Politecnico di Milano

AUTOMATION AND CONTROL ENGINEERING

# Study and Control of the Rotary Flexible Joint

Course

**Automation and Control Laboratory**

Student

**Andrea Archetti – 10616682**
**Alp Recep Dayan – 10823110**
**Alessandro Firetto – 10633148**
**Jesucristo Torres Toledo – 10822036**

**Academic Year 2022 – 2023**

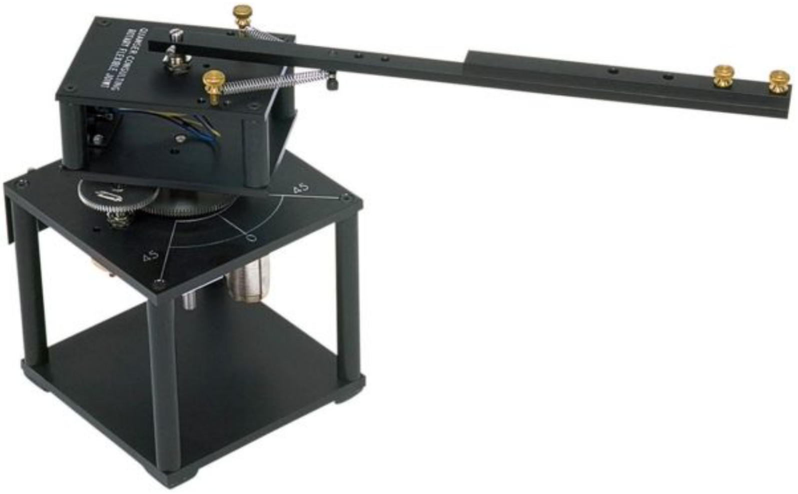
# Contents

# Problem Description

This report describes our model of the "rotary flexible joint" device made by Quanser and the control schemes we designed to control it.

The system is composed by a DC motor that provides torque to a metal base, over which a metal arm is fixed with an hinge and two symmetrical springs.

The length of the arm, hence its inertia, and the equilibrium length of the springs can be modified in a variety of different configurations.



The system has several interfaces that could be connected to an acquisition system (ADC/DAC + Amplifier) to acquire measurements and provide input signal, namely:

- Actuators:

  Voltage driven DC motor;

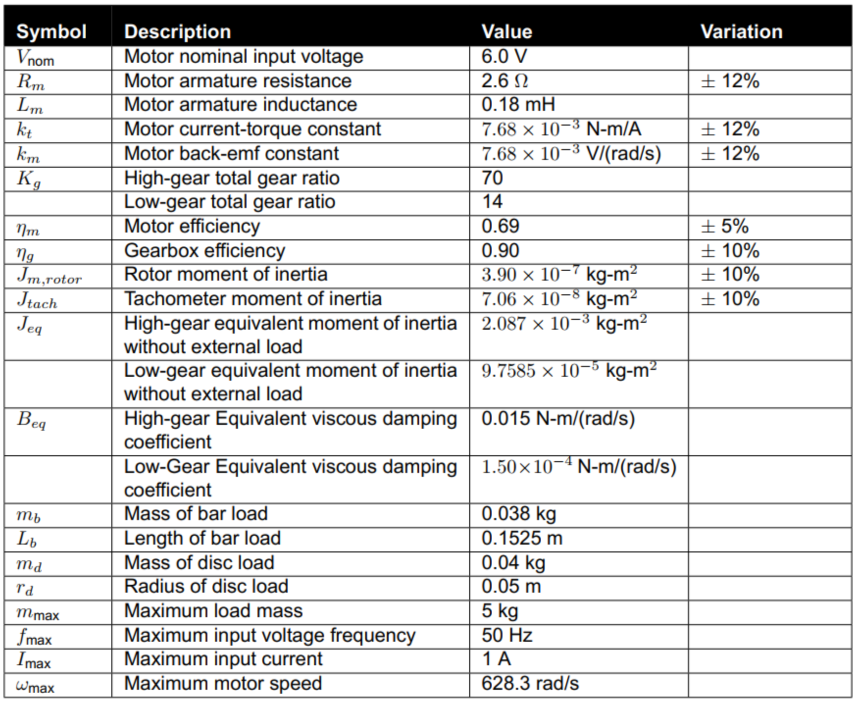
- Sensors:

  Incremental Encoder for the position of the base with respect to the global reference frame;

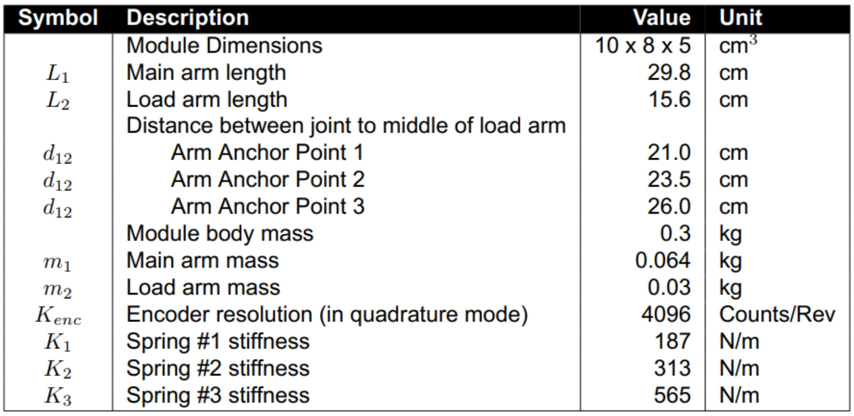
  Incremental Encoder for the relative position of the arm with respect to the base.

The acquisition system, composed of ADC/DAC + Amplifier model, is out of the scope of this report.

In the tables below we can see respectively the datasheet for the motor and for the flexible joint:



| Symbol | Description | Value | Variation |
|---|---|---|---|
| $V_{\text{nom}}$ | Motor nominal input voltage | 6.0 V | |
| $R_m$ | Motor armature resistance | 2.6 $\Omega$ | $\pm$ 12% |
| $L_m$ | Motor armature inductance | 0.18 mH | |
| $k_t$ | Motor current-torque constant | $7.68 \times 10^{-3}$ N-m/A | $\pm$ 12% |
| $k_m$ | Motor back-emf constant | $7.68 \times 10^{-3}$ V/(rad/s) | $\pm$ 12% |
| $K_g$ | High-gear total gear ratio | 70 | |
| | Low-gear total gear ratio | 14 | |
| $\eta_m$ | Motor efficiency | 0.69 | $\pm$ 5% |
| $\eta_g$ | Gearbox efficiency | 0.90 | $\pm$ 10% |
| $J_{m,rotor}$ | Rotor moment of inertia | $3.90 \times 10^{-7}$ kg-m$^2$ | $\pm$ 10% |
| $J_{tach}$ | Tachometer moment of inertia | $7.06 \times 10^{-8}$ kg-m$^2$ | $\pm$ 10% |
| $J_{eq}$ | High-gear equivalent moment of inertia without external load | $2.087 \times 10^{-3}$ kg-m$^2$ | |
| | Low-gear equivalent moment of inertia without external load | $9.7585 \times 10^{-5}$ kg-m$^2$ | |
| $B_{eq}$ | High-gear Equivalent viscous damping coefficient | 0.015 N-m/(rad/s) | |
| | Low-Gear Equivalent viscous damping coefficient | $1.50 \times 10^{-4}$ N-m/(rad/s) | |
| $m_b$ | Mass of bar load | 0.038 kg | |
| $L_b$ | Length of bar load | 0.1525 m | |
| $m_d$ | Mass of disc load | 0.04 kg | |
| $r_d$ | Radius of disc load | 0.05 m | |
| $m_{\text{max}}$ | Maximum load mass | 5 kg | |
| $f_{\text{max}}$ | Maximum input voltage frequency | 50 Hz | |
| $I_{\text{max}}$ | Maximum input current | 1 A | |
| $\omega_{\text{max}}$ | Maximum motor speed | 628.3 rad/s | |



| Symbol | Description | Value | Unit |
|---|---|---|---|
| | Module Dimensions | 10 x 8 x 5 | cm$^3$ |
| $L_1$ | Main arm length | 29.8 | cm |
| $L_2$ | Load arm length | 15.6 | cm |
| | Distance between joint to middle of load arm | | |
| $d_{12}$ | Arm Anchor Point 1 | 21.0 | cm |
| $d_{12}$ | Arm Anchor Point 2 | 23.5 | cm |
| $d_{12}$ | Arm Anchor Point 3 | 26.0 | cm |
| | Module body mass | 0.3 | kg |
| $m_1$ | Main arm mass | 0.064 | kg |
| $m_2$ | Load arm mass | 0.03 | kg |
| $K_{enc}$ | Encoder resolution (in quadrature mode) | 4096 | Counts/Rev |
| $K_1$ | Spring #1 stiffness | 187 | N/m |
| $K_2$ | Spring #2 stiffness | 313 | N/m |
| $K_3$ | Spring #3 stiffness | 565 | N/m |

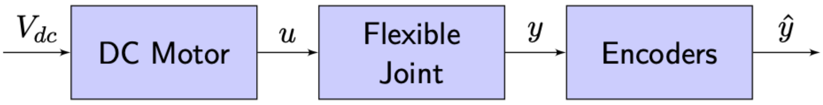
The main task of this project is to provide a basic control strategy for such system and to develop a more advanced control strategy to accommodate all possible configurations of the system in question.

This goal is divided in sub-tasks to be achieved:

1. Position control of the base, with a frequency based approach;
2. Position control of the arm tip, with a frequency based and a state space approaches;
3. Manage uncertainties and control the position of the arm tip with the system in different configurations, with a state space approach and advanced control techniques (robust or adaptive).

# Model Identification

The system could be schematized as:



The DC motor is powered by a voltage $V_{dc}$ and provides torque $u$ to the flexible joint. Finally two encoders read the angular positions $y$ of both the base and the tip and send the measured values $\hat{y}$ to the ADC converter.

## 2.1 Mathematical Model

### 2.1.1 DC motor equations

Before starting to model the DC motor we can get the time constant of its dynamics from the values in the datasheet:

$$\frac{R}{L} = \frac{2.6\,\Omega}{0.18\,mH} \approx 15\,KHz$$

As this is clearly above the frequency range of the mechanical system, we can neglect its dynamics and model only its static contribution.

