Laboratory Report 1: Position state–space control of a DC servomotor

Savian Giulio ; email: giulio.savian@studenti.unipd.it
Group: 8, Shift 2^{nd}
Components: Savian Giulio, Tortora Nicolas, Pavarin Laura, Ambrosin Gioele

June 24, 2021

1 Introduction

1.1 Activity Goal

The purpose of this document is to explain the methodologies and tests performed on the DC servomotor unit with a Simulink model that runs in real–time. In particular, the experience is divided in two main parts:

- In the first part is performed an improvement in the design of a position PID controller in continuous time that, from Laboratory 0, has been carried out in frequency domain. This is face up by introducing the concept of Anti-windup mechanism and Feedforward compensation.
- The second part is based on the state-space model of the DC servomotor unit, which leads to the possibility of implementing a more wide range of control techniques. Techniques like nominal and robust tracking design. Specifically, three different methodologies are exploited for the robust tracking design: integral action, error-space approach and extended-state estimator method

The report is set up as a continuous comparison between the numerical simulation, i.e Simulink models, and the laboratory tests data.

1.2 System and Model

The System is a DC servomotor unit that is consisting of: a Quanser SRV–02 servomotor (plant), which includes a DC motor with a built–in planetary gearbox capable of driving a mechanical load attached to its output shaft (simple disc inertia); an incremental optical encoder that is directly connected to the output shaft of the SRV–02 unit for measuring the position of the mechanical load; a potentiometer is also connected to the shaft through a gear coupling with anti–backlash mechanism; finally, a voltage driver based on a power operational amplifier that will produce the input for the plant u_{drv}.

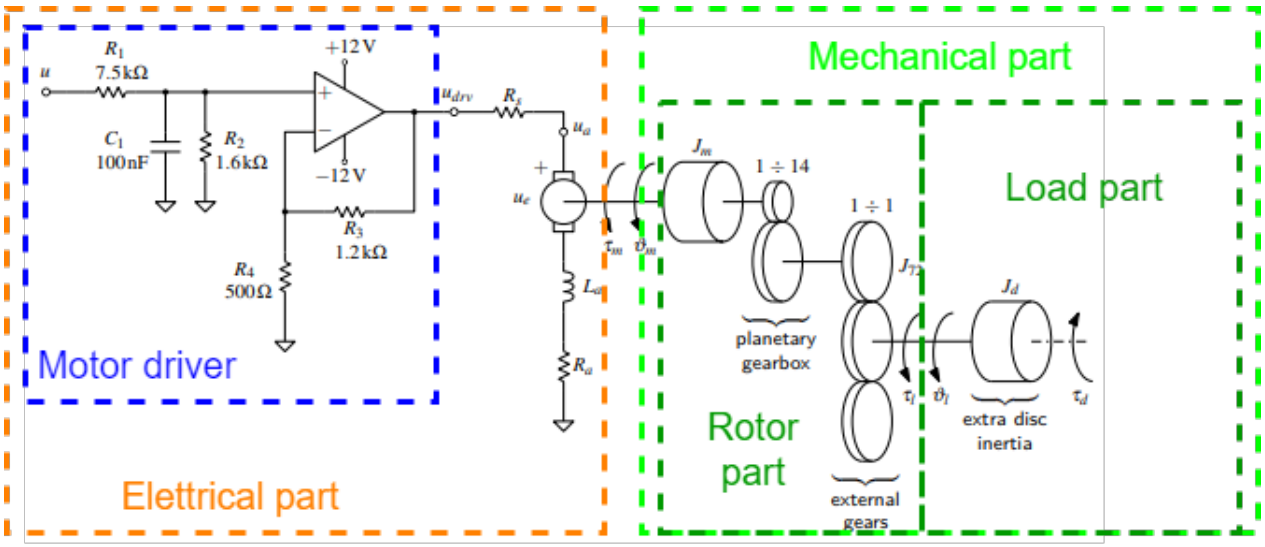


Figure 1: Lumped–element diagram of the system

The system can be schematized as in figure 1, where three parts can be distinguished: the electrical dynamics, the rotor mechanical dynamics and the load mechanical dynamics. By analyzing separately all these components we can obtain the following analytical model of the DC gearmotor with inertial load:

$$\begin{cases} L_a \frac{di_a}{dt} + (R_a + R_s) i_a = u_{drv} - k_e \omega_m \\ J_{eq} \frac{d\omega_m}{dt} + B_{eq} \omega_m = k_t i_a - \frac{1}{N} \tau_d \\ T_{drv} \frac{du_{drv}}{dt} + u_{drv} = k_{drv} - u \end{cases} \tag{1}$$

where $J_{eq} = J_m + J_l/N^2$ and $B_{eq} = B_m + B_l/N^2$. All this data values and meanings here can be find in the appendix A.3.
The figure 2 shows the model diagram of the DC servomotor unit with the formulas 1.

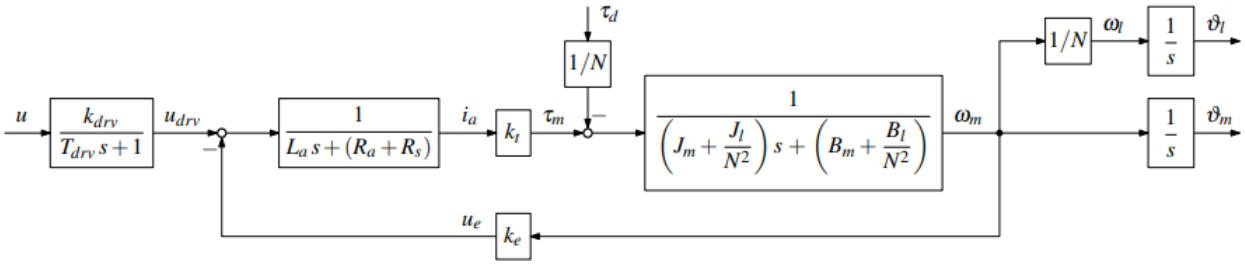


Figure 2: Model of the DC motor + Gearbox

Lastly, the position control system, i.e. the one we have to design to get the control signal $u[V]$, must perform some specifications, such as:
- perfect steady state tracking of step position (load side) references.
- perfect steady state rejection of constant torque disturbances.
- step response (at load side) with settling time $t_{s,5\%} \leqslant 0.15s$ and overshoot $M_p \leqslant 10\%$.

Therefore, to satisfy these constrains a PID controller is used whit $\alpha = T_I/T_D = 6$ obtaining the table 1 of appendix A.1, so the PID gains adopted are:

$$Kp : 8.3738 \ , \quad Ki : 86.1707 \ , \quad Kd : 0.1356 \ . \tag{2}$$

2 Tasks, Methodologies and Results

2.1 Part 1: position PID–control improvements

2.1.1 Anti-windup mechanism

The actuators of a real control system always have constraints on the amplitude of the input and output signals, in fact, a saturation block has been introduced after the PID control, for manage this problem. But, a non linear effect is introduce and negatively affects on the overshoot of the system, as can be seen, through the red line, in the step response of figure 4. The reason for this situation is due to the integrator wind-up. Namely, in presence of saturation it may happen that the actuator output does not increase, although the adjustment error and remains non-zero. It also makes the regulator inactive even when the error decreases or reverses its sign; indeed, the regulation system can only be reactivated when the control signal u falls within the linearity zone.
To avoid this scenario an anti-windup mechanism is used, figure 3, hence the integration is compensated by a signal proportional to the saturation.

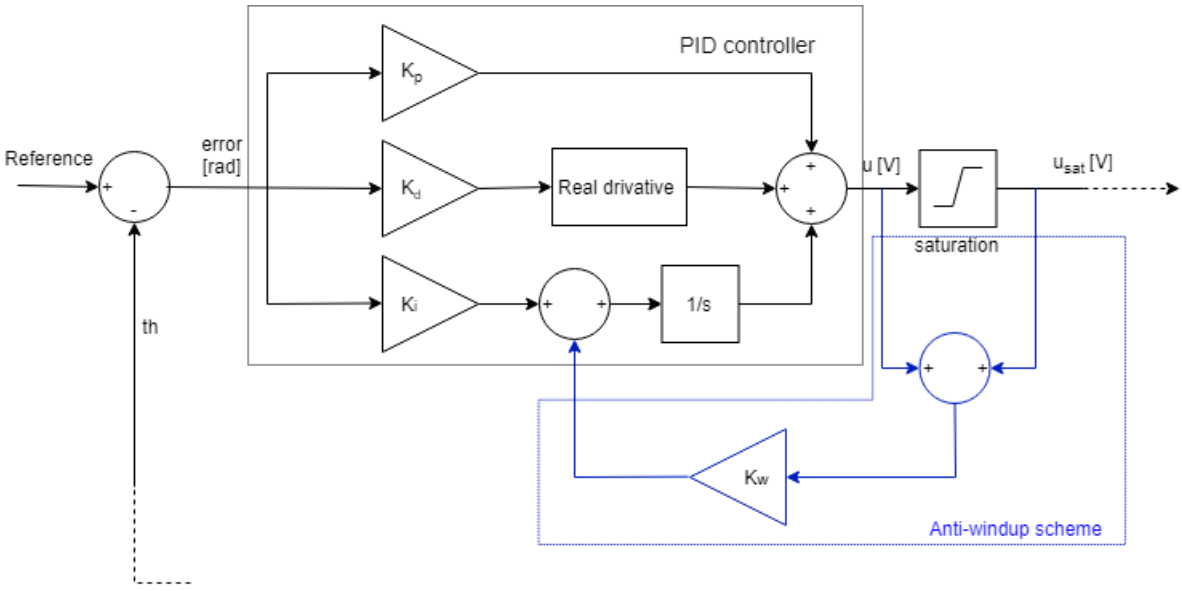


Figure 3: Model of the anti-windup mechanism

The introduction of this technique produces drastic changes on our model, reducing the overshoot and allowing to follow the reference better, as can be seen in figure 4.
The laboratory simulation is very similar to the numerical simulation and both the models with anti-windup satisfy the specification, i.e $M_p <= 396^\circ$ (pink line) and $t_{s,5\%} \leqslant 0.15s$ (green

line).

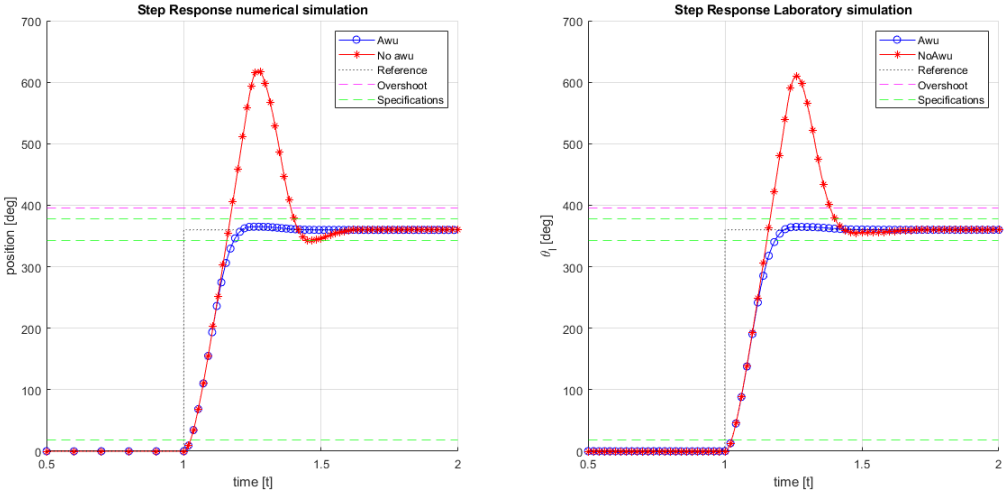


Figure 4: Step Response of amplitude $360°$ with Anti-windup

Obviously, the control signal will change and the saturation affects only the first instant ,figure 5,

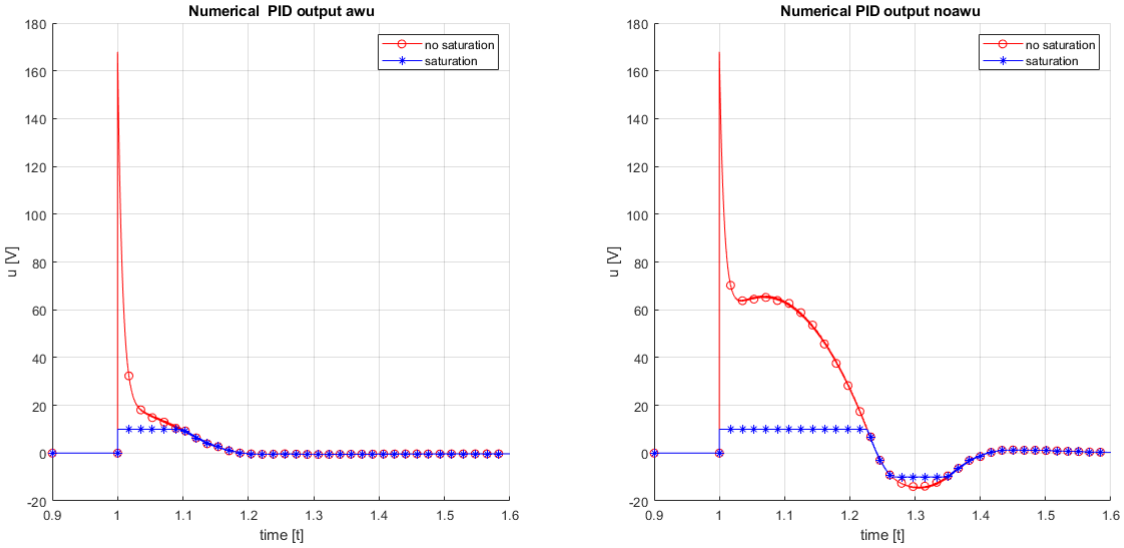


Figure 5: Control input u before and after Anti-windup

The gain K_w also carries out an important role and can't take random values. K_w must be chosen high enough for maintain the integrator input small under several errors. For this purpose Trial & Error method has been adopted and the figure 6 can show as the shape changes as a function of the gain K_w that will takes value $K_w = 33.3$ for subsequent tests.

