Control System Design

LAB REPORT

Ankit Pandey, Tigran Pletser, Emmanouil Stivaktakis, Julien van Delft

2020-2021

Pr. Emanuele Garone

Engineering Sciences

Contents

1 Introduction

The control of complex system takes a huge place in the industrial world. It can be for small inoffensive systems like the one studied in this report or for big nuclear reactor that needs a special care and where a simple mistake can kill thousands of people. Most of the systems need control in order to make them reach what is needed for the users. The control system design course is then a key course in every engineer study. Indeed, whatever the specialisation chosen, we are going to control the system on which we will work.

The aim of this project is to design a control law for a studied system. This system is a ball in a tube for which the position of the ball needs to be controlled. As the system is not known at the beginning, theoretical equations will be derived in order to predict the form of the real plant system. Three different requirements on the ball position control has been stated in the project guide.

Due to the covid-19 crisis, the real plant cannot be studied directly on site. Fortunately, a matlab simulation of the model has been given to the students in order to let them make the experiments. Our tests will be made on this simulation model.

First, the real system will be broken down into two different systems, the first one is the motor system and the second is the transformation of the fan speed into the ball position in the tube. In this report, the same methodology will be used for both system. The real system will be identified, then the created model will be validated by comparing it to the real plant and finally the control law that fulfills the requirements will be computed. The first system is going to be studied and controlled and after that, the second will follow the same path.

Finally, some problems will be enlighten and a possible refinement of the project is proposed in order to allow to go further in the future.

2 Statement of the problem

2.1 Plant description

The plant studied in this project is called " The ball in the tube". It consist of a ball placed in a tube with a current driven DC motor used as a fan at the bottom. This fan creates an air flow that rises the ball up the tube. The aim of this project is to control the vertical position of the ball.

Figure 1: The ball in the tube

The sensors used are one tachometer to measure the velocity of the fan and one infrared sensor (SHARP GP2YOD02YKOF) to measure the position of the ball. Their characteristic are shown in the figures 2 and 3 below.

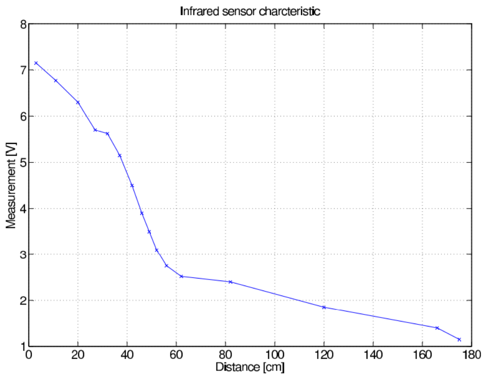

Figure 2: Infrared sensor characteristic

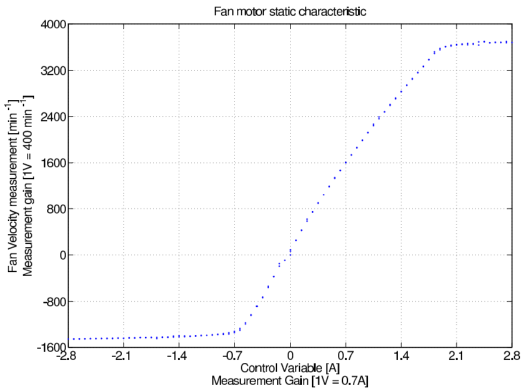

Figure 3: Tachometer characteristic

The figure 2 shows two zones of linearity that will need to be taken into account in order to have a good regulation.

2.2 Requirements description

In this project, there are one basic requirement that is necessary to fulfill and two more advanced requirements that would be good to fulfill if the first is achieved.

2.2.1 Basic requirements

Stabilization of the ball at a fixed position without any steady state error

2.2.2 More advanced requirements

1. Switch between two different points with a maximum of 1cm overshoot.
2. Switch between two different points with a maximum of 1cm overshoot in maximum 2s.

3 Theoretical considerations

3.1 Physical relationships

3.1.1 First system

First, the inner loop relationship is given by the relationship between the motor velocity and the current in the motor. We know from other courses that the motor speed is driven by this equation:

$$J_t\ddot{\theta}_m + F\dot{\theta}_m = K_i i_a \quad \Longrightarrow \quad J_t\dot{v} + Fv = K_i i_a$$

Now, using the Laplace transform on this equation, we can find the theoretical form of the first system.

$$J_t V(s)s + FV(s) = K_i I_a(s) \quad \Longrightarrow \quad V(s) = \frac{K_i}{J_t s + F} I_a(s)$$

As we know that $V(s) = G_1(s)I_a(s)$, we can conclude that $G_1(s)$ is equal to:

$$G_1(s) = \frac{K_i}{J_t s + F}$$

The first system has then the form of a simple first order system.

3.1.2 Second system

Now, we are interested in finding the form of the second system. We know that the ball position is driven by this equation:

$$m\ddot{x} = F(x) - c\dot{x} - mg$$

But, as we want to study the motion of the ball around its equilibrium point (its equilibrium point being the value at which the ball levitates), it is needed to define the relative position $\bar{x} = x - x_{eq}$ or $x = \bar{x} + x_{eq}$. Where x_{eq} is define as the point at which $\dot{x}$ is equal to 0.

$$m\cancelto{0}{\ddot{x}_{eq}} = F(x_{eq}) - c\cancelto{0}{\dot{x}_{eq}} - mg \quad \Longrightarrow \quad F(x_{eq}) = mg = F_{eq}$$

For simplicity, we are going to define the force $F(x)$ as the sum of the constant force at the equilibrium and a small variation of force around it.

$$m(\ddot{\bar{x}} + \cancelto{0}{\ddot{x}_{eq}}) = \cancel{F_{eq}} + \Delta F(\bar{x}) - c(\dot{\bar{x}} + \cancelto{0}{\dot{x}_{eq}}) - \cancel{mg}$$

The derivative of the point of equilibrium are zero by definition. The equilibrium force cancels out the gravity contribution. And finally, we get an equation in term of variation around the equilibrium point only.

$$m\ddot{\bar{x}} = \Delta F(\bar{x}) - c\dot{\bar{x}}$$

Once again, we need to take the Laplace transform of this equation.

$$ms^2\bar{X}(s) = \Delta F(s) - cs\bar{X}(s) \quad \Longrightarrow \quad \bar{X}(s) = \frac{1}{ms^2 + cs)}\Delta F(s)$$

Or transforming the coefficient into a classical form of transfer function:

$$\bar{X}(s) = \frac{\frac{1}{c}}{s(\frac{m}{c}s + 1)}\Delta F(s) \quad \Longrightarrow \quad G_2(s) = \frac{\frac{1}{c}}{s(\frac{m}{c}s + 1)}$$

The second system is formed of the combination of a simple integrator and a first order system.