The physical equations of the DC Motor then become:

$$\begin{cases} V_a = R_a I_a + E \\ E = k_m \dot{\theta} \\ \tau = k_t I_a \end{cases}$$

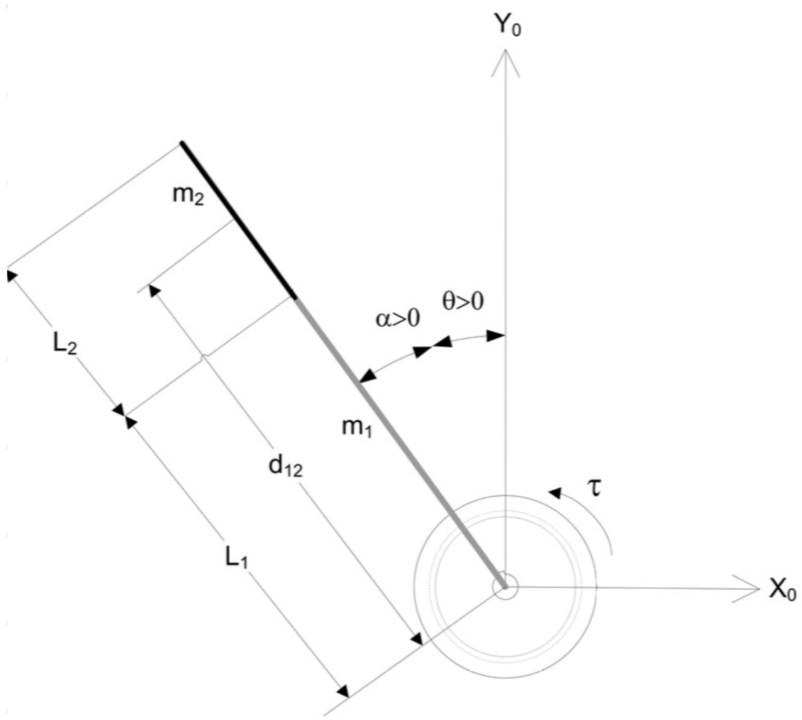
After several mathematical steps and considering the gearbox ratio $K_g$ and the conversion efficiencies we get:

$$\tau = \frac{\eta_m \eta_g k_t K_g (V - K_g k_m \dot{\theta})}{R_m}$$

Where $\tau$ is the torque applied to the system and $\theta$ is the angular position of the base.

### 2.1.2 Flexible joint equations

We can represent our system as follows:



Here we have a 2-dofs mechanical system that can be modelled using a Lagrangian approach.

The 2 deegrees of freedom are:

- $\theta$: the absolute angular position of the base;
- $\alpha$: the relative angular position of the arm with respect to the base.

The kinetic energy:

$$T = \frac{1}{2}J_{eq}\dot{\theta}^2 + \frac{1}{2}J_L(\dot{\theta} + \dot{\alpha})^2$$

where $J_{eq}$ refers to the equivalent inertia of the motor + gearbox, and $J_L$ refers to the inertia of the arm.

The potential energy:

$$V = \frac{1}{2}K_s\dot{\alpha}^2$$

where $K_s$ refers to the linearized stiffness of the equivalent torsional spring.

The dissipative function:

$$D = \frac{1}{2}B_{eq}\dot{\theta}^2 + \frac{1}{2}B_L\dot{\alpha}^2$$

where $B_{eq}$ and $B_L$ refer respectively to the equivalent friction of the motor + gearbox and the equivalent friction of the arm.

The virtual work:

$$\delta W = \tau \cdot \delta\theta$$

The dynamics of the system can be found applying the Euler-Lagrange equations for each deegree of freedom :

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}}\right) - \left(\frac{\partial T}{\partial x}\right) + \left(\frac{\partial D}{\partial \dot{x}}\right) + \left(\frac{\partial V}{\partial x}\right) = \left(\frac{\delta W}{\delta x}\right)$$

Finally we get the following system of equation:

$$\begin{cases} J_{eq}\ddot{\theta} + J_L(\ddot{\theta} + \ddot{\alpha}) + B_{eq}\dot{\theta} = \tau \\ J_L(\ddot{\theta} + \ddot{\alpha}) + B_L\dot{\alpha} + K_s\alpha = 0 \end{cases}$$

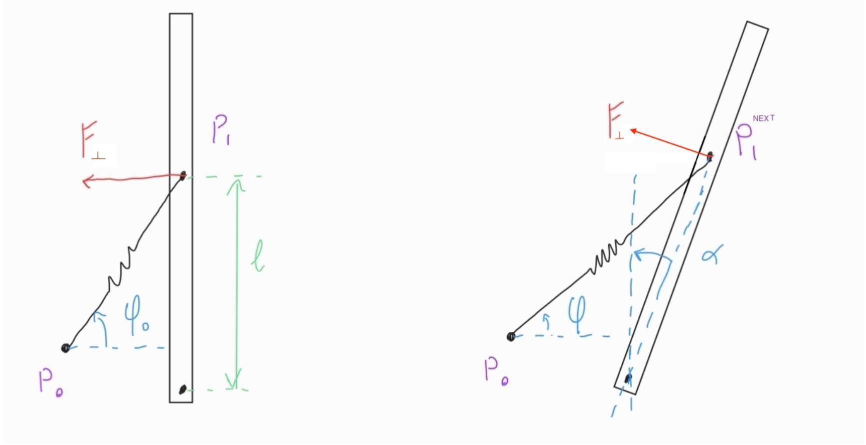
### Non-linear model of the springs

Until now we have considered the two linear springs as an equivalent torsional one in order to reduce the complexity of the model.

To prove the validity of this assumption we can consider Hooke's law:

$$F = -K_s(x_k - x_0)$$

Thanks to the symmetry of the system, studying the behaviour of a single spring is enough to model both of them.



In the left figure the equilibrium position is $x_k = x_0$, where:

$$\varphi_0 = atan\left(\frac{P_{1_y}}{P_{0_x}}\right)$$

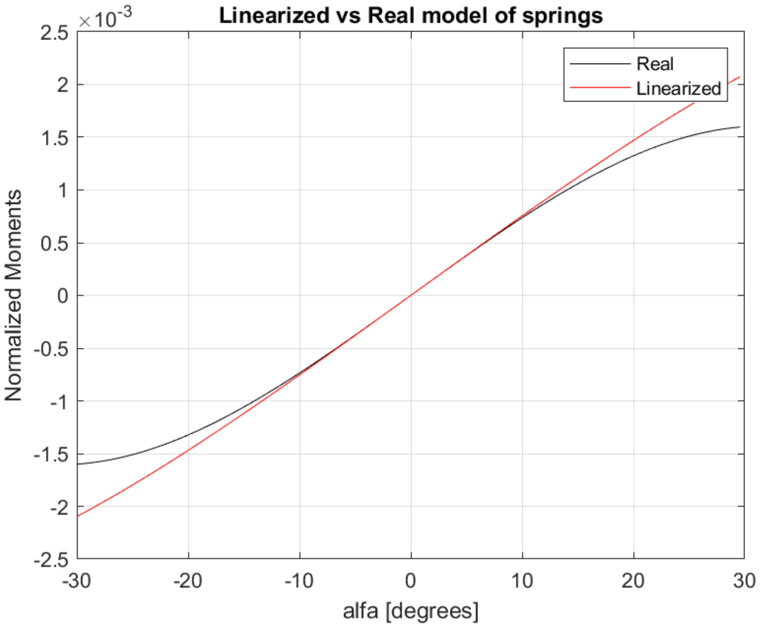
$$F_\perp = F \cdot cos(\varphi_0)$$

$$x_0 = \sqrt{(P_{1_x} - P_{0_x})^2 + (P_{1_y} - P_{0_y})^2}$$

When we perturb the system we get:

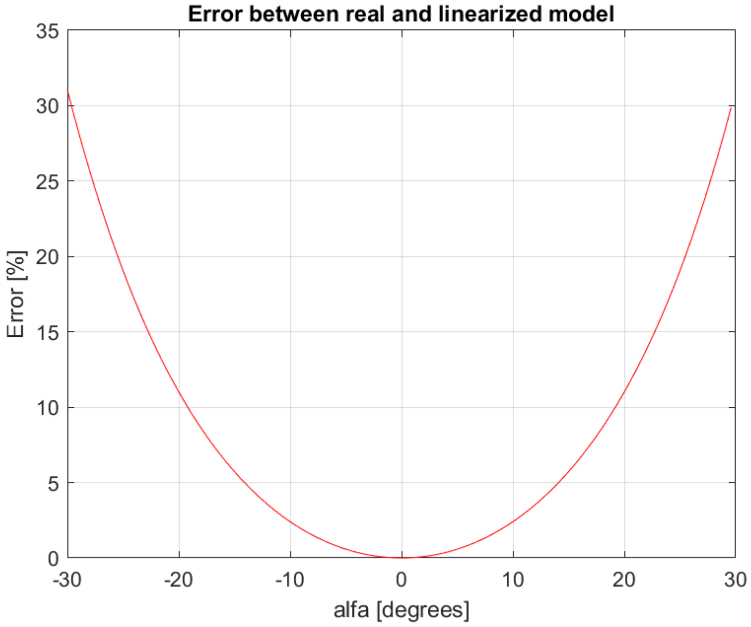
$$x_k = \sqrt{(P_{1_x} - P_{0_x})^2 + (P_{1_y} - P_{0_y})^2} \rightarrow \Delta x_k = x_k - x_0$$

$$P_1^{NEXT} = \begin{bmatrix} l \cdot sin(\alpha) \\ l \cdot (1 - cos(\alpha)) \end{bmatrix} \rightarrow F_\perp = F \cdot cos(\varphi + \alpha)$$



The plot above represents the value of the $F_\perp$ as function of the angle $\alpha$ and its linear approximation at the origin.

The error between the two curves is expressed by the curve below:



As we can see, while $\alpha$ is below 20° the error between the two models is less than 10%. As the tip's angle always remained below 10° in our measurements, the linear model introduced before is precise enough to represent our system.

### 2.1.3 State space representation

#### Continuous time

Starting from the aforementioned equations:

$$\begin{cases} J_{eq}\ddot{\theta} + J_L(\ddot{\theta} + \ddot{\alpha}) + B_{eq}\dot{\theta} = \frac{\eta_m \eta_g k_t K_g (V - K_g k_m \dot{\theta})}{R_m} \\ J_L(\ddot{\theta} + \ddot{\alpha}) + B_L \dot{\alpha} + K_S \alpha = 0 \end{cases}$$

we develop the state space system in continuous time, having as state the following array:

$$\begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \\ \dot{\alpha} \end{bmatrix}$$

the A matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\eta_m \eta_g k_t k_m K_g^2 + B_{\mathrm{eq}} R_m}{J_{\mathrm{eq}} R_m} & \frac{K_s}{J_{\mathrm{eq}}} & \frac{B_L}{J_{\mathrm{eq}}} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\eta_m \eta_g k_t k_m K_g^2 + B_{\mathrm{eq}} R_m}{J_{\mathrm{eq}} R_m} & -K_S \left( \frac{J_{\mathrm{eq}} + J_L}{J_{\mathrm{eq}} J_L} \right) & -B_L \left( \frac{J_{\mathrm{eq}} + J_L}{J_{\mathrm{eq}} J_L} \right) \end{bmatrix}$$

and the B matrix:

$$\begin{bmatrix} 0 \\ \frac{\eta_m \eta_g k_t K_g}{R_m J_{\mathrm{eq}}} \\ 0 \\ -\frac{\eta_m \eta_g k_t K_g}{R_m J_{\mathrm{eq}}} \end{bmatrix}$$

And having the angular positions $\theta$ and $\alpha$ as the outputs of the system.

#### Discrete time

In order to apply optimization techniques on our data to find the parameters of the system, we need a state space representation in discrete time. To do so we apply the forward Euler method considering a sampling time $\Delta$.

Our discrete time A matrix becomes then:

$$\begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 - \Delta\frac{\eta_m \eta_g k_t k_m K_g^2 + B_{eq} R_m}{J_{eq} R_m} & \Delta\frac{K_s}{J_{eq}} & \Delta\frac{B_L}{J_{eq}} \\ 0 & 0 & 1 & \Delta \\ 0 & \Delta\frac{\eta_m \eta_g k_t k_m K_g^2 + B_{\mathrm{eq}} R_m}{J_{\mathrm{eq}} R_m} & -\Delta K_S \left( \frac{J_{\mathrm{eq}} + J_L}{J_{\mathrm{eq}} J_L} \right) & 1 - \Delta B_L \left( \frac{J_{\mathrm{eq}} + J_L}{J_{\mathrm{eq}} J_L} \right) \end{bmatrix}$$

whereas the B matrix becomes:

$$\begin{bmatrix} 0 \\ \Delta\frac{\eta_m \eta_g k_t K_g}{R_m J_{\mathrm{eq}}} \\ 0 \\ -\Delta\frac{\eta_m \eta_g k_t K_g}{R_m J_{\mathrm{eq}}} \end{bmatrix}$$

while the outputs are still $\theta$ and $\alpha$.

## 2.2 Identification Techniques

### 2.2.1 The CVX package

Constrained numerical optimization is a technique used in system identification to estimate the parameters of a mathematical model while considering specific constraints on the parameter values. In system identification, these constraints can be imposed to ensure physical realizability, stability, or other desired properties of the identified model.

Optimization programs require solvers to provide a solution. In our case, we choose to use the CVX package, because CVX is useful for solving convex optimization problems with convex objective functions and convex constraints.

To use CVX for system identification and state-space representation, we need to formulate an optimization problem that captures our specific modeling objectives. The problem formulation typically involves defining an objective function and system constraints that must be physically satisfied. In this case, the objective function is to minimize the error between the estimated states and real ones, while the constraints are based on the physical equations in the state space matrix. Adding prior knowledge about the physical relationship between the elements in the matrices leads to more realistic and meaningful models.

### 2.2.2 Training dataset

We used two different datasets to train our model. One for the identification at lower frequencies, a square wave voltage input, and one for higher frequencies, a chirp signal from 0.1 Hz to 30 Hz. Doing so, we ended up having two state space models that we merged by simply taking the average of the two state space matrices. One other possible approach would have been feeding the two datasets together and optimizing for both at the same time.

### 2.2.3 The obtained model

Here are the numerical values for the state space matrices estimated from the optimization. This state space model is in discrete time where $\Delta = 0.002$ seconds.

$$A = \begin{bmatrix} 1 & 0.002 & 0 & 0 \\ 0 & 0.9283 & 1.162 & 0.0017 \\ 0 & 0 & 1 & 0.002 \\ 0 & 0.07169 & -1.931 & 0.9972 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0.1066 \\ 0 \\ -0.1066 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

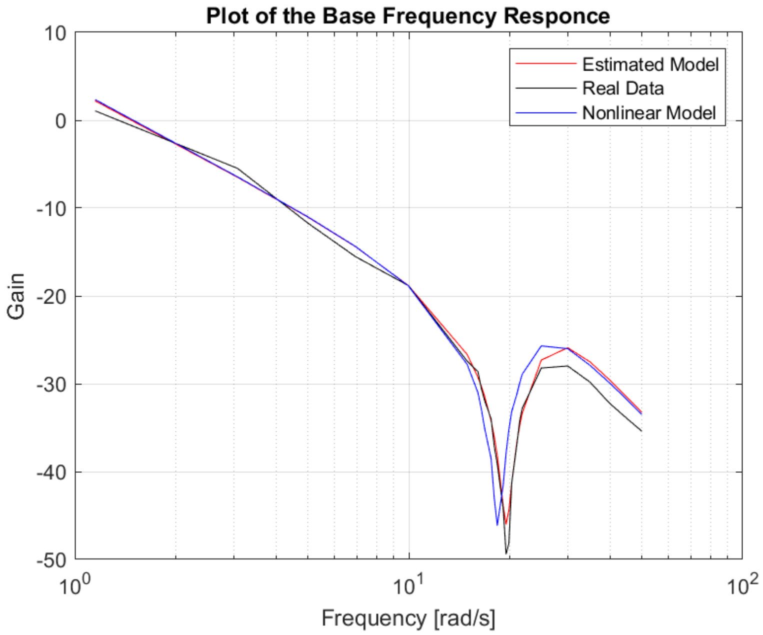
$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

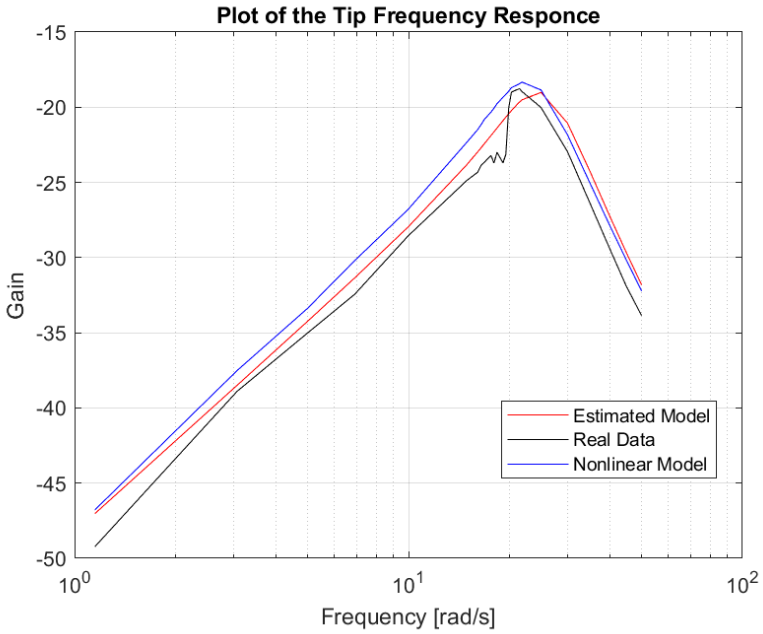
**Algorithm 1** Optimization Program

```
load $\theta$, $\alpha$ from real model
Estimate $\dot{\theta}$ and $\dot{\alpha}$ by Euler and smooth by gaussian filter
Define regressor with prediction error method as:
    $z(k) = \begin{bmatrix} \theta & \dot{\theta} & \alpha & \dot{\alpha} \end{bmatrix}$
    $\phi = \begin{bmatrix} z(k) & 0..0 & 0..0 & 0..0 & u(k) & 0 & 0 & 0 \\ 0..0 & z(k) & 0..0 & 0..0 & 0 & u(k) & 0 & 0 \\ 0..0 & 0..0 & z(k) & 0...0 & 0 & 0 & u(k) & 0 \\ 0..0 & 0..0 & 0..0 & z(k) & 0 & 0 & 0 & u(k) \end{bmatrix}$
    $Y(k) = z(k+1)$
optimization variables are created as follows:
$\hat{\theta}$ = [a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 ...
        a33 a34 a41 a42 a43 a44 b1 b2 b3 b4]
Set constraints according to known state space model

    a11 = 1                          a33 = 1
    a12 = 0.002                      a34 = 0.002
    a13 = 0                          a41 = 0
    a14 = 0                          a42 related with a22
    a21 = 1                          a43 related with a23
    a22 = 1-a42                      a44 = is free
    a23 = -(a23)*constant1           b1 = 0
    a24 = -(a44-1)*constant1         b2 = -b4
    a31 = 0                          b3 = 0
    a32 = 0                          b4 is related to b2

    constant1 = 1/((Jeq+JL)/(JL*Jeq))/Jeq
minimize (Y-$\phi * \hat{\theta}$) with given constraints
```

### 2.2.4 Validation

The estimated system from the previous part needs to be validated in the frequency domain, in this way the overall behavior of the system can be validated. On top of the estimated model, the non-linear model is also included for comparison. Also, the resonance frequency can be observed from the measurements.





### 2.2.5 Parameters estimation

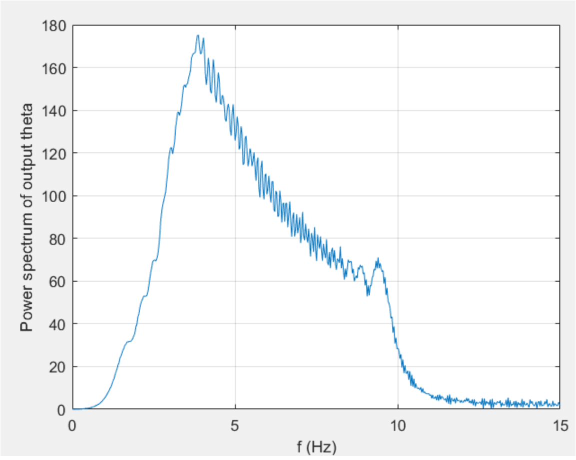
Knowing that, the shape of the matrix is the one expressed in section 2.1.3 and knowing the following relationships for the natural frequency and the damping coefficient of the system:

$$\begin{cases} \omega_n^2 = \frac{K_s}{J_{eq}} \\ 2\zeta\omega_n = \frac{B_L}{J_{eq}} \end{cases}$$

We obtain:

$$\begin{cases} \omega_n = 24.1 \, \frac{rad}{s} \\ \zeta = 0.0176 \end{cases}$$

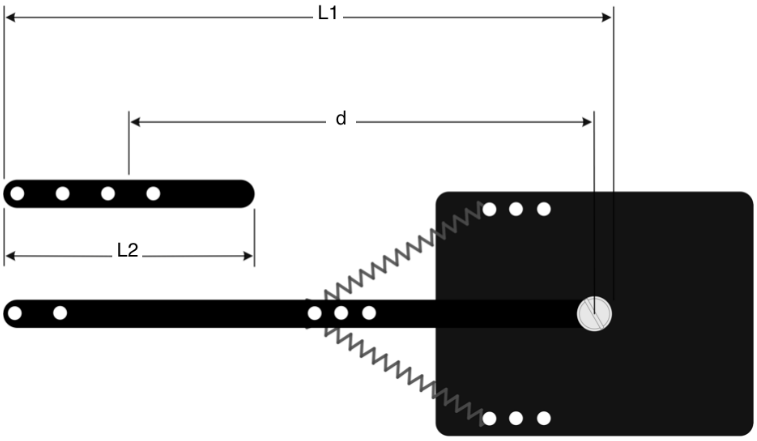
To validate our results we probed the system with a chirp signal with frequency varying from 1 Hz to 10 Hz and we extracted the Fourier transform of the tip position.



In the plot above we can see a peak in the amplitude at the resonance frequency $f_n = 3.846\,Hz$ which gives us $\omega_n = 24.27\frac{rad}{s}$, almost the same we got with the optimization. This means that our model is able to describe the resonance of the actual system.

For the parameters of the arm we can compute the values of the inertia, by knowing its geometry:

$$J_L = m_1 \cdot \frac{L_1^2}{3} + m_2 \cdot \frac{L_2^2}{12} + m_2 \cdot d^2 = 0.0032 Kgm^2$$



Knowing the inertia of the arm and using the experimental value of the resonance frequency we can find the equivalent torsional stiffness as follows:

$$K_s = J_L \cdot \omega_n^2 = 1.8426\,Nm$$

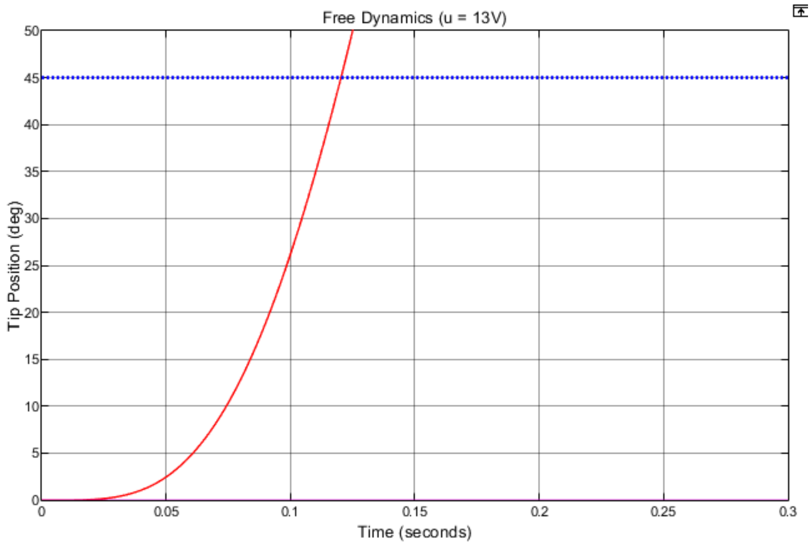
The value of the friction coefficient is supposed to be negligible:

$$B_L = 0$$

### 2.2.6 Control limits

By applying a control scheme to our system we can make it track a set point and also increase its performances in terms of speed. The speed we can achieve is limited by the maximum voltage we can provide to the motor, which in our case is $v_{max} = 15\,V$. Just to be on the safe side and provide margin for control purposes, we assume the maximum effective voltage to be $v_{crit} = 13\,V$.

By letting the system run freely at $v_{crit}$ we can see how long the tip takes to reach 45°, which is the test case that will be used throughout the controller design processes.



As we can see from the plot above, the tip reaches 45° in less than 0.15 s, if we provide the system a constant value of $v_{crit}$ voltage. Knowing this, we can find that the critical frequency of the system is:

$$\omega_{crit} = \frac{2\pi}{0.15}\,\frac{rad}{s} = 41.9\,\frac{rad}{s}$$

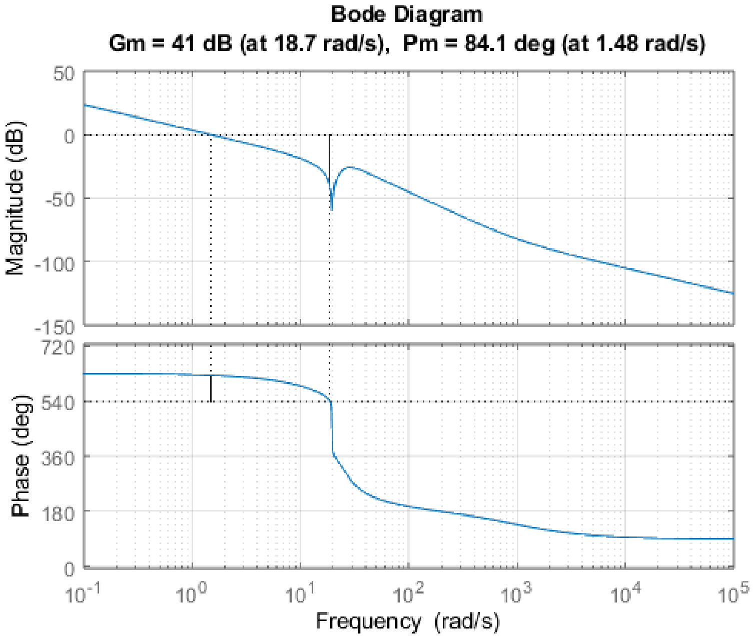
For this reason $\omega_{crit}$ will be the upper bound of the achievable bandwidth in our control schemes.

# Position Control of the Base

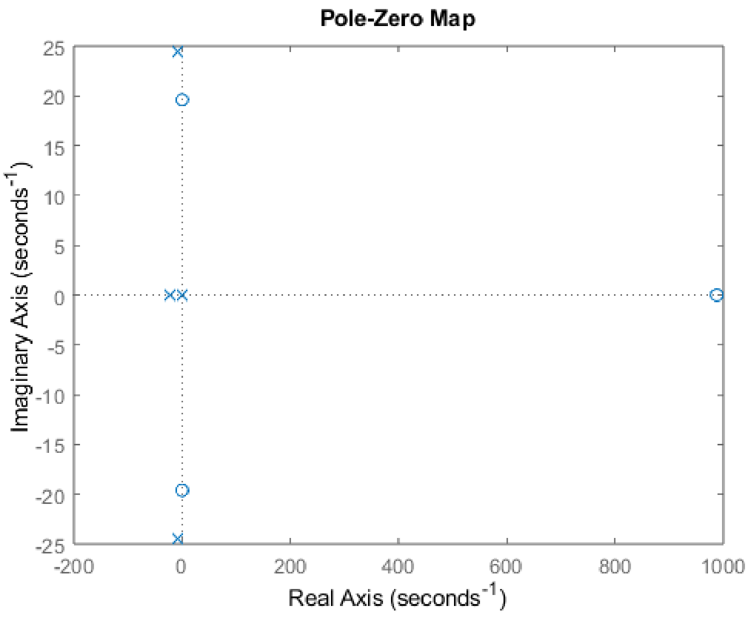
Our main objective regarding the control of the base is to achieve a set point tracking as fast as physically possible, while maintaining a phase margin $\phi_m \approx 60°$ to ensure robustness.

## 3.1 Controller Design

To do so we designed a controller based on the Bode diagram of the model of the system in open loop, represented in the image below:



By looking at the pole-zero map below we can see that the dominant poles are the complex conjugate ones, which must be cancelled in order to speed up the response of the system.



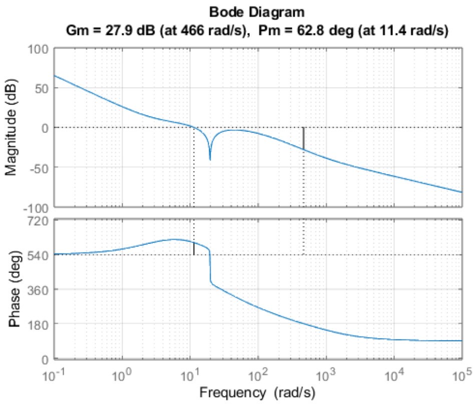
Even though the model itself presents a pole in the origin, an integral action in the controller is still needed in order to grant a zero steady state error, but doing so we drastically reduce the phase and make the system unstable. To fix this issue we added two poles right before the anti resonance in order to get the desired phase margin and, for feasibility reasons, two poles right after the resonance which also have the function to lower the peak at $\approx 30\frac{rad}{s}$.

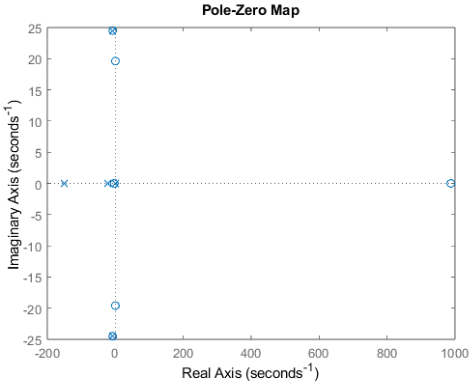
Finally the overall gain of the controller is tweaked such that the crossing frequency is right before the anti-resonance, to get as much bandwidth as possible.

The controller transfer function is thus:

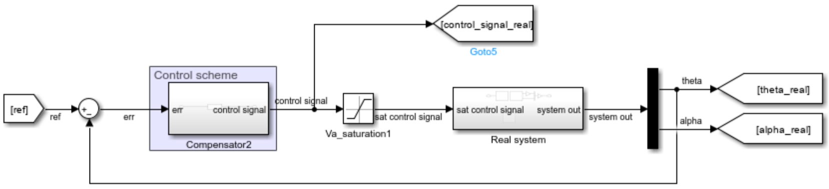
$$C = 150\frac{(s + 7.43 - 24.5i)(s + 7.43 + 24.5i)(s + 3)^2}{s(s + 22)^2(s + 150)} \tag{3.1}$$

And the Bode diagram of the controlled system in open loop and its pole-zero map are respectively:





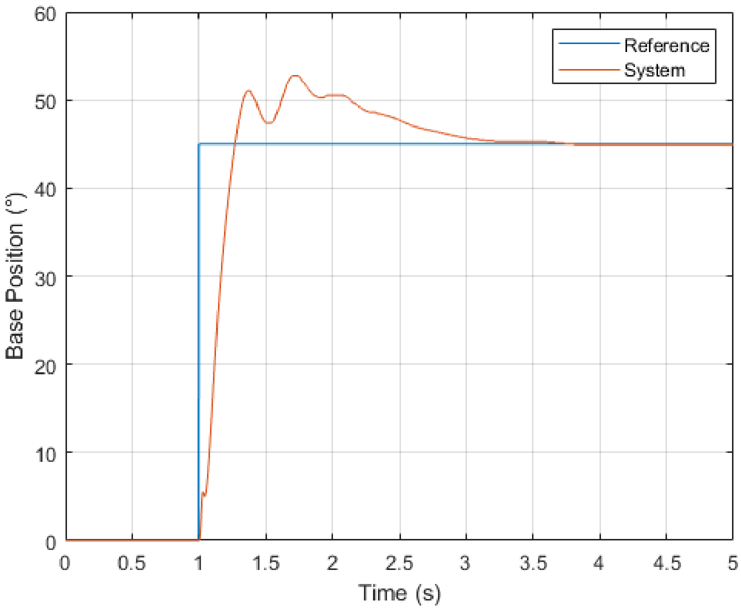
The overall control scheme is then:



## 3.2 Validation

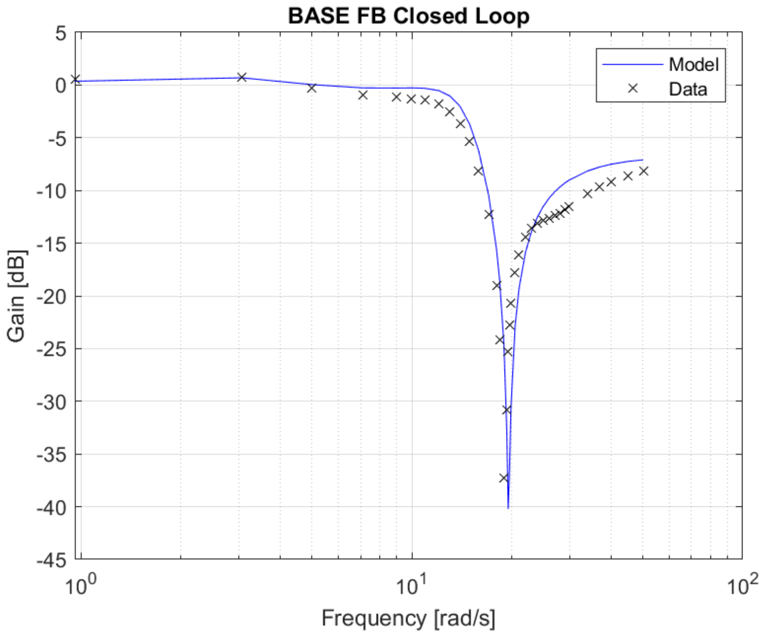
Now that we have developed a control scheme, we have to verify that the closed loop system behaves accordingly to the specification imposed.