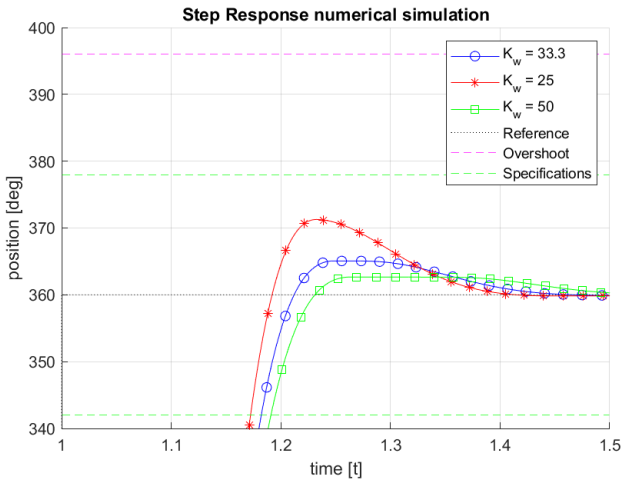


Figure 6: Step Response with different anti-windup gains

2.1.2 Feedforward compensation

As well as anti-windup mechanism, feedforward compensation is used for improve the performance of a PID controller. For instance, we can use it for: pre-filtering the reference signal, feed-forward compensation of the reference signal and cancellation of known input and output disturbance.

To this aim, a feed-forward compensation of the reference signal is used by choosing:

$$u_{FF,1} = \frac{NR_{eq}J_{eq}}{k_{drv}k_t}\frac{d\omega_l^*}{dt} + \frac{N(R_{eq}B_{eq} + k_t k_e)}{k_{drv}k_t}\omega_l^* \tag{3}$$

and the cancellation of known input disturbance is implemented as well, using:

$$u_{FF,2} = \frac{R_{eq}}{k_{drv}k_t N}\tau_{sf} sign(\omega_l^*). \tag{4}$$

Joining formulas (3) and (4) we obtain the compensation of three terms divided, in equation 5, by the sums. Those values are respectively: the inertia compensation, the friction compensation and the back electromotive force (BEMF):

$$u_{FF} = \frac{NR_{eq}J_{eq}}{k_{drv}k_t} + \frac{R_{eq}}{k_{drv}k_t N}\left[N^2 B_{eq}\omega_l^* + \tau_{sf} sign(\omega_l^*)\right] + \frac{Nk_e}{k_{drv}}\omega_l^*. \tag{5}$$

The figure 7 shows an example of a feedforward implementation, where the compensation terms are present.

Without the estimation of the static friction τ_{sf}, viscous friction coefficient B_{eq} and equivalent inertia J_{eq} we can't carry out our implementation. Hence, the values are reported in Table 2 for completeness.

Equivalent Inertia J_{eq}	Static Friction τ_{sf}	Viscous Friction B_{eq}
8.035e-07	0.0071	1.071e-06

Table 2: Parameters estimate from Laboratory 0

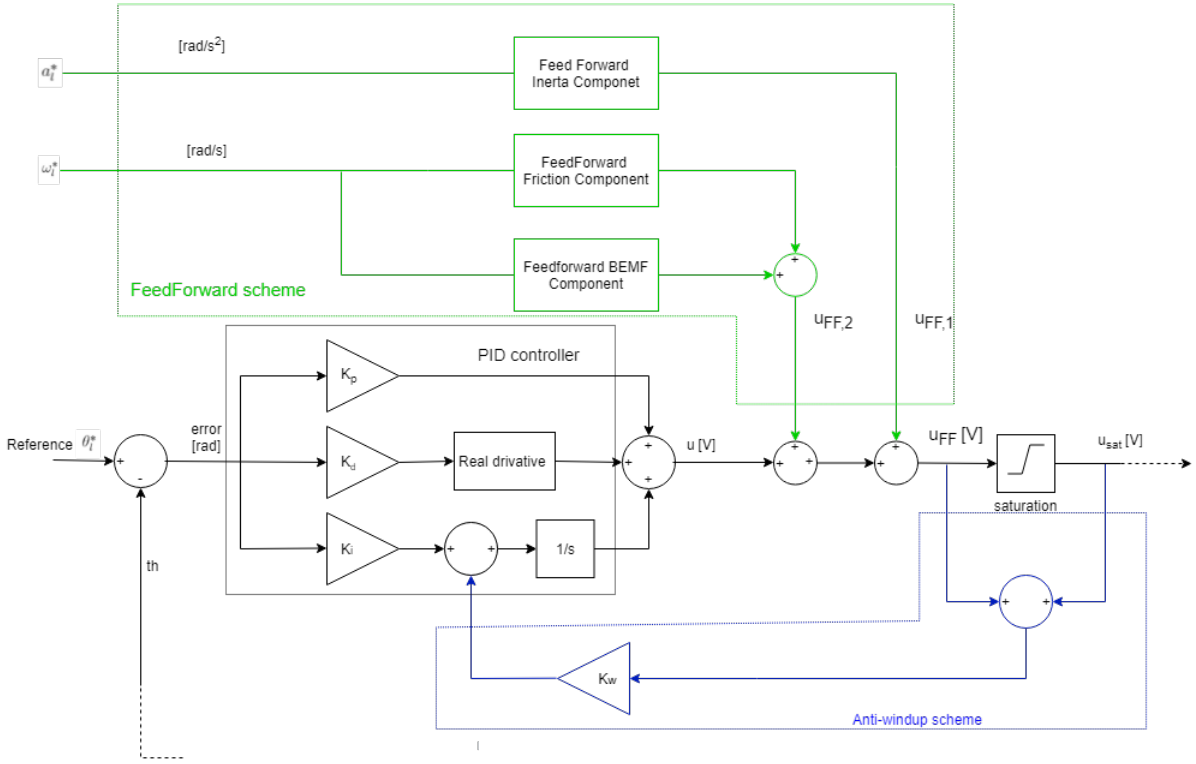


Figure 7: Controller scheme with feedforward and anti-windup

Implementing this scheme in our system allow us, to perfectly track references that with a simple feedback control is not always possible. In our simulation we are considering a periodic acceleration reference, defined ad follows:

$$a_l^* \triangleq \begin{cases} 900rpm/s & if \quad 0s \leqslant t < 0.5s \\ 0rpm/s & if \quad 0.5s \leqslant t < 1s \\ -900rpm/s & if \quad 1s \leqslant t < 1.5s \\ 0rpm/s & if \quad 1.5s \leqslant t < 2s \\ 900rpm/s & if \quad 2s \leqslant t < 2.5s \end{cases} . \tag{6}$$

while the angle θ_l^* and the speed ω_l^* are obtained by the usual integration starting from a_l^*.

In the figure (8) are represented the results of the feedforward compensation. Both the experiments confirm that the feedforward scheme is able to track better the black line, i.e. the reference. Additionally, can be seen from the two acceleration plots that in practise the simulation steel has the same problems in the tracking, since we are in a non ideal situation when different factor can disturb the system or also the estimation of the parameters can be a bit different from the true values.

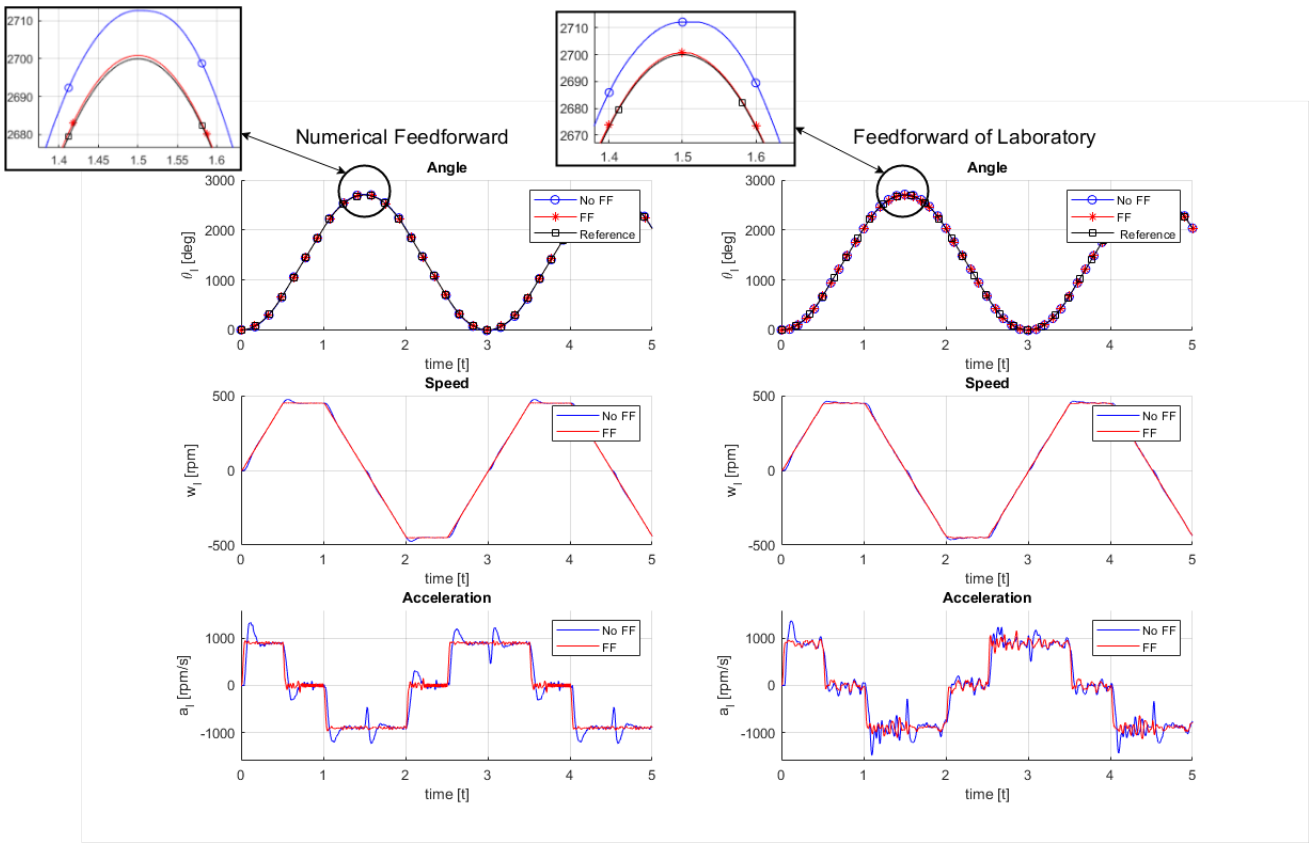


Figure 8: Feedforward compensation results

From the tracking error plot, figure 9 , the improvement can be seen better, in the first graph there is a non linear effect due to the encoder in the numerical simulation, this problem can be solved by simply by considering the output at the end of the plant and use it for compute the error and have a better view of the performance.

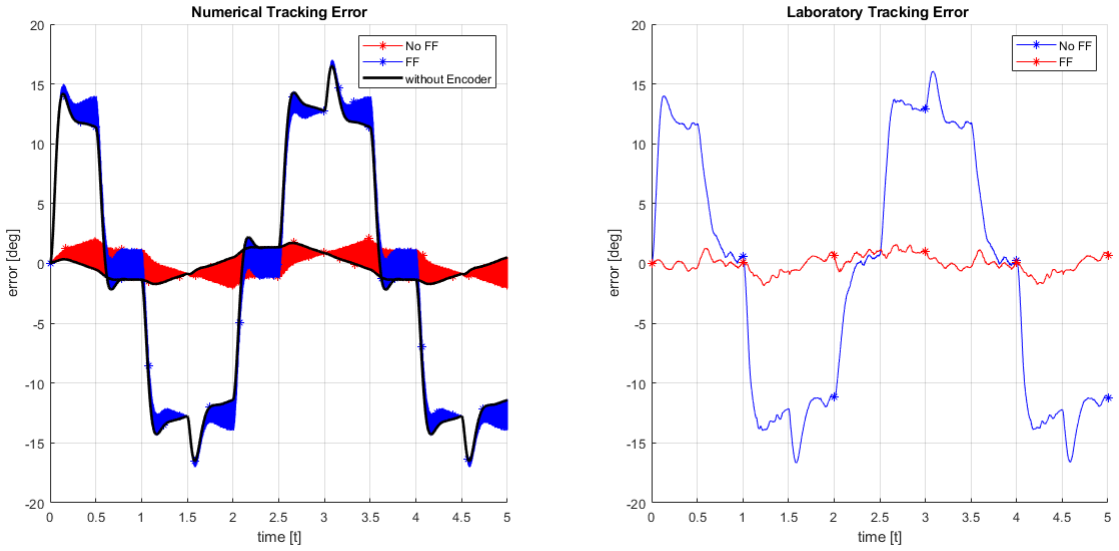


Figure 9: Tracking error with and without feedforward, in numerical and practise simulation

2.2 Part 2: position state–space control design

In this part of the report we will see technique based on the state space model of the DC gearmotor, $\Sigma = (A, B, C, 0)$ with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$ and $C \in \mathbb{R}^{1\times n}$. A possible state-space realization can be computed using the reachable canonical form, as follows:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{R_{eq}B_{eq}+k_t k_e}{R_{eq}J_{eq}} \end{bmatrix} , \qquad B = \begin{bmatrix} 0 \\ \frac{k_{drv}k_t}{NR_{eq}J_{eq}} \end{bmatrix} , \qquad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{7}$$

with state vector $x = [\theta_l, \omega_l]^T$.