3.2 Methodology

The system is going to be identified using a Gray box modelling. Indeed, in the previous section, the physical equations were derived, therefore the order of each system is known. For both system, we are going to identify them, then validate the model by comparing it with the real plant and finally design the needed control for the requirements on the model. After that, the control laws designed will be tested on the real plant to validate the control.

For the first system, the given Matlab identification function is used as it is a simple first order system.

But for the second system, the easiest way is to consider only the integrator with a gain in a first approach. Naturally, the integrator has its pole in zero then it will have a bigger impact in comparison with the pole probably very far from the origin (as the mass of the ball is small). To find the gain of the integrator, the slope of the response to a simple step will be studied.

4 Types of Control Systems

In the Control System Design course, two different types of control were studied. The *Feedforward control* and the *Cascade control* which are both industrial control used in lots of real-life processes. Those controls are very good at reference tracking and disturbance rejection.

4.1 Feedforward control

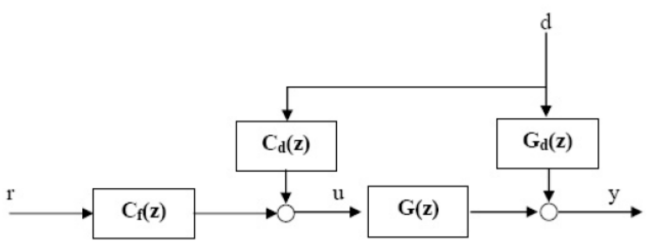

Figure 4: Feedforward control

Feedforward control can be used in very specific cases where the disturbance and the system are perfectly known. As we will probably need to use approximation on the system and we do not know the disturbances, it is not suited for the control in this project.

4.2 Cascade control

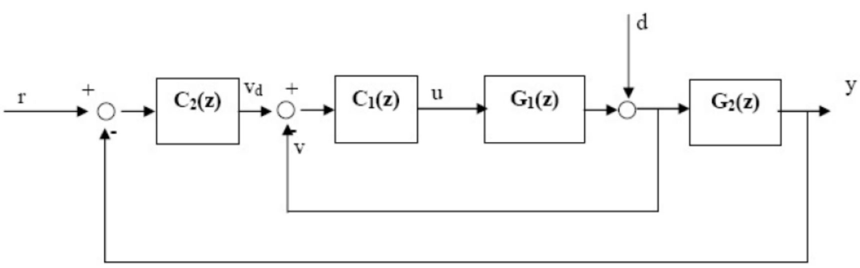

Figure 5: Cascade control

This control type is vary useful to reduce a multivariable problem (as the ball in the tube system) to n simple SISO systems. This simplifies also the design of the control law and it can be used for nonlinear control and both in discrete-time and continuous-time. We have an input (the current) and two outputs (the fan velocity and the ball position). Therefore, the cascade control seems more suited for project. On top of that, as the system current-fan velocity is faster than the global system current-position, the inner loop will have a small impact on the outer loop.

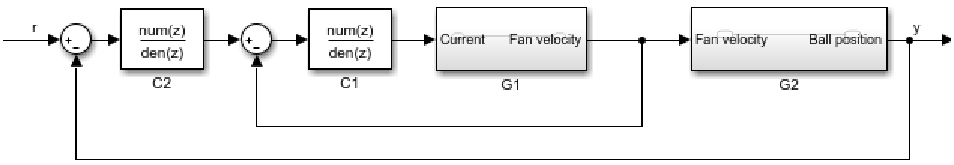

Figure 6: Cascade control of the project

The cascade control that will be designed for the control of the ball position in the tube will have this form. The first system (the inner loop) consists of a system transforming the input current into the fan velocity through the motor and the second system (the outer loop) is transforming the fan velocity into the ball position.

5 Inner loop

5.1 Identifying the first system

First, the inner system needs to be identified. To find an approximate transfer function of the system transforming the current into the velocity of the fan, a step is applied at the input of the system. The inputs and outputs are taken to the workspace and by using the Identification function, an approximate function can be found. Before finding the function, it is needed to determine the dead zone if it exists and the saturation zone.

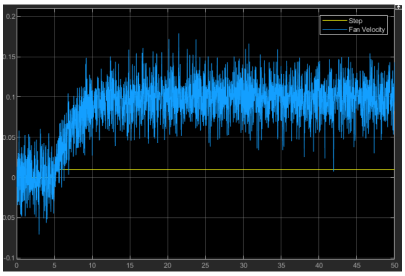

Figure 7: Response of the fan to a 0.01 amplitude step

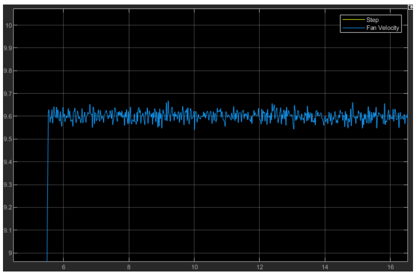

Figure 8: Upper saturation to a big positive step

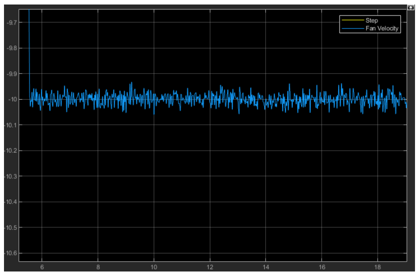

Figure 9: Lower saturation to a big negative step

From those tests, we can assure several things. There are no dead zone as the fan reacted to a small positive and negative steps with a non zero response. Consequently, there are no operating points for the inner loop. The saturation positive and negative values are respectively of 9.6V and -10V. But, by looking to the fan motor characteristic, the smallest saturation value is - 3.6V. Also, by looking to the response of the system to a step, it is clear that the system order is of 1 as we have no overshoot in the response to a step. The transfer function may be computed now by putting zero offset on both inputs and outputs and the system order to 1. The transfer function is given by the Identification function provided in the lab codes:

$$G_1(s) = \frac{4000}{2.497s+1}$$

The inner loop model gets constructed as followed, with the saturation zone on the output as mentioned before and for the inputs the voltage source can only provide between -10V and 10V. Finally, to transform the fan velocity into the tachometer voltage in order to proceed the voltage regulation, the velocity is divided by 400 as the tachometer has an output voltage of 2.5V for 1000rpm or 1V for 400rpm.