### 3.2.1 Step response



### 3.2.2 Frequency validation

By testing the system with sinusoidal references at fixed amplitudes and different frequencies we were able to reconstruct a Bode diagram for the actual closed loop system. The comparison between the experimental one and the theoretical one can be seen in the plot below:



As we can see, at low frequencies the system is able to track the reference with small amplitude reduction up until $\approx 10\,\frac{rad}{s}$ right before the anti-resonance, as expected.

# Position Control of the Tip

In order to control the position of the tip we tried two different approaches:

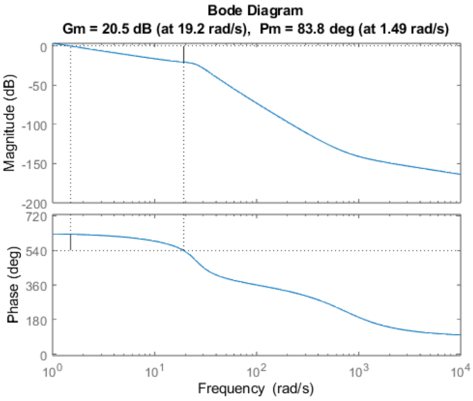
- frequency-based, as we did for the control of the position of the base;
- full-state feedback control scheme, following two methods and comparing which one has the better performance

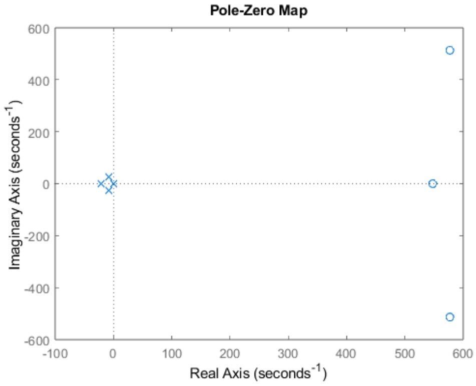
## 4.1 Frequency Based Approach

Our goal is once again to track a set point for the position of the tip granting a phase margin $\approx 60°$ for robustness. This time we have a trade-off between speed of the controlled system and overshoot of the $\alpha$ angle and we decided to focus more on the latter option.

### 4.1.1 Controller design

First of all we can have a look at the Bode diagram of the transfer function from the control signal to the tip position and its pole-zero map:



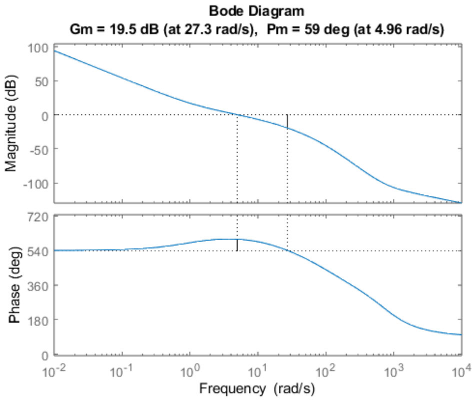


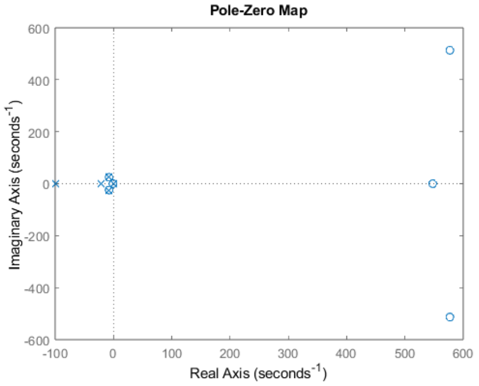
As we did for the base, we cancelled the two complex conjugate poles to reduce the oscillations in the controlled system and added an integrator to grant zero steady state error. This time there are no limitation regarding the speed of the controller due to the absence of the anti-resonance, so theoretically we could speed up the system up to $\approx \omega_{crit}$. As our main goal was to reduce the overshoot of the tip, we opted for a slower, more conservative design.

$$C = 51.28\frac{(s+7.43-24.5i)(s+7.43+24.5i)(s+1)}{s(s+100)^2}$$

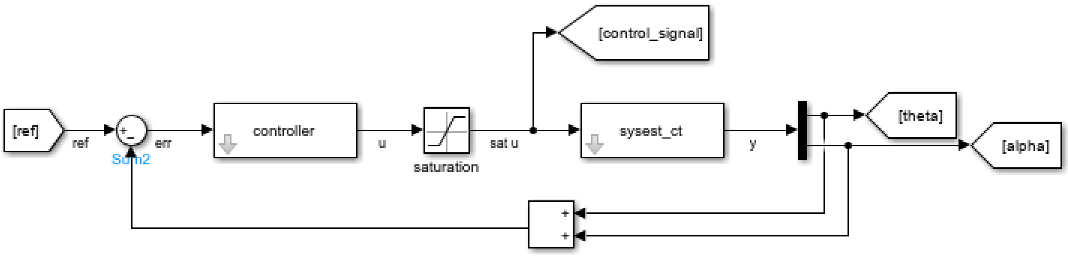
We introduced a zero just before the crossing frequency, like we did for the controller of the base, to boost the phase margin, in order to meet our specifications, whereas the two high frequency poles were added for feasibility reasons. The gain was found in such a way to get the maximum possible phase margin and a bandwidth in the order of $\approx 5\,\frac{rad}{s}$, which is a reasonable trade-off between speed and tip overshoot.

Here we can see the Bode diagram and the pole-zero map of the open loop controlled system.





The overall scheme is then:

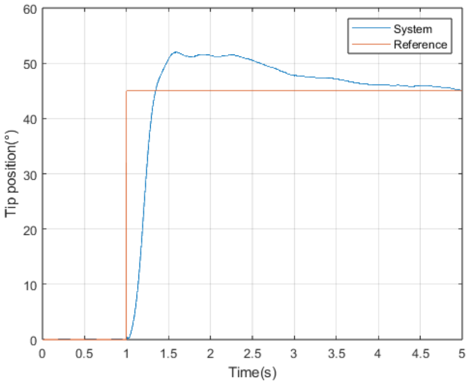


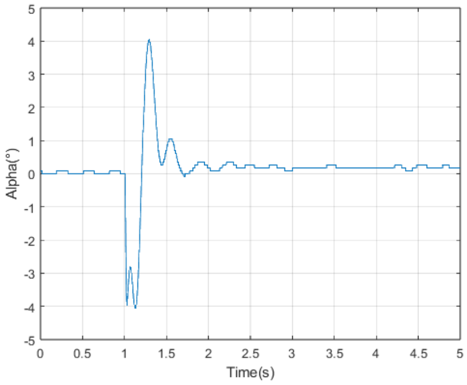
## 4.1.2 Validation

### Step response

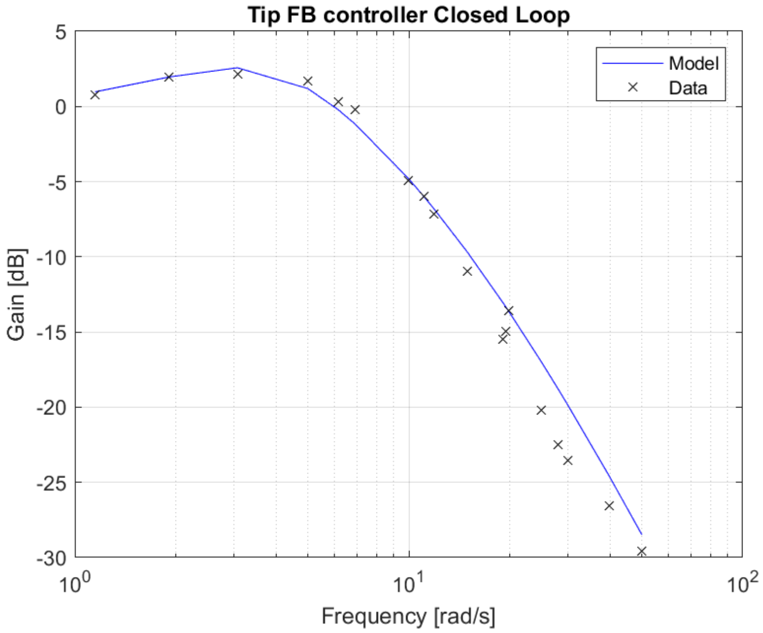
Below we can see on the left the step response of the controlled system to a 45° reference signal: the tip position quickly increases, the rising time is less than 0.5s, then the integral part of the controller lets the system reach the reference signal in less than 4s.

On the right, instead, we can see the measurements of the $\alpha$ angle, the angle between the arm and the base, which always remains below 4°, which satisfies the assumptions we made regarding the linearization of the system.





**Frequency validation**



Finally, once again, we can see how the Bode diagram of the physical system in closed loop matches the expected behaviour.

## 4.2 State Space Based Approach

To improve the performances of our controlled system we adopted a full state feedback control scheme to able to better change the dynamics of the overall SIMO system.

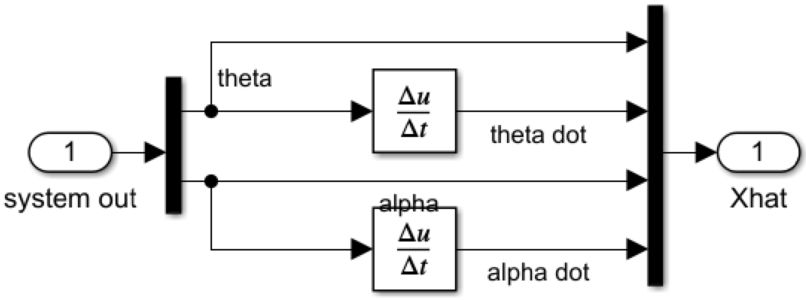
### 4.2.1 State estimators

In order to apply such control scheme we must be able to know or estimate the state of our system knowing only its outputs and inputs. For this reason we developed different kinds of state estimators.

To evaluate the performance of each one of them we need to check how correct they are. For this reason we qualitatively compared the states from the estimators with $\theta$ and $\alpha$ available from sensor measurements and their derivatives, $\dot{\theta}$ and $\dot{\alpha}$, calculated by the central difference method and smoothed with a Gaussian smoothing filter to remove spikes caused by the differentiation.

