

POLITECNICO DI MILANO

Master of Science in Automation and Control Engineering
School of Industrial and Information Engineering



AUTOMATION AND CONTROL LABORATORY – A.Y. 2023/2024

CNC MACHINE

Supervisor: Prof. La Bella
Group Name: PIZZA FESENJOON

Davide Jarik De Rosa
Email: davidejarik.derosa@mail.polimi.it
Student Number: 10933770

Amanda Piccaluga
Email: amanda.piccaluga@mail.polimi.it
Student Number: 10672448

Sina Garazhian
Email: sina.garazhian@mail.polimi.it
Student Number: 10903611

Romina Saljooghian
Email: romina.saljooghian@mail.polimi.it
Student Number: 10888850

Contents

1	Introduction	1
1.1	System Description	1
1.1.1	Auxiliary Equipment	1
1.2	Control Variables	2
2	System Kinematics	2
2.1	Forward Kinematics	2
2.2	Inverse Kinematics	3
3	Operation Area	4
4	Model Identification	6
4.1	Nominal Model Identification	6
4.2	Step Response Model Identification	8
4.2.1	Encoders Verification	8
4.2.2	Parameters Estimation - Encoder A	8
4.2.3	Parameters Estimation - Encoder B	9
4.2.4	Parameters Estimation - Limit Configurations	9
4.2.5	Dead Zone identification	10
4.3	Final Identified Model	10
4.4	Validation	11
4.4.1	Time Validation	11
4.4.2	Frequency Validation	11
4.5	System Analysis	14
4.5.1	SISO nominal model	14
4.5.2	SISO estimated model	14
4.5.3	MIMO estimated model	15
4.5.4	Sensitivity to Parameters	15
5	Motor Position Control	17
5.0.1	Specifications	17
5.1	Proportional Controller Design	17
5.2	PV Controller Design	18
5.3	Validation	19
5.3.1	Time Validation	19
5.3.2	Frequency Validation	20
5.4	End Effector Position Control	20
5.4.1	Analysis of Frictional Effects	20
5.4.2	Response to Disturbance and Noise	21
5.5	Trajectory Tracking	23
6	State Space Control	24
6.1	Derivative of The Output	24
6.1.1	Pole Placement	25
6.1.2	Linear Quadratic Control	25
6.2	Luenberger Observer	26
6.2.1	Pole Placement	26
6.2.2	Linear Quadratic Control	27
6.3	Kalman Filter	28
6.3.1	Pole Placement	28
6.3.2	Linear Quadratic Control	29
6.4	Comparison Between Observers	29
6.5	Validation	29
6.5.1	Time Validation	30
6.5.2	Frequency Validation	31
6.6	Response to Disturbance and Noise	32
6.7	Trajectory Tracking	33

7 Advanced Controller	34
7.1 Model Predictive Control	34
7.1.1 Recursive Feasibility	35
7.1.2 Stability	35
7.1.3 Convergence	35
7.2 MPC in Matlab 2019/b	36
7.2.1 MPC for Motor Position Control	36
7.2.2 MPC for Trajectory Tracking	37
7.2.3 MPC for Trajectory Tracking With Soft Constraints	38
A Appendix: Extra Material	39

1 Introduction

The goal of this project is to develop a model and design controllers of a coupled MIMO nonlinear system “CNC Machine”. In particular, the CNC Machine used in the laboratory sessions is a pantograph-type robot, as shown in Figure 1. It is composed of two rotary servos connected with a four-bar linkage. The control objectives for this process are the following:

- Motor Position Control,
- End Effector 2D Position Control,
- 2D Trajectory Tracking.

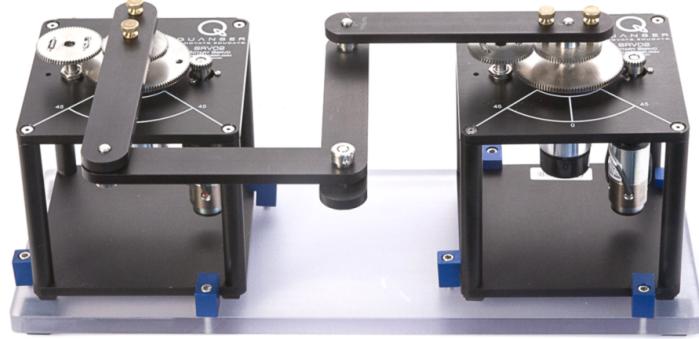


Figure 1: The CNC Machine

1.1 System Description

As already mentioned, the specific plant used in this project is a 2-DOF pantograph-type robot. This architecture is designed to manipulate the X-Y position of the End Effector, which is the tip of the robot links. The system is planar and consists of 2 actuated and 3 unactuated revolute joints. The robot is controlled by two servo motors mounted at a fixed distance, which control two arms coupled via two non-powered two-link arms. These arms are mounted on the Quanser SRV02 rotary servo plant, pictured in Figure 2, which consists of a DC motor that is encased in a solid aluminum frame and equipped with a planetary gearbox. The motor has its own internal gearbox that drives external gears. The SRV02 is equipped with an encoder, which is the only utilized sensor in the experiments.



1.1.1 Auxiliary Equipment

Other devices used in the lab sessions are the Q2-USB Data Acquisition Board, Figure 3a (the data acquisition solution for our application), that registers the signals of the encoder, and the VoltPAQ-X2 Amplifier, Figure 3b, a multi-channel, linear voltage-based power amplifier.



(a) The 2-USB Data Acquisition Board



(b) The VoltPAQ-X2 Amplifier

Figure 3: Supplementary devices that support system’s operations

1.2 Control Variables

The system is quite complex: every motor experiences both friction and an inertial effect. Additionally, the linkage between the angle of the motors and the End Effector is non-trivial. Despite this, we can visualize the basic, oversimplified plant scheme in Figure 4.



Figure 4: The basic schema of the system

Where:

- V_{dc} is voltages applied to both motors;
- u is the resulting torques acting on the chain;
- y is the coordinates of the End Effector;
- \hat{y} is the motors' relative angular position.

2 System Kinematics

In Figure 5 the CNC machine is schematized from a top-down perspective and it represents the home position of the robot, which means that both actuated joints are at $\theta_A = 0$ and $\theta_B = 0$. The system Kinematics analysis is necessary to:

- Calculate the global Cartesian coordinates of the robot End Effector from the joint angles - Forward kinematic;
- Determine the motor angles required to achieve a particular position of the End Effector - Inverse kinematic.

All the bars in the linkage share the same length L_b , the joints have all X and Y components and they are labeled with capital letters: A and B are actuated by the motors, while C and D are unactuated joints directly connected to the motors, and E denotes the End Effector of the system. The positive direction of rotatory motion is counter-clockwise with respect to the origin of the Cartesian plane (joint A position). Finally, the angles: θ_A is the angle between the AC bar and the X axis, θ_B is the angle between the BD bar and the Y axis; they represent the output shaft angles of the servos.

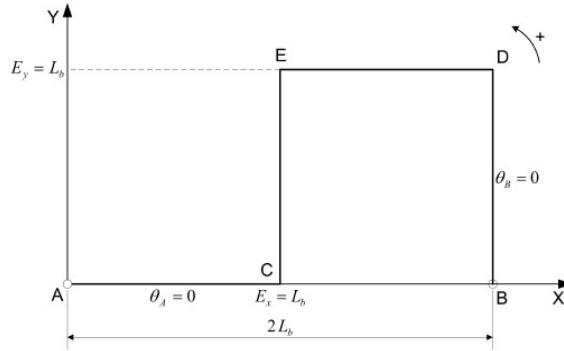


Figure 5: The default or starting position of the End Effector

2.1 Forward Kinematics

In Forward Kinematics analysis, as mentioned previously, mathematical equations are used to calculate the position of the End Effector. This is needed to derive its motion along the Cartesian axis. To write the equation, we designate p the line segment that connects C and D joints and β is the angle between p and the X axis, additionally α represents the angle between bar CE and p (see Figure 6).

We start by writing the position of the fixed joint A and B :

$$A_x = 0, \quad A_y = 0, \quad B_x = 2L_b, \quad B_y = 0.$$

And the relationships between C, D joints positions and θ_A, θ_B are:

$$C_x = L_b \cos \theta_A, \quad C_y = L_b \sin \theta_A, \quad D_x = 2L_b - L_b \sin \theta_B, \quad D_y = L_b \cos \theta_B.$$

We can proceed to calculations, we first derive a formulation for α :

$$p = \sqrt{(D_x - C_x)^2 + (D_y - C_y)^2} = 2L_b \cos \alpha \quad (1)$$

$$\alpha = \arccos \left(\frac{p}{2L_b} \right) = \arccos \left(\frac{\sqrt{(2 - \sin \theta_B - \cos \theta_A)^2 + (\cos \theta_B - \sin \theta_A)^2}}{2} \right) \quad (2)$$

Instead, for β :

$$\tan \beta = \frac{\sin \beta}{\cos \beta} = \frac{D_y - C_y}{D_x - C_x} \quad (3)$$

$$\beta = \arctan \left(\frac{\cos \theta_B - \sin \theta_A}{2 - \sin \theta_B - \cos \theta_A} \right) \quad (4)$$

At this point, we can write the formula of the End Effector position in Cartesian plan as:

$$E_x = C_x + L_b \cos(\alpha + \beta) = L_b \cos \theta_A + L_b \cos(\alpha + \beta) \quad (5)$$

$$E_y = C_y + L_b \sin(\alpha + \beta) = L_b \sin \theta_A + L_b \sin(\alpha + \beta) \quad (6)$$

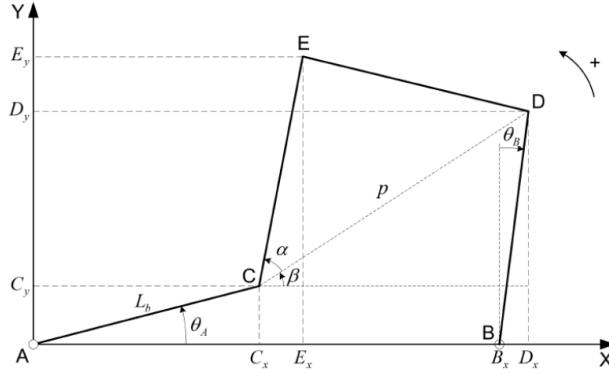


Figure 6: The forward kinematics analysis

2.2 Inverse Kinematics

In the Inverse Kinematics analysis, we start from the known quantities E_x and E_y . The goal is to obtain a formulation for the angles θ_A and θ_B that would enable us to achieve a particular position of the End Effector. We first have to define some new quantities. We name m the line segment that connects A and E joints and n the line segment that connects E and B joints. For the angles: σ_A is the angle between m and the X axis, σ_B is the angle between n and the Y axis, ϕ_A is the angle between the AC bar and m , ϕ_B is the angle between the BD bar and n , λ_A is the angle of the triangle ACE between the CE segment and the AC segment, λ_B is the angle of the triangle BDE between the ED segment and the DB segment (see Figure 7). The length of m and n is:

$$m = \sqrt{(A_x - E_x)^2 + (A_y - E_y)^2} = \sqrt{E_x^2 + E_y^2} \quad (7)$$

$$n = \sqrt{(B_x - E_x)^2 + (B_y - E_y)^2} = \sqrt{(2L_b - E_x)^2 + E_y^2} \quad (8)$$

We get an expression for λ_A :

$$\frac{m}{2} = L_b \sin \frac{\lambda_A}{2} \quad \& \quad m^2 = 2L_b^2(1 - \cos \lambda_A) \quad (9)$$

$$so : \quad \lambda_A = \arccos \left(\frac{2L_b^2 - m^2}{2L_b^2} \right)$$

and for λ_B :

$$\frac{n}{2} = L_b \sin \frac{\lambda_B}{2} \quad \& \quad n^2 = 2L_b^2(1 - \cos \lambda_B) \quad (10)$$

$$so : \quad \lambda_B = \arccos \left(\frac{2L_b^2 - n^2}{2L_b^2} \right)$$

Now we can explicit the new angles as:

$$\phi_A = \frac{\pi - \lambda_A}{2}, \quad \phi_B = \frac{\pi - \lambda_B}{2}, \quad \sigma_A = \arctan \left(\frac{E_y}{E_x} \right), \quad \sigma_B = \arctan \left(\frac{2L_b - E_x}{E_y} \right).$$

With these, we can compute the position of joint A in function of the End Effector coordinates:

$$\sigma_A = \theta_A + \phi_A, \quad \sigma_B = \theta_B + \phi_B \quad (11)$$

$$\text{and so: } \theta_A = \sigma_A - \phi_A = \arctan \left(\frac{E_y}{E_x} \right) - \phi_A \quad (12)$$

$$\text{and: } \theta_B = \sigma_B - \phi_B = \arctan \left(\frac{2L_b - E_x}{E_y} \right) - \phi_B \quad (13)$$

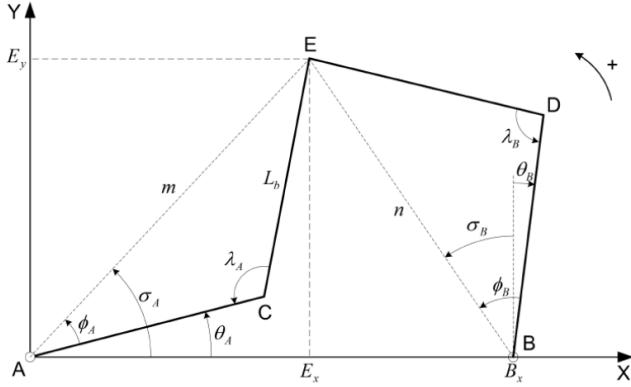


Figure 7: The inverse kinematics analysis

3 Operation Area

All the mathematical relations showed in the previous section must be considered for the specific angles configuration designed in the Figure 7. So we have to avoid, when the motors are working, the singularities and the consequential change of the robot configuration. In order to obtain this, we have defined a constrained working region, in which the End Effector can operate. First, we identified the possible singularities, which occur when the angle between two links surpasses 180° , causing the system to lose control over one degree of freedom.

$$\lambda_A = 180^\circ, \quad \lambda_B = 180^\circ \quad \text{and} \quad \alpha = 0^\circ$$

In order to take a logical safe margin from the singularities, we have defined the following constraints:

$$\lambda_A = \arccos \left(\frac{2L_b^2 - m^2}{2L_b^2} \right) \leq 150^\circ, \quad \lambda_B = \arccos \left(\frac{2L_b^2 - n^2}{2L_b^2} \right) \leq 150^\circ$$

That in function of the End Effector coordinates E_x and E_y (Equations 7 & 8) are:

$$150^\circ \geq \arccos \left(\frac{2L_b^2 - E_x^2 - E_y^2}{2L_b^2} \right), \quad 150^\circ \geq \arccos \left(\frac{2L_b^2 - (2L_b - E_x)^2 + E_y^2}{2L_b^2} \right)$$

The final constraints are:

$$CONSTRAINT \ 1: \quad E_x^2 + E_y^2 \leq (1 - \cos 150^\circ) 2L_b^2 = 3.73L_b^2 \quad (14)$$

$$CONSTRAINT \ 2: \quad E_x^2 + E_y^2 - 4L_b E_x \leq -0.27L_b^2 \quad (15)$$

For the constraint on the angle α , we select also in this case a logic margin to avoid the risk of encounter singularities, so we select: $\alpha \geq 20^\circ$, and we get (Equation 2):

$$\alpha = \arccos \left(\frac{\sqrt{(2 - \sin\theta_B - \cos\theta_A)^2 + (\cos\theta_B - \sin\theta_A)^2}}{2} \right) \geq 20^\circ$$

That in function of the motors angles is:

$$CONSTRAINT \quad 3 : \quad (2 - \sin\theta_B - \cos\theta_A)^2 + (\cos\theta_B - \sin\theta_A)^2 \geq (2\cos 20^\circ)^2 = 3.53 \quad (16)$$

Constraint 1, 2 and 3 are the fundamental constraints for our working region, but we decide to add also other two constraints, on the angles λ_A and λ_B , that impose a minimum amplitude, in order to avoid that the bars of the robot collapse onto each other:

$$\lambda_A \geq 10^\circ, \quad \lambda_B \geq 10^\circ$$

Again using equations 7 and 8 we get:

$$CONSTRAINT \quad 4 : \quad E_x^2 + E_y^2 \geq 0.03L_b^2 \quad (17)$$

$$CONSTRAINT \quad 5 : \quad E_x^2 + E_y^2 - 4L_b E_x \leq -3.96L_b^2 \quad (18)$$

We use these five constraints in order to define a workspace that guarantees the correct functioning of the CNC robot. Figure 8 illustrates the largest operational area (indicated by blue dots), considering the robot nonlinear behavior and non-ideal factors such as static friction. The End Effector workspace is shown on the left, while the corresponding joint space, which depicts the angular positions of servos A and B, is shown on the right.

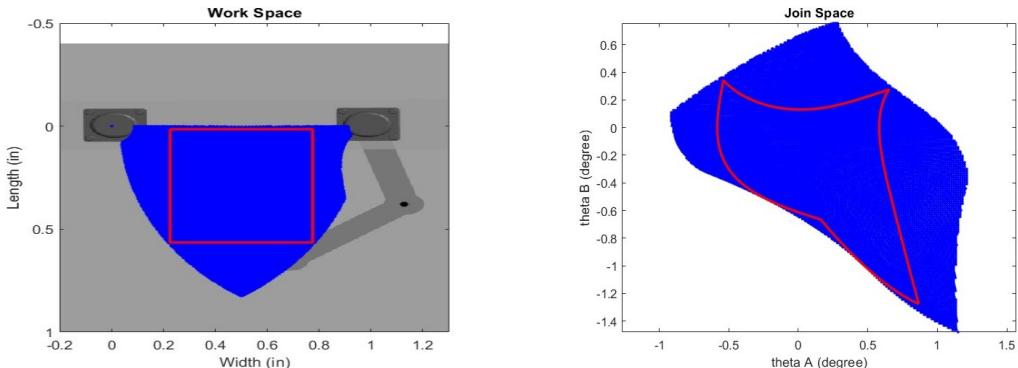
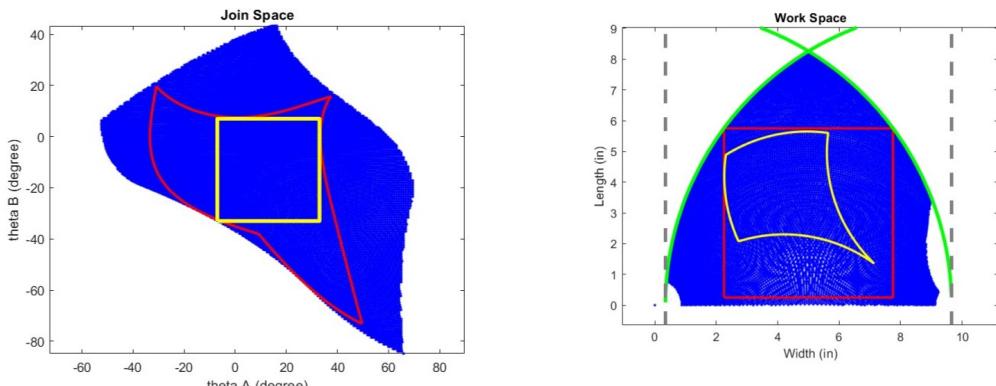


Figure 8: Mapping a square space from workspace to joint space, outlined by a red boundary.