2.2.1 Nominal tracking design

The introduction of the state-space of the system allow as to solve typical regulation problem either in nominal condition or in presence of external disturbance, robust tracking. The design is based on a static state feedback where the state $x = [\theta_l, \omega_l]$ is fully accessible. Since the state ω_l is not computed by the encoder, it is estimate through a continuous high-pass-filter:

$$H_1(s) = \frac{\omega_c^2 s}{s^2 + 2\delta\omega_c s + \omega_c^2} \qquad with \qquad \omega_c = 2\pi 50 \quad and \quad \delta = 1/\sqrt{2} \tag{8}$$

Using the model in figure 10 and computing the gains N_x, N_u and the feedback gain $K \in \mathbb{R}^{1\times 2}$ such that allocates two closed-loop eigenvalues in $\lambda_{1,2} = -\delta\omega_n \pm j\omega_n\sqrt{1-\delta^2}$, see appendix A.2, Code 1. The following results are obtained and everything is ready to perform the tests:

$$N_x = [1\ 0] \ , \ N_u = 0 \ , \ K = [8.6927\ 0.1139] \ , \ poles = -20 \pm 27.2875j \tag{9}$$

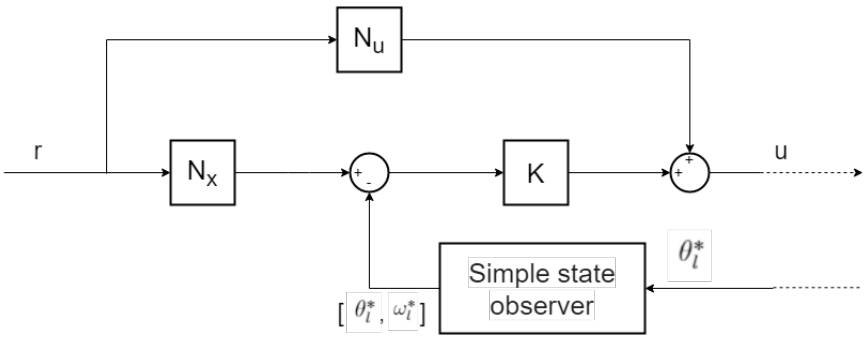


Figure 10: Model of the Nominal Tracking Design

The results of the step response are plotted in figure 11, where we can notice that without static friction τ_{sf} the nominal design is able to instantly and asymptotically track the step reference input even with different amplitude. This will cost an increment on the overshoot that almost touch its threshold. Considering ,instead, the static friction, that affects as a disturbance entering at the plant level, the model is not able to guarantee perfect steady state tracking, due to the presence of the static friction and parameter uncertainties.

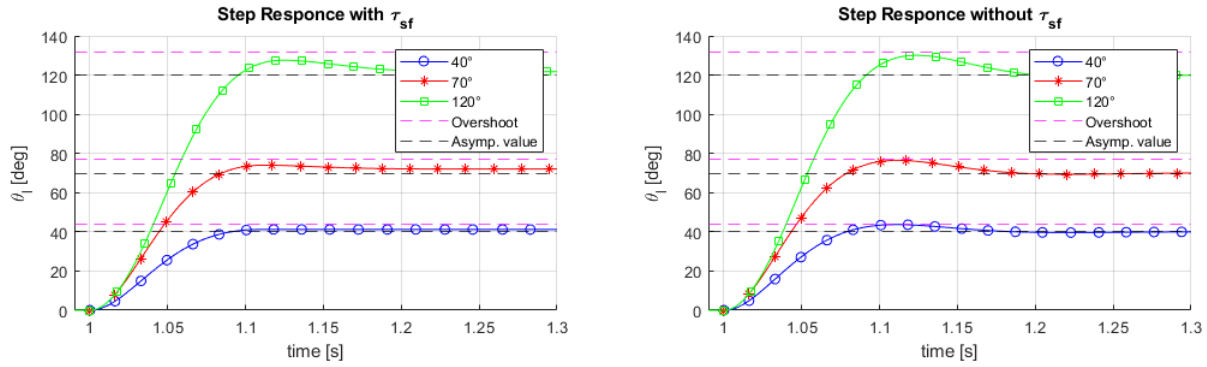


Figure 11: Step Responses in nominal tracking design with and without static friction

Even in practice the nominal design well perform, it does not break the threshold of both the settling time and the overshoot. Comparing the two simulations can be noticed that the laboratory simulation has a slightly less overshoot with respect to the numerical simulation, but this impact on the settling time that will be less.

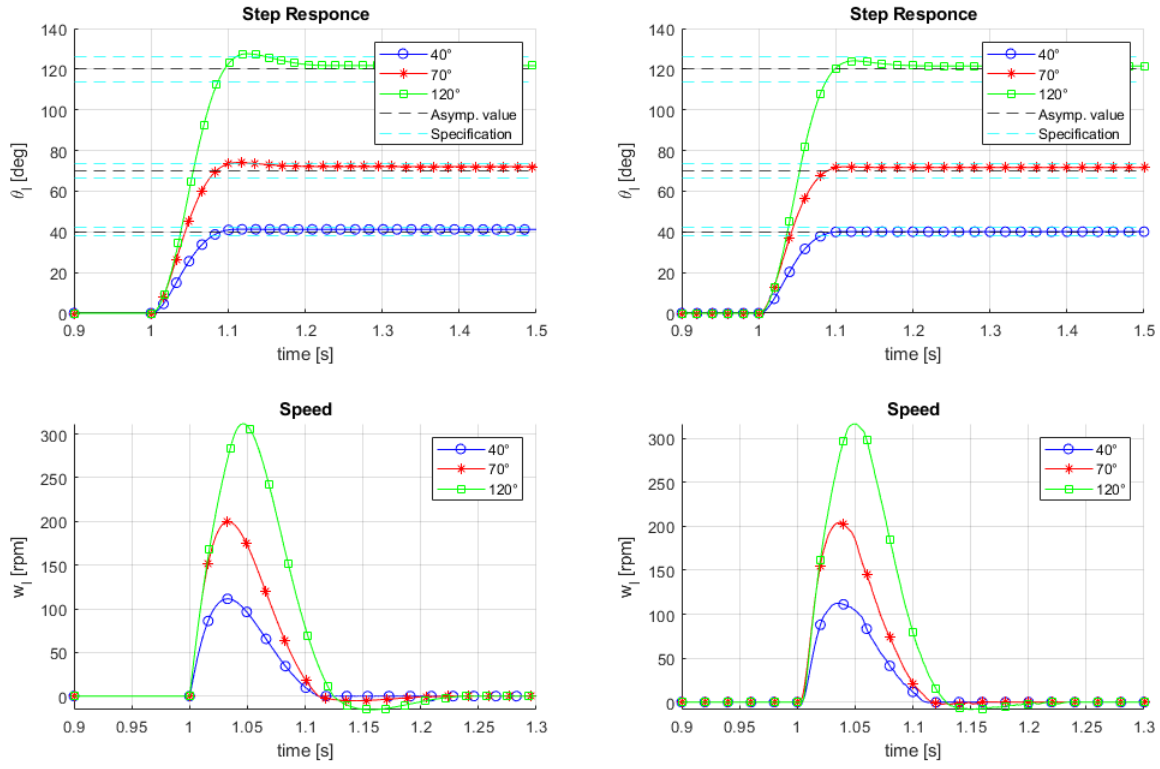


Figure 12: Numerical simulation and laboratory simulation of the nominal tracking design

2.2.2 Robust tracking design with integral action

In case of model uncertainties and constant disturbances entering at the plant level, like the static friction τ_{sf}, we can resort to the robust tracking design with integral action.
This method extends the state of the system $x_e = [x_i \ x]^T$ introducing a integrator state x_i. Obviously, whit this change, the all state-space model will increase its size and hence

the pole allocation problem needs another pole, while the feedback gain will be of the form: $K = [K_i \ K] \in \mathbb{R}^{1\times 3}$.
The scheme of the figure 13 and the code 2 in appendix A.2 are capable to handle this problem and allow to reach perfect tracking conditions.
Nevertheless, the choice of the closed-loop eigenvalues still a crucial point that can mess up the design and force the system to assume divergent behaviour. To make the best choice it is common to make several simulations and compare them, i.e. four different types are considered:

- Poles 1: $\lambda_{c,\{1,2\}} = \sigma \pm j\omega_d,\ \lambda_{c,\{3\}} = \sigma$
- Poles 2: $\lambda_{c,\{1,2,3\}} = \sigma$
- Poles 3: $\lambda_{c,\{1,2\}} = 2\sigma \pm j\omega_d,\ \lambda_{c,\{3\}} = 2\sigma$
- Poles 4: $\lambda_{c,\{1,2\}} = 2\sigma \pm j\omega_d,\ \lambda_{c,\{3\}} = 3\sigma$

where σ and ω_d are the real and imaginary parts of the eigenvalue in formula (9).

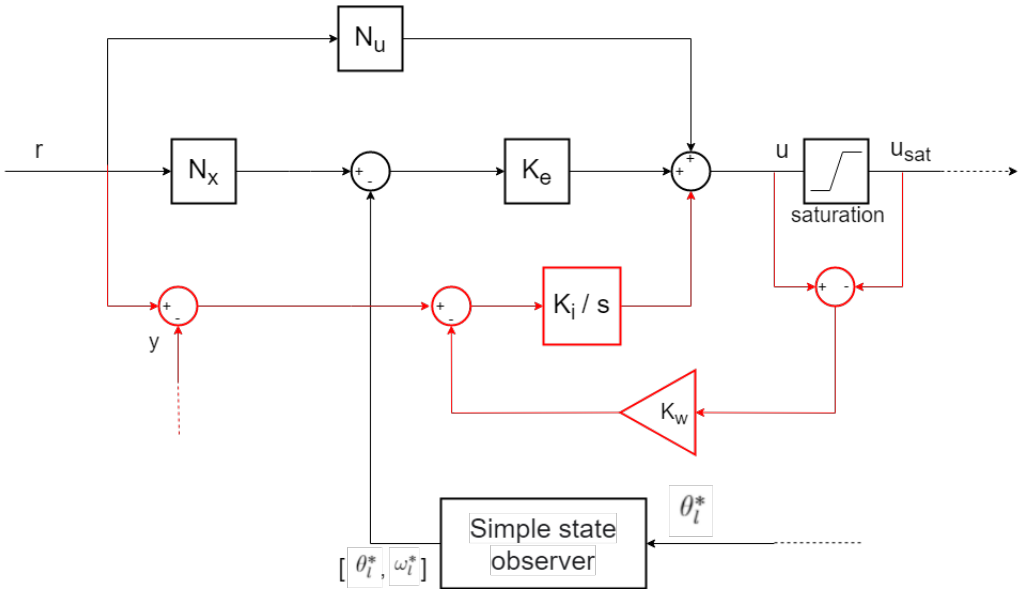


Figure 13: Model of the robust tracking design

From a first observation, without simulation, we can expect that the poles 4 will have better result. Multiple eigenvalues are very "fragile" and imply a slow convergence, additionally in a continuous time system placing the modes on the left side of the complex plan and distance from the imaginary axis will give a fast convergence. Indeed, the choice 4 has the third eigenvalue more negative with respect to the other, justifying our selection.
The numerical simulation of figure 14 and the results in table 3, show how the poles 1 and 2 can be discarded, since both have an overshoot bigger than 10% of the reference ($40\,\mathrm{deg}$) and they also break the settling time threshold. ($0.15s$). The specifications are not satisfied. In the other hand, both poles 3 and 4 have good results and from table 3 they are very similar.

Poles 3 has a better overshoot but less settling time and then pole 4 has the opposite.

Eigenvalues	Overshoot, M_p [deg]	Settling time $t_{s,5}$, [s]	Rising time [s]
Poles 1:	50.5331	0.2016	0.0360
Poles 2:	47.5867	0.4183	0.0582
Poles 3:	40.2897	0.1015	0.0743
Poles 4:	40.4454	0.0938	0.0698

Table 3: results for different poles

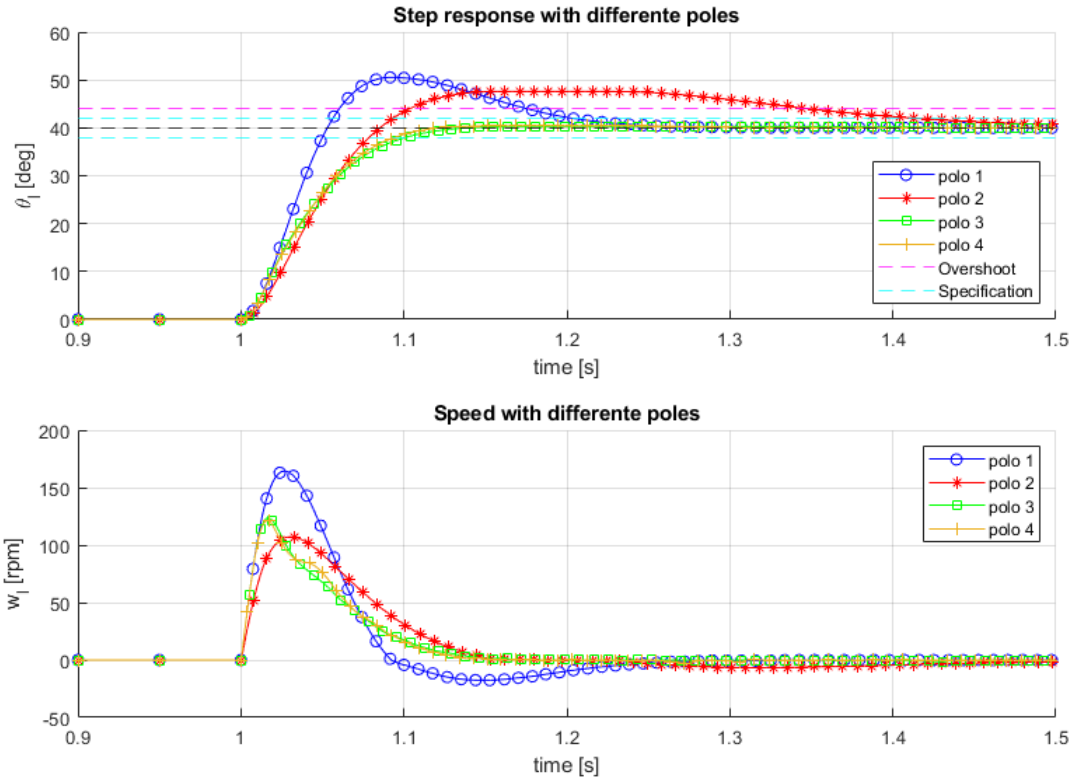


Figure 14: Plots of the robust tracking design results with different poles allocations

Notice that, as we have done in the anti-windup improvement section, the gain K_w can be optimize for achieve little improvement, up till now $K_w = 33.3$ has been used.
Focusing on the poles 4, we can see from figure 15 that the robust tracking design is able to track the constant position set-point with zero-steady state error even in presence of static friction, due to the presence of the integral arm. Note that without the static friction the convergence steel faster.

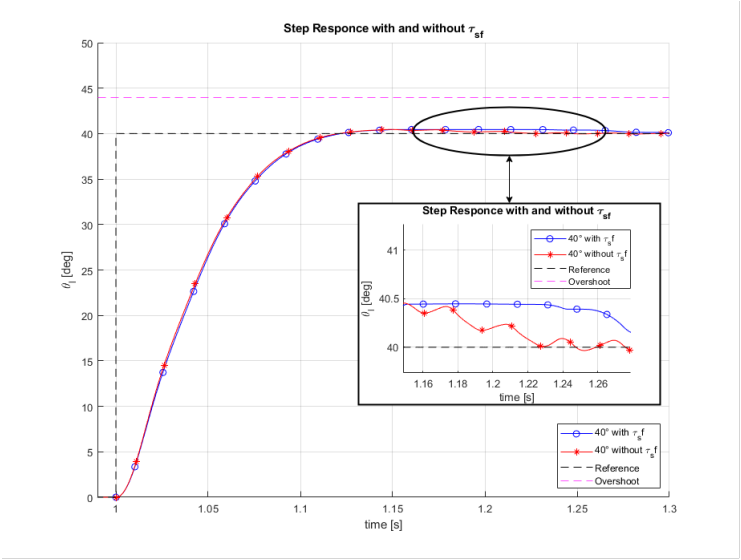


Figure 15: Step response comparison with and without τ_{sf}

The tracking condition can be seen also in figure 16, where the comparison of the numerical and practical simulation is performed. The two simulations are very similar and have good behaviour even if the reference increase its amplitude from 40° to 70° until 120°.

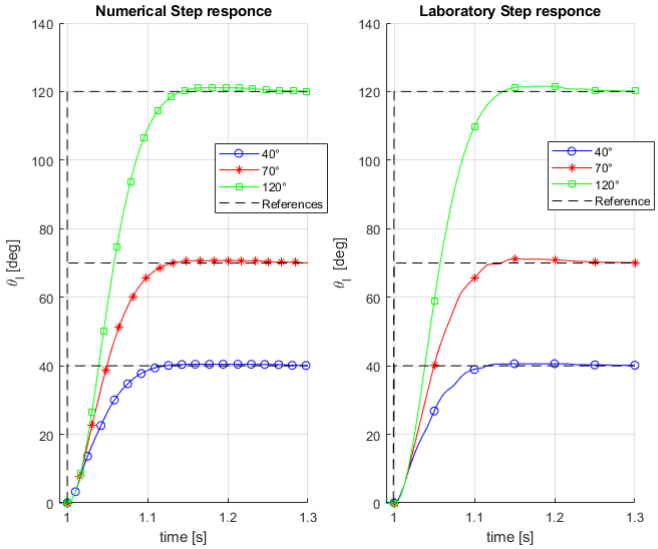


Figure 16: Step Response with different amplitude either with Simulink and in laboratory

In order to conclude this section, and confirm the improvement obtained by using the robust tracking design, instead of the nominal, is plotted in figure 17 a comparison that shows the fast convergence of the robust tracking to the reference with different position values.

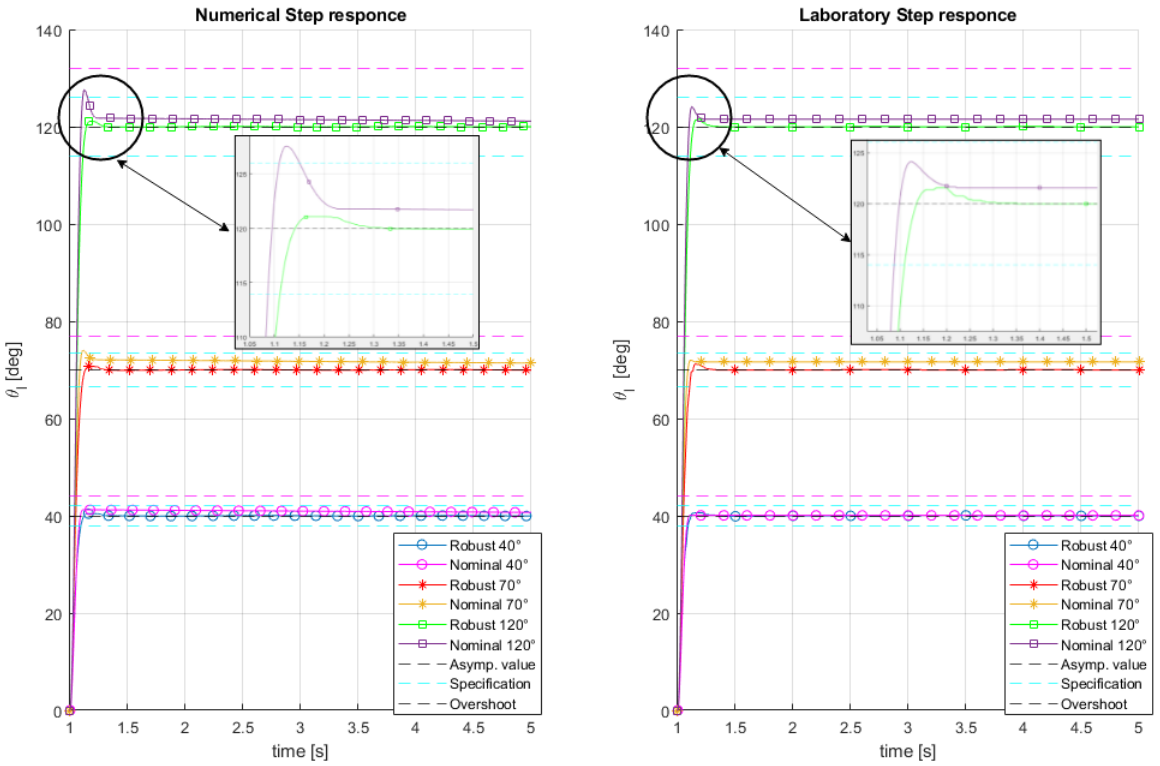


Figure 17: Comparison between nominal and robust tracking

Robust tracking with integral action have a cons, with general signal, like sinusoidal, it is not able to guarantee steady state tracking and perfect disturbance rejection. The following two methods exploit the internal model principle for achieve this purpose.
In the next experiments we will use a sinusoidal reference r and a constant disturbance w. From the ODEs of this two signals, can be derived the autonomous state-space model or exo-models, essential for the application of the internal model principle.
Exo-system : $\Sigma_z = (A_z,\ C_z)$

$$A_z = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_0^2 & 0 \end{bmatrix}, \qquad C_z = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{10}$$

2.2.3 Robust tracking with error-space approch

In the place of solving the problem of tracking the reference we reformulate it as the problem of tracking the error to zero. After defining the error-space state ξ and control, the dynamic equation for the error-space state and the extended state $z = [e, e^{(1)}, e^{(2)}, ..., e^{(q-1)}, \xi]^T$ we are able to compute $\dot{z}$. In this way we can compute the two matrices $\bar{A}$ and $\bar{B}$ that defines the state dynamics of z.
Note that, since the starting matrix A and B are reachable also the new couple will be reachable, then, we can free allocate the eigenvalues of $\bar{A} - \bar{B}K_z$, where $K_z = [K_0, ..., K_q, K_\xi]$.
For implementing this control we need a control signal $u = -K_\xi x - H(s)e$ in which $H(s)$,

formula 11, contains in the denominator the models of the the sinusoidal reference signal and the constant disturbance to reject.

$$H(s) = \frac{K_z(3)s^2 + K_z(2)s^1 + K_z(1)}{s^3 + w_0^2 s} \ . \tag{11}$$

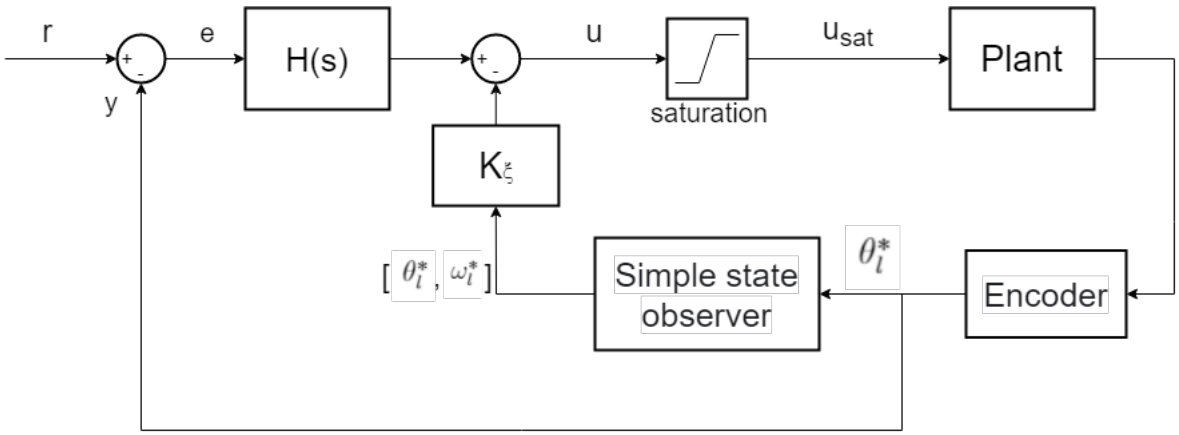


Figure 18: Model designed with robust tracking with error-space methods

The state of this system has five component, so we need to allocate five closed-loop eigenvalues, consider the choice:

$$\lambda_{c,\{1,2\}} = \omega_n e^{j(-\pi \pm \pi/4)}, \ \lambda_{c,\{3,4\}} = \omega_n e^{j(-\pi \pm \pi/6)}, \ \lambda_{c,\{3,4\}} = -\omega_n \tag{12}$$

where the value ω_n it the natural frequency of the dominant closed-loop dynamics approximated with a second order system used in section 2.2.
The performance of the entire method and the correct allocation problem can be tested by applying the design for different periods of the sinusoidal reference. For this aim, consider four periods: $T_r = [0.15s, \ 0.25s, \ 0.5s, \ 1s]$.
In figure 21 the results confirm the method and solve the perfect steady state robust tracking even with general references. As can be seen from the tracking error plot the design is able to achieve the goal also for small periods, in which there are fast changes.

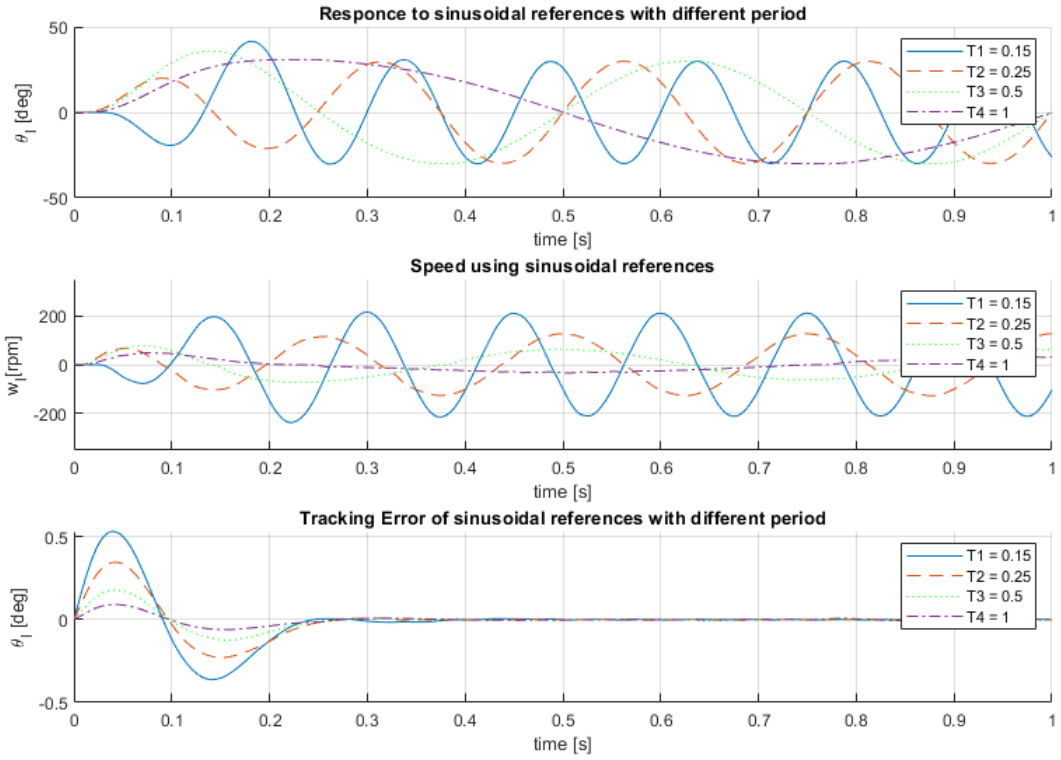


Figure 19: Robust tracking with error-space methods with different periods

Focusing on the period $T_r = 0.5$, the controller can be tested either in a real situation, with static friction, or in an ideal situation, without static friction. The figure 20 show that the differences are negligible, implying that the controller can handle constant disturbance as we have said before.

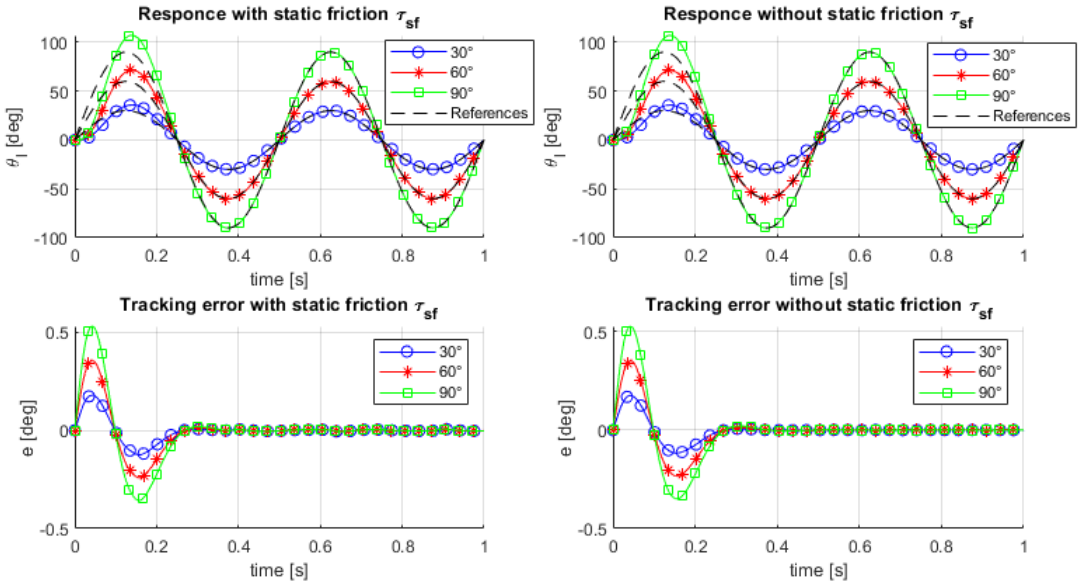


Figure 20: Robust tracking with error-space methods in real and ideal scenario

Confirmed the excellent result of the numerical simulation, now the laboratory test can be

performed. The test uses three different amplitudes of the sinusoidal inputs $(30°, 60°, 90°)$ in order to have a more general view of the experience.
Data are both very similar, except for the control input in which have a non linear behaviour of high frequency. This difference is due only for the number of samples chosen for each experiment, indeed the number of samples used in laboratory is ten times less with respect to the numerical case.

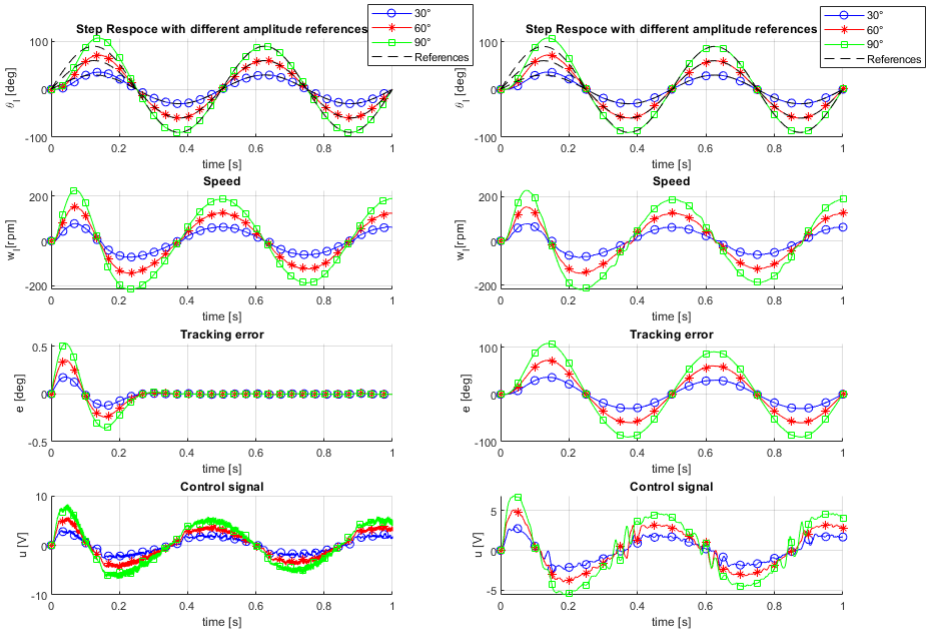


Figure 21: Robust tracking with error-space methods, using different period for the sinusoidal reference

If we repeat the test for difference position reference input signals without changing the exo-model in the previous design, the system will be no more able to robust asymptotically track the reference.
For explain better this concept repeat the test with a sinusoidal reference with $T_r = 0.1$, leaving unchanged the design that takes care of sinusoidal signals of $T_r = 0.5$. What we obtain is like a linear combination of the modes of both the sinusoidal signals, since we are introducing a sinusoidal with $T_r = 0.5$ which is not present. Hence it will impact directly the mode given by the reference with $T_r = 0.1$.
In figure 22 is showed the tracking error for both the inputs, but same design, and is evident that the model can't work with different periods from that used in the exo-system.

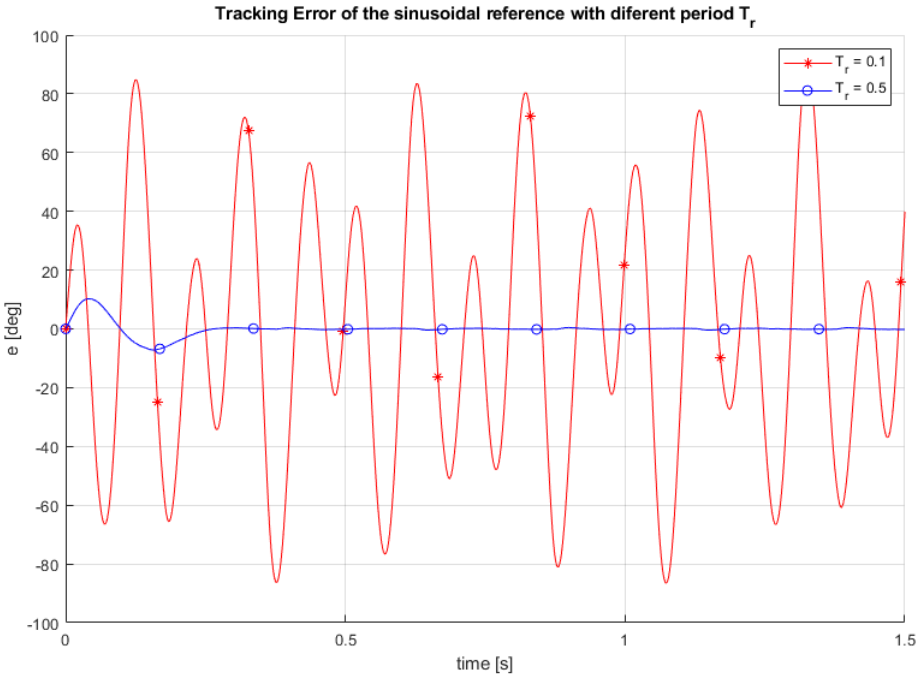


Figure 22: Robust tracking with error-space methods, using different period for the sinusoidal reference

The previous reasoning can be apply not only to different period reference signals but, more in general, to other inputs that has not taken into account in the previous design. For example, in the figure 23 is showed the test with a step input of $40°$. Both the tests, numerical and practical, can't reach an asymptotic steady state value in both an oscillation still present, due to the introduction of a compensation mode that is not present. This will impact also on the performance, in fact, the overshoot and settling time threshold is exeded.

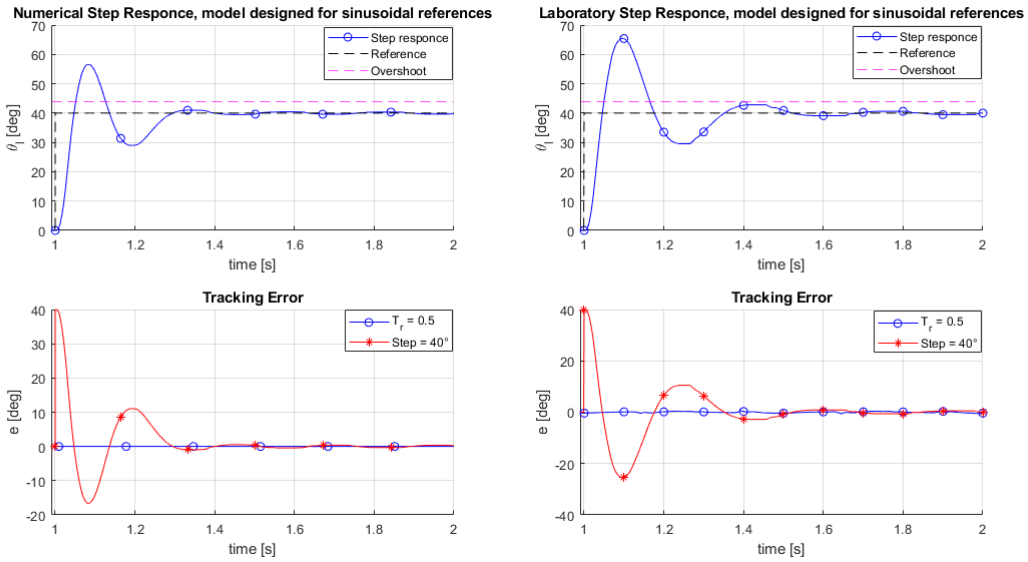


Figure 23: Robust tracking with error-space methods, using different period for the sinusoidal reference

2.2.4 Extended-state estimator method

The second method that exploit the internal model principle for achieve robust steady state tracking and perfect disturbance rejection is the extended-state estimator approach. Basically with this approach we use a full order state estimator $\hat{\Sigma}_e$ for estimate both the reference input and the load disturbance, in this way, they can be used for reference and disturbance feedforward compensation. First of all, the design of the extended estimator requires to place two eigenvalues for the controller and five for the estimator. The following choice has been taken:

$$\lambda_{c,\{1,2\}} = \delta\omega_n \pm j\omega_n\sqrt{1-\delta^2} \tag{13}$$

$$\lambda_{e,\{1,2\}} = 2\omega_n e^{j(-\pi\pm\pi/3)}, \quad \lambda_{e,\{3,4\}} = 2\omega_n e^{j(-\pi\pm\pi/6)}, \quad \lambda_{e,\{3,4\}} = -2\omega_n \ . \tag{14}$$

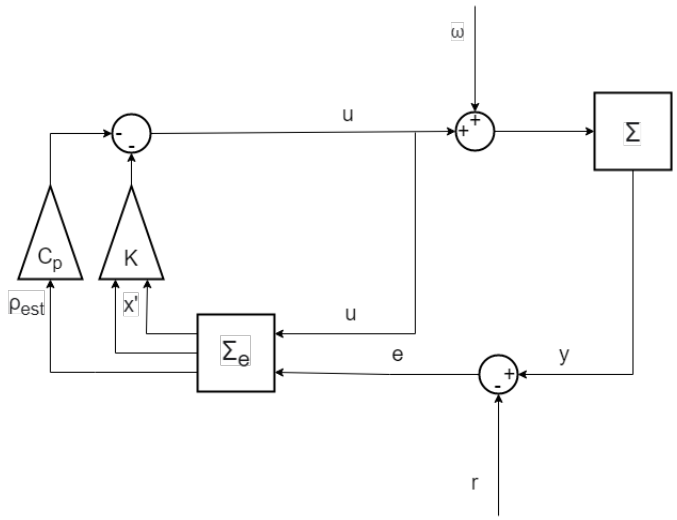


Figure 24: Model of the extended-state estimator method

After the computation of the matrix K_e for solve the allocation problem for the controller $(A-BK)$ and the matrix L_e for the estimator $(A_e - L_e B_e)$, the last one considering the dual system, the tests can be executed.
In the first test is consider the previous position sinusoidal reference input with $T_r = 0.5$ and three different amplitude for the numerical simulation: $30°$,$60°$,$90°$; while for the laboratory simulation as been considered only a amplitude of $30°$.
From both plots of figure 25 the robust steady state tracking and perfect disturbance rejection has been reached, in particular the tracking error after a first phase when try to stabilize it stay near zero.

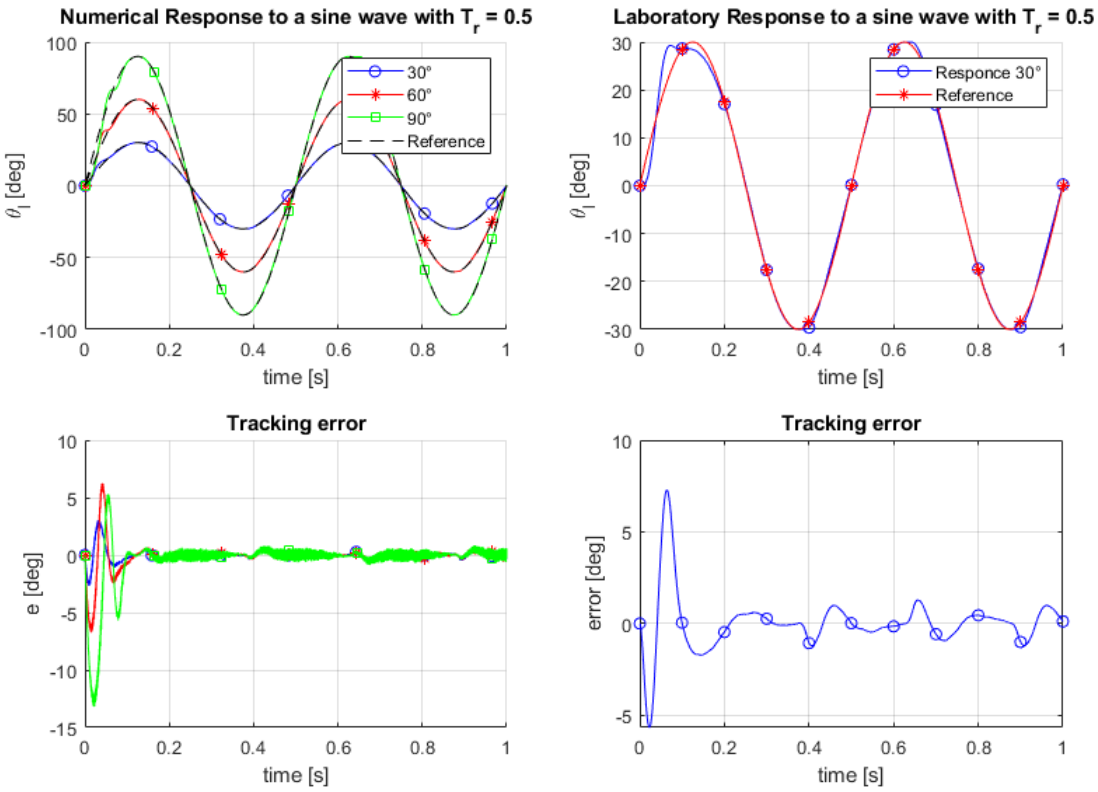


Figure 25: Robust tracking with extended-state estimator method, using different period for the sinusoidal reference

In the second test we want to analyze if the model designed with the previous reference can solve the robust tracking problem also for reference with different period, like in section 2.2.3. A new sinusoidal reference with $T_r = 0.1$ has been considered but we have the same problem of the modes, in particular we are feedforward compensating a reference which is no present, hence it will sum whit the other sinusoidal signal producing a linear combination of two modes this period 0.5 and 0.1. This can be easy seen in the response of figure 26.
By the tracking error plot this method is not able to perform asymptotic robust tracking for reference whit different periods.

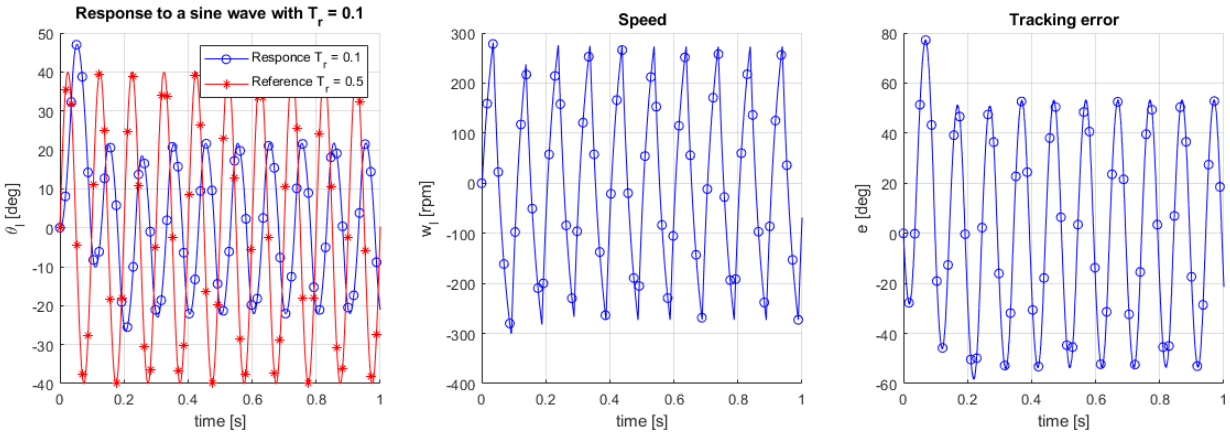


Figure 26: Extended-state estimator method designed with $T_r = 0.5$ but reference signal with $T_r = 0.1$

The second test can be extended to general reference signal in which was not taken into account in the design phase. In figure 37 a step reference with amplitude 40° has been used, indeed we have a not suitable step response. The Overshoot is too high for both the plots and after that a permanent sinusoidal around the steady state value is present. Notice that, in the first ascent of the step response a non linear effect is presence, it is due to the saturation of the control signal u that increase faster.

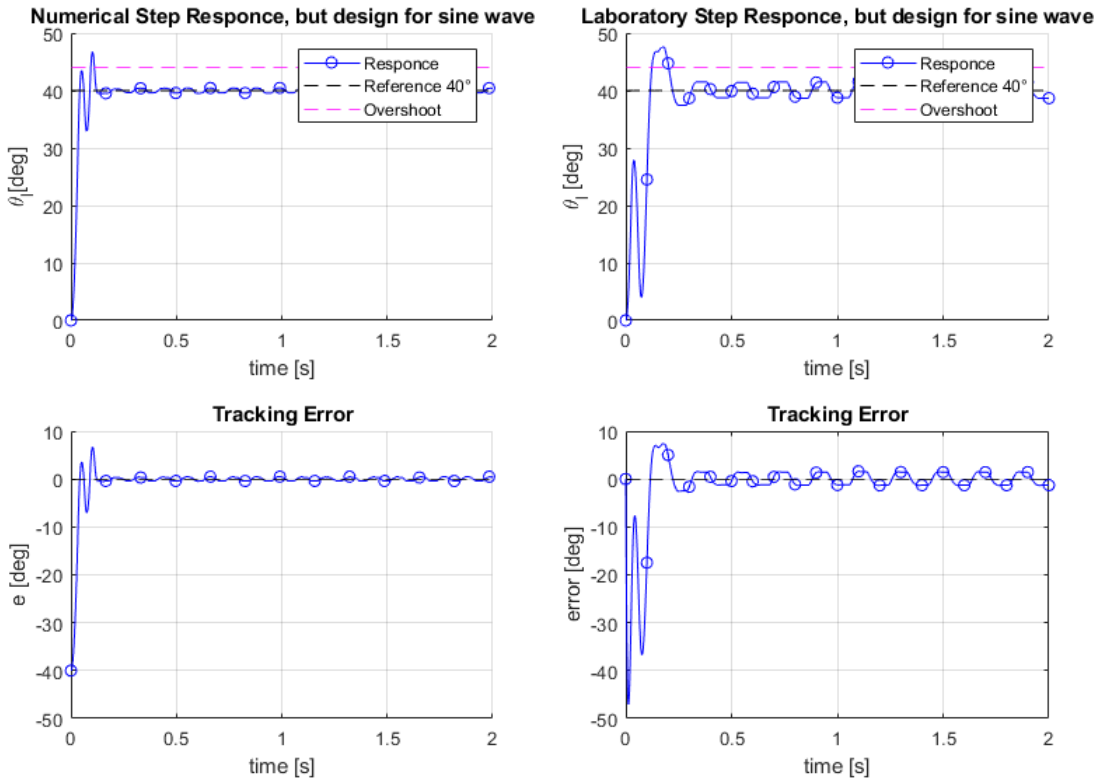


Figure 27: Extended-state estimator method designed with $T_r = 0.5$ but step reference

A Appendix

A.1 Detailed calculations for PID design

Starting from the specification on section 1.2 we can compute the damping factor δ and the fase margin ϕ_m whit the formulas:

$$\delta = \frac{\log(1/M_p)}{sqrt\pi^2 + \log^2(1/M_p)} \ , \quad \phi_m = \arctan \frac{2\delta}{\sqrt{\sqrt{1+4\delta^4} - 2\delta^2}} \tag{15}$$

The damping factor can be used for compute the crossover frequency $\omega_{gc} \approx 3/(\delta t_{s,5\%})$ used for the frequency response of the system $P(j\omega gc)$.

$$\Delta K = |P(j\omega_{gc})|^{-1} \ , \quad \Delta\phi = -\pi + \phi_m - \angle P(j\omega_{gc}) \tag{16}$$

In formula 15 the latest two parameters needed for the gains has been computed. In fact using formulas 17 we can finally compute the proportional gain K_p, the integration time T_I and the derivative time T_D.

$$\begin{cases} K_p = \Delta K \cos\Delta\phi \\ T_D = \frac{\tan\Delta\phi + \sqrt{(\tan\Delta\phi)^2 + 4/\alpha}}{2\omega_{gc}} \\ T_I = \alpha T_D \end{cases} \tag{17}$$

Then, $K_D = K_P T_D$ and $K_I = K_P/T_I$. α is like a tuning parameter that usually is major or equal to 4. We have considered an $\alpha = 6$, this change will increase a little bit the integral gain and the derivative gain that impact directly on the overshoot and on the settling time allow us to achieve better performances. Value of the parameters necessary to compute the PID gains:

Parameter:	$\alpha = T_I/T_D = 6$	$\alpha = T_I/T_D = 4$
Settling time ($t_{s5\%}$)	0.15	0.15
Overshoot (M_p)	0.1	0,1
Damping factor (δ)	0.5912	0.5912
Phase margin (ϕ_m)	1.0226	1.0226
Crossover frequency (ω_{gc})	33.8321	33.8321
Frequency responce (H(s))	-0.0822 - 0.0819i	-0.0822 - 0.0819i
Proportional gain (Kp)	8.3738	8.3738
Derivative gain (Kd)	0.1356	0.1575
Integration gain (Ki)	86.1707	111.2674
Derivative time (Td)	0.0162	0.0188
Integration time (Ti)	0.0972	0.0753

Table 1: PID parameters

A.2 Matlab code

Code 0: PID gains

```
ctrl.ts5    = 0.15;         % Settling-time
ctrl.Mp     = 0.1;          % Overshoot
ctrl.alpha  = 6;            % Alpha value

% Compute PID gain
ctrl.damping_factor = log(1/ctrl.Mp)/(sqrt(pi^2+(log(1/ctrl.Mp))^2));   %damping factor
ctrl.phi_m = atan(2*ctrl.damping_factor/(sqrt(sqrt(1+4*ctrl.damping_factor^4)-2*ctrl.damping_factor^2))); %(41)
ctrl.w_gc = 3/(ctrl.damping_factor*ctrl.ts5);                           %crossover frequency
ctrl.frq_resp = freqresp(Ps,ctrl.w_gc);
abs(ctrl.frq_resp);
angle(ctrl.frq_resp);
ctrl.Delta_K = 1/(abs(ctrl.frq_resp));
ctrl.Delta_phi = -pi + ctrl.phi_m - angle(ctrl.frq_resp);
ctrl.Kp = ctrl.Delta_K*cos(ctrl.Delta_phi);                             %proportional gain
ctrl.Td = (tan(ctrl.Delta_phi)+sqrt((tan(ctrl.Delta_phi))^2 +4/ctrl.alpha))  /(2*ctrl.w_gc);
ctrl.Ti = ctrl.alpha*ctrl.Td;
ctrl.Ki = ctrl.Kp/ctrl.Ti;                                              %integral gain
ctrl.Kd = ctrl.Kp*ctrl.Td;                                              %derivative gain
ctrl.T_L = 0.25/ctrl.w_gc;
```

Code 1: Nominal tracking design

```
%   non-encapsulated SS data
A = [0, 1; 0, -1/model.Tm];
B = [0; model.km/(gbox.N*model.Tm)];
C = [1, 0];
D = 0;
%   SS data encapulated in a SS object
sys = ss(A, B, C, D);

%compute N_x, N_u
S = [A,B;C,0];
b=[0 0 1]';
N = linsolve(S,b); %Sx=b
N_x = [N(1) N(2)];
N_u = N(3);

%specs
spec.ts5    = 0.15;         % Settling-time
spec.Mp     = 0.1;          % Overshoot

%poles to place
Ts = tf([spec.wn^2],[1, 2*spec.damp*spec.wn,spec.wn^2]);
p=pole(Ts);
K=place(A,B,p); %Compute feedback matrix
```