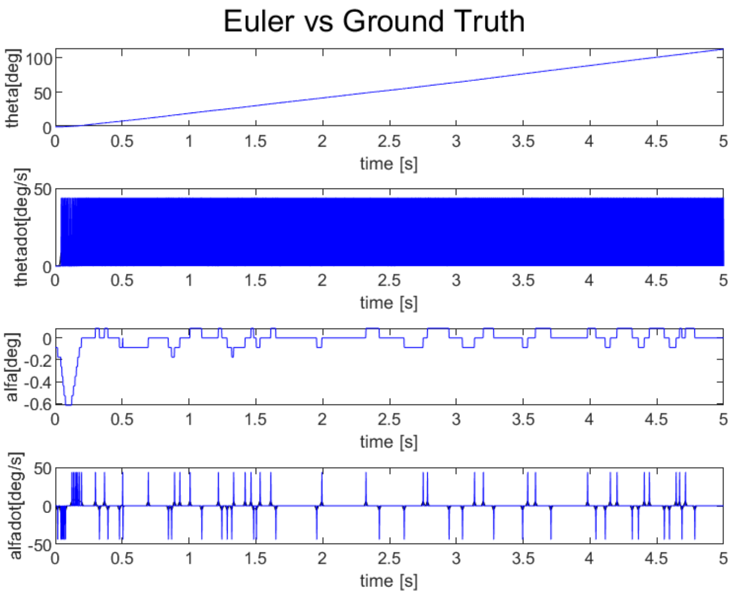
**State extractor**

As our output is the array $\begin{bmatrix}\theta\\ \alpha\end{bmatrix}$ and the full state is $\begin{bmatrix}\theta\\ \dot{\theta}\\ \alpha\\ \dot{\alpha}\end{bmatrix}$, we could just apply a derivative to the output to get the full state, as expressed in the simulink scheme below.



The benefit of such observer is that the estimation of the state is instantaneous and there is no dynamics for the error between the real and the estimated state.

The main drawback is that the derivative is done numerically, so high frequency noises in the measurements could lead to wrong state estimations. This can be clearly seen by looking at $\dot{\theta}$ in the plot below.



### Luenberger observer

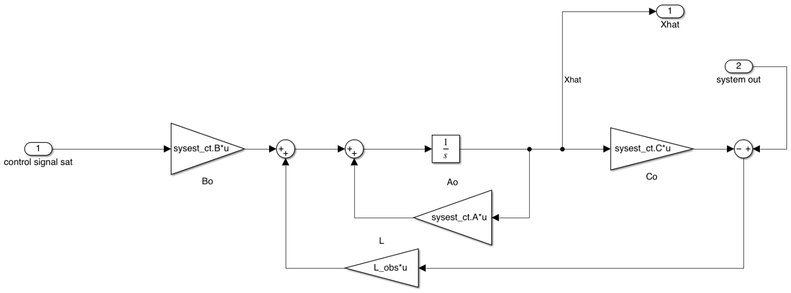
In order to provide an estimation for the states we compute an observer that has the goal to imitate the behavior of the system. We obtained the gain of the observer by imposing the poles of the state error dynamics to be ten times faster than the ones of the pole placement, further information in the control section.

$$s = -230; \qquad s = -250; \qquad s = -270; \qquad s = -290;$$

To compute the static gain of the observer we used the following matlab command:

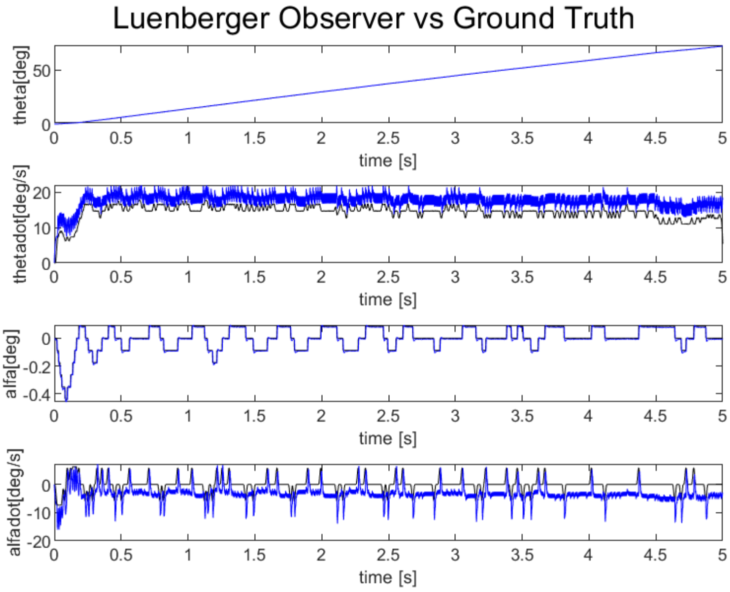
```
L_obs = place(sysest.A',sysest.C', obs_poles)';
```

The Simulink scheme is then:



The dynamics of the observer is not affected by the saturation limits of the control signal, for this reason we were able to place the dominant pole at such high frequency, which grants a fast enough estimation of the states.

The main drawbacks of this estimator are the need of a perfect knowledge of the system and the low reliability in case of unknown noises, disturbances and uncertainties.



### Kalman filter

Lastly we used an optimal observer, namely a Kalman filter, to optimize the estimated state considering noise on the measurements and external disturbances on the output.

Measurement noise refers to the inherent uncertainty or errors in the measured output of a system. It captures the discrepancy between the actual output of the system and the output predicted by the model.

This kind of noise can arise due to various factors such as measurement inaccuracies, sensor noise, or unmodeled dynamics. In our case, our sensors are positional encoders with high resolution and high sampling time. This leads to pretty good measurements with low variance.

Resolution of encoder: $\frac{2\pi}{4096} = 0.0015$
Variance of measurement: $0.0015^2 = 2.25 \times 10^{-6}$

$$R = \begin{bmatrix} 2.25 \times 10^{-6} \end{bmatrix}$$

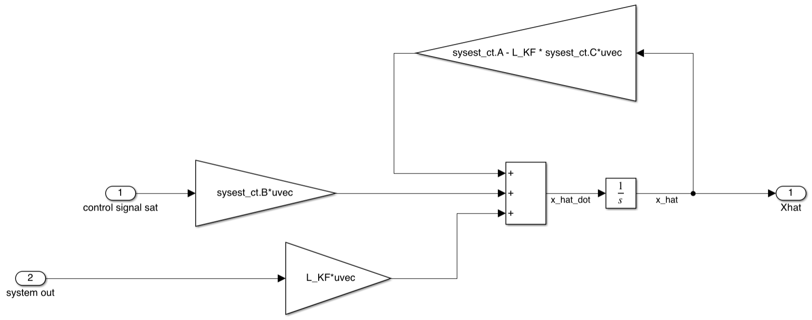
We can assume the same variance on the state as well, by supposing the state to be measured with the same encoders:

$$Q = 2.25 \times 10^{-6} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

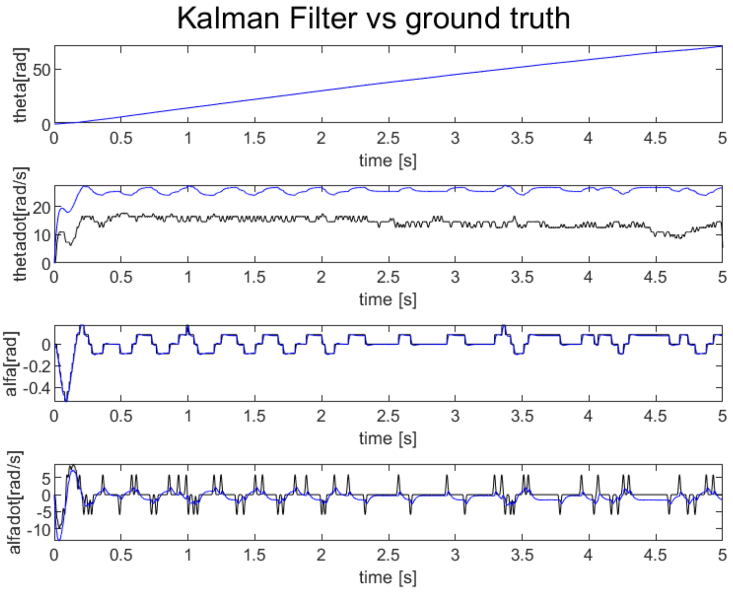
For the computation of the Kalman filter's static gain we used the command:

```
L_KF = lqr(sysest.A.’, sysest.C.’, Q_KF, R_KF).’
```

The Simulink Scheme:



As we saw, the Kalman filter is an optimal observer, which means that it’s able to estimate the states even when the system is subjected to disturbances, but it relies on Gaussian noises with known average and variance and also a good knowledge of the system to estimate. Theoretically it is more precise than the Luenberger observer, but also more computationally heavy.



## Comparison

To sum everything up, here is the numerical comparison of the three different estimators. The ground truth of states for each time step is $x_{gt}$, each estimator is represented as $x_{estimator}$. N is the number of data samples in the validation dataset. Units of the states are in degrees and degrees per seconds.

$$error = \frac{1}{N}\sum_{k=1}^{N} |x_{gt}(k) - x_{estimator}(k)|$$

Table 4.1. Mean absolute error between Ground truth vs Estimators

| | |
|---|---|
| State extractor | 5.8855 |
| Kalman filter | 3.1898 |
| Luenberger observer | 1.7703 |

The state extractor has the largest error due to derivative operation which causes sudden spikes in the estimation. On the other hand, the Kalman filter has smoother behavior but has a shift in the estimation of $\dot{\theta}$.

## 4.3 Pole Placement

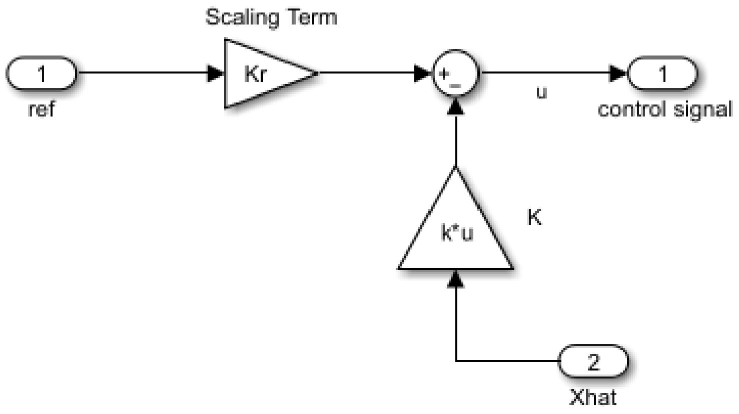
Pole placement is a control technique used to design a controller that can shape the response of a system by placing its poles at desired locations in the complex plane. The control law of the pole placement controller is:

$$u = -K_{PP}x$$

The controller gain matrix $K_{PP}$ can be calculated with the Ackermann formula by specifying the desired pole locations in the complex plane. The overall state of the system, or its estimation, is needed for the implementation of this control scheme.

### 4.3.1 Controller design

The control scheme is the following:



In order to remove the oscillations in the controlled system we decided to place all the poles on the real axis. We wanted to reduce as much as possible the overshoot of the tip with respect to the base both for design choices and also to grant the validity of our linear model. By doing so we chose to place the dominant pole at $\omega_{dom} = 23 \, \frac{rad}{s}$ and the other poles at higher frequencies.

$$Poles_{PP} = [-23, -25, -27, -29]$$

In this way the controlled system is faster than the uncontrolled one, but is well below the limit introduced by $\omega_{crit}$ (obtained in section 2.2.6); in fact we could make the system faster, but doing so the overshoot was no more negligible, for this reason we preferred a more conservative controller.

