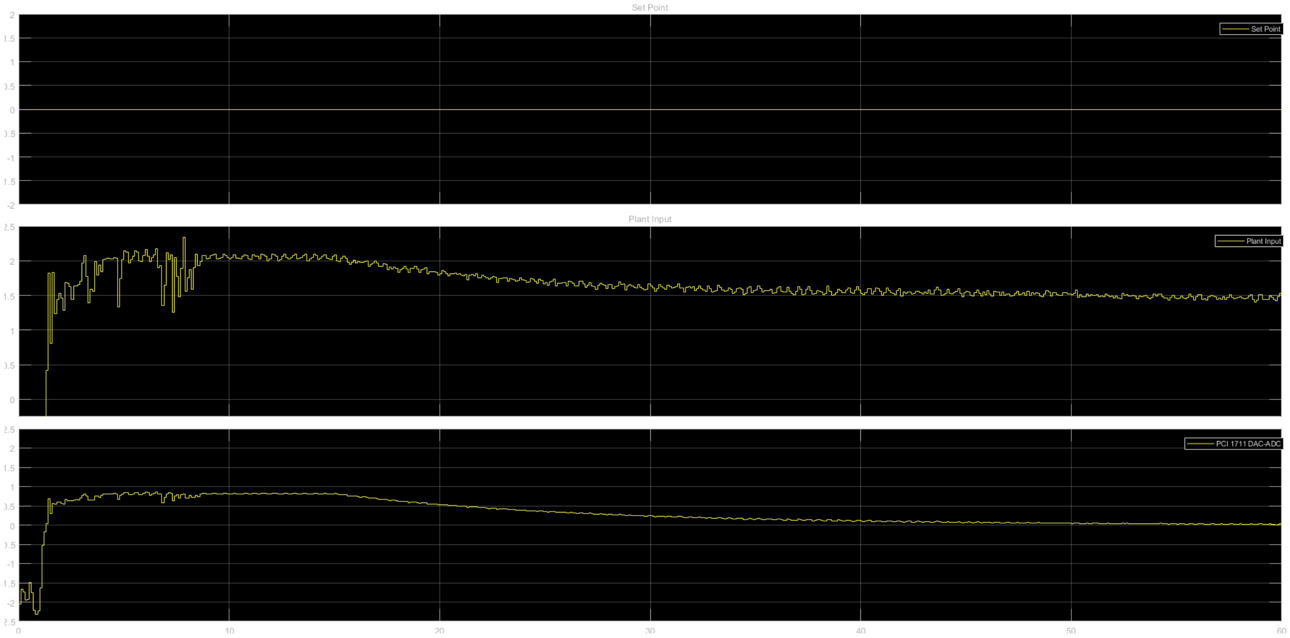

Figure 9: Setpoint : $r(t) = 0$. Sampling time : $0.01s$

Through these solutions, it is possible to obtain two different open-loop transfer functions.

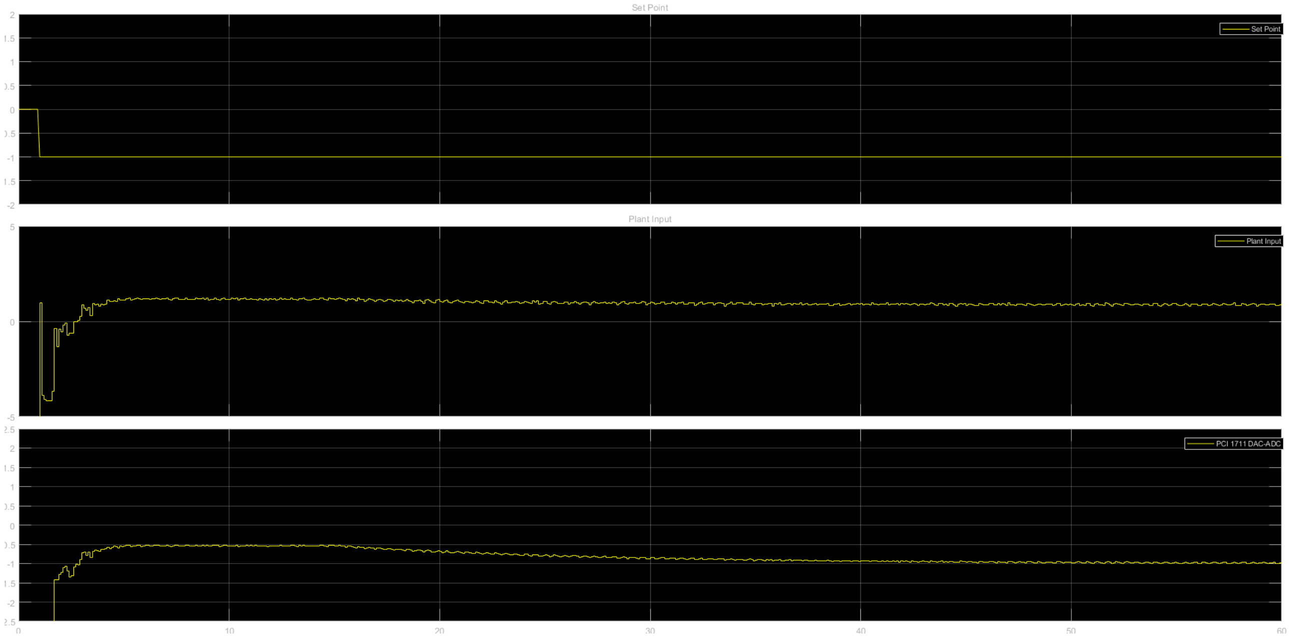

Figure 10: Setpoint : $r(t) = -1$. Sampling time : $0.01s$

References

[1] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: Theory and applications," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 26–41, 1998.

[2] "Designing a model predictive controller for a simulink plant." (), [Online]. Available: https://nl.mathworks.com/help/mpc/gs/designing-a-model-predictive-controller-for-a-simulink-plant.html.

[3] "Understanding disk margin — robust control, part 2." (), [Online]. Available: https://www.youtube.com/watch?v=XazdN6eZF80.

[4] "Proportional-integral-derivative controller." Accessed: October 2023. (), [Online]. Available: https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller.