Laboratory activity 3: Position control of a DC servomotor with resonant load

Riccardo Antonello* Francesco Ticozzi*

May 1, 2022

1 Activity goal

The goal of this laboratory activity is to design a position controller for a DC servomotor driving a *resonant* mechanical load. For such purpose, the original *inertial* load (disc inertia) of the DC gearmotor available in laboratory is replaced with a rigid beam connected to the motor through a flexible coupling (elastic joint). The whole setup actually resembles a conventional two–mass system, which is the representative model of practically every motion system with an elastic transmission. Both the PID and state–space control techniques are taken into account for the design. The state–space design is performed with either conventional eigenvalues placement methods, or by using optimal techniques based on the Linear Quadratic Regulator (LQR).

NOTE: A new black-box model is available for this experiment. If you are choosing online-only attendance, you will need it also for the estimation part of the assignment. Please read carefully the instructions on how to "lock" the hub to perform those tests.

LAB 3 CHALLENGE: the only things you need to write down and send over are described in the last subsection of this document. The DEADLINE for this challenge (that counts for presence for on-line only attendance, potential extra point in the final grade) is MAY 17, 2022 before midnight.

2 Laboratory assignments: numerical simulations

2.1 Simulink model of the DC gearmotor with resonant load

1. Implement a Simulink model of the DC gearmotor with resonant load available in laboratory. A possible Simulink implementation is reported in Fig. 2 and Fig. 3. The parameters used in the Simulink model are defined in Listing 1.

   Note that the Simulink model of the DC gearmotor is identical to that used in the previous laboratory activities, except for the structure and parameters of the attached mechanical load. Set the model parameters as follows:

   – for the DC gearmotor, choose the nominal values reported in the handout of the laboratory activity 0 (or the introductory guide to the experimental setup). Rememeber that the equivalent inertia J_{eq} must be computed by using the hub inertia J_h, and not the disk

*Dept. of Information Engineering (DEI), University of Padova; email: {antonello, ticozzi}@dei.unipd.it

Hub moment of inertia	J_h	$6.84 \times 10^{-4}\,\mathrm{kg\,m^2}$ (1)
Hub viscous friction coefficient	B_h	estimated in previous lab activity (2)
Hub side Coulomb friction	τ_{sf}	estimated in previous lab activity
Beam moment of inertia	J_b	$1.4 \times 10^{-3}\,\mathrm{kg\,m^2}$ (1)
Beam viscous friction coefficient	B_b	to be estimated
Joint stiffness	k	to be estimated

(1) The parameters refer to the "aluminum hub". For the "Quanser hub", the parameters are:

- hub moment of inertia: $J_h = 5.1 \times 10^{-4}\,\mathrm{kg\,m^2}$
- beam moment of inertia: $J_b = 2.0 \times 10^{-3}\,\mathrm{kg\,m^2}$

(2) It is assumed equal to the coefficient B_l of the inertial load case. Therefore, the coefficient B_{eq} is equivalent to the quantity $B_{eq} = B_m + B_l/N^2$ estimated in the laboratory activity 0.

Table 1: Resonant load parameters.

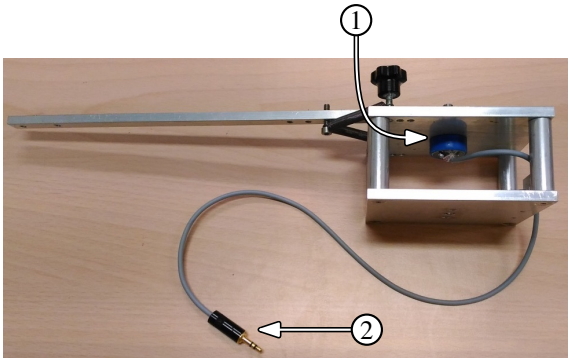
inertia J_l, as done in the previous laboratory activities.

– for the friction parameters (i.e. B_{eq} and τ_{sf}), use the values estimated in the laboratory activity 0.

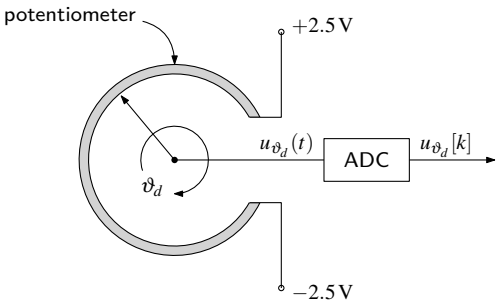
– for the resonant load parameters, use the values reported in Tab. 1. Since the beam viscous friction coefficient B_b and the joint stiffness k are unknown at this point, use the following tentative values for the numerical simulations:

$$B_b = 3.4 \times 10^{-3}\,\mathrm{Nm/(rad/s)}\,, \qquad k = 0.83\,\mathrm{Nm/rad} \tag{1}$$

2. Implement a model of the electric circuit used to measure the beam displacement ϑ_d. The circuit consists of a potentiometer (see (1) in Fig. 1a) and an analog–to–digital converter (ADC), as schematically shown in Fig. 1b. The beam is rigidly connected to the output shaft of the potentiometer. The potentiometer acts as an absolute angular position sensor, that converts the beam angular displacement into an equivalent analog voltage signal; the voltage signal is then acquired with the analog–to–digital converter (ADC) of the multi–purpose acquisition board installed on the PC workstation.



