Practical work 1
Group 2- Team B01

Control of Dynamic Systems

Ana Luisa Fogaca Nogueira
Pedro SacramentoXavier Barreto Rosa
Victor Freitas Teodoro

Metz
CentraleSupelec

CentraleSupélec

October 14, 2022

Table of content

1 Objectives

The purpose of this practical work is to open servos to control a real device (Figure 1) to different specifications. The calculations and modeling were carried out within the framework of the concepts seen in the Dynamic Systems Control courses, and the simulations were obtained through the SIMULINK extension of MATLAB.

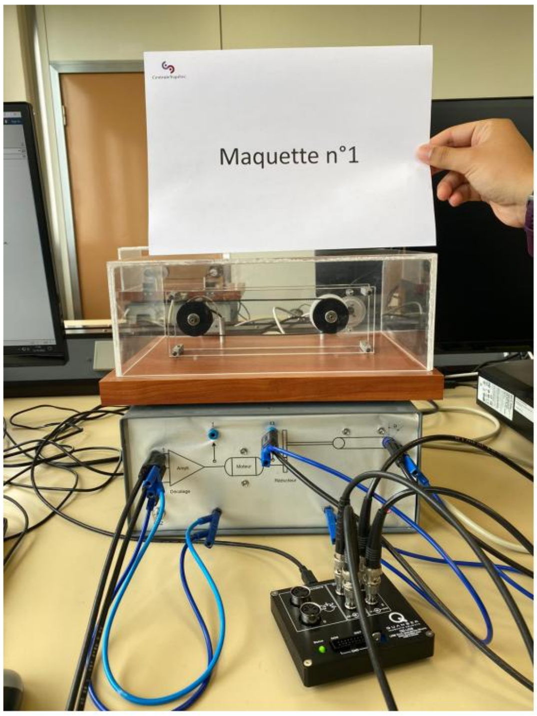


Figure 1: model nº1.

2 Approaches and results

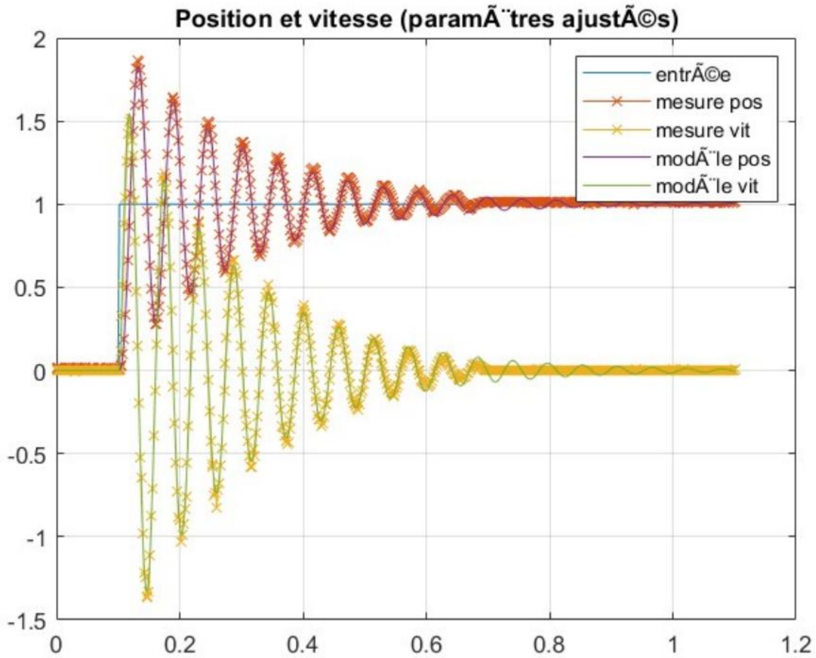
2.1 Identification

The model chosen to study the real system is fundamentally governed by 5 parameters, (speed constant), mechanical time), (electrical time constant), (ratio (constant of unknowns at first. They are: between ( ) and ( ) and (ratio between ( )). Without knowing the values of these parameters, we can ( ) neither make simulations nor create the necessary controls. We then use standard values, provided in the subject, and carry out a comparison between the simulation of the model and the real system. An algorithm which minimizes the difference between the two responses by varying the values of the parameters will provide approximations for the latter (figure 2). The standard values and the values provided by the algorithm are found in Table 1.

	Standard Value	Estimated Value
(/ /) (15)	35	45.6
(3) (/)	15	15.6
(/ /)		2.8
	3	0.5
	0.5 0.001	0.009

Table 1: System Parameters

.



real system	modeled system	cosign

Figure 2: Setpoint and Position Measurement without offset.

2.2 Serial Correction

For the open-loop system, it is verified that there is an undesirable response from the system during a change in the setpoint, either for too great a displacement, or for too great a rise time. For this reason, the addition of a continuous corrector to the system is necessary to respect the specifications imposed by the statement. To set up such a corrector, the proper pulsation of the system was calculated from equation 1 extracted from the analysis of the chart illustrated in figure 3.

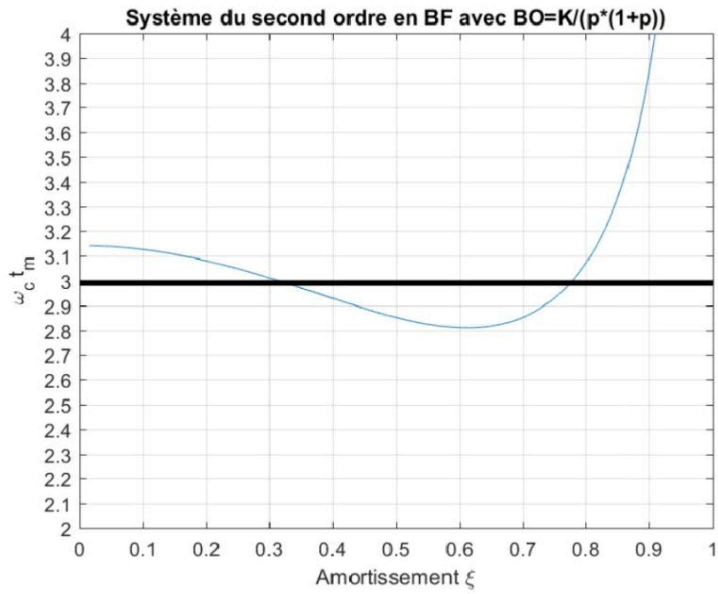


Figure 3: Abacus with the relation of Damping and .

$$= 3 \quad (1)$$

As, according to the specifications, = 40 , we obtain = 75 / . In addition, with the desired displacement value of 15%, the phase margin of = 55 is calculated from the chart contained in figure 4 .