Our goal is to not only identify singularities but also to pinpoint the specific area in which servos are decoupled, which will ultimately aid in the design of our controller. This involves analyzing a square in joint space to verify the decoupling. The concept of the decoupled working space relies on the assumption that all system deficiencies are either negligible or adequately resolved, thereby leading to a linear relationship between input and output parameters.



(a) Joint area of angles represent in degrees

(b) Work area represented in inches. The green line illustrates the theoretical bounding of area find by Constraints

Figure 9: Decoupled area of workspace specified by yellow line in both joinspace & workspace.

Figure 9 demonstrates the decoupled area of the workspace. The decoupled regions are identified by observing the movement restrictions imposed when deviating from the home position. This observation reveals that Servo A exclusively induces motion in the X direction, while Servo B induces motion in the Y direction. This separation of influence confirms the decoupling of the actuated joints, ensuring that each servo motion does not interfere with the other one.

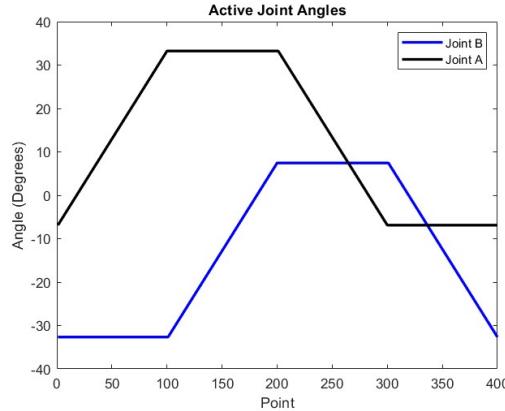


Figure 10: Variation of Active Joints θ_A & θ_B which relate in order to servo A and servo B

The figure 10 highlights the decoupling effect, demonstrating that changes in Joint A angle do not influence Joint B angle, and vice-versa. Each joint independently follows its trajectory, establishing the exclusive motion induction in their respective X and Y directions. This decoupling is crucial for accurate and efficient control of the 2-DOF pantograph-type robot.

4 Model Identification

In this section, we model our plant. We first get an analytical expression for the transfer function between voltage and angular position of the single rotary servo plant. Then, by means of experiments, we estimate the parameters. We analyze the single plant with the parameters we found. Finally, we get a representation of the full linearized MIMO system.

4.1 Nominal Model Identification

For the parameters value and their variation, please refer to the SRV02 User Manual and the CNC User Manual. At first we study the electrical model, denoting as $\omega_m(t)$ the motor angular speed. Applying the Kirchhoff's Voltage Law on the motor circuit, where the back-emf voltage is $k_m \omega_m(t)$ and the inductance can be neglected (we assume that the current through the armature is in phase with the voltage, so the armature behaves like a resistor), we get:

$$V_m(t) = R_m I_m(t) + k_m \omega_m(t)$$

$$I_m(t) = \frac{V_m(t) - k_m \omega_m(t)}{R_m}$$

For the mechanical model, the equation of motion for the motor shaft is

$$J_m \dot{\omega}_m + B_m \omega_m = \tau_m - \tau_{lm}$$

Exploiting the equation $\tau_{lm} K_g \eta_g = \tau_l$, we can rewrite

$$J_m \dot{\omega}_m + B_m \omega_m = \tau_m - \frac{\tau_l}{K_g \eta_g}.$$

Starting from $\theta_m = K_g \theta_l$, we can obtain the following relationships

$$\omega_m = K_g \omega_l \quad \dot{\omega}_m = K_g \dot{\omega}_l$$

that combined with $\tau_l = J_l \dot{\omega}_l + B_l \omega_l$ allows to get the expression

$$(J_m K_g^2 \eta_g + J_l) \dot{\omega}_l + (B_m K_g^2 \eta_g + B_l) \omega_l = \tau_m K_g \eta_g.$$

Imposing $J_{eq} = J_m K_g^2 \eta_g + J_l$ and $B_{eq} = B_m K_g^2 \eta_g + B_l$, follows the final expression

$$J_{eq} \dot{\omega}_l + B_{eq} \omega_l = \tau_m K_g \eta_g.$$

We can now combined the electrical model with the mechanical model. The motor torque is related to the voltage applied to the motor, and so to the motor current, by means of the relationship

$$\tau_m = k_t \eta_m I_m(t).$$

So the previous formula can be rewritten as

$$\begin{aligned} J_{eq} \dot{\omega}_l + B_{eq} \omega_l &= K_g \eta_g k_t \eta_m \frac{V_m - k_m K_g \omega_l}{R_m} \\ J_{eq} \dot{\omega}_l + \left(B_{eq} + \frac{K_g^2 \eta_g k_t \eta_m k_m}{R_m} \right) \omega_l &= K_g \eta_g k_t \eta_m \frac{V_m}{R_m}. \end{aligned}$$

By naming

$$B_{tot} = B_{eq} + \frac{K_g^2 \eta_g k_t \eta_m k_m}{R_m}, \quad A_m = \frac{K_g \eta_g k_t \eta_m}{R_m}$$

the final expression comes out:

$$J_{eq} \dot{\omega}_l + B_{tot} \omega_l = A_m V_m.$$

Now it is possible to consider the full structure of the SRV02 with the Laplace formulation. Assuming the motor is initially steady (no initial angular speed), we can write

$$s J_{eq} \Omega_l(s) + B_{tot} \Omega_l(s) = A_m V_m(s)$$

from which we obtain the expressions of the transfer function between the motor input voltage and the load shaft angular speed

$$\frac{\Omega_l(s)}{V_m(s)} = \frac{\frac{A_m}{B_{tot}}}{\frac{J_{eq}}{B_{tot}} s + 1} = \frac{K}{\tau s + 1}$$

and between the motor input voltage and the load shaft angular position

$$\frac{\Theta_l(s)}{V_m(s)} = \frac{\frac{A_m}{B_{tot}}}{s \left(\frac{J_{eq}}{B_{tot}} s + 1 \right)} = \frac{K}{s(\tau s + 1)}$$

where

$$A_m = \frac{K_g \eta_g k_t \eta_m}{R_m} = \frac{70 \cdot 0.9 \cdot 7.68 \cdot 10^{-3} \cdot 0.69}{2.6} = 0.1284$$

$$B_{tot} = B_{eq} + \frac{K_g^2 \eta_g k_t \eta_m k_m}{R_m} = 0.015 + \frac{70^2 \cdot 0.9 \cdot 7.68 \cdot 10^{-3} \cdot 0.69 \cdot 7.68 \cdot 10^{-3}}{2.6} = 0.084$$

$$J_{eq} = 3.59 \cdot 10^{-3}$$

So we can set,

$$K = \frac{A_m}{B_{tot}} = 1.53, \quad \& \quad \tau = \frac{J_{eq}}{B_{tot}} = 0.043$$

Of course the nominal model doesn't include the delay and several quantities used for the computations present variations with respect to their nominal value. For instance, if we re-compute K applying the maximum positive variation for each quantity we get

$$K = \frac{70 \cdot 0.9 \cdot 1.1 \cdot 7.68 \cdot 1.12 \cdot 10^{-3} \cdot 0.69 \cdot 1.05}{0.015 \cdot 2.6 \cdot 1.12 + 70^2 \cdot 0.9 \cdot 1.1 \cdot 7.68 \cdot 1.12 \cdot 10^{-3} \cdot 0.69 \cdot 1.05 \cdot 7.68 \cdot 1.12 \cdot 10^{-3}} = \frac{0.432}{0.304} = 1.42$$

so the model can experience a variation due to uncertainties. Still the nominal model is extremely important, because it allows the nominal system analysis and the sensitivity to parameters analysis.

4.2 Step Response Model Identification

We proceed to perform the classical experimental model identification in time domain by means of the step response of the single rotary servo motor system. By construction, we know that the system has 2 poles (one in the origin) and has relative degree of 2. The nominal τ is not that small (roots are real and not repeated), the system surely presents a delay, so we assume that the single servo motor transfer function has the following structure

$$\frac{\Theta_l(s)}{V_m(s)} = G(s) = e^{-ds} \frac{K}{s(\tau s + 1)}$$

from which we have to identify the quantities d, K, τ . After the parameters estimations, we will deal with the non-linearity of the full MIMO system.

4.2.1 Encoders Verification

As mentioned before, the SRV02 encoders are the only available sensors in the full plant. It is indeed needed to verify the correct functioning of both of the encoders. The methods of verification are the following:

- Moving the actuated joints by hand to 45° and evaluating the data;
- Giving voltage as input and evaluating the correspondence between the manually measured angle and the data.

The results of the experiments ensured a correct functioning of both of the encoders.

4.2.2 Parameters Estimation - Encoder A

To analyze correctly this system, we should consider the transfer function between voltage applied and angular speed

$$G(s) = e^{-ds} \frac{K}{\tau s + 1}$$

that can be easily obtained by means of an approximated derivative based on the last 10 samples of the response. In practise, we give as an input to the system a voltage square wave of amplitude 1 that would result in the arm moving forward and backward. Starting from the encoder signals, we compute the approximated derivative from which we estimate the parameters. We now deal with a first order strictly proper transfer function of an AS system, so we can perform the well known estimation of the 3 characterizing parameters:

- The steady state gain K is the value reached after 4.6τ times the step amplitude;
- The time constant τ is the time that takes for the system to reach about 63% of its steady state value[4];
- The time delay d is the time interval from the application of a control signal to any observable change in the process variable[5], in our case measured on the position, not on the approximated velocity.

To perform the most correct and precise estimation, we would need to disassemble the links that connect the 2 servo motors. This is not allowed to us, so we carry out the experiments on the servo motor A, while the other is still at initial position - $\theta_B = 0$. In Table 1 we see the estimation of the measured parameters, resulted from the scope analysis of several experiments, like in Figure 11. Notice that for each experiment the measured parameters are already averaged several times, since we measured them in several points.

Nr. Experiment	SW Amplitude	Measured K	Measured τ	Measured d
1	1 V	1.534	0.046	0.01
2	1 V	1.718	0.043	0.01
3	0.6 V	1.533	0.044	0.01
4	0.6 V	1.406	0.042	0.01
5	0.6 V	1.483	0.042	0.01
6	0.6 V	1.534	0.044	0.01
7	1.5 V	1.533	0.046	0.01

Table 1: Encoder A Resulting Parameters

It is relevant to point out that τ is averaged with respect to time, since we must average the speed to avoid noise. Considering an average of the experiments, we could set

$$K_A = 1.534 \quad \tau_A = 0.0439 \quad d_A = 0.01$$

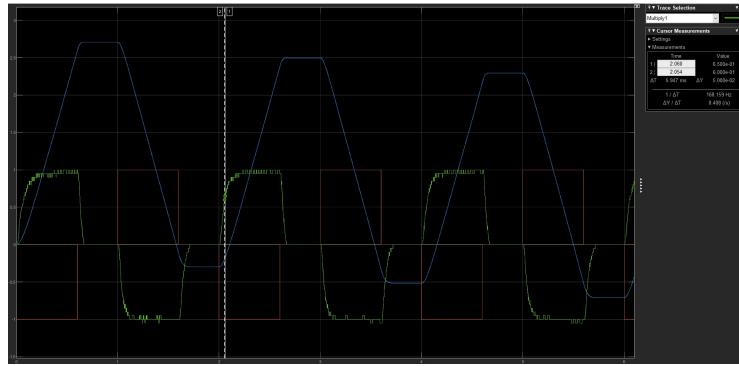


Figure 11: In blue, the scaled motor position, in green the scaled motor speed, in red the voltage input

4.2.3 Parameters Estimation - Encoder B

The same parameters estimation procedure must be performed for the other encoder. The input is again a square wave of different amplitudes, the visualization is again the one in Figure 11 and the averaging method is the same. The other encoder is still at initial position - $\theta_A = 0$. The results are shown in Table 2

Nr. Experiment	SW Amplitude	Measured K	Measured τ	Measured d
1	1 V	1.533	0.046	0.01
2	1 V	1.533	0.044	0.01
3	0.6 V	1.534	0.043	0.01
4	0.6 V	1.534	0.041	0.01
5	1.5 V	1.482	0.046	0.01
6	1.5 V	1.597	0.042	0.01

Table 2: Encoder B Resulting Parameters

Again, if we consider an average of the experiments, we can set

$$K_B = 1.535 \quad \tau_B = 0.0437 \quad d_B = 0.01$$

4.2.4 Parameters Estimation - Limit Configurations

We estimated the parameters of the transfer function between voltage and angular position of the single rotary servo plant. For both SRV02 systems we got similar results. Now we have to deal experimentally with the non linearity of the full MIMO system, in order to understand how the parameters may vary in different configuration and what is the limit in which the 2 SRV02 can be considered decoupled. In particular, we analyzed

- What happens to the step response of the first SRV02 when the second SRV02 is still and close to a singularity;
- What happens to the step response of the second SRV02 when the first SRV02 is still and close to a singularity;
- What happens when the 2 SRV02 are working together, in a step response configuration.

For the first 2 cases, we performed several experiments, all wrapped up in Table 3

Initial θ_A	Initial θ_B	Working motor	SW Amplitude	Measured K	Measured τ	Measured d
-45°	0	A	0.6 V	1.533	0.037	0.01
45°	0	A	0.6 V	1.533	0.039	0.01
0	-45°	B	0.6 V	1.533	0.036	0.01
0	45°	B	0.6 V	1.533	0.034	0.01
-30°	30°	A	0.4 V	1.534	0.036	0.01
30°	-30°	B	0.4 V	1.534	0.039	0.01

Table 3: Limit Configurations Resulting Parameters

As we can expect from the physics behind the plant, neither K nor d change when we approach singularities, τ instead seems to decrease (not significantly). This is obvious if we recall the definitions of these 2

parameters: K mainly depends on the motor resistance, the current-torque constant, the high-gear ratio, the gearbox and the motor efficiency (parameters that one shouldn't expect to be affected by the initial angles position), τ instead depends directly on the moment of inertia of the 4-bar linkage, that is more likely to be affected by the bars' position.

For what concerns the last case scenario, we performed the experiments only in the configuration $\theta_A = 0$ and $\theta_B = 0$, since we cannot move both the motors close to the singularities due to the dead zone. We give two impulses to the motors at the same time and we measure the quantities for both of them. The results are summed up in Table 4. Each experiment is done twice in order to record each motor correctly.

SW A Amplitude	SW B Amplitude	K_A	τ_A	d_A	K_B	τ_B	d_B
0.6 V	0.6 V	1.541	0.0432	0.01	1.543	0.0433	0.01
0.6 V	1.2 V	1.538	0.0431	0.01	1.533	0.0439	0.01
1.2 V	0.6 V	1.543	0.0429	0.01	1.541	0.0437	0.01

Table 4: Both Motors Working Resulting Parameters

Since our system is a coupled system and motors are somehow connected to each other through joints and bars, we are checking whether both motors impact each other or not. Note that if the output of one motor is affected by another, we should consider the other motor position as a disturbance to another motor (or another motor voltage as another input). In fact we should add an extension for our modeling part. We luckily do not see any considerable difference in this scenario. As we saw in the table above the difference of τ and K when both motors are working is under 5%. Moreover, we will show that our system and our control strategies are not too sensitive to parameters values, which means this low amount of change can be neglected. Therefore, we can consider the motors as decoupled.

4.2.5 Dead Zone identification

In the previous experiments, we noticed that there is a voltage dead-zone in both servo-motors, a sort of static friction phenomena. The results of these experiments is not explicitly reported, we just need to consider, for all the future control strategies, the presence of a dead zone for all the input voltages in the interval $[-0.25, 0.25]$ when the system is steady. Note that this is the second explicit non-linearity in our system, after the delay.

4.3 Final Identified Model

Taken into account the results of the previous experiments, we can consider the following assumptions for our model:

- The 2 motors can be considered to have the same model, which is the following one

$$\frac{\Theta_l(s)}{V_m(s)} = G(s) = e^{-0.01s} \frac{1.53}{s(0.0439s + 1)}$$

where the gain and the delay are the same for both motors, the time constant is instead the one of motor A ;

- The motors can be considered as decoupled, since we will mainly work with one motor at the time (even if we will work with 2 motors at the same time, we can neglect the very small parameters variation);
- There is a dead zone in the interval $[-0.25, 0.25]$ that is not explicitly written;
- The model is linearized around the equilibrium, that is the working region we identified previously, and we will neither exit or run close to the boundaries, so the model can be considered valid for every point in our sub-domain of interest;
- The small delay can be neglected, we will see later in System Analysis that it has a very small impact on the model, especially for our control objectives.