(a)



(b)

Figure 1: Beam–displacement readout circuit: (a) Hub (side view); (b) circuit schematic.

Potentiometer ohmic range	$0\,\Omega \div 10\,\mathrm{k}\Omega$
Potentiometer output voltage range	$\pm 2.5\,\mathrm{V}$
Potentiometer angular range	$\pm 170\,\mathrm{deg}$ [(1)]
Potentiometer output noise variance	$3.5 \times 10^{-7}\,\mathrm{V}^2$
ADC resolution	$16\,\mathrm{bits}$
ADC range	$\pm 10\,\mathrm{V}$

[(1)] The parameter refers to the "aluminum hub". For the "Quanser hub", the potentiometer angular range is $\pm 176\,\mathrm{deg}$.

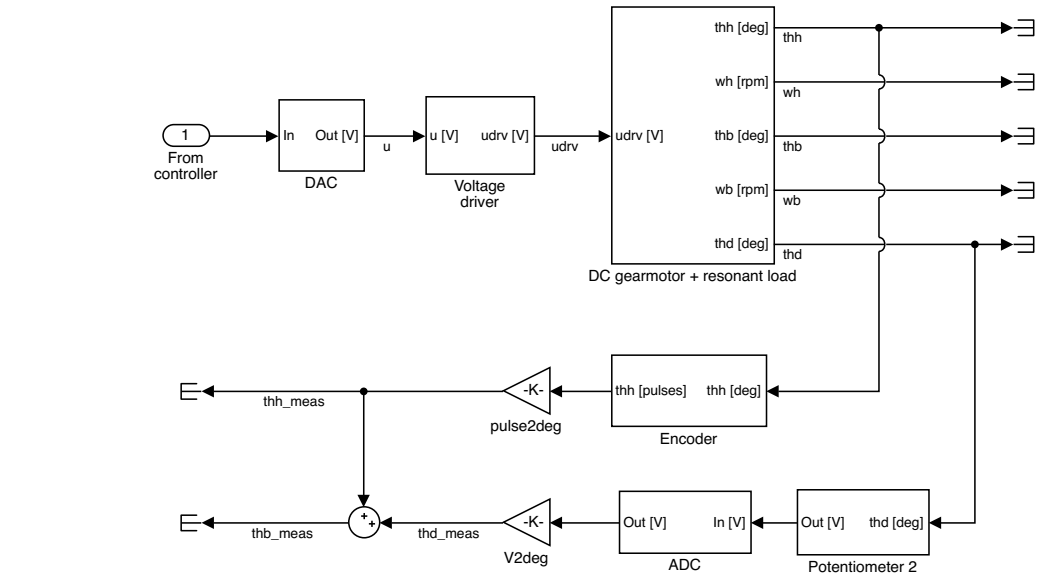
Table 2: Beam–displacement readout–circuit parameters.

The relevant parameters of the potentiometer and the ADC can be found in the introductory guide to the experimental setup ("Laboratory guide 1" - note that the potentiometer used to measured the beam position is referred there as "potentiometer 2"); for convenience, they are also reported in Tab. 2.

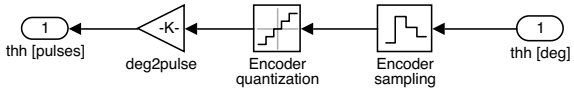
The potentiometer sensitivity, namely the angle–to–voltage conversion gain of the beam angular displacement sensor, is equal to:

$$\text{angle–to–voltage conversion gain} = \frac{\text{output voltage range}}{\text{angular range}} = \frac{2.5\,\mathrm{V}}{170\,\mathrm{deg}} \tag{2}$$

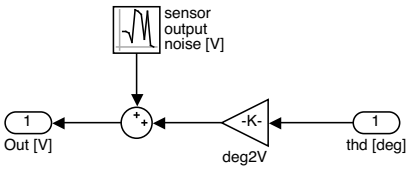
The sensor output noise is modelled with a *Sources* → *Random Number* with variance equal to $3.5 \times 10^{-7}\,\mathrm{V}^2$ (measured value) and sampling time equal to the ADC sampling time ($T_s = 1\,\mathrm{ms}$, unless otherwise stated).