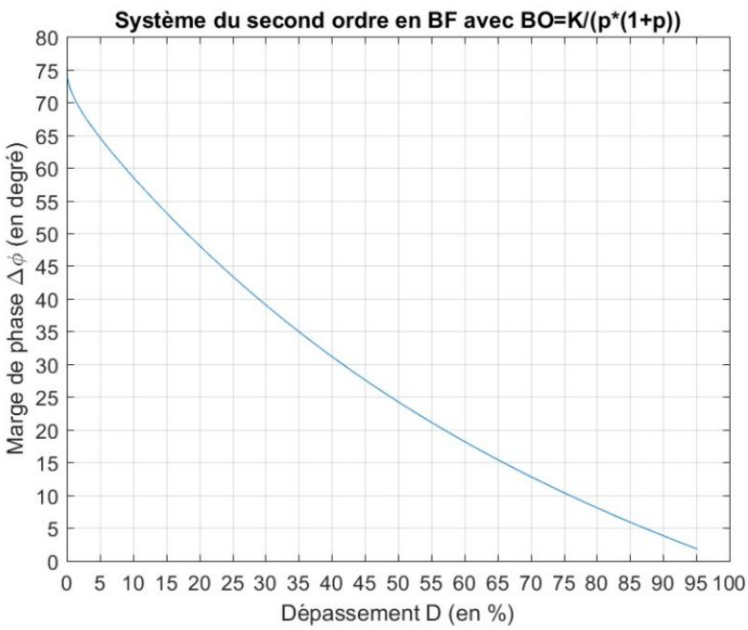


Figure 4: Abacus with the Overshoot and Phase Margin relationship.

Next, the procedure for creating a phase advance corrector, as indicated in

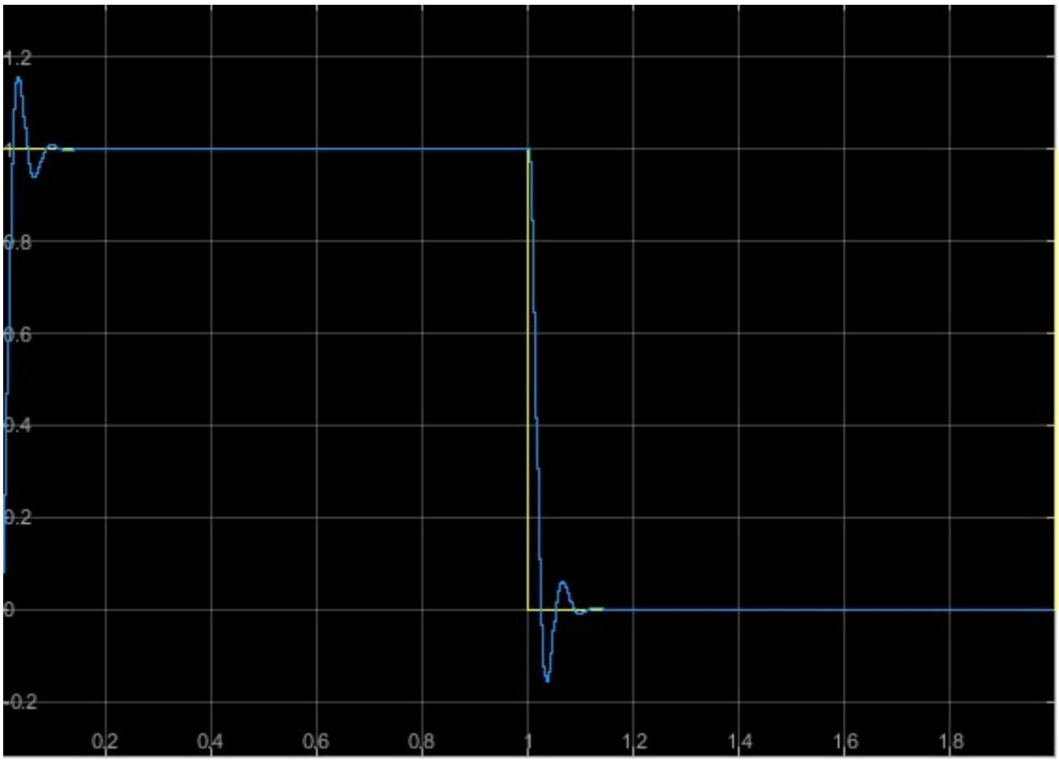
Appendix A-3 of the course handout, was made on MATLAB. The gain *G* = 0.1929 and le = ÿ151.6315 were found from the “bode” function (shown on topic item 6.6.4). The analytical form was also determined from data found on MATLAB.