Therefore, the Transfer Function between the voltage and the motor position of both SRV02 servomotors is the following

$$G(s) = \frac{1.53}{s(0.0439s + 1)}$$

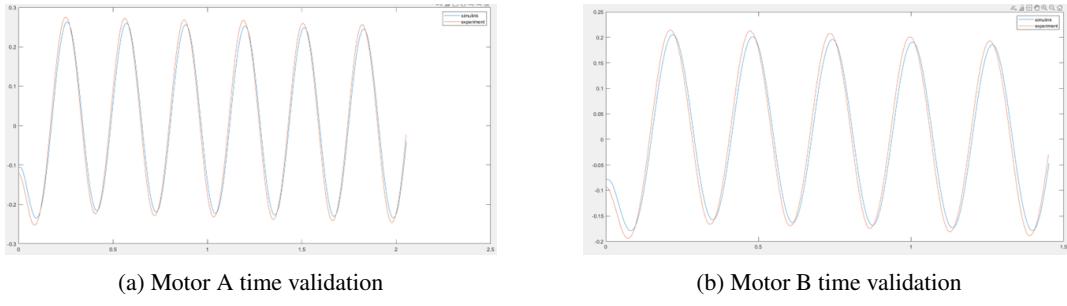
If we want to consider instead the full MIMO system, decoupled and linearized around the equilibrium, we get the following

$$H(s) = \begin{bmatrix} G(s) & 0 \\ 0 & G(s) \end{bmatrix}$$

4.4 Validation

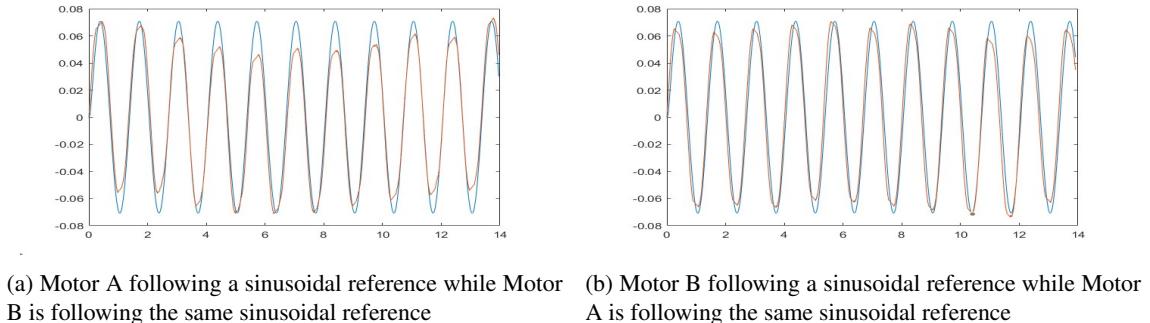
4.4.1 Time Validation

For time validation, we used for both motors a sine wave function as input.



From Figure 12a and 12b it is immediate to see that both motors models are perfectly consistent with real datas. It is important to specify that the time signals have been detrended before been shown in figures. The importance of detrending is discuss later in section 4.4.2. The model can be therefore considered successfully time validated.

Now we validate over time the configuration in which both motors are working at the same time, in order to validate empirically the hypothesis of decoupled motors.



In Figure 13a and 13b we can appreciate both motors sine wave tracking in the following configuration: at the same time, starting from home position, the 2 motors receive as a reference a sine wave of frequency $2\pi(0.75) \text{ rad/s}$ and amplitude 0.35. We can see that the error in amplitude is not very strong and apart from that the tracking is very consistent (τ variation is actually irrelevant). The only relevant parameter variation in the configuration of 2 motors working is the amplitude - less than 10%. The assumption of two motors decoupled may be considered validated empirically.

4.4.2 Frequency Validation

For model validation in the frequency domain, we use different sinusoidal waves, with more emphasis on the frequencies near the cutoff, and validate the Bode diagram (the phase and the gain) at those frequencies. However, one important point is that frequency validation requires special preprocessing, which, if ignored, could lead to unsatisfactory results.

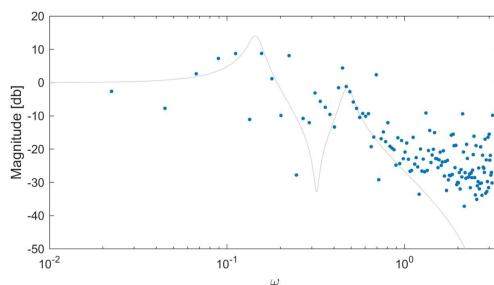


Figure 14: An example of another problem showing the results when ignoring using hanning (part of preprocess)

Our preprocessing includes detrending and windowing (or cutting data to become periodic). We will explain each of these steps.

- **Detrending:** Due to an asymmetric dead zone in the motors, a symmetric sinusoidal wave will not result in a symmetric output. In other words, after each period of symmetric sinusoidal input, we expect to return to the zero position, but we find ourselves in a positive angular position. This trend introduces other frequencies into the Fourier transform of the output, which contradicts our assumption of a Linear Time-Invariant (LTI) system (an LTI system should output a single frequency if a single frequency is input). The example below illustrates detrending.

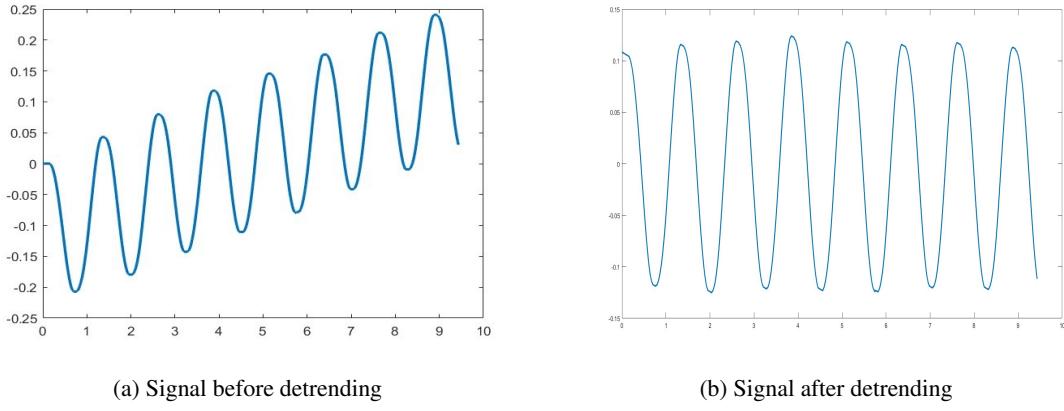


Figure 15: The process of removing long-term trends from a signal

- **Windowing:** The other problem for frequency validation arises from discretization in the frequency domain and the FFT method, which pads zeros to the signal. The Figure 14 shows an example of the problem without preprocessing (without using Hanning). To solve this, one option is to apply a Hanning window. The **Hanning window** helps ensure there is no jump at the start and end of the signal. Normally, if the signal length is not a power of 2, the algorithm pads zeros, because FFT is implemented by divide and conquer, needing a power of 2 to split into equal parts until reaching a length of one. Adding zeros at the end causes a jump if the signal's value is not zero, resulting in many unwanted elements in the frequency domain.

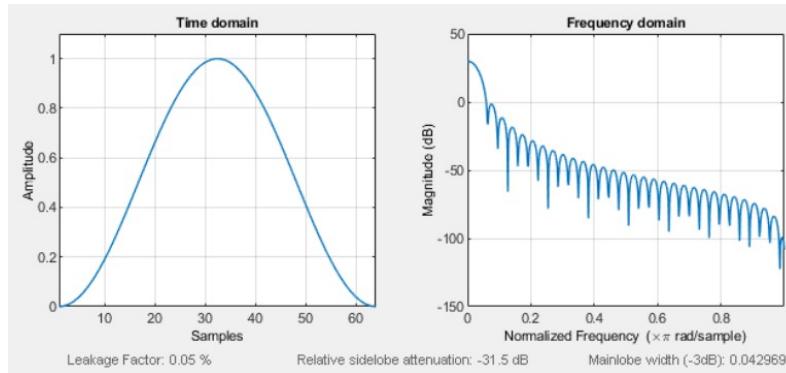


Figure 16: A normal Hanning window is multiplied to the data for better FFT results.

If the signal length is long enough and contains several periods, the Hanning window is a good option. However, in our case, the signal length was insufficient, resulting in more unwanted changes rather than removing jumps. Therefore, we used another method: **bilateral cutting**. We start the signal at the index where its value is close to zero and end it similarly. This way, we avoid any jumps in the time domain. Additionally, we consider the slope at the start and end to ensure the signal contains an integer number of periods ($N*T$). This can only be applied with the assumption of having a sinusoid signal.

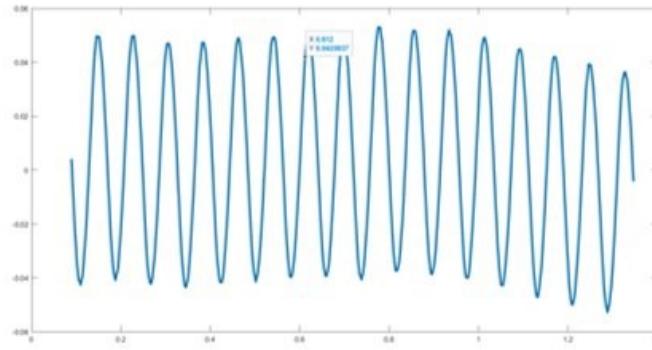


Figure 17: Preprocessed Signal: The signal is detrended, starts and ends near zero, and contains an integer number of periods.

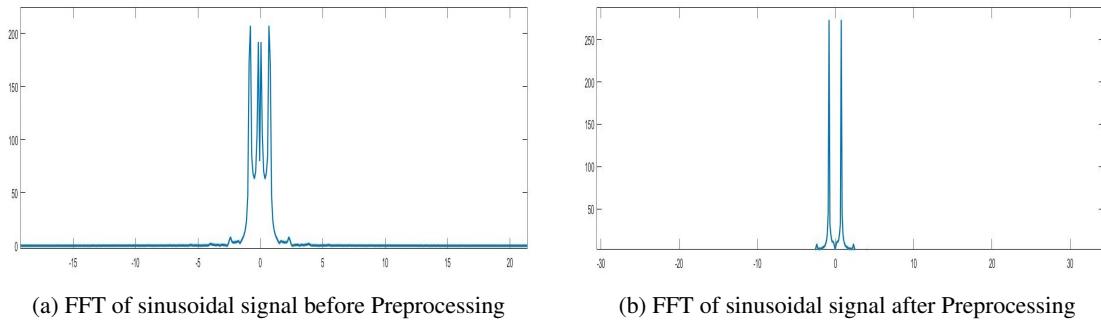


Figure 18: Illustrate Difference Between FFT of Preprocessed Signal and Non-Preprocessed Signal

Now, we get FFT from each preprocessed signal calculated the gain phase and mapped it on the Bode diagram. The result is show in figure 19.

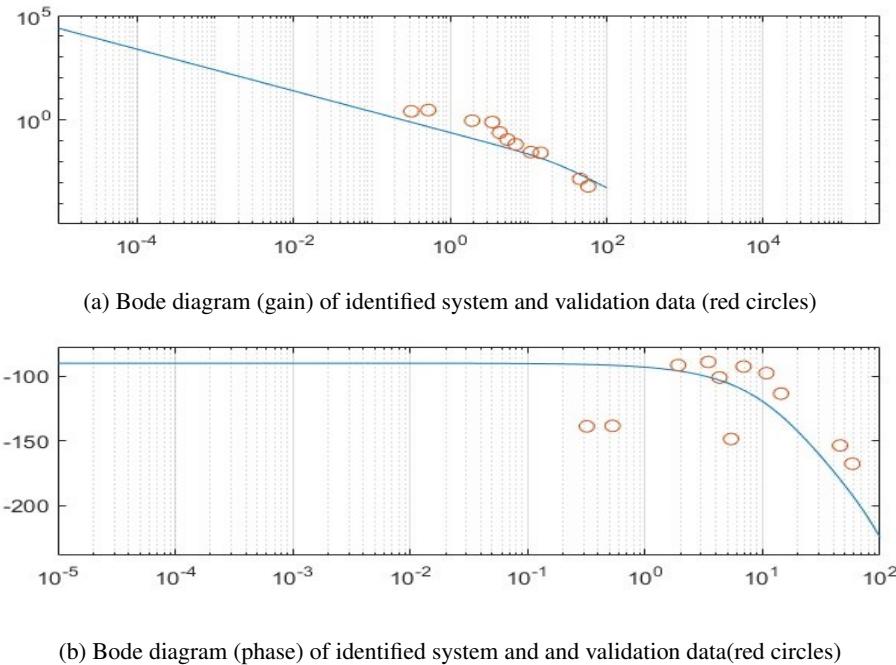


Figure 19: Frequency validation for Motor A. For the result of the other motor, refer to Appendix.

In conclusion, we can consider the system estimated in section 4.3 fully validated in time and frequency. All the assumptions we took (decoupled motors, negligible delay and linear behaviour far from singularities) may be considered valid.

4.5 System Analysis

Now that we got the model of the SRV02, obtained by means of experiments, and we have also validated it, we can proceed to system analysis. We first tackle the analysis of the SISO nominal model (single servomotor), then we do it for the estimated model by means of approximations. Finally we comment the main properties of the overall MIMO system (the full SRV02 motor structure).

4.5.1 SISO nominal model

We remind the expression for the nominal model

$$G(s) = \frac{1.53}{s(0.043s + 1)}$$

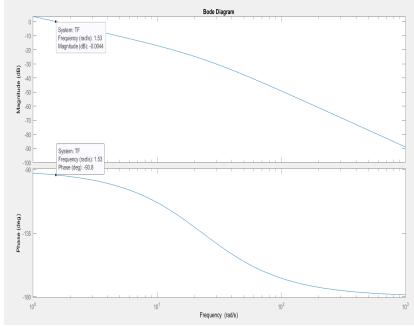


Figure 20: Bode Diagram of the nominal system

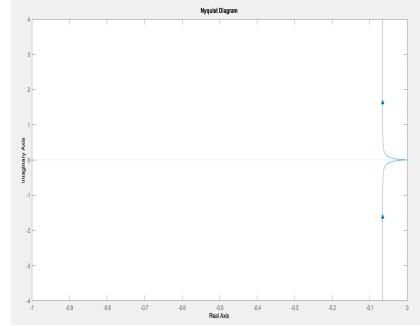


Figure 21: Nyquist Diagram of the nominal system

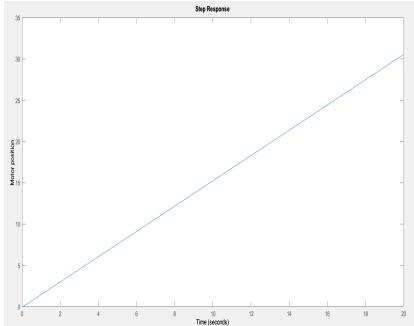


Figure 22: Step Response of the position TF

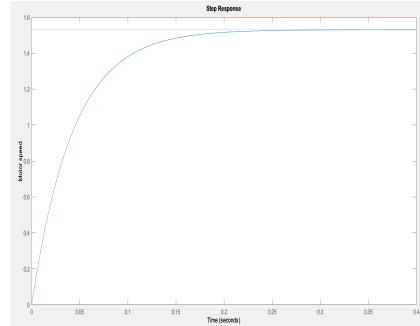


Figure 23: Step Response of the speed TF

By means of the frequency analysis from Figure 20 and 21, we get that

- The system has 2 poles: 1 in the origin and the other in -23.256;
- The relative degree is 2;
- The gain margin is infinite;
- The phase margin is 86.24° ;
- The cutoff frequency is 1.53 rad/s;

By means of the step response in Figure 22 and 23, both of the position and the velocity of the motor, we get that

- The settling time of the step response of the velocity at 1% is 0.2s;
- The rising time of the step response of the velocity at 80% is 0.069s;
- The overshoot of the step response of the velocity is 0%;

4.5.2 SISO estimated model

The expression for the estimated model

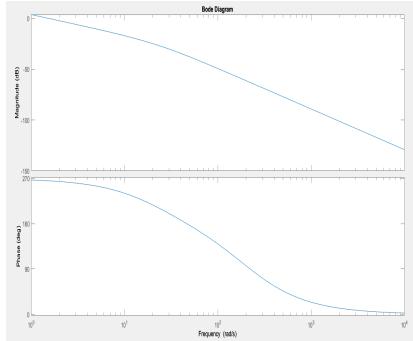
$$G(s) = e^{-0.01s} \frac{1.53}{s(0.0439s + 1)}$$

In this case, we can easily see that the model is pretty identical to the nominal one if we neglect the delay. In fact the only difference is that the pole would move slightly to the right, causing a decrease in the phase margin (the new pole is now at -22.77 , the phase margin has decreased to 86.166° , the other values are unchanged).

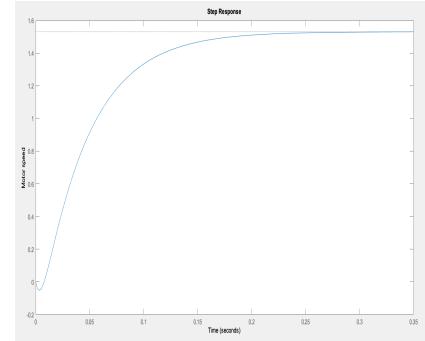
It is much more meaningful to analyze what happens if we consider instead the delay, by means of the Padé Approximation technique [3]. The Padé Approximation is a method used to approximate the behavior of a system by representing its transfer function as a ratio of polynomials. This approximation technique is particularly useful when dealing with systems with delays.

In our case, using a 1st order Padé Approximation - more than enough for our scenario - we get the following transfer function

$$G_{pade}(s) = \frac{-1.53s + 306}{0.0439s^3 + 9.78s^2 + 200s}$$



(a) Bode Diagram of the Padé approximation of the system



(b) Speed step response of the Padé approximation of the system

By means of the frequency analysis from Figure 24a, we get that:

- The phase margin in the approximated model is 85.292° ;
- The gain margin in the approximated model is 68.88 ;
- The cutoff frequency is still at 1.53 rad/s in the approximated model.