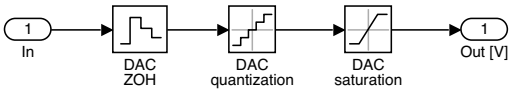
(a) Servomotor with resonant load



(b) Encoder

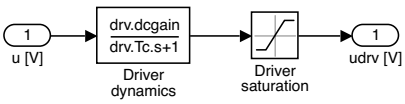


(c) Potentiometer 2



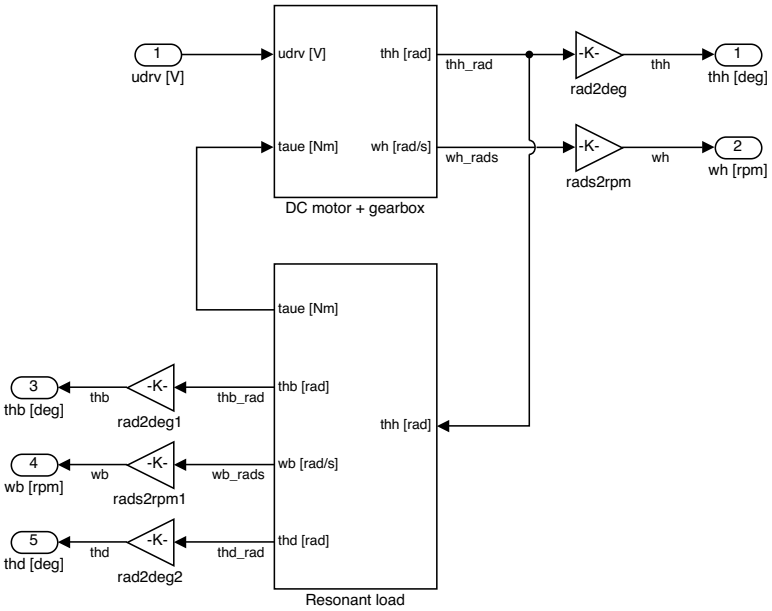
(d) DAC

(e) ADC

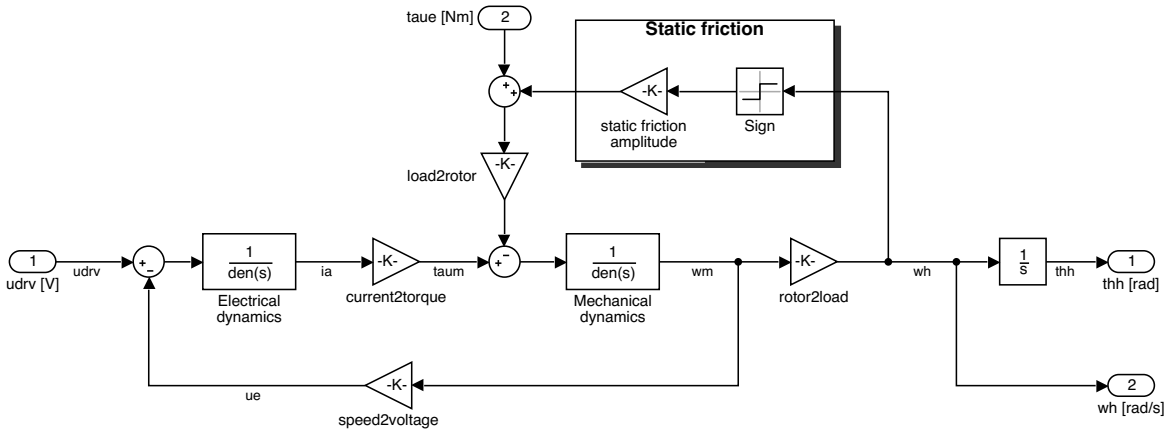


(f) Voltage driver

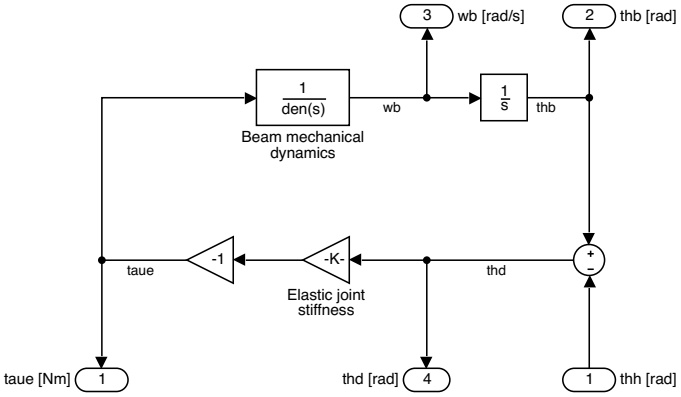
Figure 2: Simulink model implementation details: overall DC servomotor model.



(a) DC gearmotor + resonant load



(b) DC motor + gearbox



(c) Resonant load

Figure 3: Simulink model implementation details: DC gearmotor with resonant load.

Listing 1: `load_params_resonant_case.m`

```
%% 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 (mld) nominal parameters

%   hub params
mld.Jh = 6.84e-4;                   %   moment of inertia
mld.Bh = 2.5e-4;                    %   viscous friction coeff
mld.tausf = 1.0e-2;                 %   static friction (estimated)

%   beam & elastic joint params
mld.Jb = 1.4e-3;                    %   moment of inertia
mld.d  = 5.0e-2;                    %   flex joint damping coeff (estimated)
mld.wn = 24.5;                      %   flex joint natural freq (estimated)
mld.Bb = mld.Jb * 2*mld.d*mld.wn;   %   beam viscous friction
mld.k  = mld.Jb * mld.wn^2;         %   flex joint stiffness

%% 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]

%   potentiometer 2 (Spectrol 357-0-0-103) - installed on hub
sens.pot2.range.R      = 10e3;                                  %   ohmic value range
sens.pot2.range.V      = 5;                                     %   voltage range
sens.pot2.range.th_deg = 340;                                   %   angle range [deg]
sens.pot2.range.th     = sens.pot2.range.th_deg * deg2rad;       %   angle range [rad]
sens.pot2.deg2V        = sens.pot2.range.V / sens.pot2.range.th_deg;    %   sensitivity [V/deg]
sens.pot2.rad2V        = sens.pot2.range.V / sens.pot2.range.th;        %   sensitivity [V/rad]
sens.pot2.V2deg        = 1/sens.pot2.deg2V;                     %   conversion gain [V] -> [deg]
sens.pot2.V2rad        = 1/sens.pot2.rad2V;                     %   conversion gain [V] -> [rad]
sens.pot2.noise.var    = 3.5e-7;                                %   out noise variance [V^2]

%% 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
```

2.2 Position PID control

1. Design a PID controller for the *hub* position (*collocated control*) that satisfies the following specifications:
   - perfect steady–state tracking of step position references (hub side)
   - perfect steady–state rejection of constant torque disturbances (hub side)
   - step response (hub side) with settling time $t_{s,5\%} \leq 0.85\,\mathrm{s}$ and overshoot $M_p \leq 30\%$.
   - PID control structure

   Even though it has been shown in the Handout that a PD would be sufficient to stabilize the collocated system, it is evident that the integral action is required to compensate for constant disturbances entering at the plant input.