To be able to study the action of the corrector on SIMULINK, the latter is discretized through the formula **c2d,** indicated in section 6 of the subject. This allows the determination of the matrices numC and denC:

= [7.9673 ÿ7.2754] = [1 (2)

ÿ0.7795] (3)

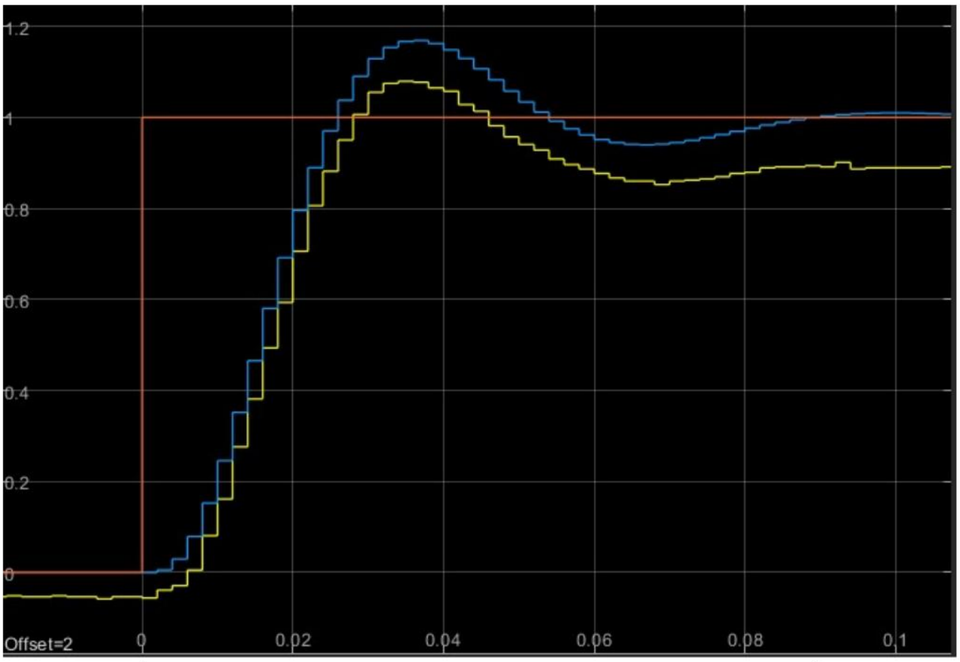
Figure 5 shows the simulation of the system under the influence of the modeled continuous corrector. We observe that the overshoot is well around 15%, the rise time is about 40ms and that the system converges towards the setpoint.



cosign answer disturbance

Figure 5: System response with continuous corrector.

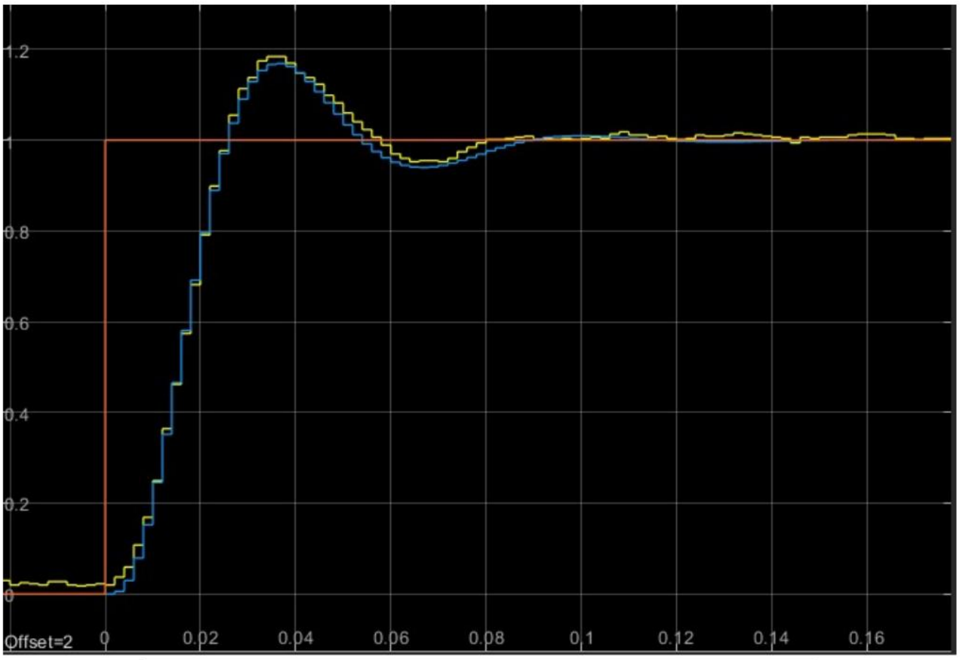
The simulation is validated by comparing the response of the modeled system with that of the real system. Figure 6 shows that there is a similarity between the formats of the two curves, but with a certain difference. This difference is due to the offset of the real system, which was not modeled.



real system modeled system cosign

Figure 6: Setpoint and Position Measurement with offset.

After making adjustments on the real system to remove the effect of the shift, we see that the deviation no longer exists and the curves are very similar, as shown in Figure 7. In the next comparisons between the model and the system carried out in this study, the effect of the shift will no longer appear, due to the adjustments carried out on the model beforehand.



real system modeled system cosign

Figure 7: Setpoint and Position Measurement without offset.

2.3 Modal command

We start with the transfer functions:

$$\frac{(\ )}{(\ )} = \frac{}{(1 + \quad)} \text{ÿ} \quad (4)$$

$$(\ ) = (\ ) + \quad (\ ) \quad (5)$$

And thanks to the use of the inverse Laplace transform:

$$(\ ) = \text{ÿ} \quad \frac{(\ )}{} + \frac{(\ )}{} \quad (6)$$

As :

$$(\ ) = (\ )\ (\ ) \quad (7)$$

$$(\ ) = \frac{}{} \quad (8)$$

$$(\ ) = \frac{(\ )}{} \quad (9)$$

It is possible to write:

$$(\ ) = \begin{bmatrix} 0 & - \\ 0\text{ÿ} & \frac{1}{} \end{bmatrix} (\ ) + [\ 0\ ] \frac{}{} \quad (\ ) \quad (10)$$

Or :

$$(\ ) = [\quad (\ )] \quad (11)$$

The controllability of the system can be determined through the Kalman criterion. We sees that the matrix is of full rank, so the system is fully controllable.

$$= (\quad) = \text{ÿÿÿ} \begin{bmatrix} 0 & \frac{}{2} \\ \frac{}{} & \frac{}{2} \end{bmatrix} \quad (12)$$