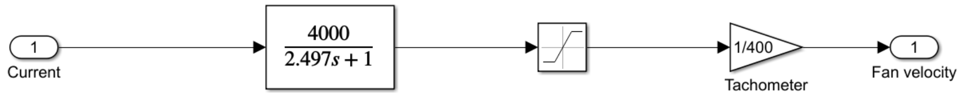

Figure 10: Inner system modelisation

The system needs to be validated. To show the correct following of the real system, the same step is applied to both inner system in open loop and the two responses are supposed to match at any time.

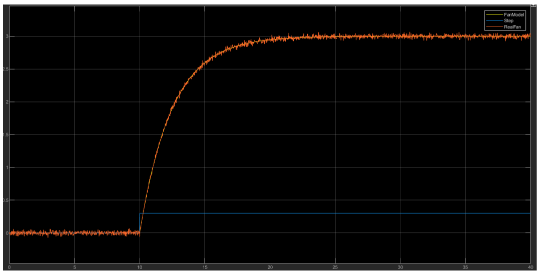

Figure 11: Comparison of the inner model with the real plant

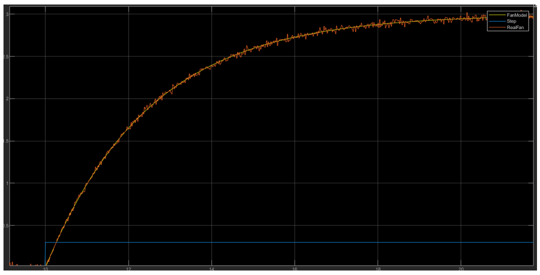

Figure 12: Zoom on the rising part showing good correlation

On those graphs, it is obvious that the yellow theoretical model follows accurately the real plant fan. The inner system is then sufficiently correct to base our calculations on it.

5.2 Regulation of the inner loop

We can now design the controller of the inner loop. The inner loop is closed and the only important regulation to be provided is that the inner loop reacts very fast to the input. We can deduce that a proportional action is enough to achieve our goal. Indeed, as the outer loop will cancel the static error, we only want a very fast inner loop allowing to consider the inner loop absent in the outer loop control. A simple MATLAB function is then used in which the proportionality factor is defined. A Zero-Order Holder has been added to the inner loop for a more realistic approach.

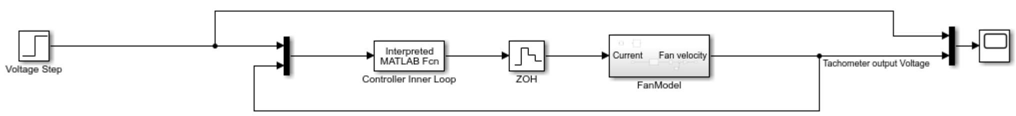

Figure 13: Closed inner loop

By taking into account the ZOH through the Pade' approximants:

$$ZOH(s) = \frac{1 - e^{sT_s}}{s} \approx \frac{1}{\frac{T_s}{2}s + 1}$$

Where T_s corresponds to the sampling time here chosen equal to 0.01.

To have a quick response, as only a proportional action is considered, the bigger the K_p, the smaller the settling time and the smaller the static error will be. But, if the K_p is to big, it can create oscillations in the response or an overshoot. To find the maximal value of K_p, the root locus is used.

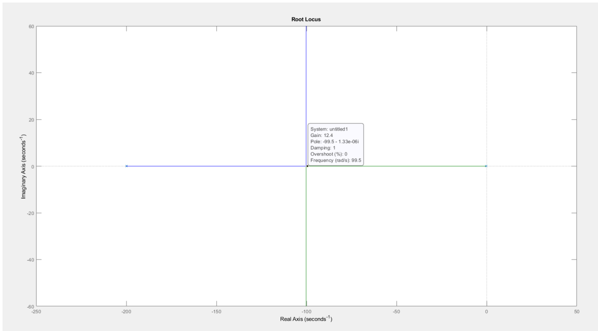

Figure 14: Root Locus of the inner loop

It is seen on the root locus that the maximum proportional gain that can multiply the error without causing overshoot is K_p = 12.4. As it is the inner loop, it is possible to have a small overshoot with a damping of minimum 0.7 which is a good compromise. Indeed, the smaller the damping ratio, the smaller the peak time is but the bigger the overshoot will be. Taking all of those into consideration, the maximum value of K_p is 25.5 which leads to a small overshoot of 4.5% and a damping ration of 0.7. The overshoot in itself is not a problem as it is going to be countered in the outer loop.

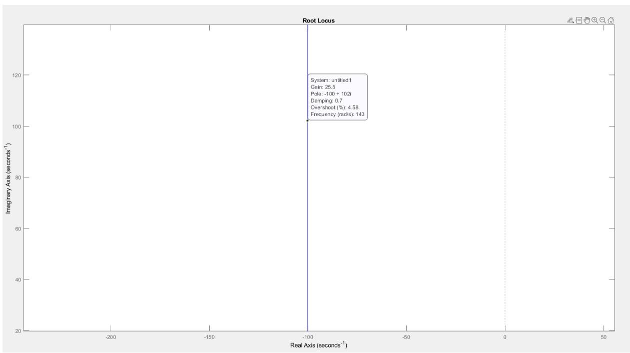

Figure 15: The maximum gain leading to a damping ratio of 0.7

Finally, the gain and phase margin are studied. It gives an indication on the maximum gain we can add to the loop without causing instability in the system. For the calculations, the Pade' approximants are discarded as the real zero-order hold function is needed. By plotting the Bode diagram of the system including the real ZOH transfer function and the inner loop transfer function, we can find more realistic gain and phase margin.

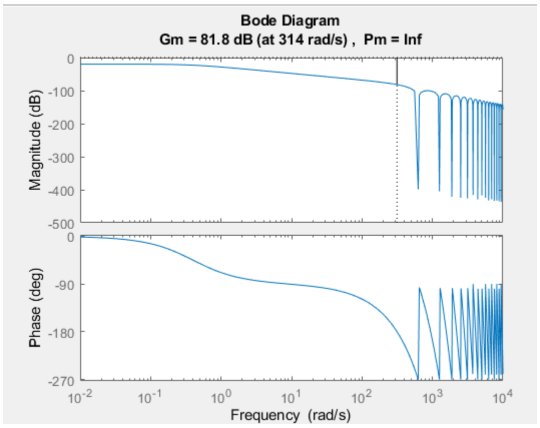

Figure 16: Gain and phase margin

It is obvious on the Bode diagram that K_p can take a very high value before causing instability. The maximum K_p in root locus is then restricting parameter.

The equilibrium point for the ball position is 6.2V in input which leads to the ball levitating. We will study the system moving around this equilibrium point. As a simple P is considered, the static error is non-zero. The input voltage for the inner loop to get to the equilibrium 6.2V at the output is approximately 6.4V. Therefore, the maximal value we can put in the input after the equilibrium is 10-6.4 = 3.6V. To avoid saturation in the motor, the K_p can have the maximum value of $\frac{10}{3.6} = 2.777$. The value of K_p is then set to 2.775.