By means of the step response analysis from Figure 24b, we get that:

- The settling time of the step response of the velocity at 1% is $0.21s$;
- The rising time of the step response of the velocity at 80% is $0.079s$;
- The overshoot of the step response of the velocity is still 0%;

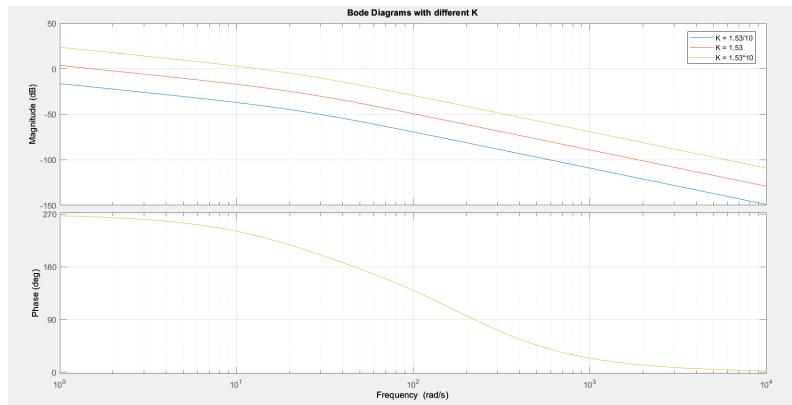
It is clear that the delay influences the transfer function at high frequencies: the cutoff frequency remains the same, the gain and phase margin decrease with the delay. For the time domain, the setting and rising time are the same shifted by the delay amount. We utilized a first order Padé approximation, if we try a second or a third order approximation the results would be extremely similar.

4.5.3 MIMO estimated model

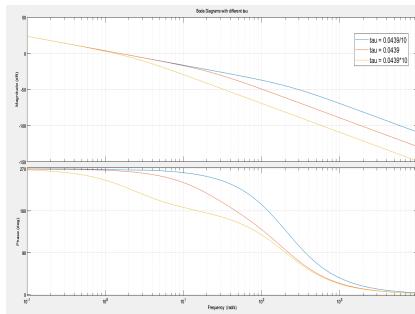
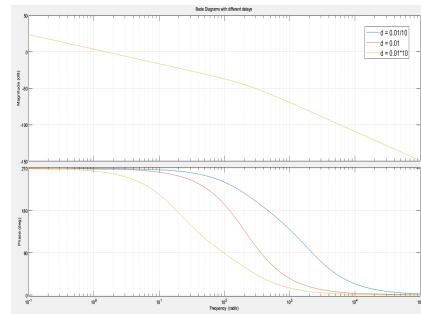
In our case, we assumed that the two motors are always decoupled, so it is meaningless to perform a MIMO analysis of the overall 2 servomotors system: $G_{11} = G_{22}$ and $G_{12} = G_{21} = 0$, so the poles will be the ones of the SISO system (of second order), there will not be any invariant zeros and the principal gains will be equal (the condition number is equal to 1 for every frequency).

4.5.4 Sensitivity to Parameters

Both in the nominal and the estimated model, it exists a certain grade of uncertainty about the 3 main parameters K , τ and d . Therefore, it is reasonable to analyze how the frequency behaviour of the voltage-to-position transfer function is affected by the 3 parameters. We use the 1st order Padé Approximation for our goal.

Figure 25: The Bode Diagram at different K

As we can appreciate from Figure 25, decreasing K implies decreasing both the gain and the phase margin. For the cutoff frequency instead, a higher K means a higher cutoff frequency. A good news is that the phase margin remains around 60% even when we increase a lot the gain, so we can make the system much faster with control.

(a) The Bode Diagram at different τ (b) The Bode Diagram at different d

Observing Figure 26a, increasing τ implies decreasing both the gain and the phase margin. The cutoff frequency instead remains practically the same: τ affects only the high frequencies. From Figure 26b, we see that increasing d implies decreasing significantly the gain margin and slightly the phase margin. Concerning the cutoff frequency, it is totally untouched: d affects only the high frequencies.

We now use a numerical method in order to analyze the TF sensitivity to parameters: we try every possible combination in which the three parameters (K , d , τ) are experiencing a numerical variation, according to the elements of the following test vector:

$$\text{testvector} = [0.95, 0.9, 0.8, 0.75, 1.1, 1.2, 1.25]$$

In the algorithm, we set

$$K_{\text{test}} = K_{\text{original}} * \text{testvector}(\text{element})$$

$$\tau_{\text{test}} = \tau_{\text{original}} * \text{testvector}(\text{element})$$

$$d_{\text{test}} = d_{\text{original}} * \text{testvector}(\text{element})$$

At every iteration we evaluate the gain margin and the phase margin, uploading the minimum values in case

$$\text{currentphasemargin} < \text{minphasemargin}$$

$$\text{currentgainmargin} < \text{mingainmargin}$$

We obtain the worst case scenario (minimum values for gain/phase margin in all possible combinations). The result is the following:

- The smallest phase margin is 82.67° ;
- The smallest gain margin is 44.08 .

It is therefore obvious that even considering terrible estimation of parameters, the system remains very robust.

5 Motor Position Control

In this section, we will discuss the process of implementing a PID-based control system for adjusting the angular position of each robot joint. Initially, we considered a PID controller for the system. However, due to the integral action in the voltage position of the servo motor, we ultimately decided to use a P or PD-type controller. In fact, adding another integrator in the closed-loop system can cause problems with stability, where it could decrease the phase margin.

Let's start by discussing some control theory background. We will be focusing on the standard second-order transfer function. The response characteristics are determined by the values of ω_n and ζ . Typically, the damping ratio (ζ) influences the shape of the response, while the natural frequency (ω_n) affects the speed of the response. The closed-loop transfer function can be expressed as:

$$\frac{\Theta_l(s)}{\Theta_d(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (19)$$

Also, the step response is characterized by two key properties: **overshoot** and **peak time**

$$PO = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}, \quad t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \quad (20)$$

By specifying these parameters, we can calculate the transfer function and subsequently develop a controller.

5.0.1 Specifications

The time-domain specifications chosen for controlling the load shaft position in each servo system are:

$$t_p = 0.3, \quad PO = 0.2\%, \quad |e_{ss}| \leq 0.5deg$$

Through the adjustment of the parameter t_p , we enhanced the system's speed, leading to optimized performance and stability. Additionally, adjusting the proportional overshoot results in a lightly damped response and in a reduced settling time.

When aiming to enhance system speed, there is often a trade-off between achieving faster response and significant control signal. In such scenarios, it becomes imperative to strike a balance between system speed and input voltage. To address this trade-off, parameters are selected with careful consideration, as illustrated in Appendix.

5.1 Proportional Controller Design

Proportional control is highly esteemed for its simplicity, effectiveness, and versatility. Given these features, along with the favorable system gain margin observed in our analysis, we decided to first implement the proportional controller. The control law is given by:

$$V(s) = K_p (\Theta_d(s) - \Theta_l(s)) \quad (21)$$

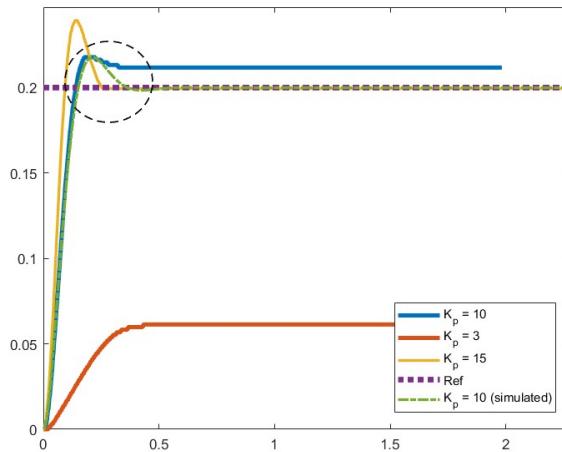


Figure 27: Closed loop system step response to Ref. of 0.2 rad of Proportional Controller

The proportional controller provides a simple way to correct the position error but does not inherently address steady-state errors. The experimental results reveal the following:

- With $K_p = 15$, the real system shows considerable overshoot before settling close to the reference.
- There is a steady-state error for $K_p = 3$ and $K_p = 10$ in the experiments, while the simulation response for $K_p = 10$ aligns with the reference without such error.

We initially expected no steady-state errors in the system by tuning a proper K_p , due to the internal integral action of the system, as shown by the simulated model response for $K_p = 10$. The observed variation may be attributed either to actuator delays or more likely to system nonlinearities (a dead zone for low values of voltage). To address this issue and improve performance, we added first an integral action (wrong solution) and then a derivative term (final solution).

5.2 PV Controller Design

In cascade control, the system utilizes two feedback loops: an inner loop and an outer loop. The inner loop is responsible for controlling the speed, while the outer loop is responsible for controlling the position. This arrangement helps to improve the overall performance and stability of the control system.

The PV (Proportional-Velocity) controller enhances the transient response by incorporating a derivative term in addition to the proportional term. The inclusion of the derivative term helps to predict and counteract the rate of change of the error, thereby improving the response time of the system and reducing the overshoot.

The controller structure is given by:

$$V_m(s) = K_p(\Theta_d(s) - \Theta_l(s)) - K_v s \Theta_l(s) \quad (22)$$

Kindly keep in mind the following details: K_p represents the proportional control gain, K_v represents the velocity control gain, $\Theta_d(s)$ represents the setpoint or reference load shaft angle, $\Theta_l(s)$ represents the measured load shaft angle, and $V_m(s)$ represents the motor input voltage (the control signal). The block diagram of the PV control can be found in Figure 28.

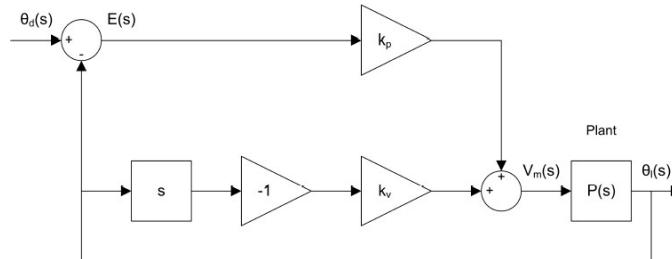


Figure 28: Block diagram of PV controller.

Refer to Appendix for the implementation of the derivative with an high pass filter and simulink schema.

The overall open-loop transfer function is:

$$L(s) = \frac{P(s)K_p}{1 + P(s)K_v s} \quad (23)$$

The overall closed-loop transfer function is

$$\frac{\Theta_l(s)}{\Theta_d(s)} = \frac{KK_p}{\tau s^2 + (1 + KK_v)s + KK_p} \quad (24)$$

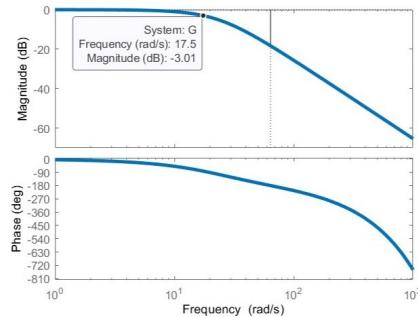
Note that in the nominal open-loop transfer function a critical cancellation occurs, but this does not happen in practice, because in the real controller do not use a pure derivative. Upon revisiting the standard second-order transfer function in equation 19 and equating it to the aforementioned expression, we derive the expressions to obtain K_p and K_v . With the given design specifications $PO = 0.2$ and $t_p = 0.3$, it follows that:

$$K_v = 0.535(V/rad), \quad K_p = 15.5(V/rad)$$

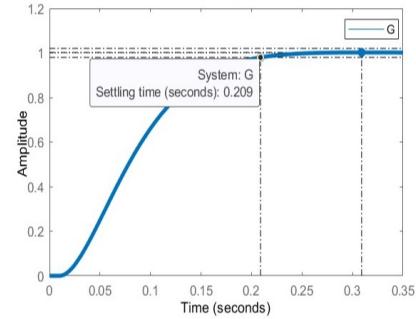
With this controller, the system has:

- 2 stable poles located at $-20.7 \pm j10.5$. These poles indicate a stable and well-damped response;
- Achieves a settling time at 1% of approximately 0.2 seconds, ensuring rapid response;

- Practically null overshoot;
- Gain Margin (G_m): $+\infty$, indicating a robust system;
- Phase Margin (P_m): 73.2, providing good phase stability.

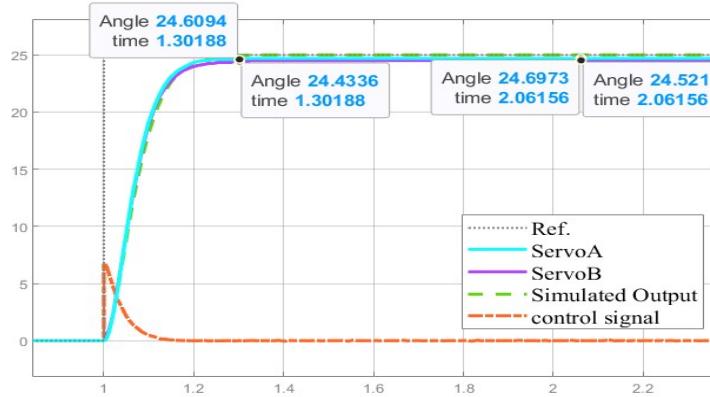


(a) Bode Diagram of the Closed Loop TF



(b) Step Response of Closed Loop system

Lab Experiment

Figure 30: System step response to Ref. of 25°

The system's performance in reaction to a reference input of 0.4369 rad (25°) is illustrated in Figure 30. The step response analysis reveals the following:

- **System Stability:** The placement of the poles in the left half of the s-plane confirms the stability of the closed-loop system.
- **Peak Response:** Both ServoA and ServoB reach peak values of approximately 24.6° and 24.43° , respectively, around 1.3 seconds after the reference input is applied. This slight overshoot is within acceptable limits.
- **Steady-State Convergence:** ServoA stabilizes at 24.69° and ServoB at 24.52° , which are very close to the reference input of 25° , demonstrating good tracking performance and minimal steady-state error.
- **Steady-State Error:** The error is less than 0.5° , indicating precise control and minimal deviation from the desired angle.
- **Control Signal:** The control effort, shown by the orange line, indicates the input provided to the system. The initial control signal is significant but remains below the 10V threshold, ensuring the actuator operates within safe limits.

5.3 Validation

5.3.1 Time Validation

For this particular FB controller, in Figure 30 it is shown the step response to a 0.4369 rad reference over time for both the real and the simulated system. As it is evident by the graph, the responses are almost identical with the exception of a small delay in the real system, which has been discussed previously. Therefore, the controller can be considered successfully time validated.

5.3.2 Frequency Validation

We decided to validate in frequency domain only for one encoder. The reason is clear: since the motor models are the same and they are both successfully validated in time and frequency there is practically no need to validate them both again in control. Therefore, with the same pre-processing steps previously discussed in section 4.4.2, we validate in frequency PV control for Motor B.

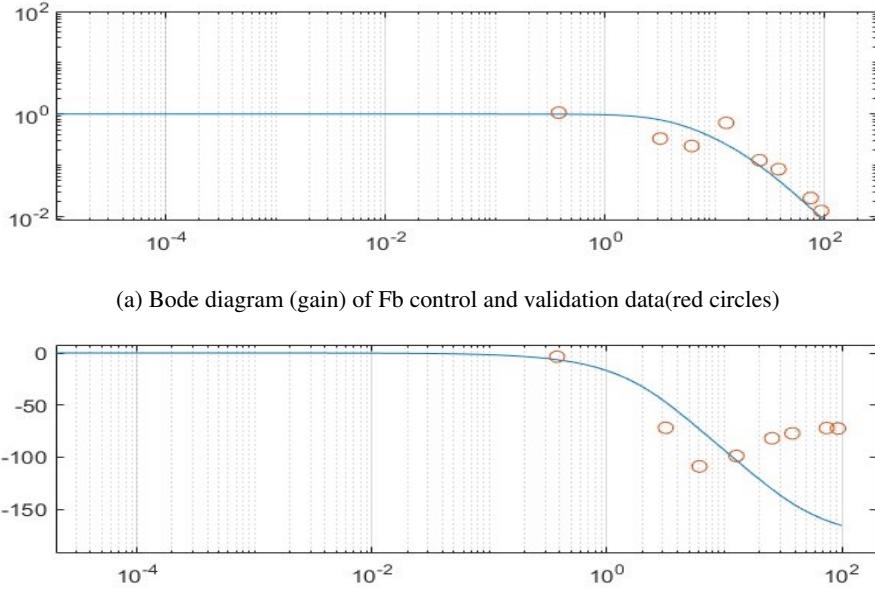


Figure 31: Frequency validation for Motor B

Figure 31 show very satisfying results. The control system can be considered successfully frequency validated.

5.4 End Effector Position Control

In the absence of a sensor on the robot End Effector, we rely on inverse kinematic equations to calculate and control the position of the End Effector. We have tracked 3 specific points inside the workspace identified in Figure 9: (5.5, 2), (3, 3) and (2, 4), with the X and Y values of each point denoted in inches. In this section, we illustrate the coordinates for points (3, 3) and (2, 4). As illustrated in Figure 32a and Figure 32b, both angles exhibit a rapid convergence to the desired values with minimal overshoot, highlighting effective and consistent control dynamics. Figure for the remaining point tracking look in Appendix.

5.4.1 Analysis of Frictional Effects

During the analysis of point (2, 4), it was observed that additional force was required to initiate motion, likely due to static friction within the servo motor B. This effect was more pronounced with smaller step inputs, suggesting that minor adjustments could lead to increased steady-state error.

- **Static Friction:** When the control signal is below 0.25V (the dead zone), static friction can impede the initial rotation of the servo motor.
- **Overshoot and Stability:** To counteract static friction, it is advisable to allow a controlled amount of overshoot to ensure movement. However, excessive overshoot must be avoided to prevent instability and undesired oscillations.