Code 2: Robust tracking design with integral action

```
%poles
sigma=-spec.damp*spec.wn;
omega_d=spec.wn*sqrt(1-spec.damp^2);
p4 = [2*sigma+omega_d*j , 2*sigma-omega_d*j , 3*sigma ];
```

```

%space model with augmented state
zer = zeros(2,1);
Ae=[0 C; zer A];
Be=[0; B];
Ce=[0 C];

%4st pole
Ke = acker(Ae,Be,p4);
K2 = [Ke(2) Ke(3)];
Ki = Ke(1);
```

Code 3: Robust tracking design with error space approach

```
%poles
pa=spec.wn*exp(j*(-pi+pi/4));
pb=spec.wn*exp(j*(-pi-pi/4));
pc=spec.wn*exp(j*(-pi+pi/6));
pd=spec.wn*exp(j*(-pi-pi/6));
pe=-spec.wn;
p=[pa pb pc pd pe];

% Design of poles for different period
T = [0.15 0.25 0.5 1];
figure('Renderer', 'painters', 'Position', [10 10 900 600])
for i = 1:4;
    Tr=T(i);
    w0= 2*pi/Tr;
    %compute Az, Bz of the extended model in error-space
    Ar=[0,1,0;0,0,1;0,-w0^2, 0];
    zer=zeros(2,3);
    colonna = [0,0;0,0; C];
    Az= [Ar colonna;zer A];
    Bz=[0;0;0;B];
    Kz=acker(Az,Bz,p);
```

Code 4: Robust tracking with extended state estimator method

```
%"controller" eigenvalues and "estimator" eigenvalues
pc1=-spec.damp*spec.wn + j*spec.wn*sqrt(1-spec.damp^2);
pc2=-spec.damp*spec.wn - j*spec.wn*sqrt(1-spec.damp^2);
pc = [pc1 , pc2];

pe1=2*spec.wn*exp(j*(-pi+pi/3));
pe2 = conj(pe1);
pe3=2*spec.wn*exp(j*(-pi+pi/6));
pe4 = conj(pe3);
pe5=-2*spec.wn;
pe =[pe1 pe2 pe3 pe4 pe5];

% reference
T = [0.15 0.25 0.5 1];
Tr = 0.5;
w0 = 1/Tr;
Ar = 30;

% Define system
```

```
Ap = [0 1 0 ; 0 0 1 ; 0 -w0^2 0];
Cp = [1 0 0];
A = [0, 1; 0, -1/model.Tm];
B = [0; model.km/(gbox.N*model.Tm)];
C = [1, 0];
D = 0;
zer = zeros(3,2);
BCp = B*Cp;

Ae = [Ap , zer;
      BCp , A ];
Be = [0 ; 0 ; 0 ; B ];
Ce = [0 , 0 , 0 , C];
K3 = place(A,B,pc);
Le = place(Ae.',Ce.',pe).';

Aest = Ae - Le*Ce;
Best = [ Be Le];
Cest = eye(5);
Dest = zeros(5,2);
```

A.3 Data sheet

Meaning of the main parameters:

Parameter:	Symbols definition
J_m, B_m	rotor moment of inertia and viscous friction coefficient
J_l, B_l	load moment of inertia and viscous friction coefficient
R_a, L_a	resistance and inductance of the armature circuit
u_a, i_a	supply voltage and current to the armature circuit
u_e	back electromotive force (BEMF)
k_t, k_e	torque and electric (BEMF) constants
τ_m, ω_m, θ_m	motor side torque, speed and position
τ_l, ω_l, θ_l	load side torque, speed and position
τ_d	disturbance torque applied to the load inertia
N	planetary gearbox reduction ratio
R_s	shunt resistance
u, u_{drv}	voltage driver input and output voltages
k_{drv}, $f_{c,drv}$	voltage driver attenuation gain and cut–off frequency
T_{drv}	voltage driver time constant
J_{eq}, B_{eq}	total inertia and viscous friction as “seen” at motor side

Data listing 1:

```
%% General parameters and conversion gains

 % conversion gains
 rpm2rads = 2*pi/60; % [rpm] -> [rad/s]
 rads2rpm = 60/2/pi; % [rad/s] -> [rpm]
 rpm2degs = 360/60; % [rpm] -> [deg/s]
```

```
degs2rpm = 60/360; % [deg/s] -> [rpm]
deg2rad = pi/180; % [deg] -> [rad]
rad2deg = 180/pi; % [rad] -> [deg]
ozin2Nm = 0.706e-2; % [oz*inch] -> [N*m]

%% DC motor nominal parameters

% brushed DC-motor Faulhaber 2338S006S
mot.R = 2.6; % armature resistance
mot.L = 180e-6; % armature inductance
mot.Kt = 1.088 * ozin2Nm; % torque constant
mot.Ke = 0.804e-3 * rads2rpm; % back-EMF constant
mot.J = 5.523e-5 * ozin2Nm; % rotor inertia
mot.B = 0.0; % viscous friction coeff (n.a.)
mot.eta = 0.69; % motor efficiency
mot.PN = 3.23/mot.eta; % nominal output power
mot.UN = 6; % nominal voltage
mot.IN = mot.PN/mot.UN; % nominal current
mot.tauN = mot.Kt*mot.IN; % nominal torque
mot.taus = 2.42 * ozin2Nm; % stall torque
mot.w0 = 7200 * rpm2rads; % no-load speed

%% Gearbox nominal parameters

% planetary gearbox Micromotor SA 23/1
gbox.N1 = 14; % 1st reduction ratio (planetary gearbox)
gbox.eta1 = 0.80; % gearbox efficiency

% external transmission gears
gbox.N2 = 1; % 2nd reduction ratio (external trasmission gears)
gbox.J72 = 1.4e-6; % inertia of a single external 72 tooth gear
gbox.eta2 = 1; % external trasmission efficiency (n.a.)

% overall gearbox data
gbox.N = gbox.N1*gbox.N2; % total reduction ratio
gbox.eta = gbox.eta1*gbox.eta2; % total efficiency
gbox.J = 3*gbox.J72; % total inertia (at gearbox output)

%% Mechanical load nominal parameters

% inertia disc params
mld.JD = 3e-5; % load disc inertia
mld.BD = 0.0; % load viscous coeff (n.a.)

% overall mech load params
mld.J = mld.JD + gbox.J; % total inertia
mld.B = 2.5e-4; % total viscous fric coeff (estimated)
mld.tausf = 1.0e-2; % total static friction (estimated)

%% Voltage driver nominal parameters

% op-amp circuit params
drv.R1 = 7.5e3; % op-amp input resistor (dac to non-inverting in)
drv.R2 = 1.6e3; % op-amp input resistor (non-inverting in to gnd)
drv.R3 = 1.2e3; % op-amp feedback resistor (output to inverting in)
drv.R4 = 0.5e3; % op-amp feedback resistor (inverting in to gnd)
drv.C1 = 100e-9; % op-amp input capacitor
drv.outmax = 12; % op-amp max output voltage

% voltage driver dc-gain
drv.dcgain = drv.R2/(drv.R1+drv.R2) * (1 + drv.R3/drv.R4);

```

```
% voltage driver time constant
drv.Tc = drv.C1 * drv.R1*drv.R2/(drv.R1+drv.R2);

%% Sensors data

% shunt resistor
sens.curr.Rs = 0.5;

% Hewlett-Packard HEDS-5540#A06 optical encoder
sens.enc.ppr = 500*4; % pulses per rotation
sens.enc.pulse2deg = 360/sens.enc.ppr; % [pulses] -> [deg]
sens.enc.pulse2rad = 2*pi/sens.enc.ppr; % [pulses] -> [rad]
sens.enc.deg2pulse = sens.enc.ppr/360; % [deg] -> [pulses]
sens.enc.rad2pulse = sens.enc.ppr/2/pi; % [rad] -> [pulses]

% potentiometer 1 (Spectrol 138-0-0-103) - installed on motor box
sens.pot1.range.R = 10e3; % ohmic value range
sens.pot1.range.V = 5; % voltage range
sens.pot1.range.th\_deg = 345; % angle range [deg]
sens.pot1.range.th = sens.pot1.range.th$_$deg * deg2rad; % angle range [rad]
sens.pot1.deg2V = sens.pot1.range.V / sens.pot1.range.th$_$deg; % sensitivity [V/deg]
sens.pot1.rad2V = sens.pot1.range.V / sens.pot1.range.th; % sensitivity [V/rad]
sens.pot1.V2deg = 1/sens.pot1.deg2V; % conversion gain [V] -> [deg]
sens.pot1.V2rad = 1/sens.pot1.rad2V; % conversion gain [V] -> [rad]

%% Data acquisition board (daq) data

% NI PCI-6221 DAC data
daq.dac.bits = 16; % resolution (bits)
daq.dac.fs = 10; % full scale
daq.dac.q = 2*daq.dac.fs/(2$^$daq.dac.bits-1); % quantization

% NI PCI-6221 ADC data
daq.adc.bits = 16; % resolution (bits)
daq.adc.fs = 10; % full scale (as set in SLDRT Analog Input block)
daq.adc.q = 2*daq.adc.fs/(2$^$daq.adc.bits-1); % quantization
```

A.4 Simulink models and implementation

System and model:

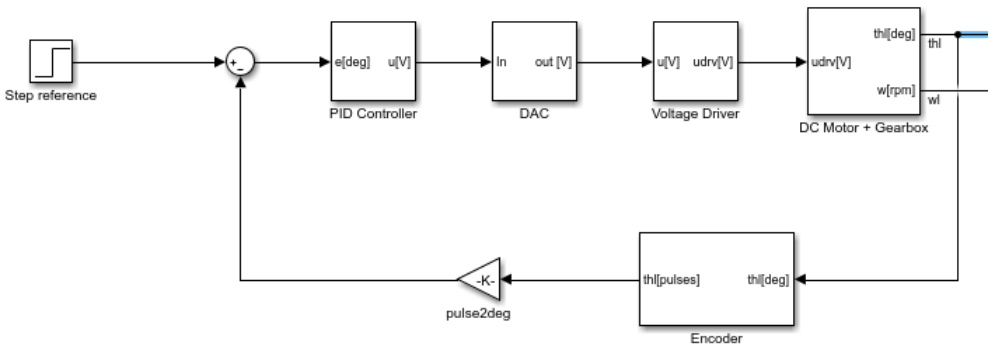
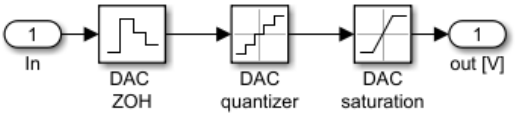


Figure 28: Position control-loop

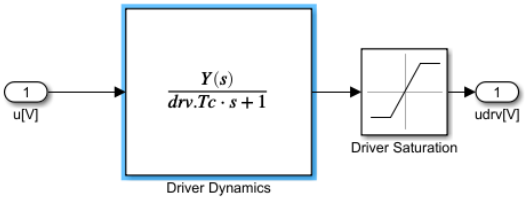
The digital to analog converter comprises a discrete zero–order hold (sampling time $T_s = 0.001$), a quantizer (quantisation interval $= 20/(216-1)V$) and the saturation ($\pm 10V$). The block takes the continuous-time control signal $u[V]$ as input and produce the saturated input for the voltage driver that have a lowpass dynamics defined by:

$$H_{drv}(s) = \frac{k_{drv}}{T_{drv}s + 1} \tag{18}$$

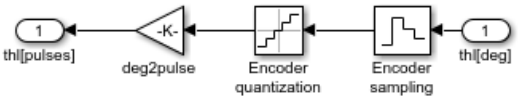
Moreover, another saturation block is needed for limit $u_{drv}[V]$ voltage between $\pm 12V$, that will enter directly on the plant. The encoder is fundamental for reconstruct our output, namely the angle θ_l, and is formed by: a discrete zero–order hold , a quantizer (numerical quantization interval $= 360°/(500 \times 4)$, while in laboratory $360°/4096$) and a gain to perform the conversion from degrees units to pulse count.