Using Ackermann's formula we got the following gain:

$$K_{PP} = [21.1801, 2.6433, -34.8132, 1.3692]$$

When using a pole placement controller, it is sometimes necessary to introduce a scaling term to the reference value in order to reach it effectively. This scaling term ensures that the control control input generated by the controller matches the magnitude and dynamics of the reference signal.

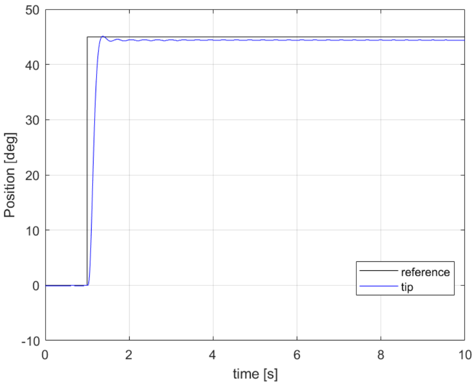
$$\begin{cases} K_{dc} = D + CA^{-1}B \\ K_r = \frac{1}{K_{dc}(1)} \end{cases}$$

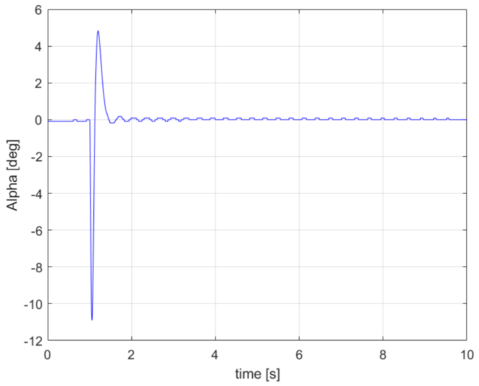
Using the continuous-time DC gain equation above and inverting it to get the scaling term, we got $K_r = 21.1801$.

### 4.3.2 Validation

**Step response**

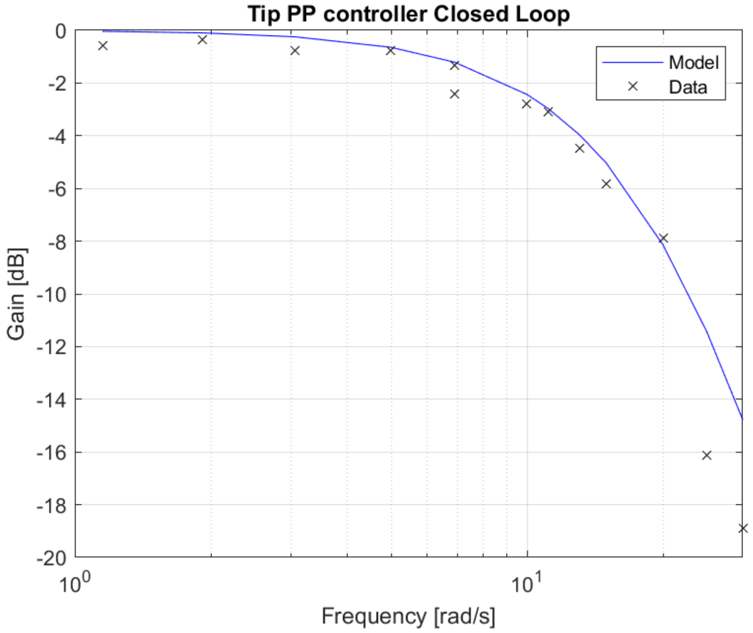
As we can see, the system is able to perfectly reach the reference with negligible oscillations and without overshooting, also the $\alpha$ angle always remains inside the linearization limits.





**Frequency validation**

By testing the closed loop system at different sinusoidal input we were able to reconstruct the Bode plot of the system with a pole placement controller.



As can be seen, the model simulates very well the results obtained at low frequency in the real system.

## 4.4 Linear Quadratic Regulator

The LQ regulator is an optimal full state feedback control scheme which has as control law:

$$u = -K_{LQR}x$$

Where $K_{LQR}$ is found by solving this minimization problem:

$$\min_{u(\cdot)} \int_0^{t_1} \left( x^T Q x + u^T R u \right) dt$$

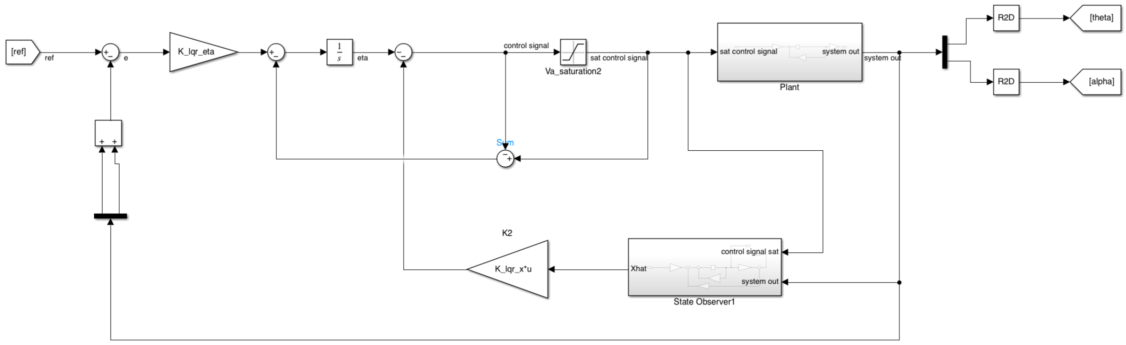
subject to $\dot{x} = Ax + Bu$

Where Q and R are matrices which represent respectively the weight on the state and on the inputs.

### 4.4.1 Controller design

We implemented a Linear Quadratic Regulator with an integral action in an outer feedback loop, in this way the LQ will provide stability and the integral action will provide a zero steady state error between the tip position and the reference.

The control scheme is the following:



To mitigate possible issue related to the saturation limit and the integral action of the enlarged system we also implemented an anti wind-up configuration.

In order to speed up the response of the system we designed the LQ controller by prescribing a convergence rate faster than an exponential:

$$y = e^{-\tau t}$$

To do so we added a weight on the A matrix as follows:

$$\tilde{A} = A + \tau I$$

In this way we imposed the real part of all the poles of the controller to be smaller than $-\tau$.

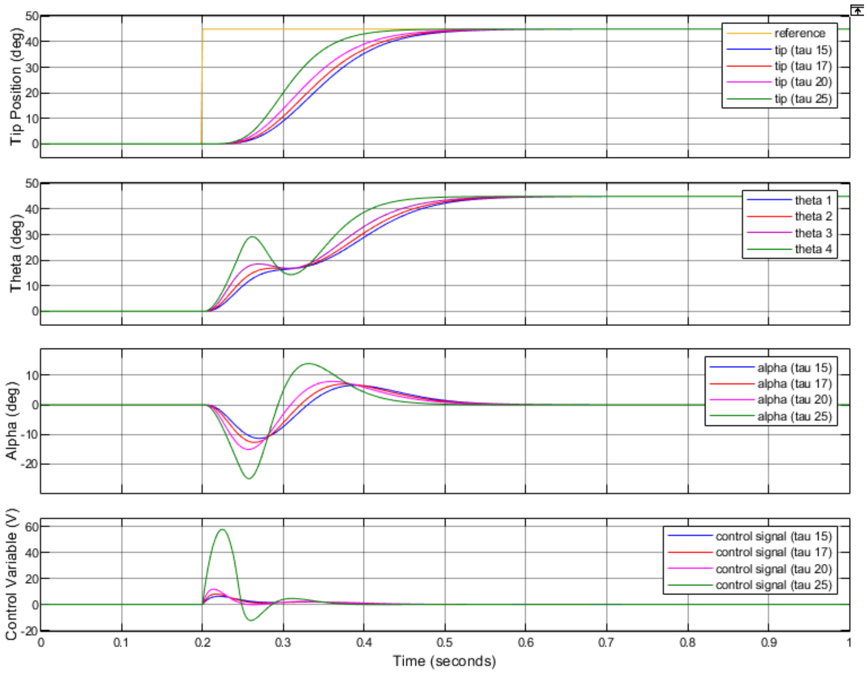
At first we defined the Q and R matrices as identity matrices and tested different values of $\tau$ to see how they would impact the response of the overall system.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = 1$$

These results are obtained with a step response for the tip of 45°:

- $\alpha$ overshoot [%] ($\alpha$ OS);
- Tip overshoot [%](Tip OS);
- Settling Time [s](ST);
- Rising Time [s](RT);
- Max Control Variable [V](Max V);
- Integral of the Control Variable [Vs](I V);

| $\tau$ | $\alpha$ OS | Tip OS | ST | RT | Max V | I V |
|---|---|---|---|---|---|---|
| 15 | 11.405 | 0.000 | 0.732 | 0.158 | 6.315 | 0.528 |
| 17 | 12.725 | 0.000 | 0.611 | 0.152 | 8.050 | 0.528 |
| 20 | 15.268 | 0.000 | 0.455 | 0.144 | 11.905 | 0.537 |
| 25 | 25.194 | 0.000 | 0.301 | 0.108 | 59.267 | 2.432 |



In the plot above we can clearly see how increasing $\tau$ we can have a faster response, but also the control signal required increases as well, going over the saturation limits.

By choosing $\tau = 20$ we can mitigate the trade-off between fast response and saturation limits.

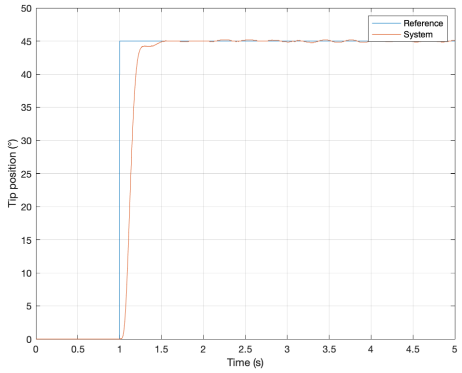
For this value of $\tau$ we noticed that changing the weights on the main diagonal of the Q matrix doesn't have an impact on the performances of the controlled system, so we kept them as identity matrices.

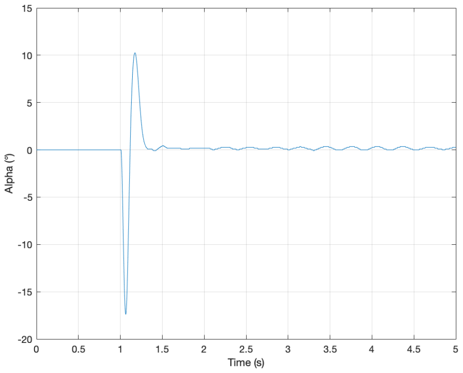
Their contribution start to be relevant for lower values of $\tau$, but as the response of the system was too slow in that configuration we decided to keep the value of $\tau = 20$.

## 4.4.2 Validation

### Step response

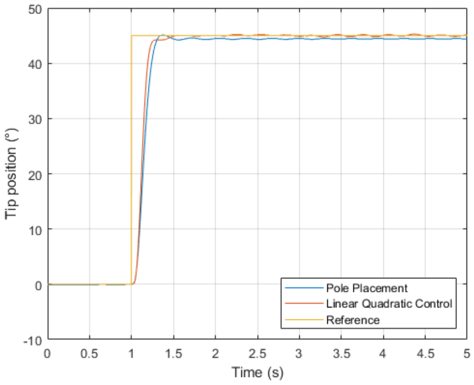
Below we can see on the left the step response: the rising time is less than 0.13s and the settling time is slightly more than 0.5 seconds. On the right, instead, we can see the measurements of the $\alpha$ angle which has a couple of big peak, the highest of 17°, which still satisfies the assumptions we made regarding the linearization of the system.