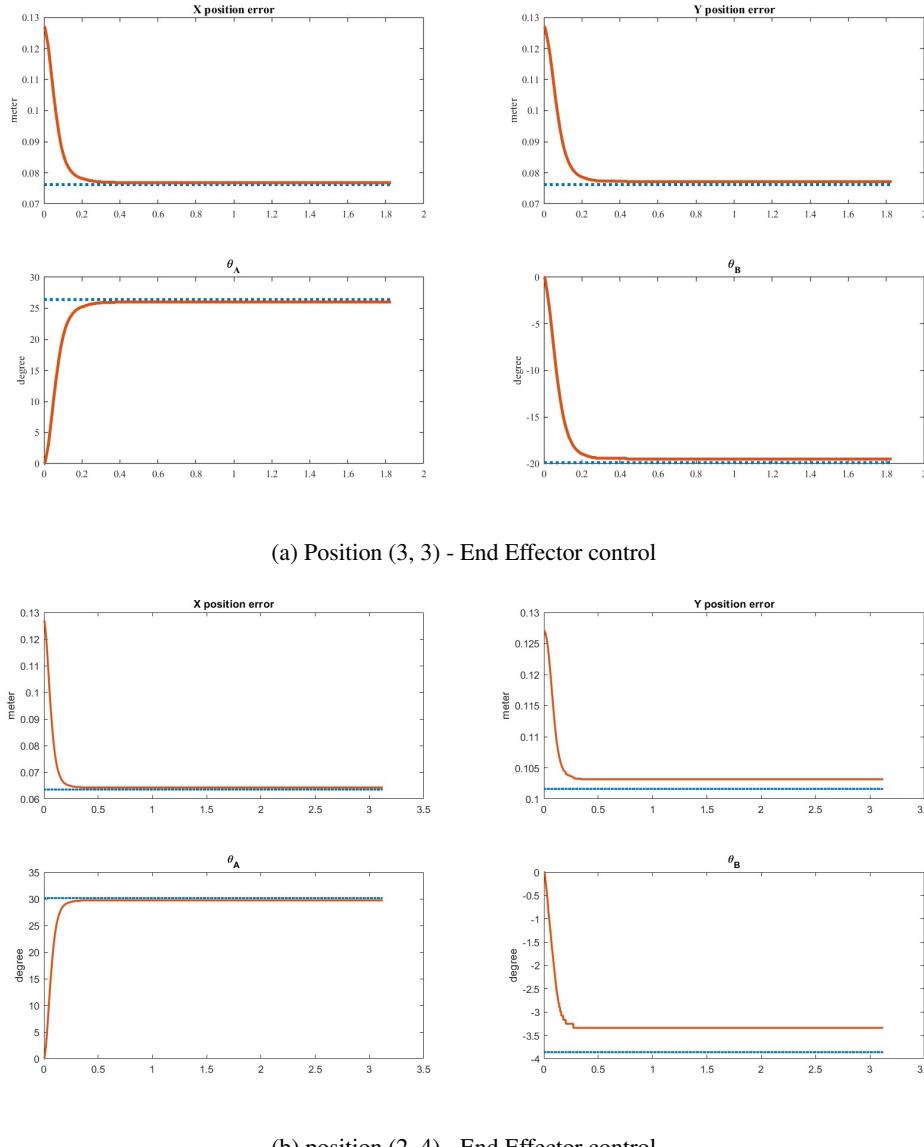
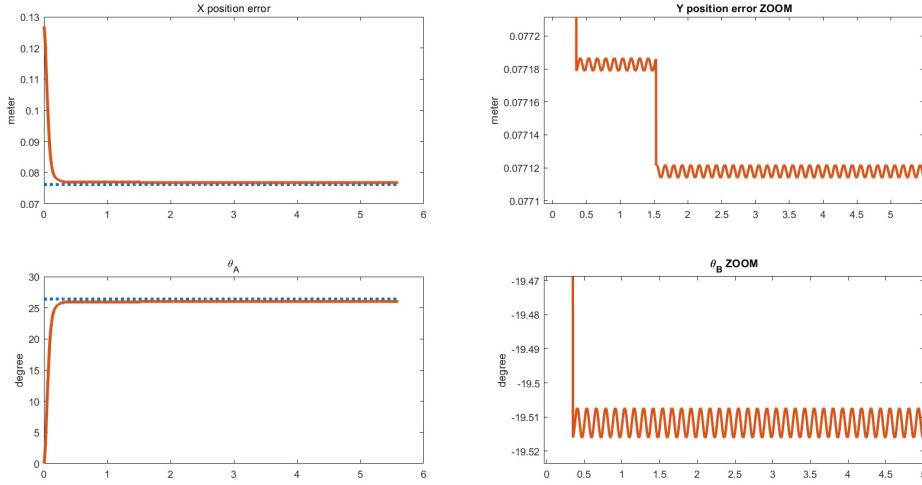


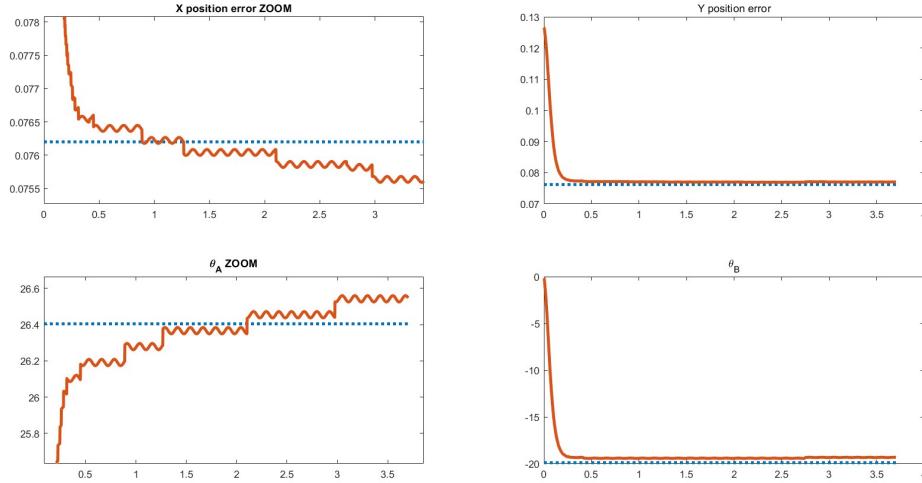
Figure 32: The robot movement from the home position to the designated destinations is illustrated, showcasing the system ability to accurately reach specified points within the workspace. The angle θ_B in End Effector positioning in (2, 4) has a bigger convergence error because it is very small.

5.4.2 Response to Disturbance and Noise

The inclusion of a derivative term in a control system has several effects, such as enhancing system response speed, improving disturbance rejection, and increasing damping. However, it also introduces amplification of high-frequency noise within the system. To address this issue, we can explore the use of a low-pass filter to reduce high-frequency noise or fine-tuning the derivative time constant to balance noise reduction and system performance. Reducing the derivative time constant may diminish sensitivity to noise but could compromise the effectiveness of the derivative action. By configuring the derivative time constant to 50rad/s, we effectively mitigate the amplification of high-frequency noise. This adjustment helps maintaining control accuracy while reducing the impact of noise on the system.



(a) Amplitude of disturbance and noise: 0.005 - End Effector control



(b) Amplitude of disturbance and noise: 0.02 - End Effector control

Figure 33: The response of the system to a sinusoidal disturbance of 0.5rad/s and to a noise at 50rad/s

In Figure 33a and Figure 33b, it is shown the system response to low frequency disturbance (at 0.5 rad/s) and to high frequency noises (at 50 rad/s). Both of signals are simulated with sinusoids with the same amplitude in Simulink. We have to be sure that the results are consistent with the transfer functions. We remind that the disturbances are attenuated by the sensitivity transfer function:

$$D(s) = \frac{1}{1+L(s)} = \frac{s^2 + 41.42s}{s^2 + 41.42s + 540.2}$$

and the noises are attenuated by the complementary sensitivity function:

$$F(s) = \frac{L(s)}{1+L(s)} = \frac{23.71}{0.0439s^2 + 1.819s + 23.71}$$

By observing the transfer functions above, we expect a reduction to 4% for the disturbances and to 18% for the noises. The results obtained in high frequency are even better than the expected ones: in fact the reduction for disturbances is 5% in the first experiment and 7.5% in the second one, instead the reduction for noises is approximately 1% for both the experiments. This comes from the fact that respect to the nominal transfer functions we implemented the derivative in a more efficient way, tuning better all the parameters. Of course even better disturbance rejection are achievable (by means for example of a Loop Shaping designed controller), but it was not our main focus to build a controller super robust against low and high frequency disturbances.

In summary:

- While the derivative term enhances disturbance rejection and system stability, it is crucial to balance its benefits against the potential noise amplification. Implementing an additional filter or adjusting the derivative time constant are effective strategies for managing **derivative term and noise trade-off**.
- The overall performance of the control system depends on effectively managing both the **desired response speed and the noise** introduced by the derivative term. **Fine-tuning** these parameters ensures optimal system stability and accuracy.

5.5 Trajectory Tracking

In this extra section, a square will be the reference trajectory for the End Effector position. Referring to the workspace in Figure 9b, choose the dimensions of the square inside the decoupled area.

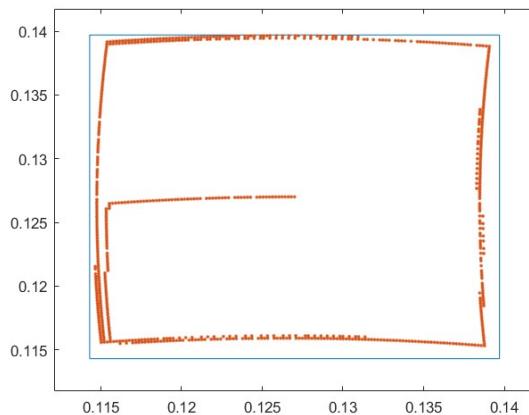


Figure 34: Illustration of the system's trajectory tracking in a two-dimensional Cartesian plane, using a reference square waveform (The dimensions of the square are 2.5 cm) - Trajectory Tracking

The system follows correctly and with small error its reference trajectory, demonstrating its capacity to adapt to changes in direction or velocity without experiencing significant overshoot. It is conceivable for the system to temporarily exceed the desired position before initiating self-correction.

Fine-tuning the frequency of the reference square wave enhances precision and reduces variations, as depicted in Figure 35.

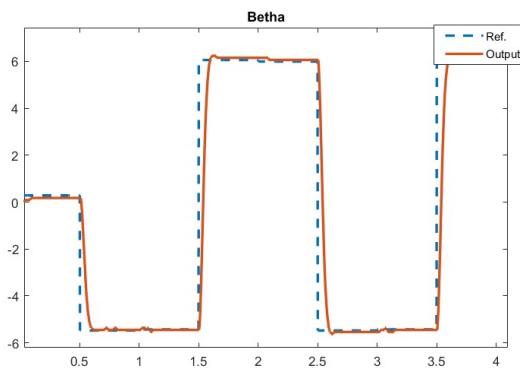


Figure 35: Visualization of the system tracking θ_B , the angle for Servo B, within the trajectory, highlighting its improved responsiveness in a two-dimensional Cartesian plane.

6 State Space Control

In this section we analyze the control strategies implemented with the addition of three different type of observers, used to estimate the internal state of the system:

- Estimation of the non-measurable state (velocity) as the derivative of the output (position)
- Luenberger Observer
- Kalman Filter

These observers play a critical role in State-Space control by facilitating the implementation of state feedback control, even when some states are not directly measurable. The implementation of the controller is based on the following State Space controller techniques:

- Pole Placement
- LQ (Linear Quadratic) control

The state-space representation of the system is chosen as:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

where the four matrices are:

$$\begin{array}{lll} A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix} & B = \begin{bmatrix} 0 \\ \frac{K}{\tau} \end{bmatrix} \\ C = [1 \ 0] & D = 0 \end{array}$$

With this particular SS representation, the first state x_1 corresponds to the motor position, x_2 is the motor speed and the output y is the first state. We also decide to include an integrator in the controller, in order to reach a steady state with zero error. To do this we need to enlarge the system: we check that the system is controllable and observable (indeed there is no derivative action in the model) and we add a new state to the system (so now the order of the system is $n = 3$).

Let's explore the implementation of these observers with different controller methods to position the End Effector accurately at the target point (3, 3) (inches) within the designated working area.

6.1 Derivative of The Output

The estimation of the velocity by the derivative of the output is a straightforward method, but it results extremely sensitive to measurement noise, leading to significant inaccuracies in state estimation. To reduce the differentiation error we introduce an high pass filter instead of a pure derivative, similarly to what we did for the PV controller, obtaining a more robust estimation against noises. In the Figure 36 we can see that the estimated state is characterized by oscillations and this highlights the limitations of this approach in our experiments.

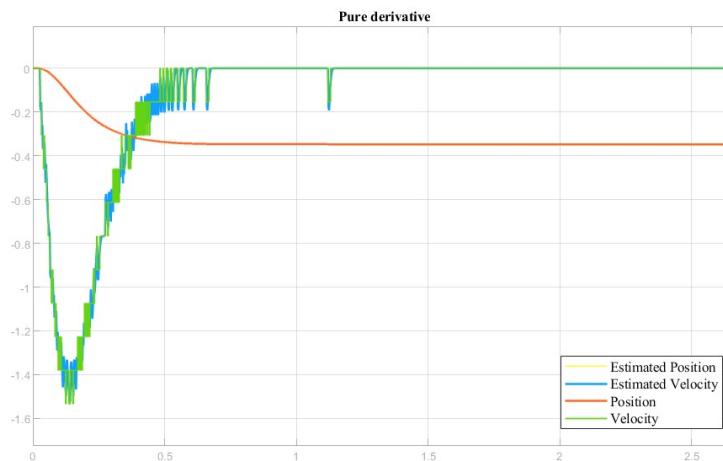


Figure 36: Performance of the State Derivative Observer. The observer attempts to estimate the state trajectories using direct differentiation of the output signal.

6.1.1 Pole Placement

We take in consideration the limitations of the performance of this very simple type of observer in the design of our controller with the Pole Placement method: in fact we defined the desired poles in such a way to obtain a good settling time but also to limit the overshoot: our first choice for the dominant pole was $s_1 = -17$, in order to guarantee stability and have a settling time of about 0,3 seconds, but the practical application of this choice showed that to be that fast we need an high overshoot to reach lower steady state error. So we set $s_1 = -13$, obtaining a safer response and a slightly higher settling time. This configuration is anyways fast enough for the hypothetical applications of our robot. We set the other two poles at an higher frequency respect to the dominant one: $s_2 = -15$ $s_3 = -17$. By means of the MATLAB function 'acker', we have obtained the following gain values:

$$K_x = [19.2522 \quad 0.6374] \quad K_v = -95.1167$$

If you compare K_x with the corresponding gains of the PV Controller, they exhibit almost similar values. This similarity indicates that the state feedback design effectively mimics the PV control behavior.

Although we incorporated an integrator in the PV controller to address steady-state error, it resulted in oscillations. The state-space design, however, provides a more robust framework to manage such trade-offs and achieve desired performance without inducing instability.

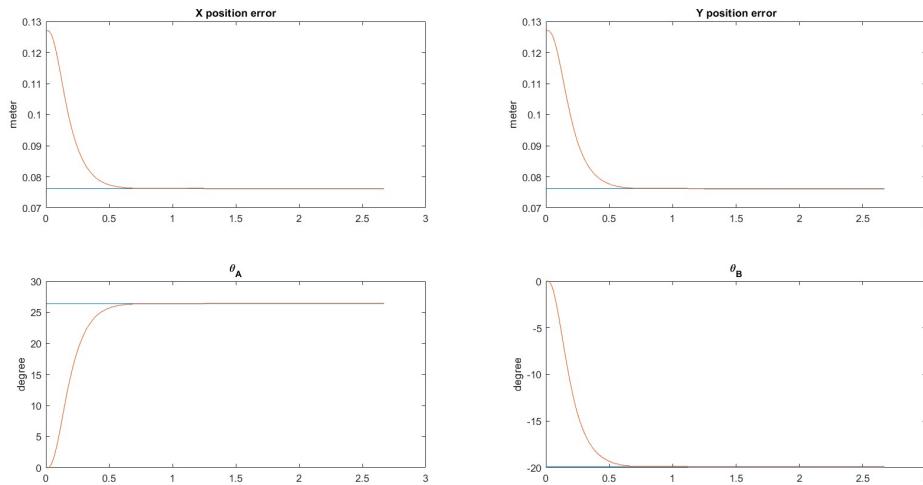


Figure 37: **State Derivative Observer with Pole Placement controller** for End Effector positioning, coordinates (3, 3). It provides moderate response and it is suitable for applications where slower settling times and small steady error are acceptable.

6.1.2 Linear Quadratic Control

The Linear Quadratic (LQ) controller is designed to minimize a cost function, which typically balances the control effort and the deviation from the desired state. For the LQ controller design, the weighting matrices Q and R define the performance criteria. Typically:

- **Q matrix:** Represents the weight on the state variables.
- **R matrix:** Represents the weight on the control input.

These matrices need to be chosen based on the desired trade-off between state accuracy and control effort. In this case we have set them:

$$Q = \text{diag}(1, 1, 1) \quad \& \quad R = 10$$

Using the identity matrix as Q is a common default choice, offering equal weighting to all states and control inputs, and simplifies the design process. However, increasing the value of R is advisable to reduce excessive control effort. By amplifying the R matrix, the focus shifts towards minimizing control effort in comparison to system performance. This adjustment aims to strike a balance between achieving desired performance and limiting control effort.

About the cutoff frequency, for the same reason of limiting the control effort, and also to guarantee a better convergence to the steady state value, we set it equal to 10rad/s .

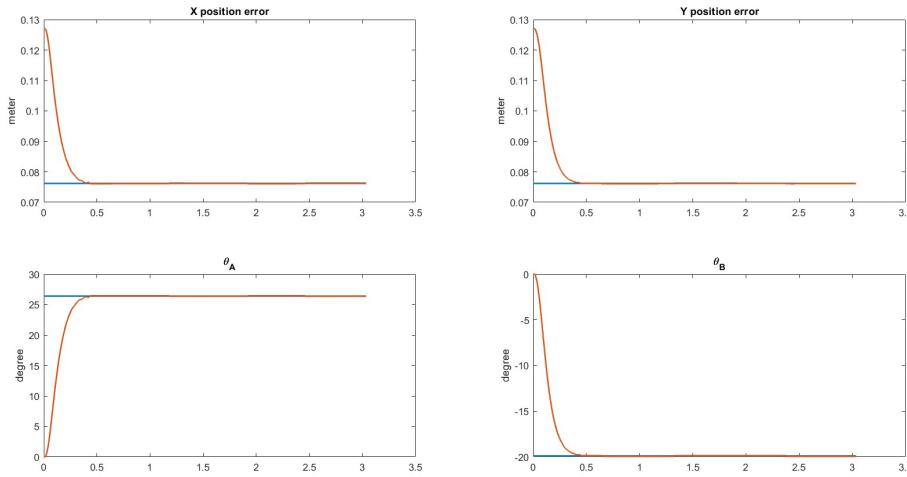


Figure 38: **State Derivative Observe with LQ Controller** for End Effector positioning, coordinates (3, 3). It is faster with respect to the pole placement implementation.