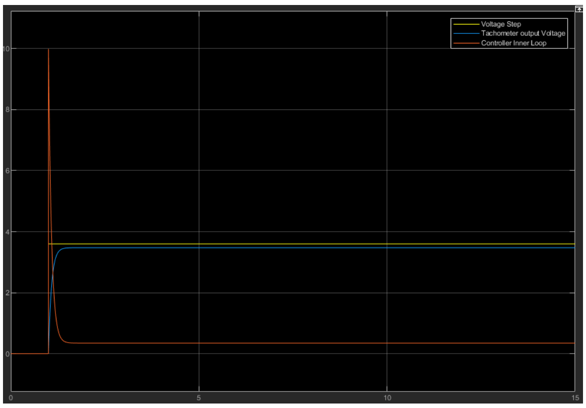

Figure 17: Response of the controlled inner loop to a big current step

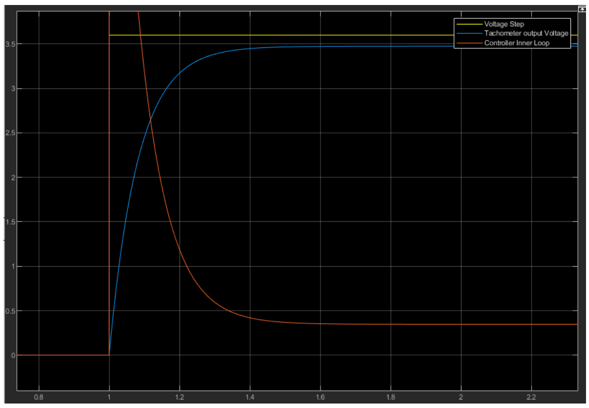

Figure 18: Zoom on the rising part showing the settling time

The saturation period is almost zero which is sufficiently small. The settling time is about 0.25 seconds, as the requirements states that the ball needs to get to the reference position within 2 seconds, 0.25 seconds is very fast compared to the global time restriction.

Last thing to do for the inner loop is to verify that the controller designed on the model works the same way on the real plant.

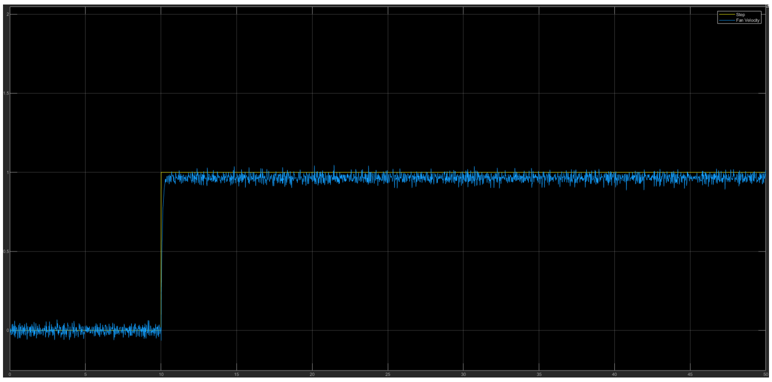

Figure 19: Real controlled inner loop on the plant

The real fan velocity follows the step with a constant static error but there are no oscillations or overshoot. We can conclude that the inner loop is correctly designed as the inner system reacts very fast to the input.

6 Outer loop

6.1 Modelisation of the distance sensor

Before making experiments and calculations on the outer loop, we need an interpolation function transforming the voltage into a meter reference and vice versa. To find it, we can use the interpolation block and place all the points of 2 in the block. This way, the voltage input from the tachometer the plant output for the position of the ball can be transformed into meters.

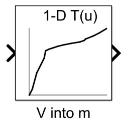

Figure 20: Interpolation function changing volts into meters

The sensor operates in two different linear zones that need to be taken into consideration. The first one between 0 and 1.2m which is the less sensitive zone with a slope approximately equal to 1 between 1.2 and 2.5V and the second zone that is very sensitive with a slope of around 8 between 1.2 and 1.8m or 2.5 and 7.3V.

6.2 Identifying the transfer function

To identify the transfer function of the system having the fan tachometer voltage as an input and the ball position as an output, the same identification function is used. We know from section 3.1 that the second system is an integrator. As a consequence, the input will be integrated leading to transforming a simple step into a ramp. A simple way to identify the function is then to put an impulse in the system and observe the response.

First, the operating point is needed. By testing, we found that the ball starts to lift up with a 6.2V.

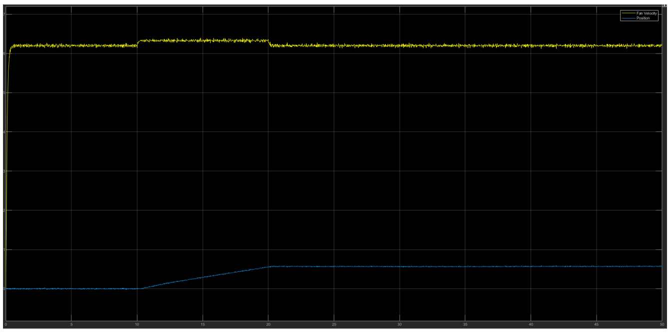

Figure 21: Equilibrium point

It is seen on this figure that for a small step at 6.2V the ball starts rising up in the tube and most important, the ball stays at its position after the step is removed and the fan tachometer is back at 6.2V. This means that the fan velocity allowing the ball to levitated but not going up is around 2500rpm. As we are going to study the motion of the ball around its equilibrium, we need to add the value leading to 2500rpm at the input of the inner loop and remove 6.2V at the input of the outer loop. The static error is non-zero (as explained previously), with a K_p of 2.775, the value leading to 6.2V at the output of the inner loop is found by testing to be equal to 6.4227V.