   For the design of the PID controller, use a ratio $\alpha = T_I/T_D = 4$ between the integral and time T_I and derivative times T_D. Implement the derivative action with a "real derivative", consisting of a first order high–pass filter of the type

$$H(s) = \frac{s}{T_L\, s + 1} \tag{3}$$

   with a cut-off frequency $1/T_L \approx 10\,\omega_{gc}$, where ω_{gc} is the gain crossover frequency selected for the PID controller design. Consider to implement the integrator anti–windup mechanism described in the handout of the laboratory activity 1. For the anti–windup gain $K_W = 1/T_W$, consider the tentative value equal to $T_W = t_{s,5\%}/5$.

2. Validate the design of point 1 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1.

   For the tests, use a step reference input with amplitude equal to 50° and 120°. Compare the overshoot in the step response obtained by enabling or disabling the anti–windup mechanism.

2.3 Position state–space control using eigenvalues placement design methods

For the state–space design methods considered in the next sections, assume that the state $\boldsymbol{x} = [\vartheta_h\,,\ \vartheta_d\,,\ \dot{\vartheta}_h\,,\ \dot{\vartheta}_d]^T$ of the plant model derived in equations (52)-(54) of the Handout is fully accessible. Indeed, this assumption is not true in practice, since the hub and beam deflection velocities are not directly measured with dedicated sensors. However, the two speeds can be determined with sufficient accuracy through high–pass filtering of the position measures. The cut–off frequency and roll–off of the high–pass filters ("real derivatives") used to estimate the hub and beam velocities must be chosen to avoid excessive amplification of the measurement noise at high frequency. Two possible choices are the continuous–time filter

$$H_1(s) = \frac{\omega_c^2\, s}{s^2 + 2\delta\omega_c s + \omega_c^2} \qquad \text{with} \qquad \omega_c = 2\pi\,50\,, \quad \delta = 1/\sqrt{2} \tag{4}$$

or the discrete–time filter

$$H_2(z) = \frac{1 - z^{-N}}{N\,T_s} \qquad \text{with} \qquad N = 10 \tag{5}$$

With such preliminary assumption, do the following tasks:

1. Design a state–space controller for the *hub* position (*collocated control*) that satisfies the following specifications:

   - perfect steady–state tracking of constant set–points in the *nominal* case (*nominal perfect tracking*).
   - step response (hub side) with settling time $t_{s,5\%} \leq 0.85\,\mathrm{s}$ and overshoot $M_p \leq 30\%$.

   Select the location of the eigenvalues of the closed–loop system by assuming that the dominant closed–loop dynamics can be approximated with that of a second order system with transfer function

   $$T(s) \;=\; \frac{\omega_n^2}{s^2 \,+\, 2\delta\omega_n\, s \,+\, \omega_n^2}\,, \qquad 0 \leq \delta < 1 \tag{6}$$

   whose eigenvalues (poles) are equal to

   $$\lambda_{1,2} \;=\; -\delta\omega_n \,\pm\, j\,\omega_n\sqrt{1-\delta^2} \tag{7}$$

   The natural frequency ω_n and the damping factor δ are related to the time–domain design specifications as follows (see handout of laboratory activity 0):

   $$\omega_n \;=\; \frac{3}{\delta\, t_{s,5\%}}\,, \qquad\qquad \delta \;=\; \frac{\log(1/M_p)}{\sqrt{\pi^2 + \log^2(1/M_p)}} \tag{8}$$

   Since the state model has four variables, it is necessary to place four controller eigenvalues. Consider the following placement for the four controller eignevalues:

   $$\lambda_{c,\{1,2\}} \;=\; \omega_n\, e^{j(\,-\pi \,\pm\, \varphi\,)}\,, \qquad \lambda_{c,\{3,4\}} \;=\; \omega_n\, e^{j(\,-\pi \,\pm\, \varphi/2\,)} \tag{9}$$

   where φ denotes the angle

   $$\varphi \;=\; \mathrm{atan}\,\frac{\sqrt{1-\delta^2}}{\delta} \tag{10}$$

   For the design of the state feedback matrix $\boldsymbol{K} \in \mathbb{R}^{1\times 4}$, use either the `place` or `acker` routines of the Control System Toolbox (CST).

2. Validate the design of point 1 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1. For the tests, use a step reference input with amplitude equal to 50°.

3. (**optional**) Modify the design of point 1 to guarantee *robust* perfect steady–state tracking of constant position set–points and perfect rejection of constant load disturbances. For such purpose, consider the design based on integral action that was introduced in the laboratory activity 1.

   The design is performed on the augmented–state system $\Sigma_e = (\boldsymbol{A}_e, \boldsymbol{B}_e, \boldsymbol{C}_e, 0)$, with state vector $\boldsymbol{x}_e = [x_I,\, \boldsymbol{x}]^T$, which includes the integrator state x_I among the system state variables. Since the state vector has five components, there are five eigenvalues to place in this case. Consider the following eigenvalues placement for the design of the state feedback matrix $\boldsymbol{K}_e = [K_I,\, \boldsymbol{K}] \in \mathbb{R}^{1\times 5}$:

   $$\lambda_{c,\{1,2\}} \;=\; \omega_n\, e^{j(\,-\pi \,\pm\, \varphi\,)}\,, \qquad \lambda_{c,\{3,4\}} \;=\; \omega_n\, e^{j(\,-\pi \,\pm\, \varphi/2\,)}\,, \qquad \lambda_{c,5} \;=\; -\omega_n \tag{11}$$

with ω_n, δ and φ as determined in point 1.

4. (**optional**) Validate the design of point 3 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1. For the tests, use a step reference input with amplitude equal to 50°.

2.4 Position state–space control using LQR methods

1. Reconsider the design problem of point 1 in Sec. 2.3, namely the design of a state–space controller for the *nominal* perfect tracking of constant hub position set–points.

   Instead of designing the state feedback matrix $\boldsymbol{K}$ by eigenvalue placement use the LQR methods. In particular, design the state feedback matrix $\boldsymbol{K}$ in order to minimise the following quadratic cost function

   $$J = \int_0^{+\infty} \vartheta_h^2(t) \, + \, r \, u^2(t) \, dt \qquad\qquad r > 0 \tag{12}$$

   The unique cost weight r should be chosen to satisfy the control specifications reported in point 1 in Sec. 2.3. This can be done by sketching the Symmetric Root Locus (SRL) associated with the LQR problem, as shown in Fig. 4a, and then selecting the gain $1/r$ of the locus to have all the controller eigenvalues (closed–loop eigenvalues) located within the s–plane region defined by the desired time–domain specifications (*allowable region*). A possible selection which complies with the design specifications is shown in Fig. 4b.

   For sketching the SRL, proceed as follows:

   - let `A` and `B` be the two state–space matrices (53) and (54) of the Handout; moreover, let `C=[1,0,0,0]`.
   - it is immediate to verify that if $\Sigma = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, 0)$ has transfer function $G(s)$, then $\Sigma' = (-\boldsymbol{A}, -\boldsymbol{B}, \boldsymbol{C}, 0)$ has transfer function $G(-s)$. Therefore, define the two LTI objects `sysG = ss(A,B,C,0)` and `sysGp = ss(-A,-B,C,0)`, that will be used for sketching the SRL.
   - the SRL is the root locus of a feedback system with loop transfer function equal to $G(-s)G(s)$. Hence, to plot the SRL, use the command `rlocus(sysG*sysGp)`.
   - remember that the SRL is parametrised in the gain $1/r$; moreover, for a certain gain $1/r$, the roots of the SRL are the zeros of the transfer function $1 + (1/r)\, G(-s)\, G(s)$ (regular property of any root locus). Using these two facts, try different values of r until all the stable roots of the SRL lie within the allowable region on the complex plane.

   To sketch the allowable region in the left half–plane (LHP), use the following facts:

   - to have a settling time within specifications, from (8) it follows that the real part $\sigma = -\omega_n \delta$ of the dominant closed–loop eigenvalues must be $\sigma \leq -3/t_{s,5\%}$.
   - to have an overshoot within specifications, the closed–loop eigenvalues must lie within the sector in LHP delimited by the two straight lines with slope angles $\pm\varphi$, with φ defined in (10).