6.2 Luenberger Observer

The Luenberger Observer is designed to improve state estimation by incorporating a model of the system dynamics along with the output measurements. This observer, compared to the previous one, significantly reduces the noise sensitivity and provides more accurate and stable state estimation. As it can be seen in Figure 39, the observer design ensures that the error dynamics, defined as $e(t) = x(t) - \hat{x}(t)$, converge to zero over time, leading to reliable state feedback control.

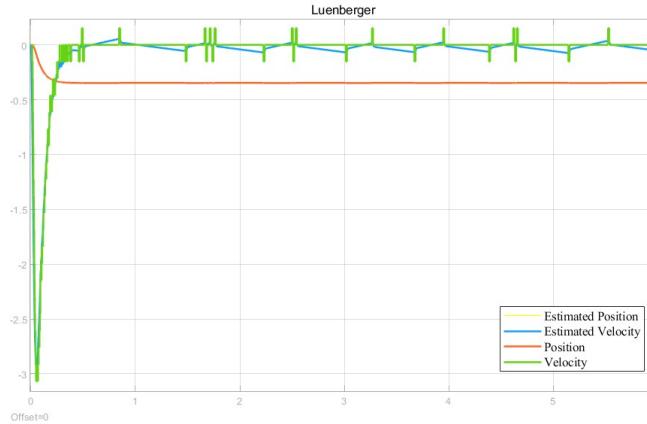


Figure 39: Performance of the Luenberger Observer. The response shows improved stability and convergence compared to the Pure Derivative Observer.

6.2.1 Pole Placement

We used the Pole Placement technique to design the poles of our resulting system. Firstly we set the dominant pole of the system to $s_1 = -17$ (as we initially tried with the derivative observer) and then we impose the other two poles in position $s_2 = -40$ and $s_3 = -60$ (higher frequencies that don't have a significant influence on the behavior of our system, because far from the cutoff frequency). The resulting gain vector of the controller is:

$$K_x = [117.6362 \quad 2.7032] \quad K_v = -1170, 7$$

We set also the poles of the Luenberger Observer with the Pole Placement method, and we impose them at a very high frequency with respect to the system ones: $S_{l1} = -400$ and $S_{l2} = -100$. The second one has a lower frequency to guarantee a better estimation on the velocity state while being less sensitive to its oscillations. **The resulting gain of the observer is:** $L = [477.21 \quad 29127]$

We computed also the close loop transfer function, from which we have identified the resulting cutoff frequency, that corresponds to 14rad/s . (Figure 49a and 49b)

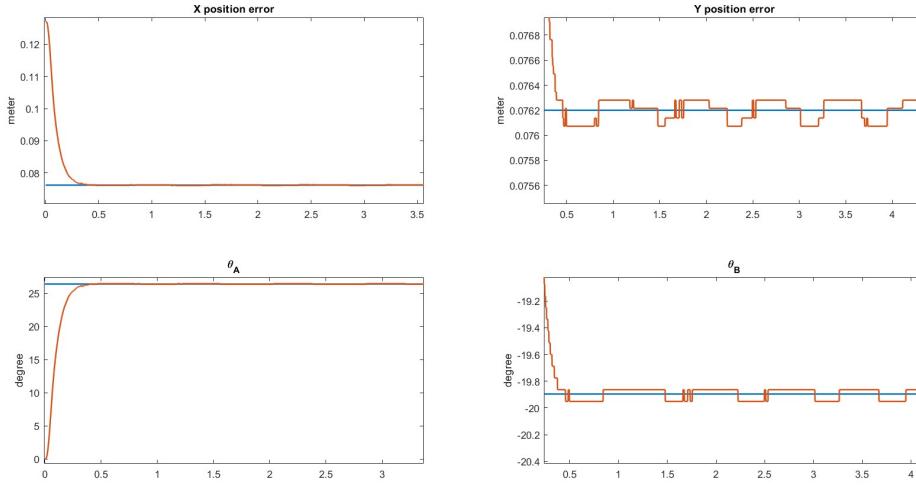


Figure 40: **Luenberger Observer with Pole Placement Controller** for End Effector positioning, coordinates (3, 3). It delivers a fast and smooth response, making it ideal for applications needing minimal oscillations and steady-state accuracy.

6.2.2 Linear Quadratic Control

In this case, we have set the two matrices of LQ controller as:

$$Q = \text{diag}(1, 1, 1) \quad \& \quad R = 1$$

For same reasoning mentioned in designing LQ for pure derivative, set the Q to identity matrix. Also by choosing $R = 1$, it normalizes the control inputs and since the system is well-behaved, there is no need to complicate the design by adjusting it. About the cutoff frequency we maintain the same choice of the one of the state derivative observer: we set it to 10rad/s . We computed also the Luenberger Observer using the LQ method:

$$Q = \text{diag}(50, 50, 50) \quad \& \quad R = 1$$

The choice of $Q = 50 * \text{eye}(3)$ prioritizes minimizing errors in state estimates, emphasizing the importance of accurate estimation. Additionally, setting $R = 1$ treats all output measurements equally, assuming uniform noise characteristics across measurements. This maintains simplicity in the observer design. **The resulting L gain of the observer is the following:** $L = [854.9463 \quad 10.7643]$

We can see in the Figure 41 that the converges to the desired position has a very good performance, without the small oscillations we get with Pole Placement.

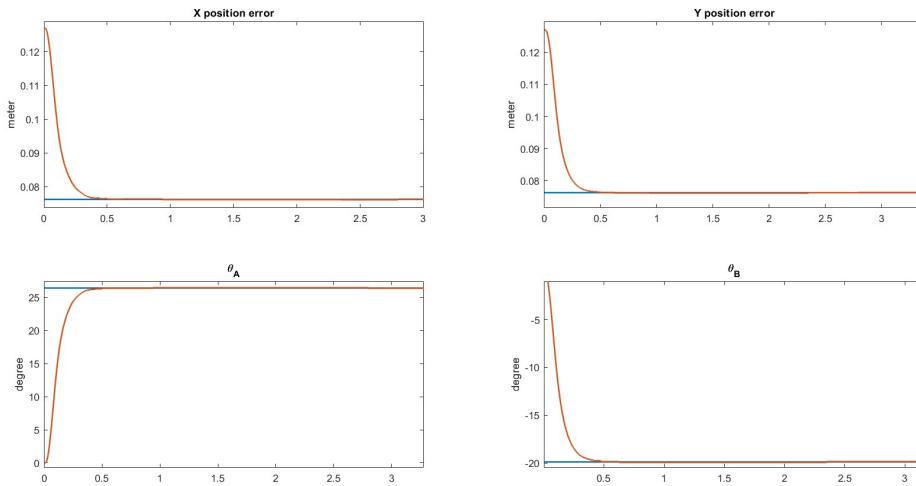


Figure 41: **Luenberger Observe with LQ Controller** for End Effector positioning, coordinates (3, 3). No oscillations with respect to Pole Placement controller.

6.3 Kalman Filter

The Kalman Filter represents the optimal state observer in the presence of stochastic noise. By utilizing statistical models of the process and measurement noise, the KF provides the best linear unbiased estimate of the states.

When implementing the KF, it is essential to define the Q and R matrices: the process and measurement noise covariance matrices. Q signifies our confidence in the model, while R represents our confidence in the sensors. Given the high accuracy of the sensors, it is advisable to use a low R . Conversely, due to the model limited accuracy within the dead zone and our desire for a consistent Q , it is recommended to use a higher Q .

$$Q = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix} \quad \& \quad R = 0.1$$

As shown in Figure 42, the KF achieves superior accuracy and robustness in state estimation, with minimal error and rapid convergence. This makes it the preferred choice for many real-world control applications.

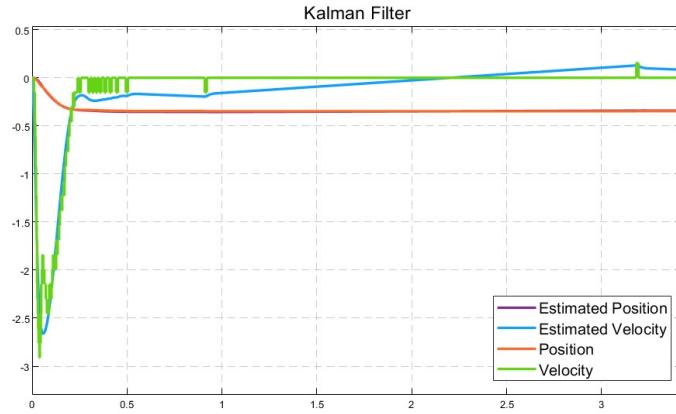


Figure 42: Performance of the Kalman Filter. The state estimates are highly accurate and exhibit minimal error.

6.3.1 Pole Placement

We designed the controller in an analogue way with respect to the Luenberger Observer: we set the poles as $s_1 = -17$, $s_2 = -40$ and $s_3 = -60$. As a result, we observed a faster response and an improvement in the stability of the system.

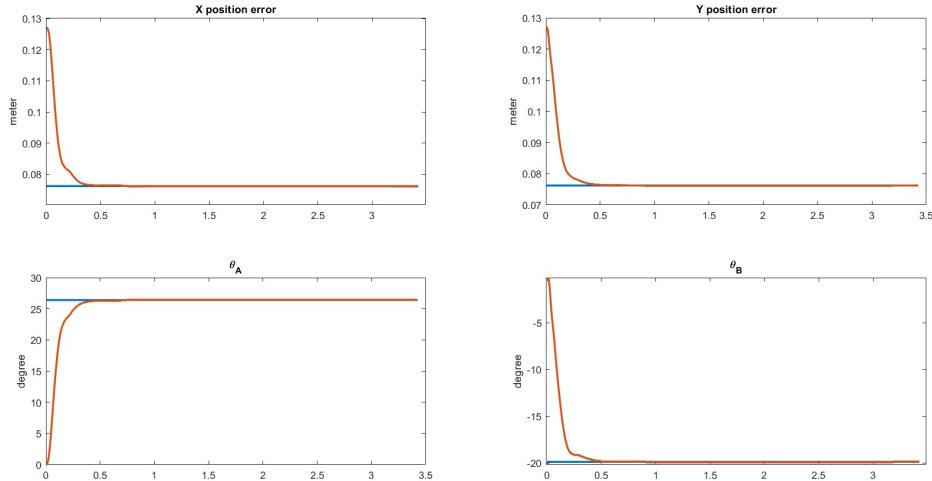


Figure 43: **Kalman Filter Observer with Pole Placement Controller** for End Effector positioning, coordinates (3, 3). It offers rapid response and it is effective for applications requiring quick corrections, high stability.

6.3.2 Linear Quadratic Control

We tested the Kalman Filter with

$$Q = \text{diag}(1, 1, 1) \quad \& \quad R = 1$$

It was observed that the control signal effort was excessively high. To address this issue, similar to the adjustment made for state estimation from the pure derivative observer, the covariance matrix was increased to $R = 5$. This adjustment improved performance while reducing the demand on control action. After successfully adjusting R , the cutoff frequency was set to 13rad/s to synchronize with the frequency obtained via Pole Placement technique, aiming for a fast system response.

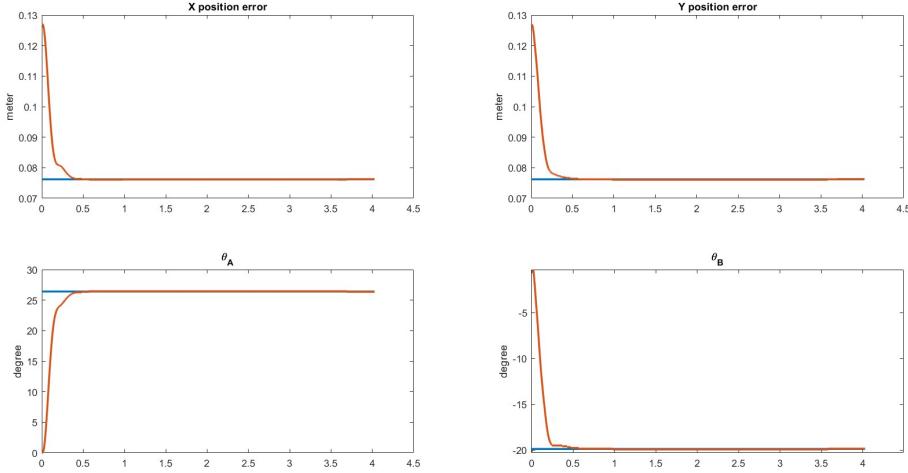


Figure 44: **Kalman Filter Observer with LQ Controller** for End Effector positioning, coordinates (3, 3). Provides optimal and smooth performance, suitable for scenarios where steady-state accuracy and minimal control effort are required.

6.4 Comparison Between Observers

- **Pure Derivative Observer:** Exhibits rapid position error convergence with a settling time at 1% of approximately 0.6 seconds with the Pole Placement technique and about 0.45 seconds with the LQ controller. Despite its quick response, it is slower compared to the other observers and may show higher sensitivity to noise, which can affect the precision. This sensitivity to noise can lead to less accurate positioning and potential instability in the presence of measurement disturbances.
- **Luenberger Observer:** Demonstrates fast and smooth position error convergence with minimal oscillations around the steady state with the pole placement and no oscillation with the LQ. The settling time, in both cases, is around 0.4 seconds. It provides a balanced approach between response speed and stability, making it suitable for applications where moderate precision and speed are required.
- **Kalman Filter Observer:** Achieves the fastest and smoothest error convergence with no oscillations, providing the most stable response. The settling time is the same of the Luenberger Observer. It is highly effective in filtering out noise and handling system uncertainties, making it ideal for high-precision operations where accuracy and stability are paramount.

The choice between these observers should be based on the specific requirements and constraints of the application at hand.

6.5 Validation

In this section we will validate the controller. We decide to validate one of the methods between LQ and Pole Placement for each observer, since the results are quite similar. The methods we validated are:

- KF with LQ control, which represents the best overall performance;
- Luenberger with pole placement, since the Luenberger observer with LQ control has similar performances to KF with LQ;
- Pure derivative with LQ control, since pure derivative with pole placement is more similar to PV controller.

6.5.1 Time Validation

The main differences between real data and simulink is due to dead zone. In reality, when the controller start with low values the error remain the same, however, in simulink by starting with low voltages the error decreases. So, whenever the controller decide to give a low value, we may counter a difference in reality and simulation. Also when we are near the convergence, we should give low values, but this low value not only may be in the dead zone and have no effects on the motors, but also have negative effects on state estimation in methods that rely on the linearity of the model (like Kalman filter). Therefore, for a method like Kalman Filter, when we are near convergence we face a problem in state estimation and also our control signal in somehow is not applied to the linear system; that is why the graphs are different in the figure 47.

To explain more in detail, for the derivative case, when we start our control in simulink with low voltage we have a low speed which effects our control strategy; however, in reality we have zero speed due to being in dead zone and we are a bit slower.

Derivative with LQ Controller

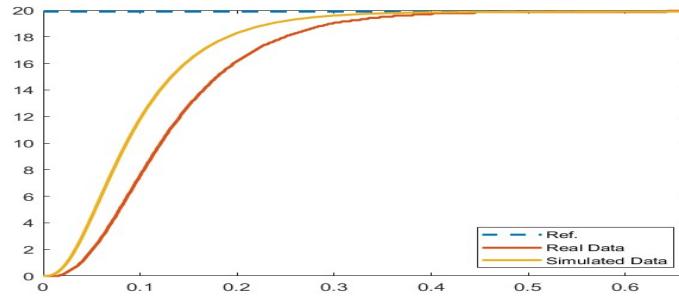


Figure 45: Time validation - Derivative as observer and LQ control - Servo B

Luenberger Observer with Pole Placement Control

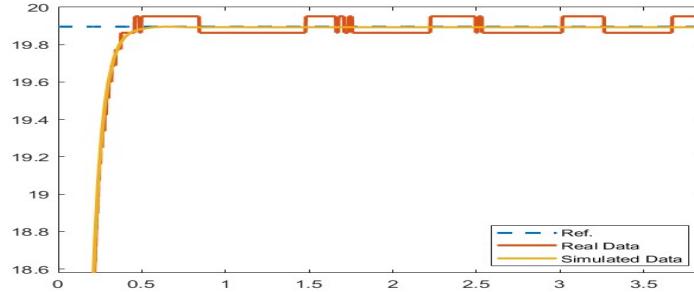


Figure 46: Time validation - Luenberger as observer and PP control - Servo B

Kalman Filter with LQ Control

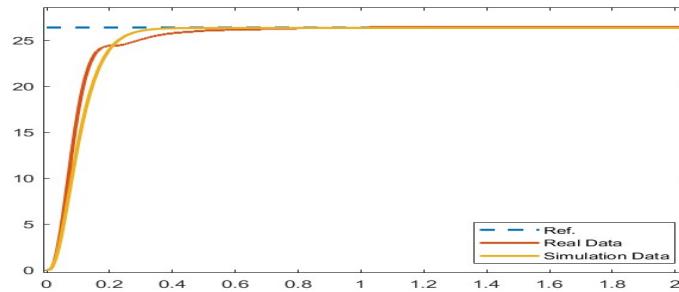


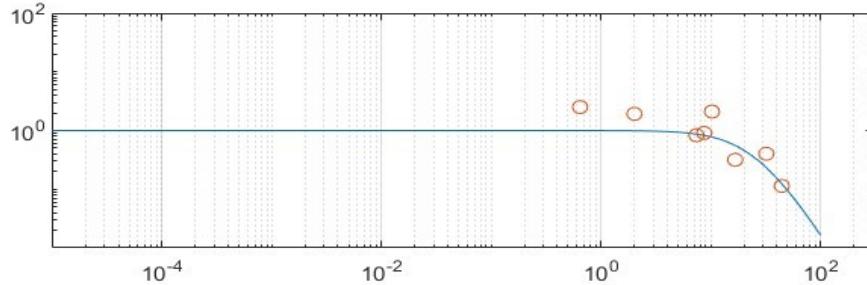
Figure 47: Time validation - KF as observer and LQ control - Servo A

Figure 46 and 47 show an almost perfect correspondence between real and simulated system. The oscillations of the Luenberger observer are so small that they can be considered negligible.