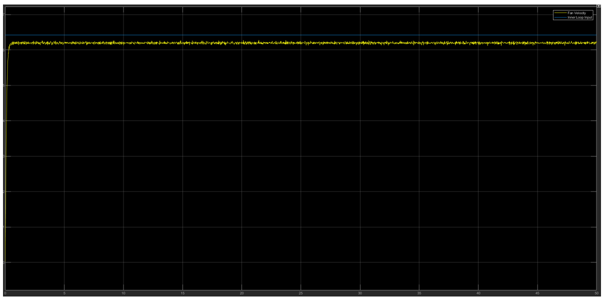

Figure 22: Response of the inner loop to a constant input of 6.4227V

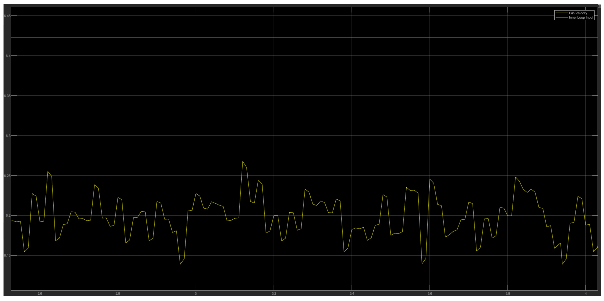

Figure 23: Zoom on the equilibrium values for the inner loop

With this point, it is possible to put a simple step of 1V in the system and observe the coefficient of the ramp of the response. As the system is approximated by an integrator with a gain, by finding the slope of the output to a step, it is possible to find the gain of the integrator. The step of 1V is transformed into a meter step through the interpolation previously made.

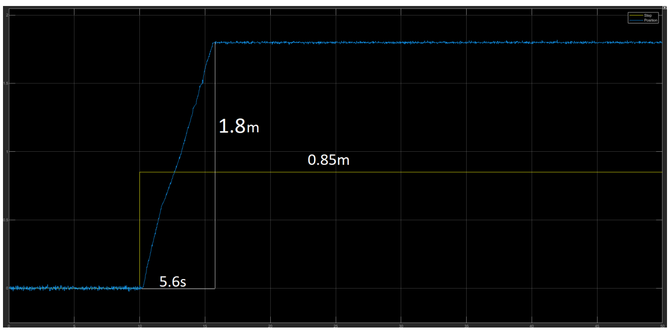

Figure 24: Integrator gain identification

The gain is then:

$$gain = \frac{\frac{1.8}{5.6}}{0.85} = 0.378151 \quad \Longrightarrow \quad G_2(s) = \frac{0.378151}{s} \tag{1}$$

The subsystem representing the second system is consequently made by:

Figure 25: Second subsystem

The saturation zones are the top and bottom positions of the ball in meter, 0 for the bottom point and 1.8m for the highest. The 1.2V is the value given for 0 meter that we need to add to the voltage reference. The 0.174 value is an experimental constant allowing a response closer to the plant. Actually, the value of 6.2V is subtracted at the input of the second system because of the equilibrium point. Then, there is a small time interval at the beginning where the second system has a negative value at the input (time for the fan to rise at 2480rpm or 6.2V). The constant added is computed to remove this small negative offset due to the rising to the equilibrium point.

6.3 Validation of the second system

Last thing to do in order to confirm the quality of the model is to put a step in both the real plant and our model and observe the matching of the responses.

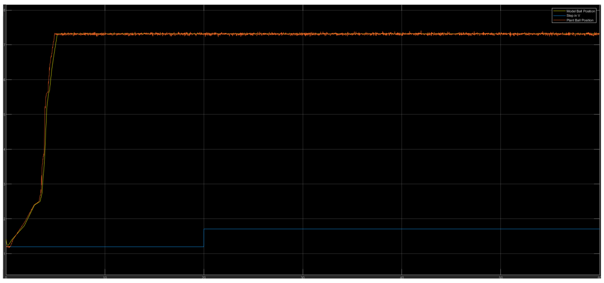

Figure 26: Validation of the model in open loop

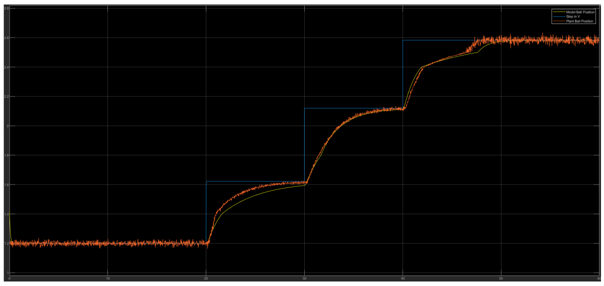

Figure 27: Validation of the model in closed loop

In open loop and in closed loop without any controller ($K_p = 1$), the model follows well the real plant. Although, there are some errors on the model, probably due to the interpolation function of the sensor which is not perfectly made. The model is sufficiently accurate to base our calculations on it.

6.4 Introduction of noise in the model

We can notice a difference between 17 and 19, in the real plant there is noise. This can be explained by the fact the both sensors produce noises.

This will be taken into account in our model by adding a *Band-Limited White Noise* block with a noise power of $0.1W$ for the inner loop and $10^{-6}W$ for the outer loop. Those values were found by comparison with the real plant.

6.5 Regulation

Several methods have been tested to approach the requirements.

6.5.1 Proportional regulation

As explained before, the distance sensor has 2 linear zones with two different slopes. We will start by selecting the upper zone (1.2m to 1.8m or 2.5V to 7.3V) which has a bigger slope of 8.03, in order to have a more accurate regulation. The first requirement is easily obtained with a P regulator as the second system is already an integrator. The steady state error will naturally tend to zero as shown on the figure 27. For the other regulations, the position of the poles needs to have a damping ratio high enough in order to have 1cm of overshoot and a real part bigger in absolute value than a given number.

The first regulation tried was a proportional regulation as seen in 28.

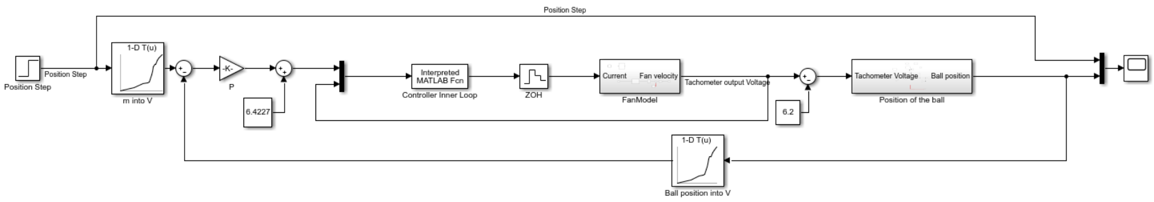

Figure 28: Proportional regulation of the outer loop

In order to find a satisfying proportional regulator the MATLAB tool *rltool* is used. It allows us to express the requirements and to move the poles to respect the requirements. As the maximum overshoot is 1cm, it corresponds for the zone between 1.2 and 1.8m to $\frac{1cm}{60cm} * 100\% = 1.666\%$. By plotting the root locus of the outer loop system (the closed inner loop with the controller with the second system transfer function and the gain of the sensor), the following root locus is obtained.