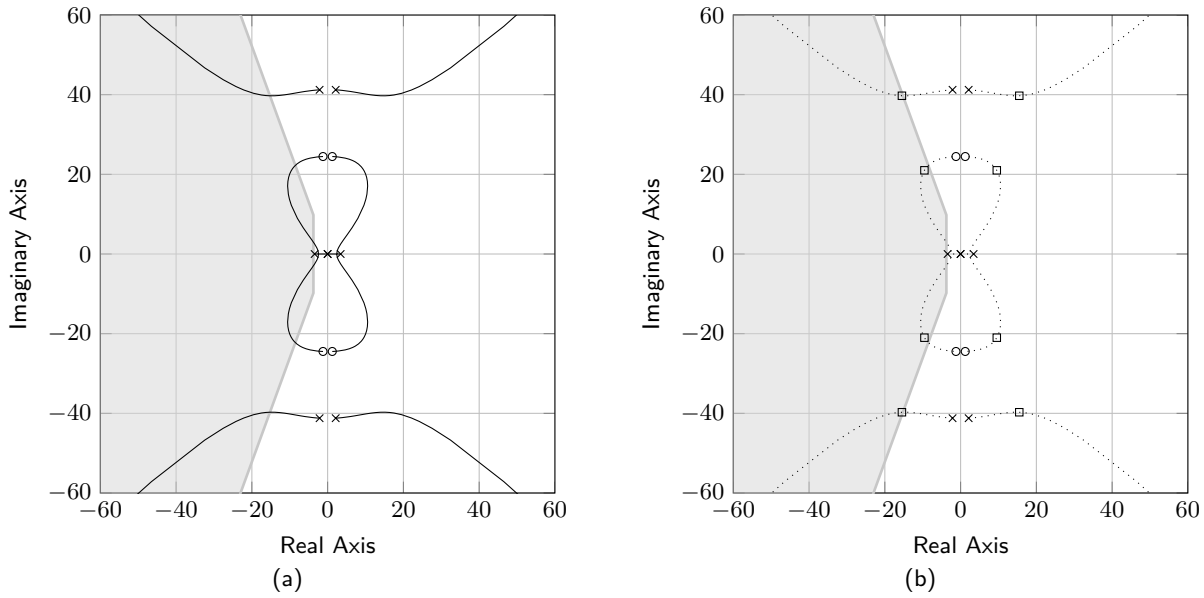


Figure 4: Symmetric root locus for the LQR design of point 1 in Sec. 2.4: (a) full SRL; (b) eigenvalues locations (square markers) for a specific r satisfying the design specifications.

Once the weight r has been determined, compute the state feedback matrix `K` by using either `K = lqr(sysG, C'*C, r)` or `K = lqry(sysG, 1, r)`.

2. Validate the design of point 1 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1. For the tests, use a step reference input with amplitude equal to 50°.

3. Reconsider the problem of point 1 of this section, which consists of designing an LQR for nominal perfect tracking of constant hub position set–points. Instead of using the cost function (12) for the LQR design, consider to use a more general cost of the type

$$J \;=\; \int_0^{+\infty} \boldsymbol{x}^T(t)\,\boldsymbol{Q}\,\boldsymbol{x}(t) \;+\; r\,u^2(t)\,dt \tag{13}$$

Select the cost weights $\boldsymbol{Q}$ and r according to the *Bryson's rule*. At steady–state, it is desired to have

$$|\vartheta_h \;-\; \vartheta_h^*| \;<\; 0.3\,\vartheta_h^*\,, \qquad |\vartheta_d| \;<\; \pi/36\;(5\,\mathrm{deg})\,, \qquad |u| \;<\; 10\,\mathrm{V} \tag{14}$$

where ϑ_h^* is the hub position set–point. Therefore, with a position reference of $50\,\mathrm{deg}$, the maximum deviations to consider for the Bryson's rule are

$$\bar{\vartheta}_h \;=\; 0.3 \times 50 \times \pi/180\,, \qquad \bar{\vartheta}_d \;=\; \pi/36\,, \qquad \bar{u} \;=\; 10 \tag{15}$$

In the $\boldsymbol{Q}$ matrix, set both the weights for the hub and beam velocities to zero.

Once the two weights $\boldsymbol{Q}$ and r have been determined, compute the state feedback matrix `K` by using `K = lqr(sysP, Q, R)`, where `sysP` is the LTI object defining the plant dynamics.

4. Validate the design of point 3 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1. For the tests, use a step reference input with amplitude equal to 50°.

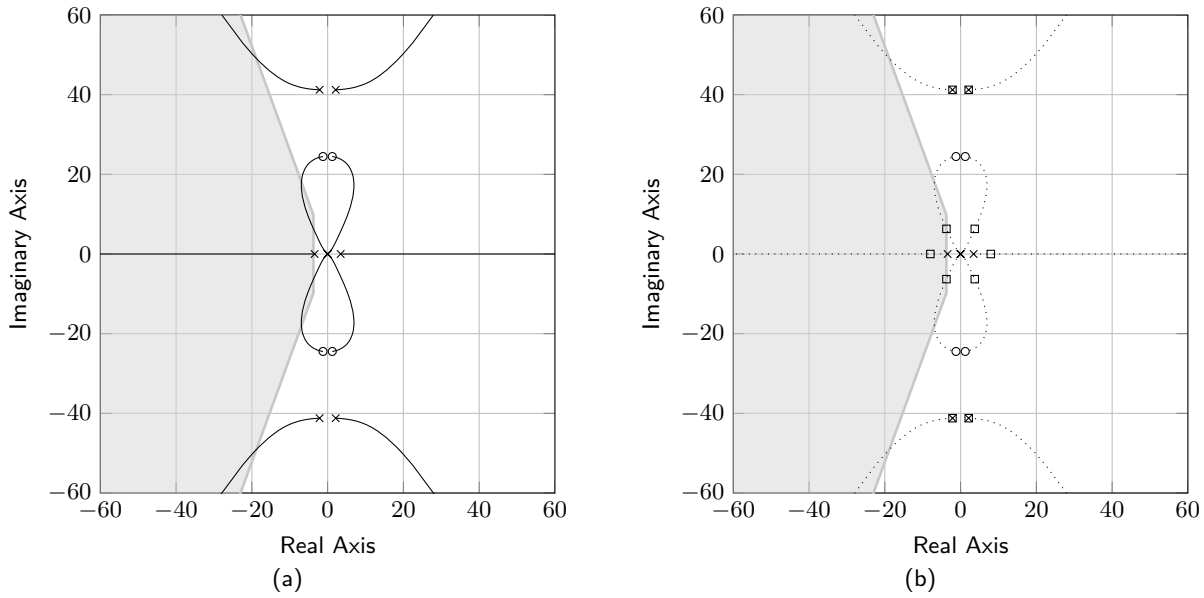


Figure 5: Symmetric root locus for the LQR design of point 5 in Sec. 2.4: (a) full SRL; (b) eigenvalues locations (square markers) for a specific r that forces the low–frequency eigenvalues to lie within the admissible region.