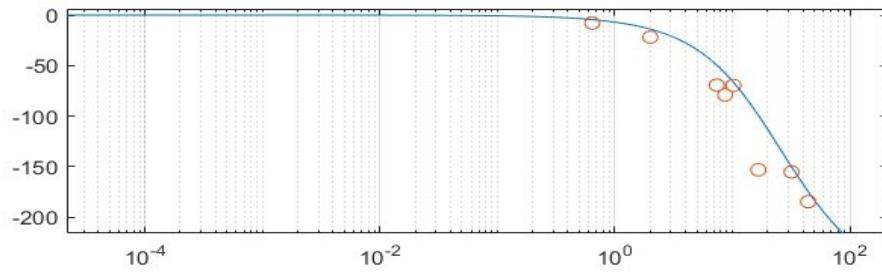
6.5.2 Frequency Validation

Just like time validation, here we do exactly the same steps that we have discussed for frequency validation of the estimated model in Section 4.4.2. The results are shown below.

Derivative with LQ Controller



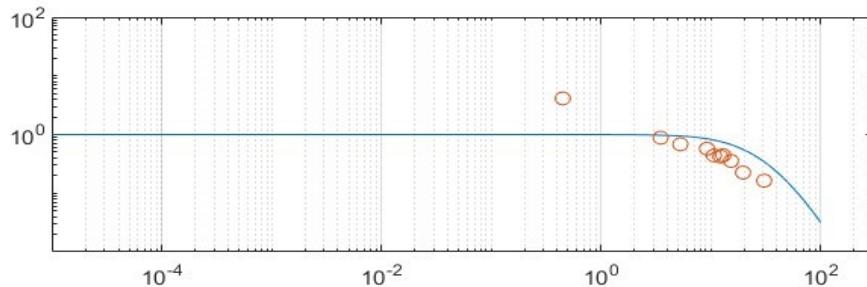
(a) Bode diagram (gain) of Derivation as observer and LQ method (closed loop) and validation data (red circles)



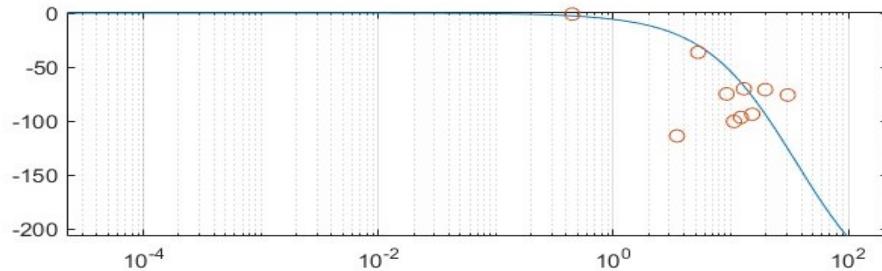
(b) Bode diagram (phase) of Derivation as observer and LQ method (closed loop) and validation data (red circles)

Figure 48: Frequency validation for Motor B

Luenberger observer with Pole Placement Control



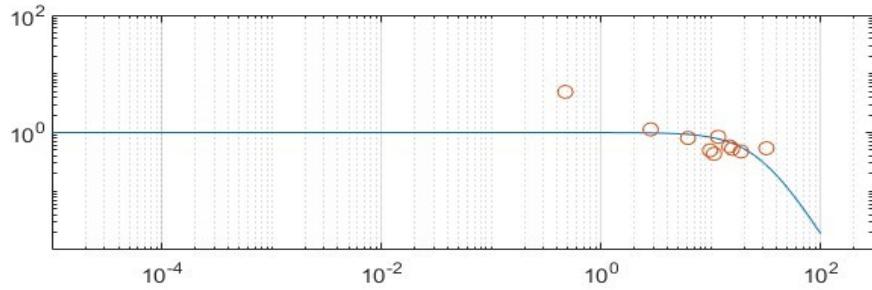
(a) Bode diagram (gain) of Luenberger with pole placement method (closed loop) and validation data (red circles)



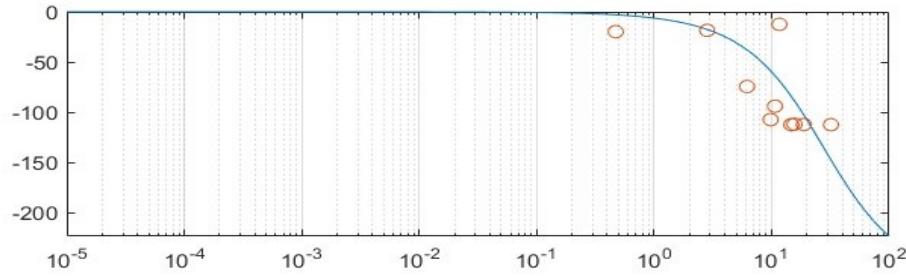
(b) Bode diagram (phase) of Luenberger with pole placement method (closed loop) and validation data (red circles)
Notice that the real system has a delay which shows its effect

Figure 49: Frequency validation for Motor B

Kalman Filter with LQ Control



(a) Bode diagram (gain) of Kalman Filter with LQ method (closed loop) and validation data (red circles)



(b) Bode diagram (phase) of Kalman Filter with LQ method (closed loop) and validation data (red circles)

Figure 50: Frequency validation for Motor A

6.6 Response to Disturbance and Noise

In Section 5.4.2, we assessed the performance of a PV controller in handling noise and disturbances, observing its capability of noise attenuation acceptable. However, the utilization of a Kalman Filter coupled with a Linear Quadratic controller demonstrates superior robustness against such uncertainties. We chose Kalman Filter with LQ control for this disturbance response experiment because between all the previous controllers it has the best filtering properties. As previously, we excite the system with 2 sinusoidal signals with the same amplitude 0.02, one at 0.5rad/s and one at 50rad/s . In this case, we expect an attenuation of 90% for high frequency noises.

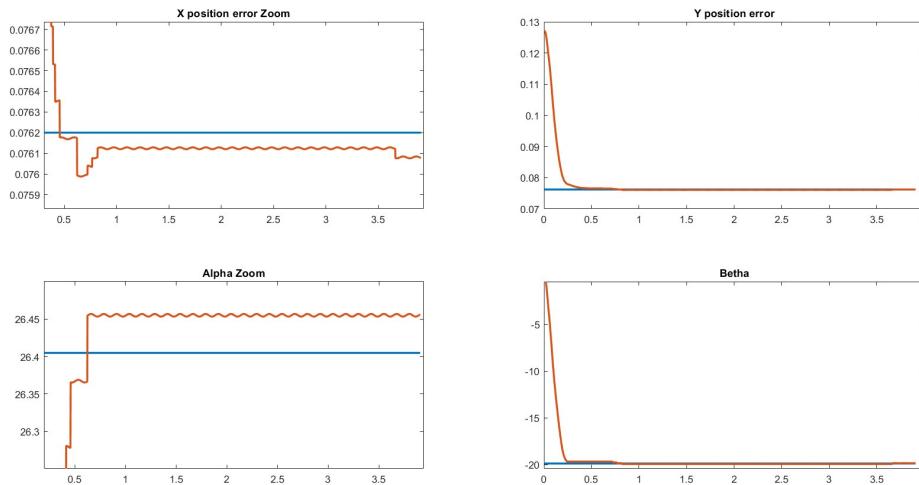


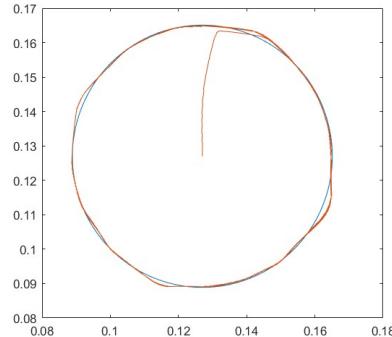
Figure 51: The response of the system to a sinusoidal disturbances, for End Effector positioning, coordinates (3, 3). Position errors reported in meter, Alpha as θ_A & Beta as θ_B are represented in degrees.

The disturbances are reduced to approximately 4% with respect to their original amplitude (far better

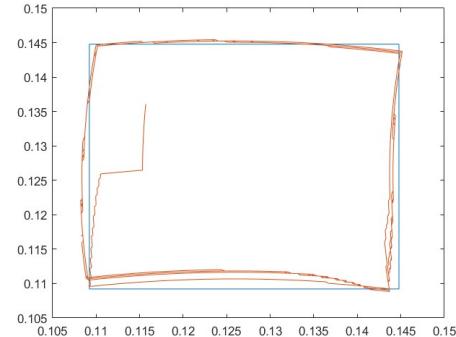
than in PV controller), while the noises are attenuated to less than 1%. The control is impressively powerful against both low frequency disturbances and high frequency noises. This is due to the fact that KF constantly updates its estimates by comparing predicted and actual measurements, allowing it to quickly adjust to changes. This ability ensures that the system accurately estimates its current state even when faced with outside disruptions, maintaining stability as shown in the figure. Additionally, The KF inherently considers both process noise and measurement noise.

6.7 Trajectory Tracking

In this section we utilize two different types of reference signals, both at 1.25rad/s : sinusoidal wave and square wave. The control strategy that is chosen is the one with Kalman Filter Observer and LQ controller.



(a) Circular trajectory, using two sinusoidal waves



(b) Square trajectory, using two square waves

Figure 52: Illustration of the system trajectory tracking in a two-dimensional Cartesian plane

The experiment reveals effective performance in tracking both circular and square paths. The system follows the square wave path with minor deviations, which are more evident at the sharp corners of the square. In comparison, the PV controller described in section 5.5 struggles with the sharp transitions of the square wave reference, resulting in more significant tracking errors.

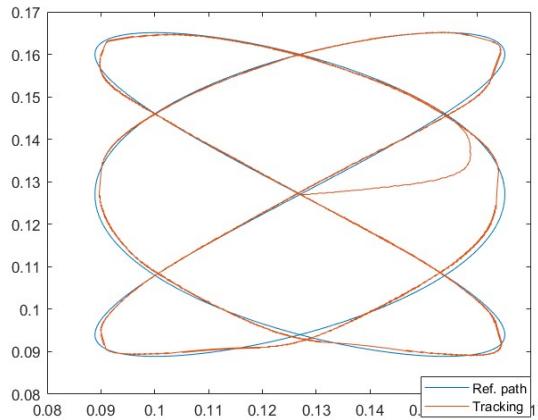


Figure 53: Illustrate different shape for the reference path with significant performance tracking

In figure 53, a crazy trajectory is imposed as a reference to the control system, nevertheless the tracking is almost perfect.

7 Advanced Controller

Model Predictive Control (MPC) is an advanced control technique that predicts future system behavior over a finite time horizon and computes the control inputs that optimize a given performance criterion, subject to system dynamics and constraints. In our case the goal of using MPC is not overcoming in terms of results the SS control, which is quite excellent already, but to design an highly adaptive control strategy that enable us to change the stress on the control action and most importantly to include the input constraints (the voltage limit) and the output constraints (the operational area and optionally the conditions in which the two servo-motors are decoupled) in an optimization problem scenario in which we arbitrarily decide which is the "penalization" for violating these constraints.

7.1 Model Predictive Control

The general formulation of the MPC optimization problem is as follows:

$$\min_{\mathbf{u}} \sum_{k=0}^{N-1} \|y(k|t) - y_{\text{ref}}(k|t)\|_Q^2 + \|u(k|t) - u_{\text{ref}}(k|t)\|_R^2$$

Subject to:

$$y(k+1|t) = Ax(k|t) + Bu(k|t)$$

$$u_{\min} \leq u(k|t) \leq u_{\max}$$

Where u_1 and u_2 are the Voltage applied to joints θ_A and θ_B , respectively. $y(k|t)$ is the predicted output at time k , $u(k|t)$ is the predicted control input at time k , $y_{\text{ref}}(k|t)$ and $u_{\text{ref}}(k|t)$ are the reference trajectory and control input, respectively, Q and R are weighting matrices, and N is the prediction horizon. A and B are the usual state space matrices of the system. $u_{\min} = -10$, the minimum voltage applicable to the motor and $u_{\max} = 10$ is the maximum one.

In addiction to this, we could optionally include these other soft constraints (violation allowed but penalized):

$$-0.1 \leq y_A(k|t) \leq 0.5 \quad -0.5 \leq y_B(k|t) \leq 0.1$$

These further constraints describe for the output the imposition of staying away from the area in which the two servomotors are not decoupled. Of course, by means of a parameter we will discuss later, we could relax these two constraints how much we want.

Moreover, since the reference is considered in some conditions, if we give references which are not in work space, the most similar path regard the cost function is followed. However, for previous methods, we could not do something like this and we totally refuse from the beginning to control and follow trajectories out of working area.

Therefore, the advantages of this method to previous ones are guarantee on safety and other from of constraint, trade off between input usage (which may increase financial cost) and error, ability to have a admissible performance on trajectories which are not in work space, and so on.

There are several MPC methods which may be used for our problem, we should select one of them. For example, we could use "**Tube-based MPC**" which gives us a tube which guarantee we always will remain in this tube. And if we are following a trajectory the tube is also moving. This method is used when we are uncertain about our model and it has good robustness features. Another popular method is "**Quasi-infinite MPC**". In this method, instead of putting a strict constraint to reaching reference point at the end of time horizon, we add a cost which can be added by a local control for a small region and we set the constrain to reach that local area. This method is very nice for nonlinear systems but for linear system this is as same as the simplest MPC format. The simplest one is indeed the "**classic MPC**".

Almost all MPC method proposed take advantage of 3 main feature:

1. Recursive feasibility: if the problem is solvable at first time instant, it will have a solution for all other time instances;
2. Stability: the loss is decreasing over time;
3. Convergence.

Since classic MPC has all these 3 main features, the other methods do not own any advantage with respect to it in our scenario. Therefore, we implemented the classic MPC. Let's start by discussing about the three mentioned main feature of MPC for the classic MPC.

First, assume a basic linear system with constraints and a quadratic cost function for as our system here is assumed to be linear.

Consider a discrete-time linear system:

$$x_{k+1} = Ax_k + Bu_k$$

with state $x_k \in \mathbb{R}^n$ and control input $u_k \in \mathbb{R}^m$. The goal is to minimize a quadratic cost function over a finite prediction horizon N :

$$J(x_k, \mathbf{u}) = \sum_{i=0}^{N-1} (x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}) + x_{k+N}^T P x_{k+N}$$

subject to state and input constraints:

$$x_{k+i} \in \mathcal{X}, \quad u_{k+i} \in \mathcal{U}, \quad \forall i = 0, \dots, N-1$$

where $Q \geq 0$, $R > 0$, and $P \geq 0$ are weighting matrices, and \mathcal{X} and \mathcal{U} are constraint sets.

7.1.1 Recursive Feasibility

Definition: The MPC optimization problem is recursively feasible if the existence of a feasible solution at time k guarantees the existence of a feasible solution at time $k+1$.

Proof Outline:

1. **Initial Feasibility:** Assume the optimization problem is feasible at time k , yielding an optimal sequence $\mathbf{u}^* = \{u_k^*, u_{k+1}^*, \dots, u_{k+N-1}^*\}$.
2. **Shifted Sequence:** Construct a shifted sequence for time $k+1$:

$$\mathbf{u}' = \{u_{k+1}^*, u_{k+2}^*, \dots, u_{k+N-1}^*, Kx_{k+N}^*\}$$

where K is a stabilizing feedback gain such that $A+BK$ is stable and $Kx_{k+N}^* \in \mathcal{U}$. Or if we have added a constraint that the state should go to zero (simple form MPC) we can easily substitute it with $u(K+N) = 0$.

3. **Feasibility of New Sequence:** Show that \mathbf{u}' is feasible at time $k+1$:

- The state x_{k+1} is given by $x_{k+1} = Ax_k + Bu_k^*$, which ensures that the initial condition for the new optimization is the result of the previous control action.
- By construction, x_{k+N}^* is the terminal state in the previous sequence, and $Kx_{k+N}^* \in \mathcal{U}$ ensures the last control input is feasible.

Thus, the problem remains feasible at $k+1$.

7.1.2 Stability

Definition: Stability in MPC implies that the state trajectory $\{x_k\}$ remains bounded and converges to the origin (or a desired equilibrium point).

Proof Outline:

1. **Cost Function Decrease:** Show that the cost function decreases at each time step.
2. **Lyapunov Function:** Use the cost function $V(x_k)$ as a Lyapunov function candidate.
3. **Decrease Condition:** Show that:

$$V(x_{k+1}) - V(x_k) \leq -\|x_k\|_Q^2 - \|u_k\|_R^2$$

This can be derived from the optimality of the control sequence and properties of the quadratic cost.

4. **Positive Definiteness:** Ensure $V(x)$ is positive definite and radially unbounded.
5. **Conclusion:** By the Lyapunov stability theorem, the system is stable if the cost function decrease condition is met.

7.1.3 Convergence

Definition: Convergence in MPC means that the state x_k converges to the equilibrium point as $k \rightarrow \infty$.

Proof Outline:

1. **Boundedness and Stability:** From the stability proof, x_k is bounded and the cost function decreases.
2. **Invariant Set:** Define a terminal invariant set \mathcal{X}_f such that if $x_k \in \mathcal{X}_f$, then the terminal control law $u = Kx_k$ keeps the state within \mathcal{X}_f .
3. **Convergence to Invariant Set:** Show that x_k converges to \mathcal{X}_f .
4. **Equilibrium Point:** Since \mathcal{X}_f includes the equilibrium point (typically the origin), the state will eventually converge to the equilibrium.

Or in other way we can use the difference between each two cost, sum them and using Barbala's Lemma to prove convergence to zero. As we saw below:

1. $J^*(x(\delta)) \leq J^*(x(0)) - \int_0^\delta L(x_{\text{MPC}}(\tau), u_{\text{MPC}}(\tau)) d\tau$
2. $J^*(x(2\delta)) \leq J^*(x(\delta)) - \int_0^\delta L(\dots) d\tau$
3. \vdots
4. $J^*(x(\infty)) \leq J^*(x(0)) - \int_0^\infty L(x_{\text{MPC}}(\tau), u_{\text{MPC}}(\tau)) d\tau$

The expression on the right of the last inequality can be broken down as follows:

$$\underbrace{J^*(x(0))}_{\geq 0 \text{ and finite}} - \underbrace{\int_0^\infty L(x_{\text{MPC}}(\tau), u_{\text{MPC}}(\tau)) d\tau}_{J_{\text{MPC}}(x(0))}$$

Therefore,

$$J_{\text{MPC}}(u(\cdot)) \text{ is finite} \implies \text{B-Lemma} \implies L \rightarrow 0$$

7.2 MPC in Matlab 2019/b

We now explain in detail how we adapted the theory of MPC to our system, in particular the Matlab 2019/b implementation of model predictive control differs a bit from the theory specified above.