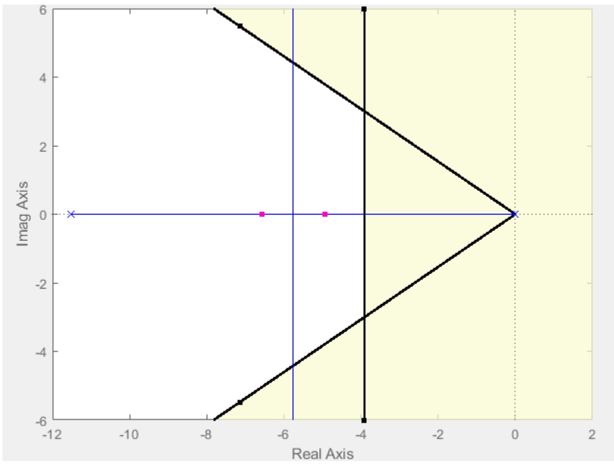

Figure 29: Root locus for the P controller

By moving the poles to the limiting values for the settling time and for the overshoot, the maximum and minimum values of K_p can be computed.

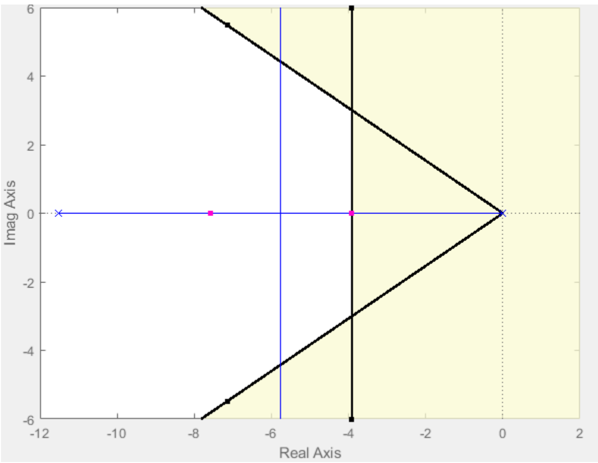

Figure 30: Minimum K_p gain of 0.88255

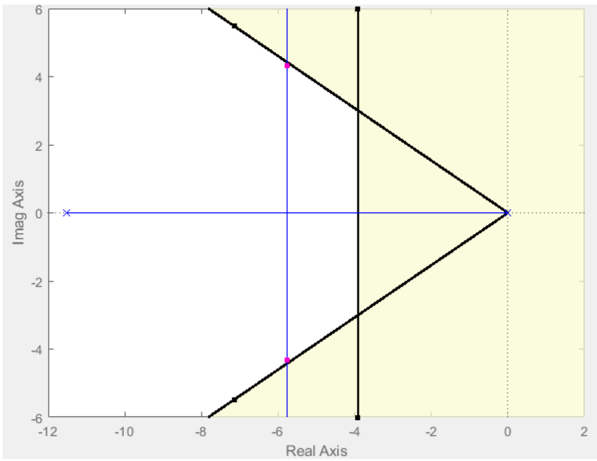

Figure 31: Maximum K_p gain of 1.5581

By choosing a K_p equal to 0.9821, the two poles are combined on the real axis which means that there is no overshoot and that the response is relatively fast. This P controller on our model gives sufficient results to fulfill all requirements.

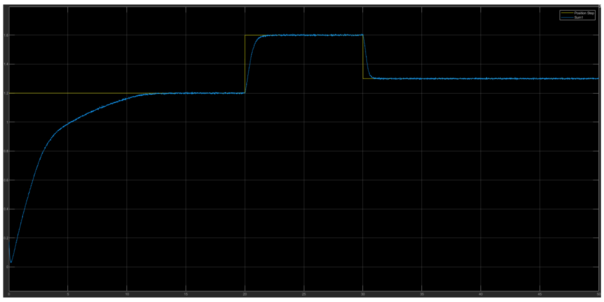

Figure 32: Test of the regulation on the model

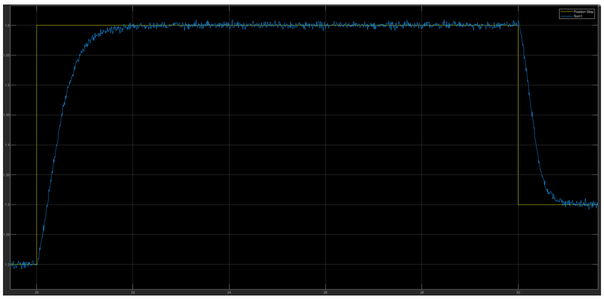

Figure 33: Zoom on the rising parts

The response is nice in the interest zone. The rising from 1.2 to 1.6m takes 1.5s and there is no overshoot. Both requirements are then fulfilled. Let's check if this regulation works the same way on our plant.

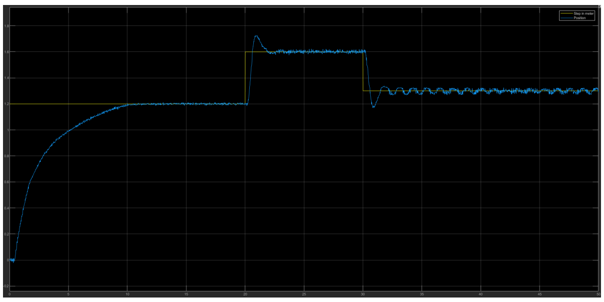

Figure 34: Plant controlled by the P controller

The settling time is a bit long and an overshoot appears. This is probably due to the simplification we made on the form of the second system which is the combination of a first order system and an integrator in reality. Therefore, this regulation does not fulfill the requirements.

6.5.2 Proportional and Derivative Regulation

In order to limit the overshoot and decrease the settling time, the derivative regulation is added. As the second system contains already an integrator, the integrated regulation is not needed for a zero static error.

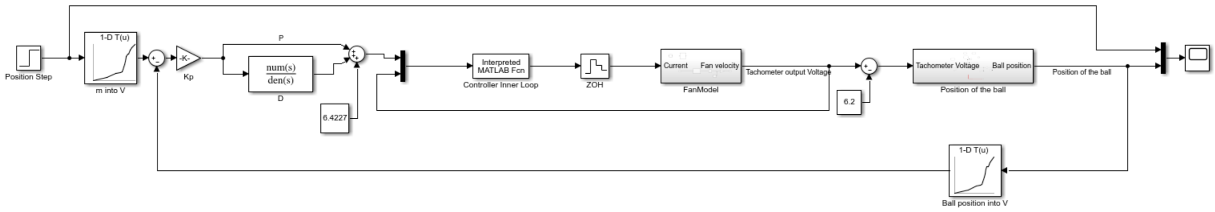

Figure 35: Proportional and derivative regulation of the outer loop

Again with *rltool* and the requirements, we add a real zero to the root locus. In order to make the response faster, the zero is placed close to the pole to have a pole-zero cancellation making the settling time smaller.