Next, we need to find the numerical equations of the state space. For this objective, we use the c2d function of MATLAB, as well as and the values present in the statement. The resulting matrices are found in [13].

$$= [1\ 0.0936\ 0\ 0.8752] \qquad = [\ 0.022\ 0.0437] \quad (13)$$

To find the desired location of the poles, the following piece of code was used:

```
% second -order damping ratio and natural frequency
ksi = abs(log(D))/sqrt(log(D)^2 + pi^2);
w0 = pi/(tm * sqrt(1-ksi^2));

% desired poles
pk = roots ([1 2*ksi*w0 w0^2]); % continuous
zk = exp(pk*T); % discreet
```

Where **tm** is the first maximum time and **D** is the overshoot. It gives:

$$_1 = \text{ÿ}63.24 + 104.72 = \qquad _2 = \text{ÿ}63.24\ \text{ÿ}104.72 =$$
$$_1\ 0.86 + 0.18 \qquad _2\ 0.86\ \text{ÿ}0.18 \quad (14)$$

Next, find the appropriate L-matrix gains. This is done using the **place** function of MATLAB. We also check the behavior of the system subjected to a step, figure 8.

```
% pole placement
L = place(stateD.A, stateD.B, zk)

% closed loop discrete system
cloopD = L(1) * feedback(stateD ,L);
step(cloopD)
```

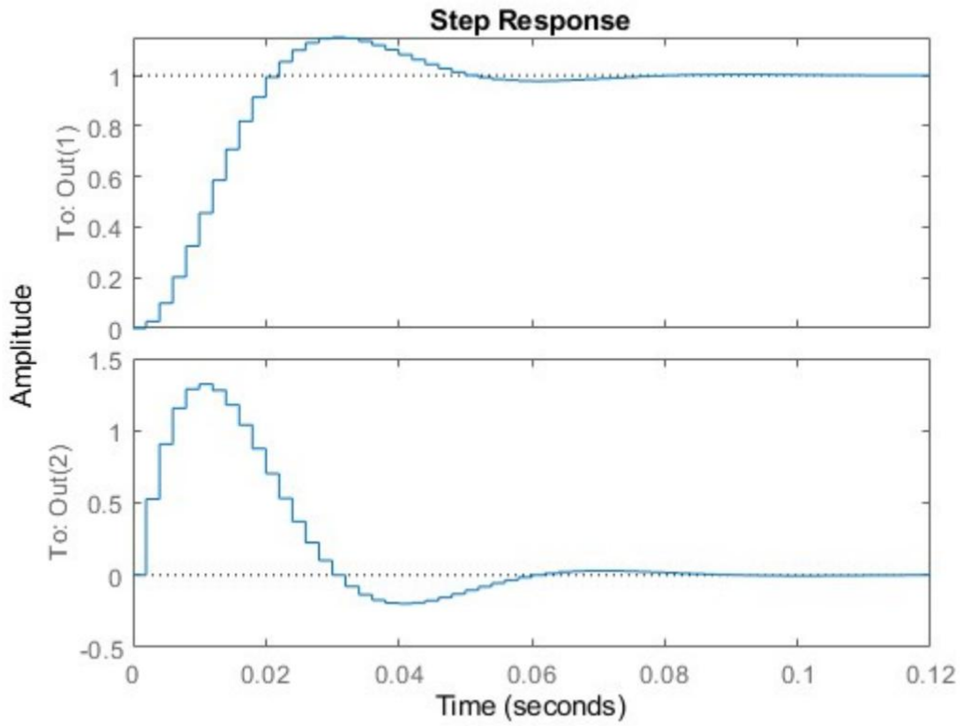
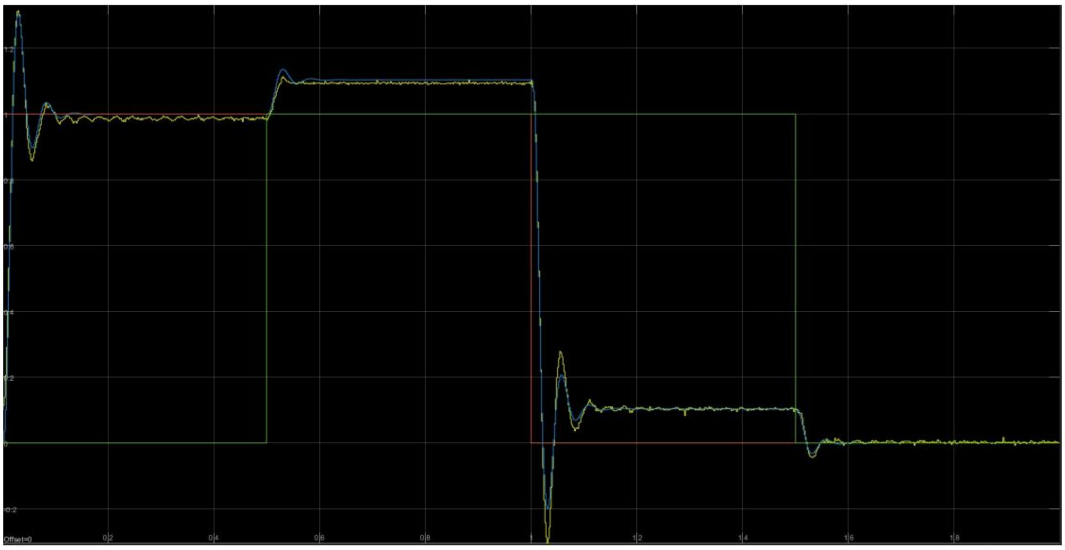


Figure 8: Step response.

$$= [12, 0457 \quad 2, 8473] \tag{15}$$

It is emphasized that in this modeling the order of the state variables is and for followed by this reason the sequence of values in the matrix can change.