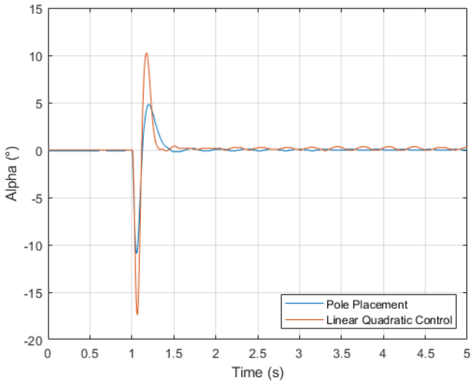




## 4.5 Comparison

To decide which controller to use, we compared their performances with a step response.





The main features are:

| Method | $\alpha$ OS | Tip OS | ST | RT | Max V | I V |
|---|---|---|---|---|---|---|
| PP | 11.405 | 0.000 | 0.791 | 0.172 | 16.298 | 0.977 |
| LQR | 15.268 | 0.000 | 0.455 | 0.144 | 11.905 | 0.537 |

Both the control schemes have a settling time lower than one second and a similar rising time, however we prefer the performances of the pole placement controller because the $\alpha$ overshoot is less.

# System with uncertainties

In the last part of our analysis we developed different control schemes to control the position of the tip when the arm had different values of inertia. As we already developed a model for the system in the nominal condition, we can design a control which considers the uncertainties of the parameters in the model and is still able to grant similar performances, and most importantly stability, for all the different system conditions.

In details we are changing the position of the tip of the arm and its value of inertia as expressed in the table below. The inertia is calculated in the same way as in paragraph 2.2.5.

| d[cm] | $J_L[Kgm^2]$ |
|---|---|
| 20 | 0.0032 |
| 37 | 0.0062 |

Whereas below we can see the A matrices in continuous time in both cases:

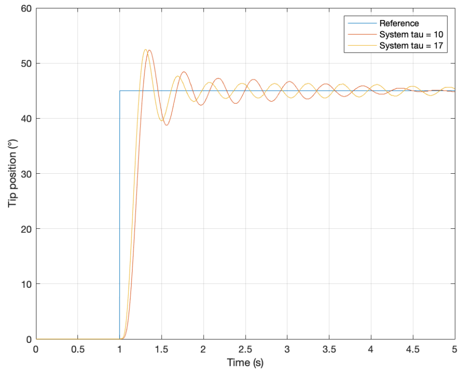
$$A_{nominal} = \begin{bmatrix} 0 & 1.0377 & -0.6104 & -7.4061e-5 \\ 0 & -37.199 & 603.06 & 0.2717 \\ 0 & -0.0377 & 0.9948 & 1.0001 \\ 0 & 37.201 & -987.46 & -0.4496 \end{bmatrix}$$

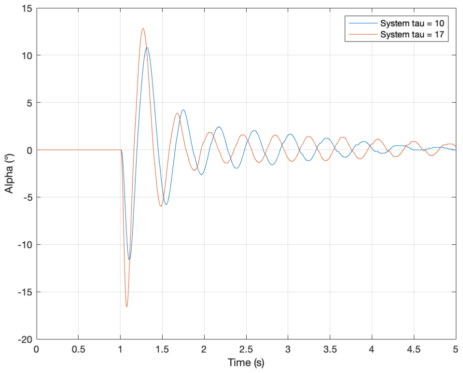
$$A_{maxInertia} = \begin{bmatrix} 0 & 1.0377 & -0.6104 & -7.4053e-5 \\ 0 & -37.199 & 603.05 & 0.2717 \\ 0 & -0.0377 & 0.9270 & 1.0001 \\ 0 & 37.201 & -919.71 & -0.4183 \end{bmatrix}$$

## 5.1 Robustness of State Space Control

Knowing that the LQR control has robustness properties, we tried to apply the same control scheme we developed before to the system with different inertia parameters, as it is more reliable than the pole placement controller. As the observer relies on a correct model of the system, we decided to slow down the controller to better ensure stability of the closed loop system.

In the plots below we can see the step response of the system with the maximum possible value of inertia in both cases, $\tau = 10$ and $\tau = 17$.





In the controller design we are not including the uncertainties related to the new system, so we cannot guarantee the same performances as in the nominal case, but still the system is able to reach the reference, although with oscillations.

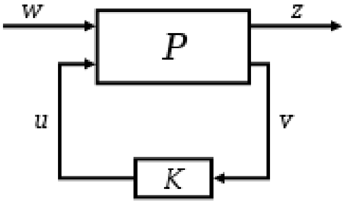
## 5.2 Robust control

If we want to consider uncertainties of the model in our design we have to resort to robust control, which means the offline development of a controller that is able to maintain the same performances of the closed loop system even if the model of the system upon which it is designed is not reliable.

### 5.2.1 $H_\infty$ control

$H_\infty$ control is a type of optimal control scheme which grants robustness if the modeled plant presents uncertainties.

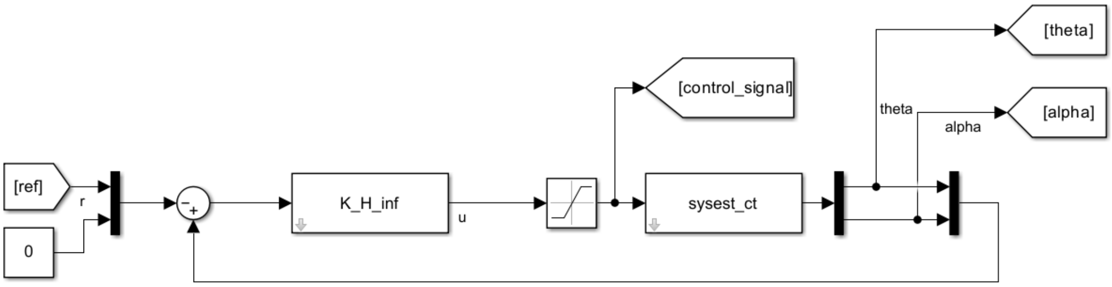
In the image below we can see how the design of the controller is done:



Where w is the array of exogenous inputs, z is the array of optimization variables, u and v are respectively the inputs and outputs of the enlarged system P and K is the $H_\infty$ controller.

In our case w is the reference signal and z is the difference between the reference and the tip position. As there was only one value to optimize, the introduction of weights was not necessary in this case.

The obtained K, which is a 2x1 array of transfer functions, is then implemented as follows:



Up until now we have a somewhat robust controller for our scheme which works perfectly fine on the nominal system, if we want to also take into consideration the information regarding the uncertainties in the model parameters, then we have to resort to $\mu$ synthesis control.

### 5.2.2 $\mu$ synthesis control

This kind of control is a generalization of $H_\infty$, the premises and the control scheme are identical but here we can also define a degree of uncertainty for our system, so after applying the controller, the robustness will be the one prescribed.

By looking at the A matrices for the tip position in the maximum inertia case at the beginning of the chapter, we can see that the elements $A(3,3)$, $A(4,3)$ and $A(4,4)$ differ from the nominal case of $\approx 7\%$.

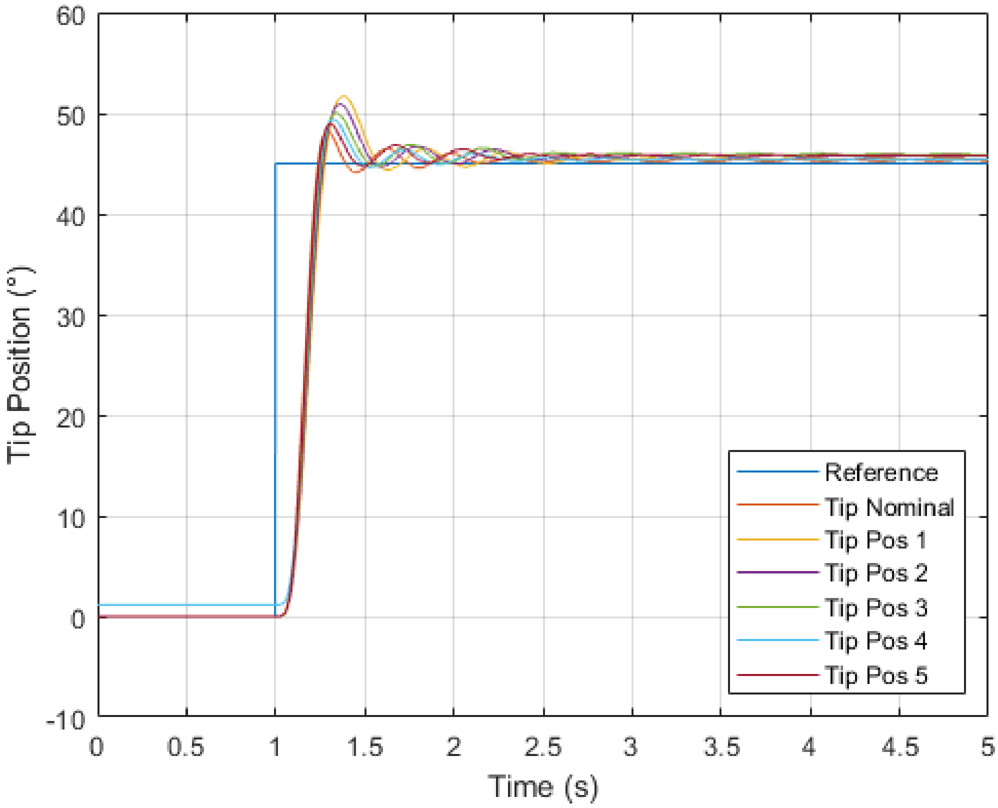
Knowing this information we defined an "uncertain state space" in Matlab having as nominal value the model we found previously and we put an 8% uncertainty on those three parameters inside the A matrix. The $\mu$ synthesis function finds the controller $K_\mu$ by applying the $H_\infty$ algorithm recursively for each value of the parameter inside the specified range and keeping the best one.

The obtained controller is thus:

$$K_\mu = \begin{bmatrix} \frac{-362.8s^3-1.381e4s^2-3.856e5s-5.643e6}{s^4+81.54s^3+3545s^2+7.229e4s+1.329e6} & \frac{-22.8s^3-1019s^2-2.894e4s-4.135e5}{s^4+81.54s^3+3545s^2+7.229e4s+1.329e6} \end{bmatrix}$$

### 5.2.3 Validation

In the plot below we can see the step response of this new control scheme where the different lines are due to the different values of inertia of the arm.



As we can see, by increasing the inertia of the system the overshoot increases as well, but still the system is stable and also the performances are roughly the same, regardless of the uncertainties with the nominal model.

Unfortunately, when collecting data one of the signal was not starting from 0°, as we can see in the plot, but the conclusions we got are not affected by it, in fact it perfectly behaves like the other signals.