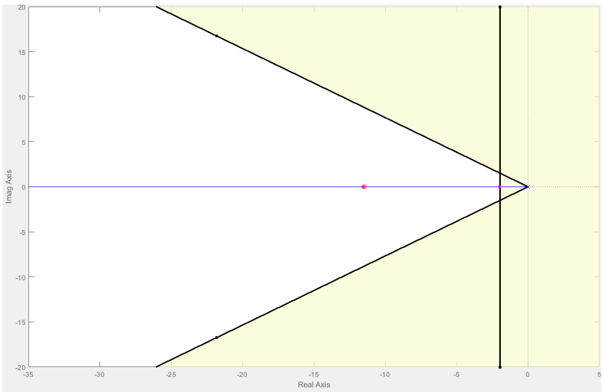

Figure 36: Zero placed on the inner loop pole

The zero is located in -11.5. But to make the derivative action into a proper transfer function, it needs a filtering pole usually placed between 3 and 10 times the zero location.

$$C_d(s) = T_d s \quad \Longrightarrow \quad C_d(s) = \frac{T_d s}{\frac{T_d}{N}s + 1}$$

The N value is arbitrarily set to 5. The derivative controller transfer function is given by:

$$C_d(s) = \frac{0.087s}{\frac{0.087s}{5}s + 1}$$

The minimum value of K_p giving a sufficiently small settling time is K_p = 0.70753. The full controller is the sum of the derivative action and 1, multiplied by K_p.

$$C(s) = 0.70753\left(\frac{0.087s}{\frac{0.087}{5}s + 1} + 1\right) \tag{2}$$

But first, the phase and gain margin need to be checked in order to ensure the good choice of K_p and add the delay to the full system.

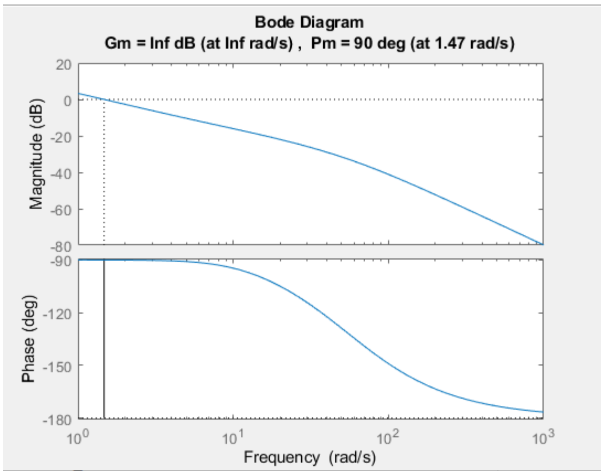

Figure 37: PD controlled full system margin

The gain margin is infinite which means that we could theoretically increase K_p to infinity and the phase margin is 90 degrees which is superior to the good practice value of 30 deg. However, increasing the K_p can create oscillations and make a bigger noise in the response. The value chosen is then $K_p = 0.70753$ in order to fulfill the requirements without problems.

This controller implemented in our model and in the real plant gives better results than the previous P.

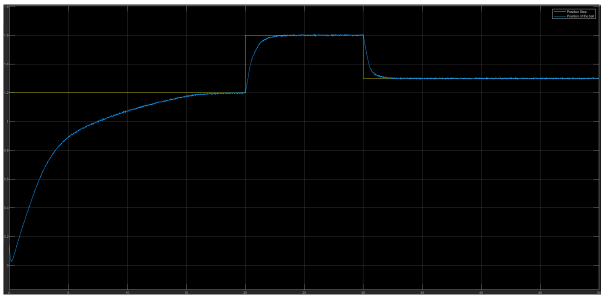

Figure 38: Test of the regulation on the model

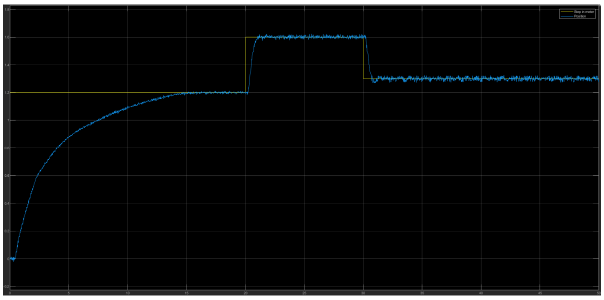

Figure 39: Test of the regulation on the plant

The settling time fulfills the requirements as the plant makes the rising from 1.2m to 1.6m in 1.3 seconds and the model is also below the 2 seconds. The plant does not produce a big overshoot as in the figure 34 anymore. Even though, there are a lot of noise around 1.3m which is probably caused by the distance sensor because 1.3m is just between its two linear zones.

6.6 Extension to the full tube zone

In order to work in the full range of the tube, we need to design another regulation since the slope of the infrared sensor characteristic is different in the 0 to 120cm zone. Since poles and zeros of the regulated system will be the same as in the 120-180cm zone, only the gain needs to be found.

6.6.1 Gain identification

To do so, we use the same method as in the section 6.5.2, with an overshoot percentage of $\frac{1cm}{120cm} * 100\% = 0.833\%$.

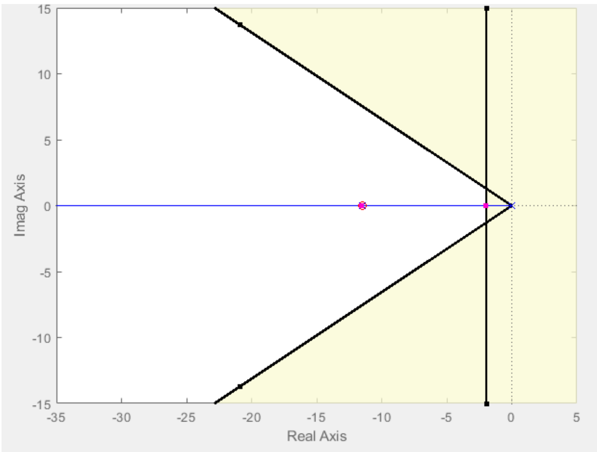

Figure 40: Root locus for the PD controller in the second zone

We find that a gain of 5.0766 is the minimal value for this zone. This regulation for the zone between 0 and 1.2m gives sufficient results on both plant and model.