5. (**optional**) Reconsider the design problem of point 3 in Sec. 2.3, namely the design of a state–space controller for the *robust* perfect tracking of constant hub position set–points (and perfect rejection of constant load disturbances), using the integral action approach.

   Instead of designing the state feedback matrix $\boldsymbol{K}_e$ by eigenvalue placement, as done in Sec. 2.3, use the LQR methods explained in Sec. **??**. For the design, use the cost function

   $$J = \int_0^{+\infty} x_I^2(t) + r\, u^2(t)\, dt \qquad r > 0 \tag{16}$$

   where x_I denotes the integrator state[1]. As done in point 1, one can resort to the SRL to find the cost weight r that allows to satisfy the design specifications. To plot the SRL, follow the instruction reported in point 1, where now $\Sigma = (\boldsymbol{A}_e, \boldsymbol{B}_e, \boldsymbol{C}_e, 0)$ with $\boldsymbol{C}_e = [1, 0, 0, 0, 0]$. The plot of the SRL, together with the admissible region determined in point 1, is shown in Fig. 5a. It can be verified that there is no value for r that allows to have simultaneously all the controller eigenvalues within the admissible region. In particular, it is required a high $1/r$ gain to move the high frequency eigenvalues within the admissible region; however, with a high gain, the low frequency eigenvalues approach the SRL zeros, thus moving outside the admissible region. Therefore, for the selection of the cost weight r, restrict the attention to the low frequency eigenvalues, and choose a value that moves them within the admissible region, as show in in Fig. 5b. Alternatively, you can use the strategy proposed in point 2 of Sec. **??**, by suitably choosing ρ and α. In this scenario, the weight r should be selected by trial–and–error. Remind that the CST routine for the discrete time LQR design is `dlqr`.

6. (**optional**) Validate the design of point 5 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1. For the tests, use a step reference input

[1] Note that a cost function like (12) cannot be chosen, since the pair $(\boldsymbol{A}_e, \boldsymbol{C}_e)$ with $C_e = [0, 1, 0, 0, 0]$ is not detectable (in fact, the unobservable subsystem is the integrator of the integral action, which is not asymptotically stable), and hence the LQR cannot be designed with the methods of Thm **??**.

with amplitude equal to $50°$.

7. (**optional**) Repeat the design of point 3 by introducing the integral action in the controller, in order to achieve *robust* tracking of constant position set–points (and perfect rejection of constant load disturbances).

   In this case, the matrix $\boldsymbol{Q}$ contains an extra weight for the integrator state. This weight must be chosen different from zero, otherwise the LQR cannot be designed (explain why). Wanting to apply the Bryson's rule, it should be necessary to identify a maximum deviation of the integrator state from its steady–state value. Unfortunately, there is no reasonable and immediate choice for such quantity. Therefore, in the LQR design, consider different choices for the weight q_{11} associated with integrator state (first component of the extended state vector $\boldsymbol{x}_e = [x_I,\, \boldsymbol{x}]^T$), such as:

   $$q_{11} \in \left\{10^{-2},\, 10^{-1},\, 1,\, 10,\, 10^2\right\} \tag{17}$$

   Choose the remaining cost weights as suggested in point 3.

8. (**optional**) Validate the design of point 7 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1. Consider all the possible choices reported in (17) for the weight q_{11} associated with the integrator state x_I. For the tests, use a step reference input with amplitude equal to $50°$.

9. (**optional**) Reconsider the design of point 3. Instead of choosing the weight q_{22} for the beam displacement ϑ_d according to the Bryson's rule, consider to introduce a *frequency dependent* weight that penalizes the oscillations at the resonant frequency of the elastic joint. The frequency dependent weight can be chosen as follows:

   $$q_{22}(\omega) \,=\, q_{22}\, \frac{\omega_0^4}{\left(\omega_0^2 - \omega^2\right)^2}\,, \qquad q_{22} \,>\, 0 \tag{18}$$

   where ω_0 denotes the resonant frequency of the elastic joint. The new cost function is

   $$J_{FS} \,=\, \frac{1}{2\pi} \int_{-\infty}^{\infty} \boldsymbol{X}^*(j\omega)\, \boldsymbol{Q}(j\omega)\, \boldsymbol{X}(j\omega) \,+\, r\, U(j\omega)\, d\omega \tag{19}$$

   where

   $$\boldsymbol{Q}(j\omega) \,=\, \mathrm{diag}\left\{\frac{1}{\bar{\vartheta}_h^2}\,,\; q_{22}(\omega)\,,\; 0\,,\; 0\right\}\,, \qquad r = \frac{1}{\bar{u}^2} \tag{20}$$

   and

   $$\bar{\vartheta}_h \,=\, 5.0 \,\times\, \pi/180 \qquad \text{and} \qquad \bar{u}_h \,=\, 10 \tag{21}$$

   Compared to (15), a smaller value has been chosen for $\bar{\vartheta}_h$, so that the hub position response becomes faster (in fact, the positioning error $|\vartheta_h - \vartheta_h^*|$ is penalized with a larger weight in the cost function), and hence the mechanical resonance is excited more (i.e. its effects on the beam displacement ϑ_d are more evident, if not properly compensated with a suitable control action).