Figure 9 shows the comparison between the behavior of the simulation and that of the real system. An approximation between the two curves is noted, even if there are certain disturbances which have not been modeled (friction, surface imperfections, etc.) and which cause noise.



real system	modeled system	cosign	disturbance

Figure 9: Setpoint and Position Measurement without offset.

2.4 Static error cancellation with integral action

Aiming to cancel the static error, we add to the system an integrator whose output is given by (ÿ) and the transfer function is given by 1ÿ ÿ1 . This is done by inserting the variable *z* into the state vector, so as to obtain an extended state representation (16 e 17) whose gain K, now composed of 3 poles, will allow the elimination of the statistical error.

$$( ) = \begin{bmatrix} 1.0000 & 0.0938 & 0 \\ 0 & 0.8750 & 0 \; 0 \\ 1.0000 & & 0 \end{bmatrix} [( )] + \begin{bmatrix} 0.0022 \\ 0.0438 \; 0 \\ \end{bmatrix} ( ) + \begin{bmatrix} 0 \\ 0 \\ ÿ1 \end{bmatrix} \quad (16)$$

$$= [1 \; 0 \; 0] [( ) ( )] \quad (17)$$

For the controllability of the system, we show that:

$$[\qquad 2] = \begin{bmatrix} 0.0027 & 0.0079 & 0.0124 \\ 0.0477 & 0.0420 & 0.0369 \\ 0 & 0.0027 & 0.0106 \end{bmatrix} ÿ ÿ 0 ÿ \quad (18)$$

So, the matrix is of full rank and the system is fully controllable.

With the aim of finding the new gain K, we first introduce equation 19 to find, respecting the specifications provided = 50 and , *D = 15%,* what the damping results in = 0.52. With this value, equation 20 is used to find the pulsation clean 0 = 73.4.

$$= \frac{| \; (\;)|}{\sqrt{ÿ ( )2 + 2}} \quad (19)$$

$$_0 = \frac{}{\sqrt{ÿ1ÿ 2}} \quad (20)$$

$$ÿ \quad 2 + 2 \quad 0 + \;^2_0 \quad (21)$$

Then, we make an approximation of the model by a second order system, whose characteristic polynomial is defined by equation 21. The roots of the polynomial are calculated and the third pole is obtained by multiplying by a factor 5 the most negative part real among the roots. The result is given by equation 22. Equation 23 gives the values of , which is the discretized vector for a numerical processing.

= ÿ0.3794 + 0.6283
ÿ0.3794 ÿ0.6283
ÿ1.8971 (22)

= = 0.9196 + 0.1162
ÿ ÿ 0.9196 ÿ 0.1162
ÿ 0.6843 (23)

Finally, using the *place* function in MATLAB with the vector matrices as arguments, we find: , and the

= [12.71047.03131.2004] (24)

Figures 10 and 11 illustrate the extended state asserv simulation_sim.slx obtained on MAT LAB using the vector L found in the extended state asserv ini file. It is possible to observe in Figure 10 that the statistical error no longer exists, even with the intervention of the disturbance, and the model seems to exhibit the desired behavior. The only problem with the integral action is the saturation of the u command, which reaches the -10V limit during the simulation, as seen in Figure 11.

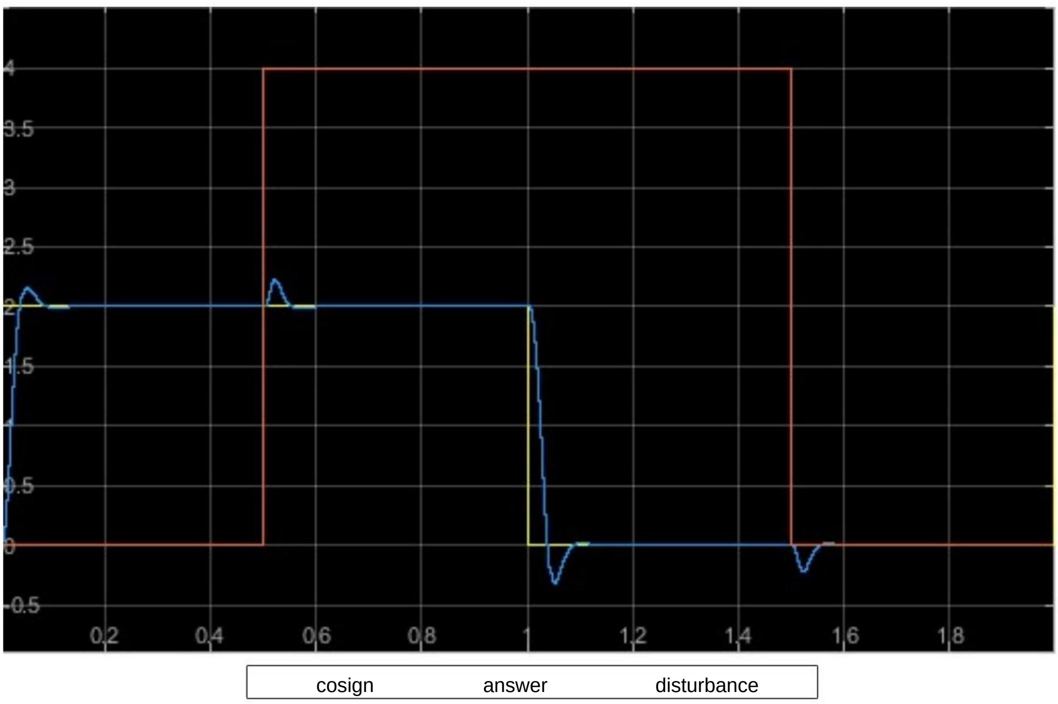


Figure 10: System response with integral action.