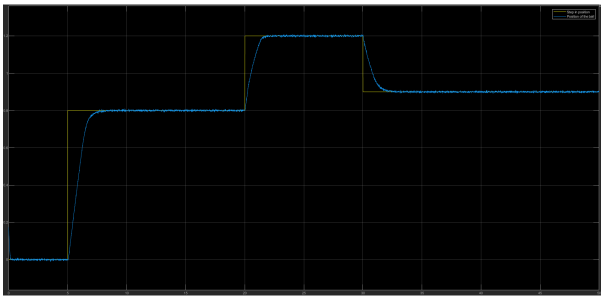

Figure 41: Test of the regulation on the model

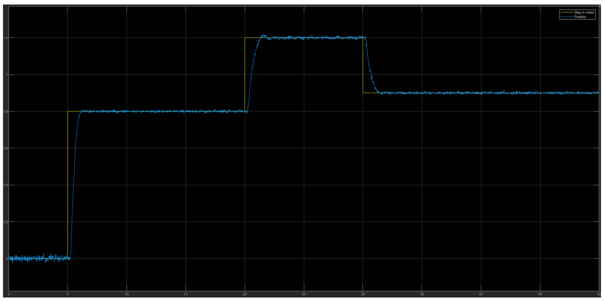

Figure 42: Test of the regulation on the plant

The settling time is sufficient in the model and very good for the plant. This small difference is explained by the first order system in the real second system of the plant not considered in the model. Also, there is a small overshoot in the plant response, close to 1cm for the jump from 0.8m to 1.2m and a smaller one for the two other steps. We can globally conclude that this regulation fulfills the requirements for both the model and the plant in this zone.

6.6.2 Regulation of the full tube

We then have 2 zones with 2 different regulations :

1. The 120-180cm zone:
$$C(s) = \frac{0.07387s + 0.7075}{0.0174s + 1} \tag{3}$$

2. The 0-120cm zone:
$$C(s) = \frac{0.53s + 5.0766}{0.0174s + 1} \tag{4}$$

7 Implementation

Previously, we only considered continuous regulation. But in reality, we need to make a discrete-time regulation. For this purpose, the block MATLAB function is used with a sample time of 0.01. In simulink, several global variables are declared and it is necessary to store the previous data allowing to make discrete-time derivative and integral regulation.

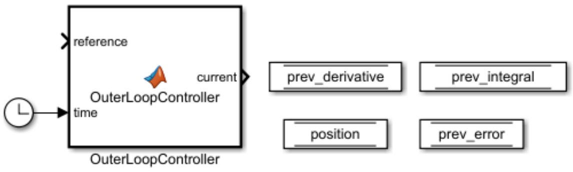

Figure 43: Matlab blocks for the discrete-time implementation

To compute the controller actions in discrete-time, the Z-transform of the controller is needed. We used the *c2d* matlab function to transform the previous controller in the equation 2 in its Z form by using Tustin approximation. The controller law in the first zone becomes:

$$C_d(s) = \frac{0.087}{0.0174s+1} \implies C_d(z) = \frac{3.884z-3.884}{z-0.5536}$$

As we know that for the derivative action: $U_d(z) = C_d(z)E(z)$, we can now compute the law in discrete-time. The final derivative law becomes then:

$$U_d(z)(z-0.5536) = E(z)(3.884z-3.884) \implies u_d(k) = 0.5536u_d(k-1)+3.884e(k)-3.884e(k-1)$$

with e(k) the error between the reference and the response at step k.

In the beginning of the code, the position is checked in order to know in which linear sensor zone the ball is. Then, the K_p value is set to the one corresponding to the right zone. To have the total regulation, the next input is computed as:

$$u(k) = K_p * (e(k) + u_d(k))$$

The final implementation on the model is in the following form:

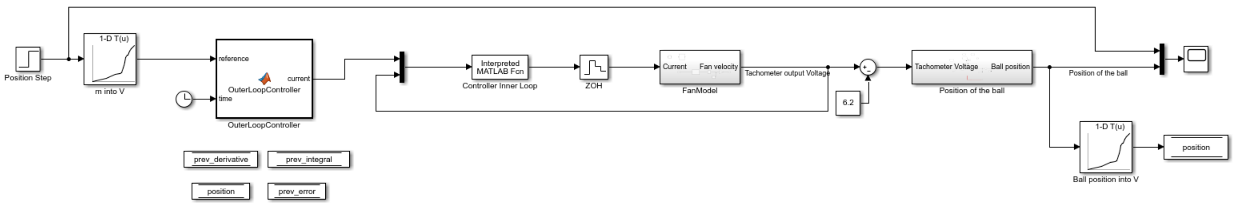

Figure 44: Discrete-time PD regulation of the outer loop

This kind of implementation allows to change the K_p of the 2 zones easily. The output seems good but to be sure that the discrete-time controller is the same as the continuous-time one, it would be useful to compare the responses to the same input.

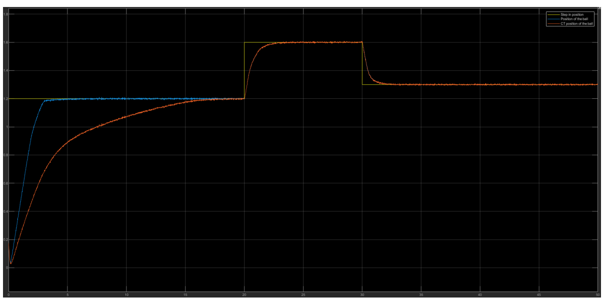

Figure 45: Difference of the model responses between continuous-time and discrete-time regulations

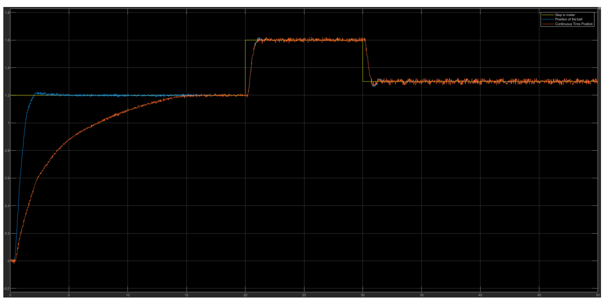

Figure 46: Difference of the plant responses between continuous-time and discrete-time regulations

The difference for the first step is explained by the changing K_p in discrete-time but for the rest, the discrete-time is in a good match with the continuous regulation.

8 Discussion

We developed 2 regulation for this plant. The first one was a proportional regulation. It achieved control of the position of the ball but did not meet the requirement cited in 2.2.
We then designed a proportional and derivative regulation which achieved control of the ball and met the requirements around our equilibrium point.
We then designed a second PD regulation to take into account the change of slope in the infrared sensor characteristic (2).
Thanks to those 2 different regulation, the zones 0-130 and 150-180 respect the requirements.
In the 130-150 zone, oscillations and noises appear which make the regulation disrespect the requirements. Our interpretation of these oscillations and noises is that they come from the non linearity of the sensor in that zone. To respect the requirements in the full tube, a third regulator should be designed for this specific zone.