(a) DAC



(b) Voltage driver



(c) Encoder

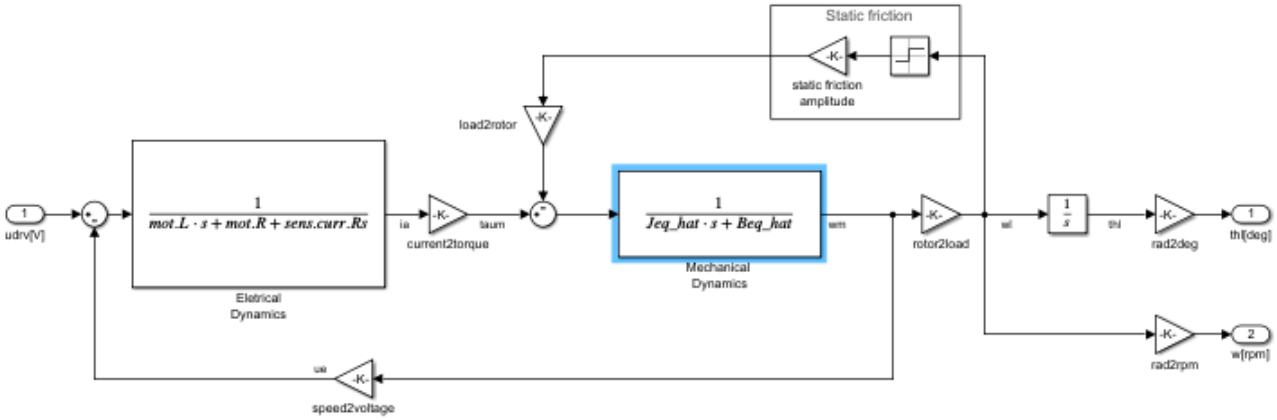


Figure 29: Dc motor + gearbox

Techniques:

- Anti-windup mechanism:

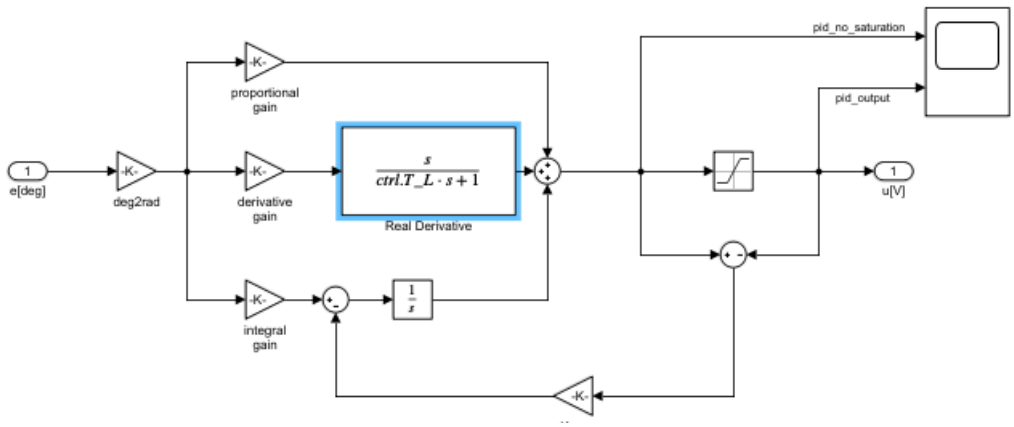


Figure 30: Anti-windup scheme of the controller

- Feedforward compensation:

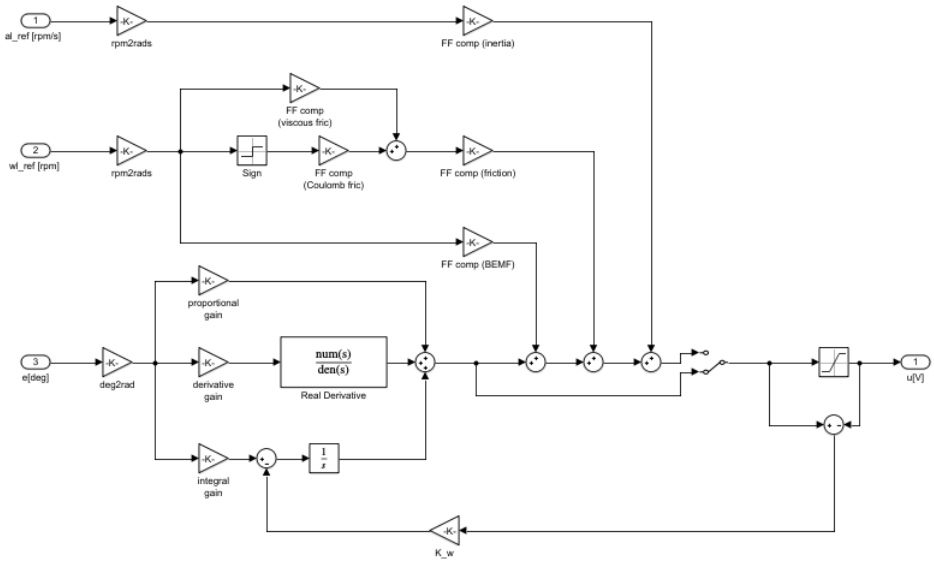


Figure 31: Feedforward scheme of the controller

- Nominal tracking design:

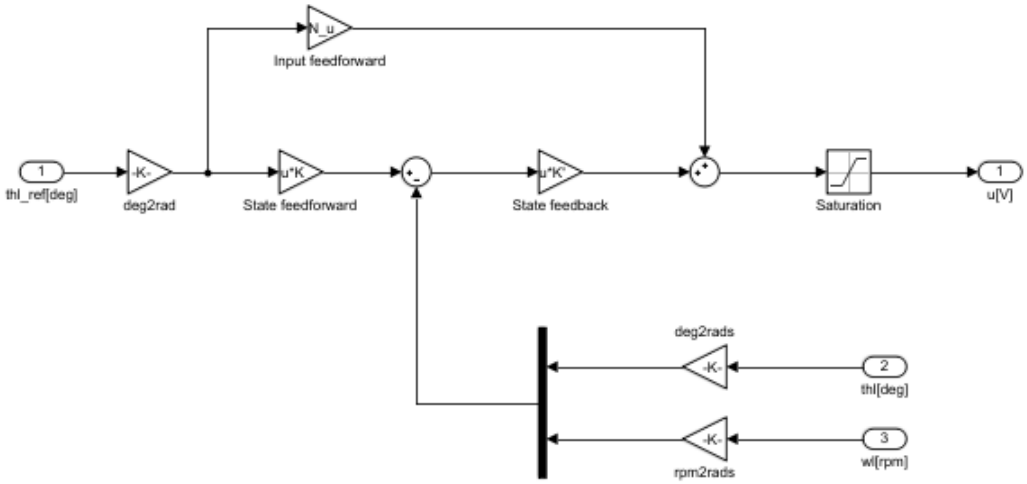


Figure 32: Nominal tracking scheme of the controller

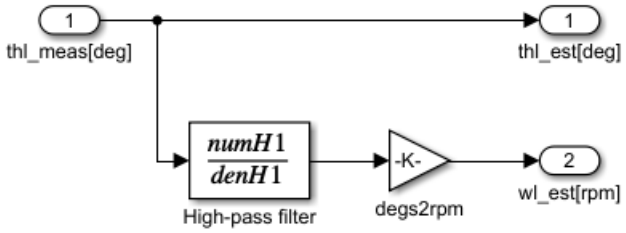


Figure 33: "Simple" state observer

- Robust tracking design with integral action (it uses the same observer of figure 33)

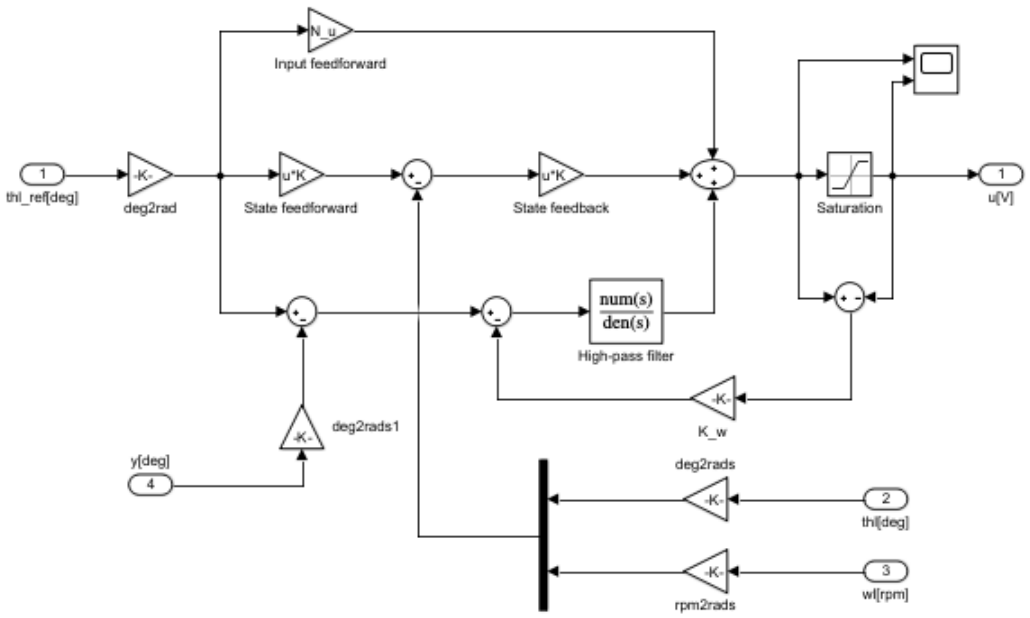


Figure 34: Robust tracking scheme of the controller with integral action

- Robust tracking design with error space approach (it uses the same observer of figure 33)

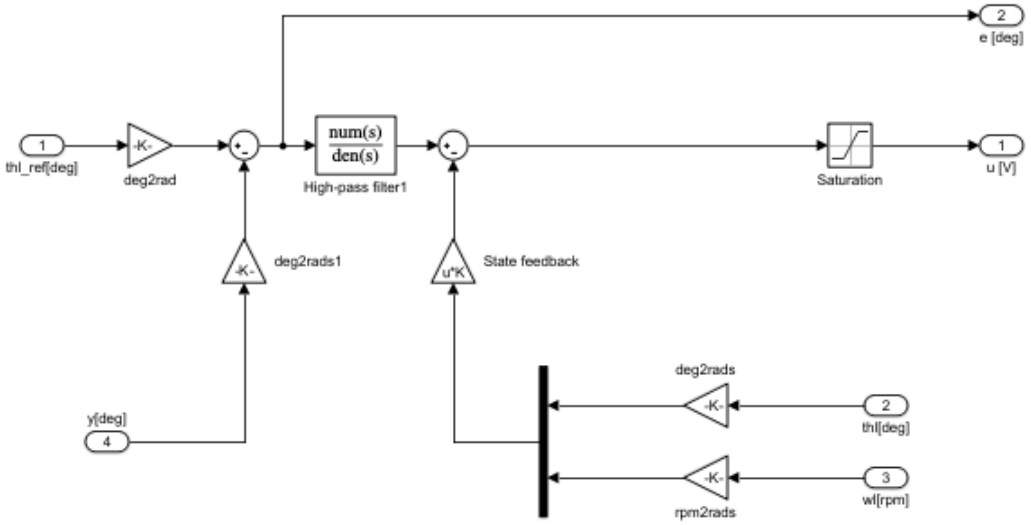


Figure 35: Robust tracking scheme of the controller with error space design

- Robust tracking design with extended-state estimator

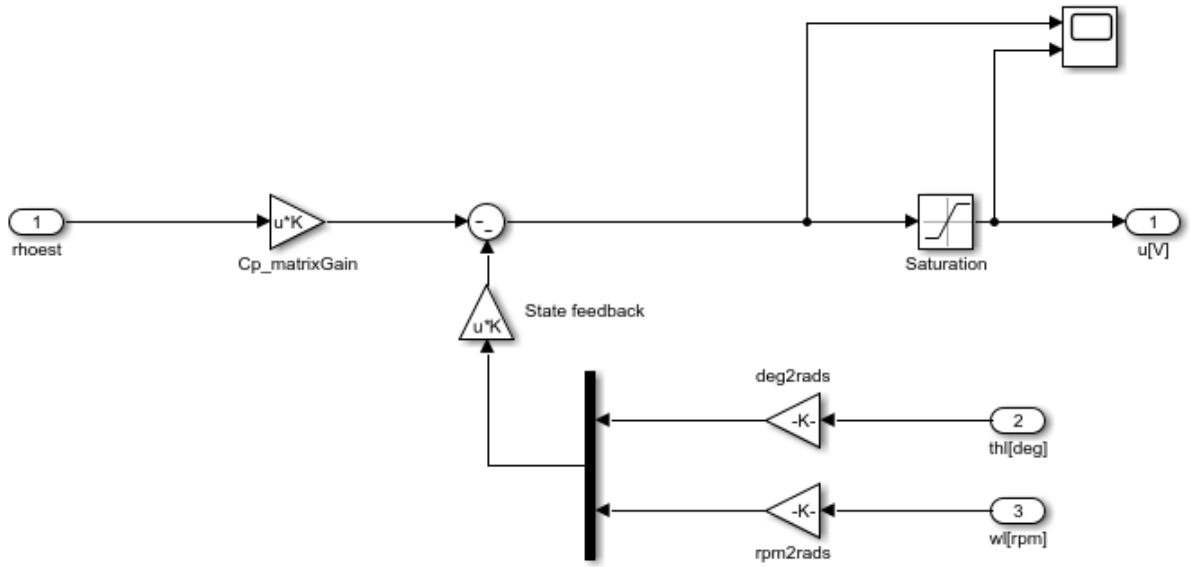


Figure 36: Position Controller

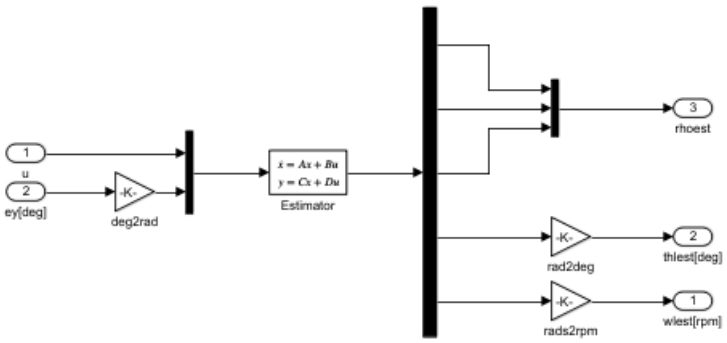


Figure 37: Observer