In principle, we defined in matlab our own MPC class. We implemented some functions to compute the equilibrium position and the optimal input sequence by means of the Casadi optimization tool [1]. The prediction horizon was $N = 3$, $Q = [100000 \ 0; 0 \ 1]$, $R = 0.01$ and the sampling time $T_S = 0.05$. We were able to reach a settling time at 1% in almost 0.2s and no overshoot at all in the simulation, considering $u_{\text{MAX}} = 10V$. Unfortunately we had problems in running the Casadi optimization tool in our environment, so we had to chose a less performative method of implementation for MPC.

We decided, at the end, to implement the standard Matlab MPC controller [2], which was modified in Matlab in order to match our requests. The configuration for our experiments was the following one:

- A, B, C, D - The state space representation is the same of State Space Control;
- $T_S = 0.05$ - The sampling time;
- $\text{mpcobj.PredictionHorizon} = 3$ - The prediction horizon;
- $\text{mpcobj.ControlHorizon} = 2$ - The control prection horizon;
- $\text{mpcobj.MV.Min} = 10$ - The upper limit for input voltage (Hard Constraint);
- $\text{mpcobj.MV.Max} = -10$ - The lower limit for input voltage (Hard Constraint);
- $\text{mpcobj.Weights.ManipulatedVariables} = 0.35$ - The weight on the input;
- $\text{mpcobj.Weights.ManipulatedVariablesRate} = 0.25$ - The weight on the rate of change of the input;
- $\text{mpcobj.Weights.OutputVariables} = 1$ - The weight on the output;
- $\text{mpcobj.Weights.ECR} = 100$ - The weight on soft constraints;

Our block includes a state estimation, done with the a Kalman Filter with the same configuration of the one utilized in section 6 for State Space control. With this configuration, we were able to obtain in simulation similar result to the Casadi MPC: settling time at 1% in 0.26s, no overshoot and a steady state error of 0.5%. The reason why we selected the previous parameters in that due to the presence of the dead zone (that we did not model in this case) we must use an input signal strong enough and a reduced prediction horizon, otherwise the steady state error may start to get relevant.

7.2.1 MPC for Motor Position Control

In Figure 54 it is shown the MPC control system response to a step reference of amplitude 23.65° . The results are the best ones we ever achieved: settling time at 1% in 0.26s, steady state error close to 0.3% and overshoot under 0.5%. This shows that the parameters were perfectly tuned and especially that in case of motor position control MPC is our best choise.

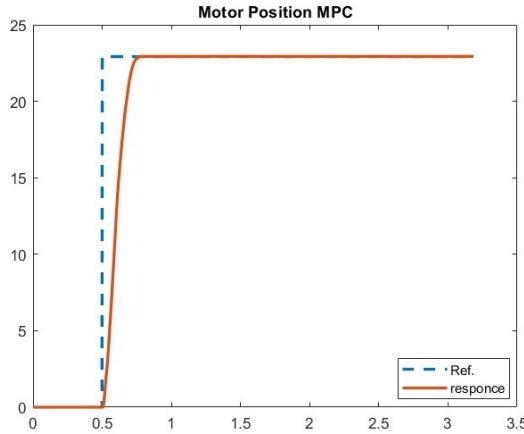


Figure 54: MPC step response

7.2.2 MPC for Trajectory Tracking

We now use the MPC control with the same configuration to track a circle reference, the same one utilized in section 6 for State Space control, the sine waves used as reference are set to 1.25rad/s frequency.

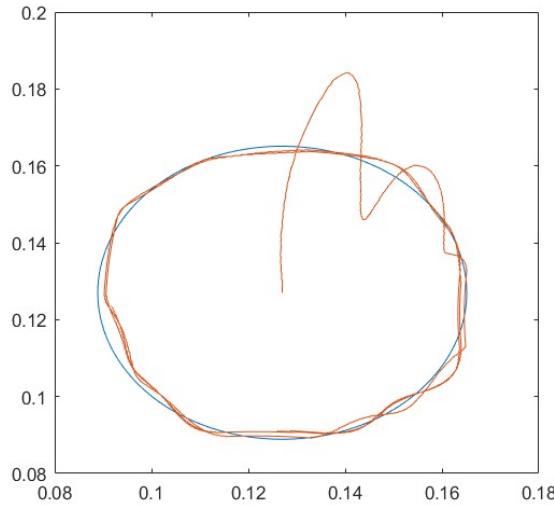


Figure 55: MPC trajectory tracking

In Figure 55, the MPC controller follows the circle trajectory almost perfectly. Unfortunately, with respect to Kalman Filter + LQ control trajectory tracking we have two problems: the trajectory is tracked with small errors and at the beginning there are strong oscillations. This of course does not mean that the MPC is bad designed: the oscillations at the beginning depend on the fact that the control action is very strong due to limitation in computational power, the errors are caused by the fact that the reference changes continuously and at every 0.05s we have to recompute the optimal control sequence.

In a nutshell, generally MPC could track trajectories with far more efficiency than State Space control, but in our specific scenario we are very limited, both in computational power and in optimization tool choices. In fact, the controller could be optimized in the following ways:

- Using a reduced sampling time - this is doable in simulation, but in our setup the computational power is extremely limited, so if we reduce the sampling period the system becomes too slow in computations and it is not able to track any trajectory;
- Using advanced optimization tools such as Casadi - so that we are faster with the same computational power;
- Implementing an advanced algorithm in which we know in advance the trajectory we want to track - doable if in our application the end-effector trajectory is known in advance;

7.2.3 MPC for Trajectory Tracking With Soft Constraints

It is extremely relevant for MPC to implement in the block the soft constraints related to the workspace region defined in Section 3. It is already written above the chosen weight for soft constraints (100 - a strong weight). We can add to the code the following commands:

- $mpcobj_A.OV(1).Min = -0.1$ - Minimum angle for motor A in radians;
- $mpcobj_A.OV(1).Max = 0.5$ - Maximum angle for motor A in radians;
- $mpcobj_B.OV(1).Min = -0.5$ - Minimum angle for motor B in radians;
- $mpcobj_B.OV(1).Max = 0.1$ - Maximum angle for motor B in radians;

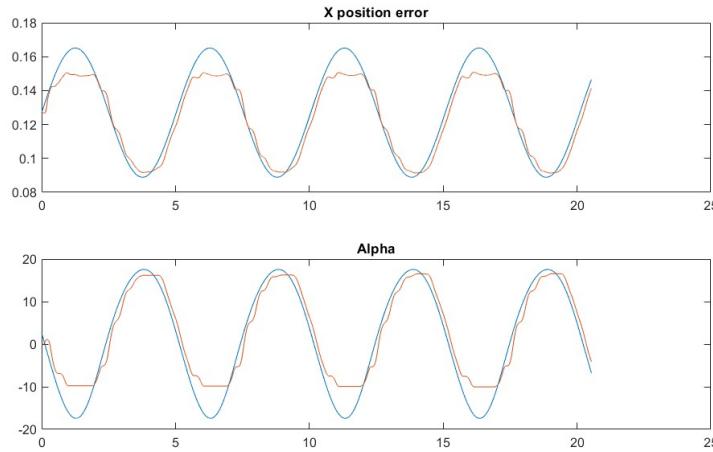


Figure 56: Reference tracking in time domain. Position error is reported in meter, and Alpha as angle for Servo A is reported in degree.

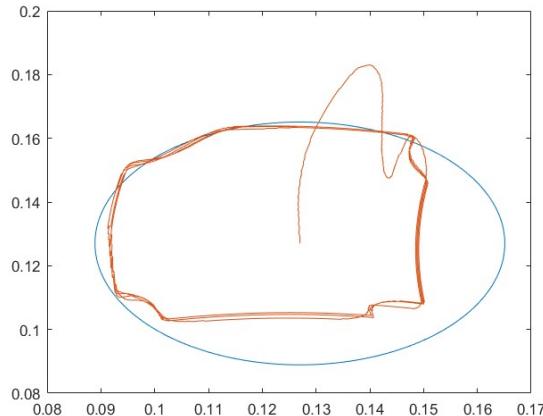


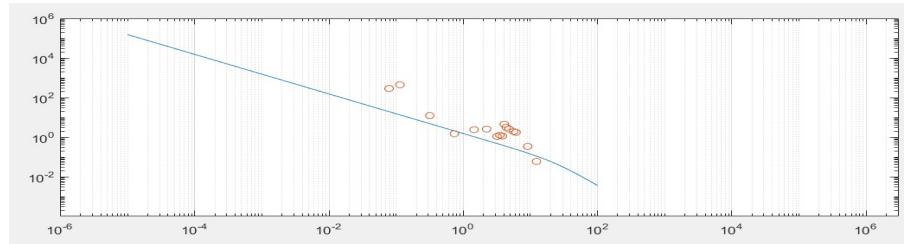
Figure 57: Trajectory tracking with constraints

As observable from Figure 56 and Figure 57, the soft constraints implementation works perfectly. This is maybe the biggest advantage in MPC: we can in practise decide and change rapidly the level of nonlinearities in the system by hardening or softening the 4 angle soft constraints.

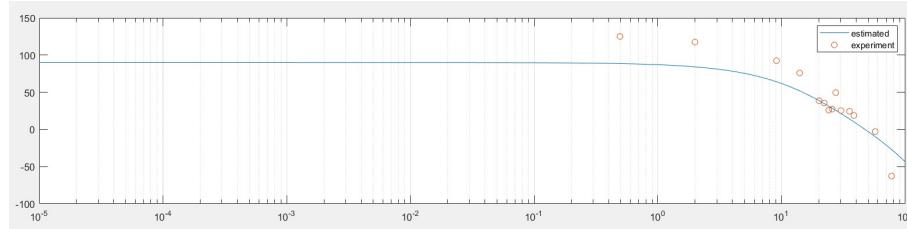
In conclusion the MPC controller is the best one we tried for motor position control, and even if in our specific configuration the MPC controller works a bit worse than KF and LQ control in trajectory tracking, it still has some key advantages and further possibilities easily implementable.

click for [Video of Tracking Circle Trajectory](#)

A Appendix: Extra Material



(a) Bode diagram (gain) of identified system and validation data(red circles)



(b) Bode diagram (phase) of identified system and validation data(red circles)

Figure 58: Motor B frequency validation

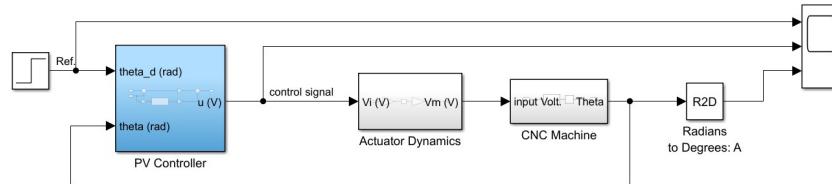


Figure 59: Simulink model for Angular Position Control

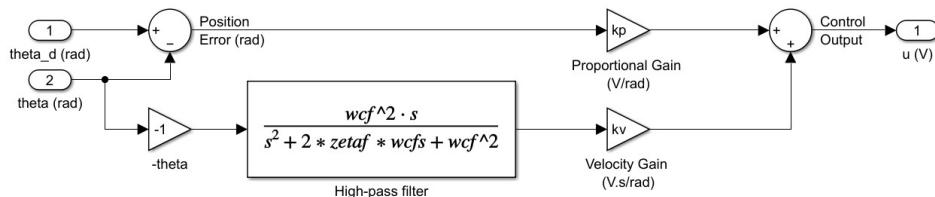


Figure 60: Simulink model illustrating a Proportional-Velocity-Controller utilizing a high-pass filter instead of a pure derivative, with a cutoff frequency of 50Hz and a damping ratio of 0.9. This modification is employed to mitigate the amplification of high-frequency noise and prevent instability caused by abrupt changes in the control signal.

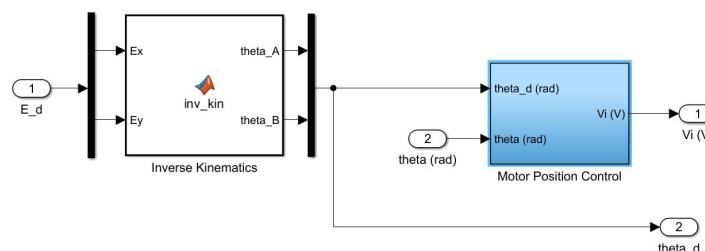


Figure 61: illustrating Controller Block in Trajectory Tracking, utilizing Inverse Kinematic to calculate the angles θ_A & θ_B

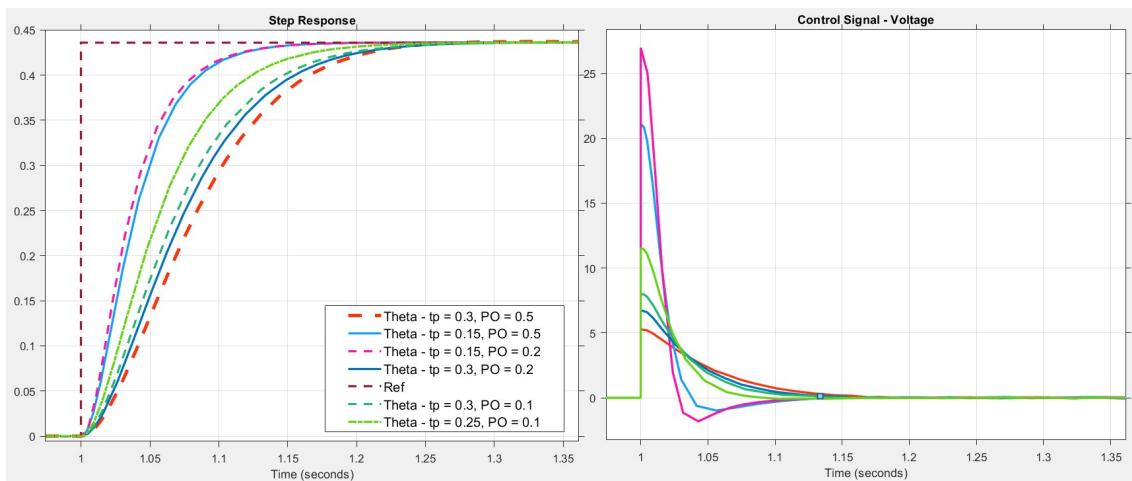
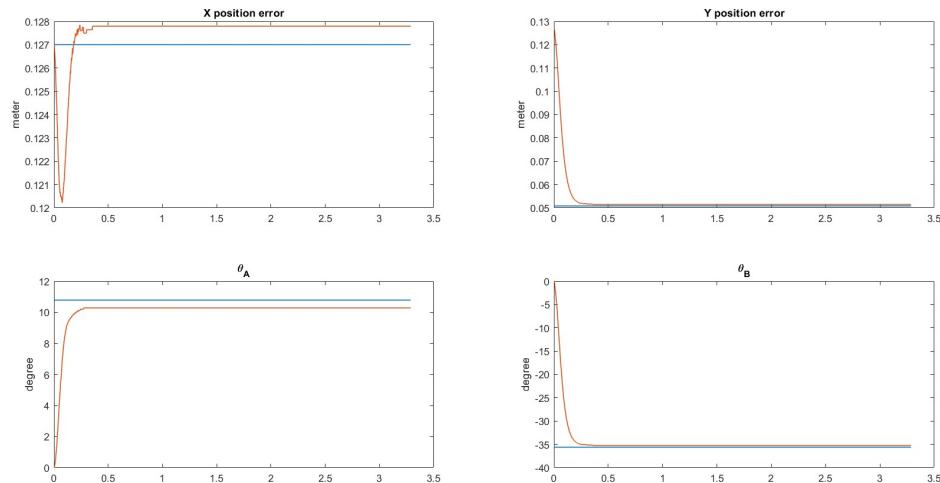
Figure 62: Control Signal Analysis for different t_p and PO

Figure 63: Implementation of PV Controller for End Effector position, point (5.5, 2) - Static friction effect within the Servo A

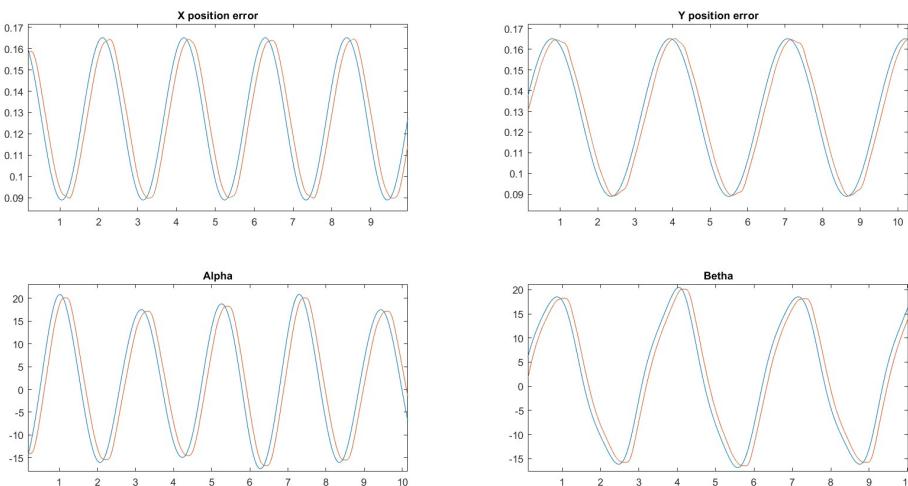


Figure 64: KF with LQ Controller for the Trajectory of Figure 53

References

- [1] Mathworks. casadi/optistack. Accessed on 2024-06-01.
- [2] Mathworks. mpc. Accessed on 2024-06-01.
- [3] M. Vajta. Some remarks on padé-approximations. *Department of Mathematical Sciences, University of Twente*, 2000.
- [4] University of Illinois. Time response of first order systems. Accessed on 2024-04-10.
- [5] Lee T.H. Tan K.K. Wang, Q.G. *Time-Delay Systems*. 1999.