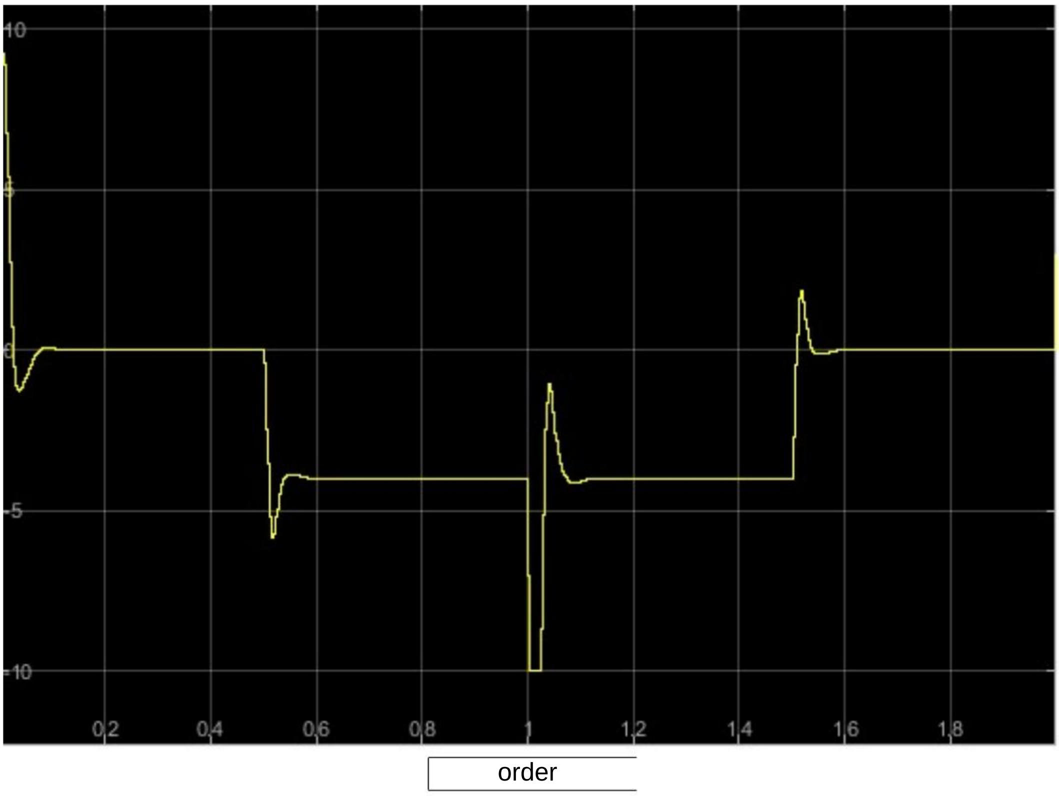


Figure 11: Intensity of the *u* command with integral action.

Finally, we observe in figure 12 that the response of the simulated system and that present for the real system follow almost the same trajectory. Even after the setpoint changes and the action of the disturbance, the curves are very similar. The experience therefore validates the implementation of the controls requested in this section.

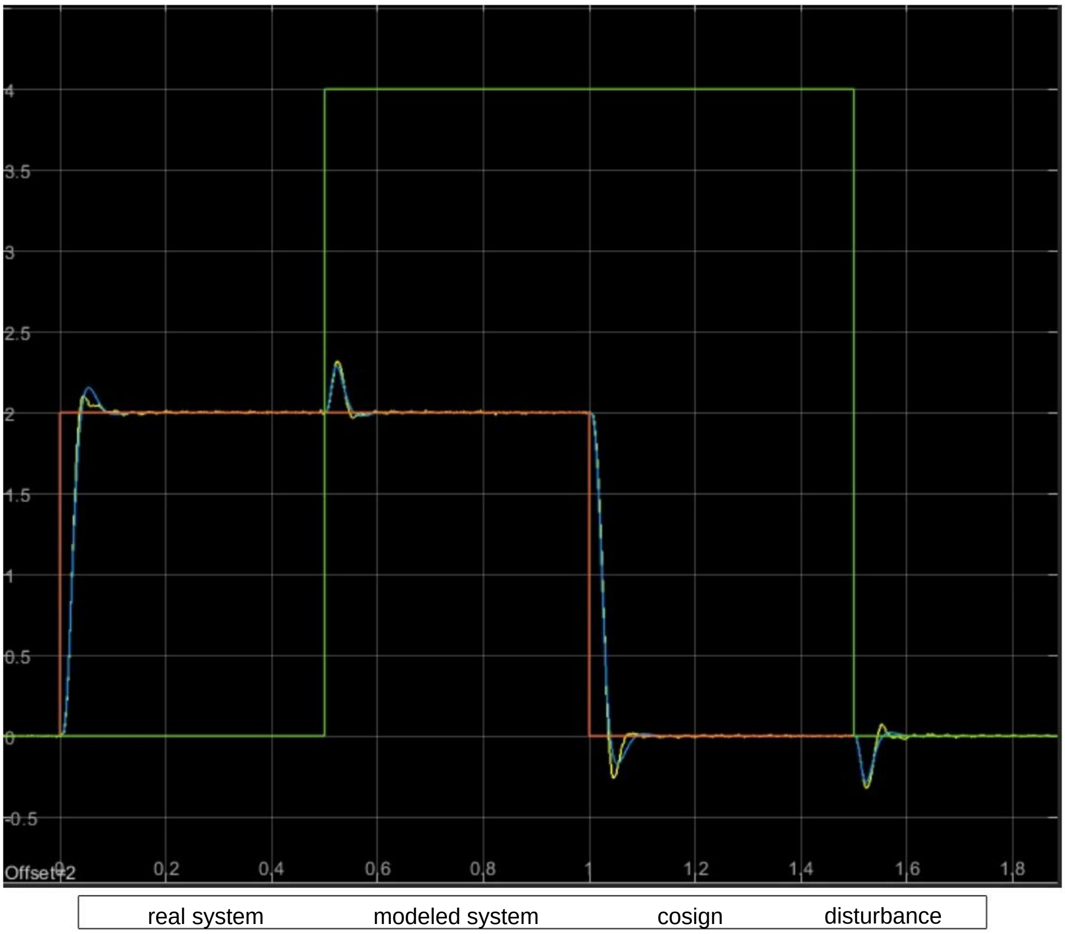


Figure 12: Validation by comparison of the system models and the real system.

2.5 Quadratic linear control with integral action

Although the integral action constructed in the previous section fixed the static error, its implementation created the command saturation problem, which prevents its use in a real system. To solve this problem, we add a Linear Quadratic command to the integral action, using the following cost function:

$$= \sum_{=0}^{ÿ} ÿ \quad 1 \quad (\quad)^2 + 2 (\quad (\quad) ÿ (ÿ))2 + 3 (\quad (\quad) ÿ (ÿ))2 + (\quad)^2$$

The conditions of existence of a solution of J are such that (Aa, Ba) is stabilisable, a fact which has been demonstrated in section 5.4, and that (Aa, Ha) is detectable, where = By setting = [1 2 3], we find:

$$(\quad) = \quad ÿ\,2\,ÿ \quad = \frac{2}{3}\,2\,(64 \quad \frac{469\;1}{4000} + \frac{469\;3\,)}{5000}$$

For that to have full rank, that is to say) ÿ 0, if just 3 ÿ 0. With this expected condition, we show that is observable, therefore detectable, and we can freely arbitrate the values of .

1 And 2

Once the 1 , 2 components of vector H are defined, just use MATLAB's *dlqr* function to find the gain vector L, as well as ensure that the value

Iteration	1 And 2		l	Max |u|
1	0	3	[6.3085 16.9278 1.6798]	>10
2	0	2 1	[4.8168 11.0349 0.8778	=10
3	0	0.85	[4.5115 9.9812 0.7527]	9.7

Table 2: Gain L found with LQ

.

absolute maximum of *u* does not exceed the imposed limit interval [-10 V, 10 V]. If this value exceeds the limit, 1 , 2 , are reset and the process is repeated iteratively. and One finds in table 2.5 the results of the iterations carried out for the system models.

Figure 13 shows the extended state asserv simulation sim.slx obtained at MATLAB using L = [4.5115 9.9812 0.7527] in the extended state asserv ini file. It can be seen that the characteristics of the response, due to the servocontrols carried out in the preceding sections, are maintained. The big difference is found in figure 14, where it is possible to observe that the commands of the system no longer saturate, ie |u| < 10. This means, therefore, that we have ended up with a system that respects the physical constraints imposed.

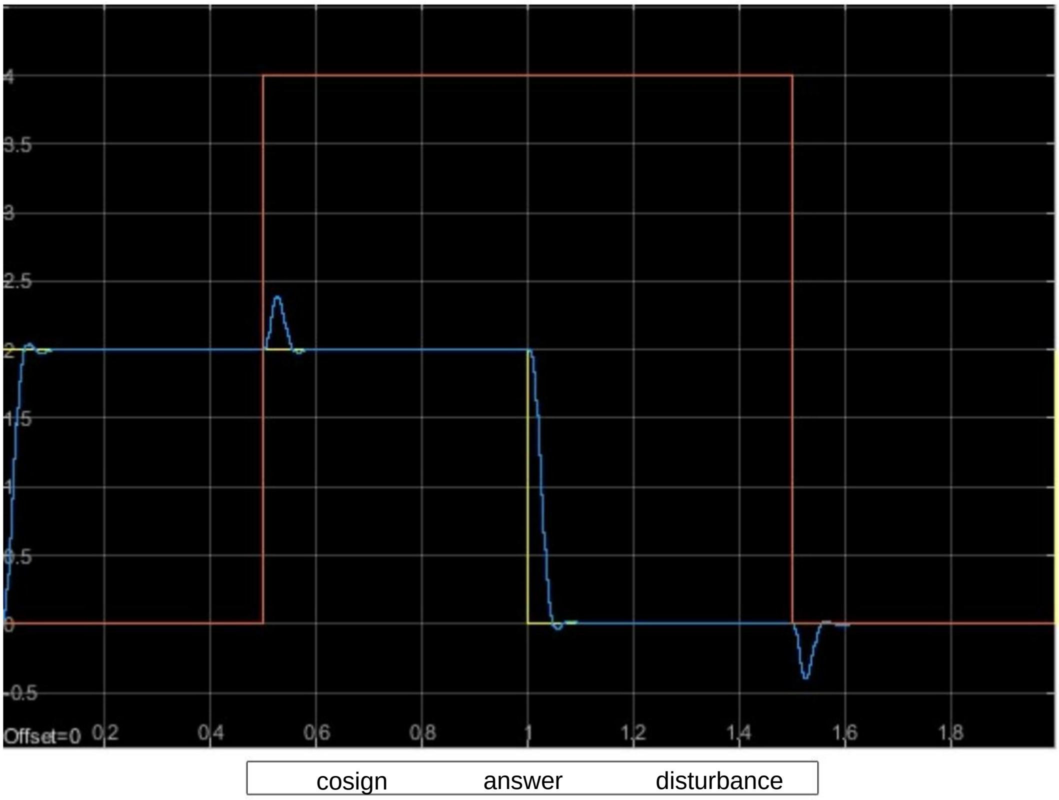
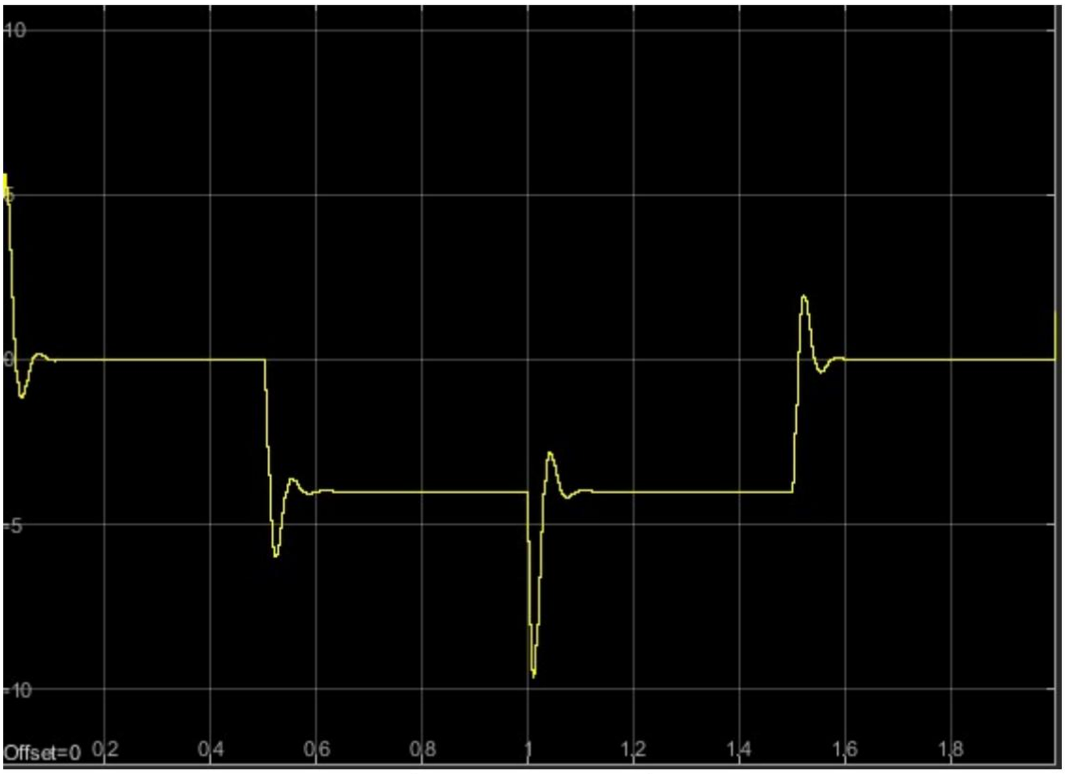


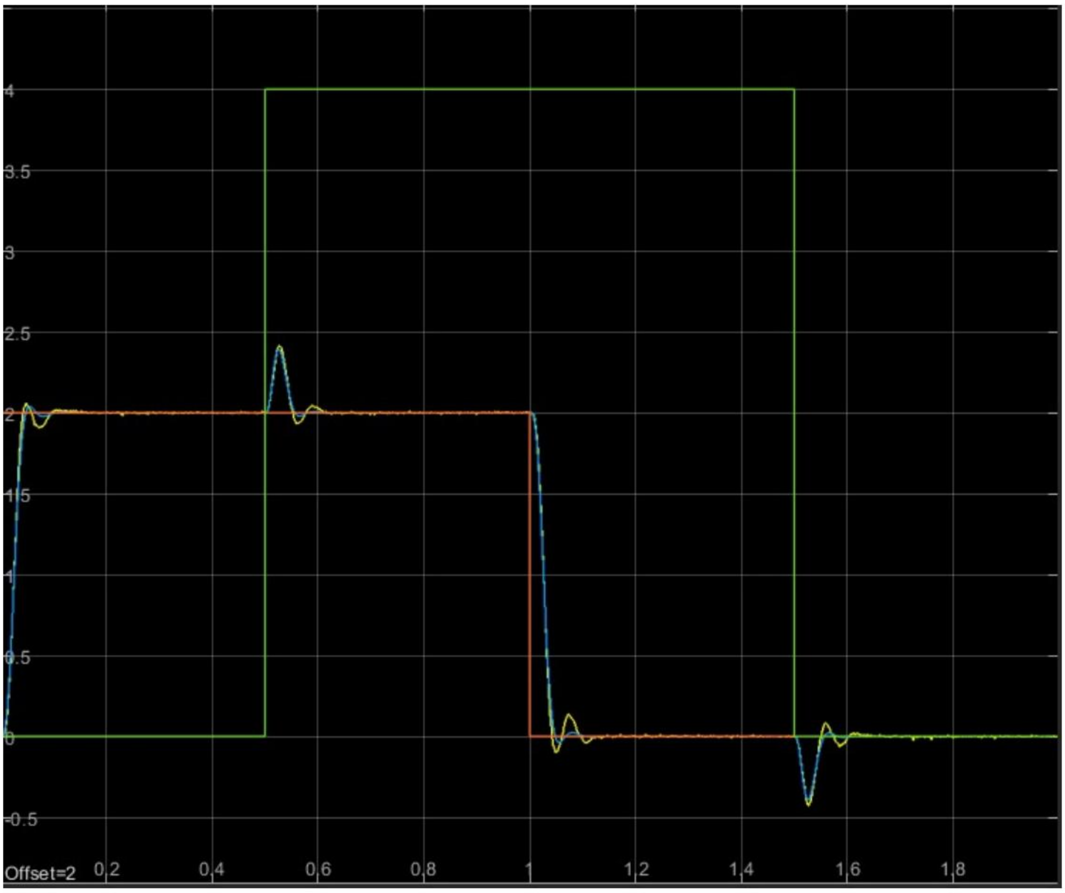
Figure 13: System response with LQ command.



disturbance

Figure 14: Command intensity u with LQ command.

Ultimately, as happened during the validation of the real model in figure 12, not only was a similar trajectory perceived between the real behavior and the simulation, but also a response very close to the disturbance and setpoint changes, as shown in Figure 15. Thus applying satisfactorily to reality.



real system	modeled system	cosign	disturbance

Figure 15: Validation by comparison of the system models the real system.

3 Conclusion

After executing the job, one can see how the steps and commands are intertwined. From the beginning, when an unwanted system response has been observed and a continuous corrector has been created to correct itÿ; followed by the equations of state of the numerical model; followed by the perception of a static error which was resolved, but resulted in a command overflow that would prevent its use in a real system; followed finally by the execution of a linear quadratic command which led to a satisfactory result, one can see how the steps are essential for an affirmative answer. Furthermore, it is found that all the comparisons between the real commands and the simulation commands are satisfactory, which shows how our modeled system applies in reality and concludes our objective.