   Design a frequency–shaped LQR (FS–LQR) to minimise the cost function (19). In particular, proceed as follows:

   – determine the resonant frequency ω_0 of the elastic joint, by computing the eigenvalues

of the plant model, and then evaluating the frequency of the pair of complex conjugate eigenvalues that give rise to the resonant mode.

– note that the weight $q_{22}(\omega)$ in (18) can be factorised as follows

$$q_{22}(\omega) \,=\, H'^{*}_{Q}(j\omega)\, H'_{Q}(j\omega) \qquad \text{with} \qquad H'_{Q}(s) \,=\, \sqrt{q_{22}}\, \frac{\omega_0^2}{s^2 \,+\, \omega_0^2} \tag{22}$$

Therefore, proceed to factorise the weight matrix $\boldsymbol{Q}(j\omega)$ as follows:

$$\boldsymbol{Q}(j\omega) \,=\, \boldsymbol{H}^{*}_{Q}(j\omega)\boldsymbol{H}_{Q}(j\omega) \qquad \text{with} \qquad \boldsymbol{H}_{Q}(j\omega) \,=\, \text{diag}\left\{ \frac{1}{\bar{\vartheta}_h}\,,\; H'_{Q}(j\omega)\,,\; 0\,,\; 0 \right\} \tag{23}$$

– determine a state–space realisation $\Sigma'_{Q} = (\boldsymbol{A}_{Q'},\, \boldsymbol{B}_{Q'},\, \boldsymbol{C}_{Q'},\, D_{Q'})$ of the transfer function $H'_{Q}(j\omega)$. A possible state–space representation is

$$\boldsymbol{A}'_{Q} \,=\, \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix}, \qquad \boldsymbol{B}'_{Q} \,=\, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{C}'_{Q} \,=\, \begin{bmatrix} \sqrt{q_{22}}\,\omega_0^2\,, & 0 \end{bmatrix}, \qquad D'_{Q} \,=\, 0 \tag{24}$$

– determine a state–space realisation $\Sigma_{Q} = (\boldsymbol{A}_{Q},\, \boldsymbol{B}_{Q},\, \boldsymbol{C}_{Q},\, \boldsymbol{D}_{Q})$ of the transfer matrix $\boldsymbol{H}_{Q}(s)$. A possible choice is

$$\begin{aligned} \boldsymbol{A}_{Q} \,&=\, \boldsymbol{A}'_{Q}\,, & \boldsymbol{B}_{Q} \,&=\, \begin{bmatrix} \boldsymbol{0}_{2\times 1} & \boldsymbol{B}'_{Q} & \boldsymbol{0}_{2\times 2} \end{bmatrix} \\ \boldsymbol{C}_{Q} \,&=\, \begin{bmatrix} \boldsymbol{0}_{1\times 2} \\ \boldsymbol{C}'_{Q} \\ \boldsymbol{0}_{2\times 2} \end{bmatrix}, & \boldsymbol{D}_{Q} \,&=\, \text{diag}\left\{ \frac{1}{\bar{\vartheta}_h}\,,\; D'_{Q}\,,\; 0\,,\; 0 \right\} \end{aligned} \tag{25}$$

– form the augmented–state model $\Sigma_{A} = (\boldsymbol{A}_{A},\, \boldsymbol{B}_{A},\, \boldsymbol{C}_{A},\, D_{A})$ with state $\boldsymbol{x}_{A} = [\boldsymbol{x},\, \boldsymbol{x}_{Q}]^{T}$ and matrices

$$\boldsymbol{A}_{A} \,=\, \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{B}_{Q} & \boldsymbol{A}_{Q} \end{bmatrix}, \qquad \boldsymbol{B}_{A} \,=\, \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{C}_{A} \,=\, \begin{bmatrix} \boldsymbol{C}\,, & \boldsymbol{0}_{1\times 2} \end{bmatrix} \qquad D_{A} \,=\, 0 \tag{26}$$

Moreover, define the cost matrices for the new cost function (**??**), i.e.

$$\boldsymbol{Q}_{A} \,=\, \begin{bmatrix} \boldsymbol{D}_{Q}^{T}\boldsymbol{D}_{Q} & \boldsymbol{D}_{Q}^{T}\boldsymbol{C}_{Q} \\ \boldsymbol{C}_{Q}^{T}\boldsymbol{D}_{Q} & \boldsymbol{C}_{Q}^{T}\boldsymbol{C}_{Q} \end{bmatrix}, \qquad \boldsymbol{N}_{A} \,=\, \boldsymbol{0}\,, \qquad R_{A} \,=\, r \tag{27}$$

– solve the standard LQR problem with cost function (**??**) for the augmented–state model Σ_{A} defined in the previous point. Use the command `K = lqr(AA, BA, QA, RA)` to compute the state feedback matrix $\boldsymbol{K}_{A}$ (with obvious meaning of the variables `AA`, `BA`, `QA`, `RA`).

– the feedforward gains $\boldsymbol{N}_{x_A} \in \mathbb{R}^{6\times 1}$ and $N_u \in \mathbb{R}$ (refer to the handout of laboratory activity 1) are computed by solving the following system of linear equations

$$\begin{bmatrix} \boldsymbol{A}_{A} & \boldsymbol{B}_{A} \\ \boldsymbol{C}_{A} & D_{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{N}_{x_A} \\ N_u \end{bmatrix} \,=\, \begin{bmatrix} \boldsymbol{0}_{6\times 1} \\ 1 \end{bmatrix} \tag{28}$$

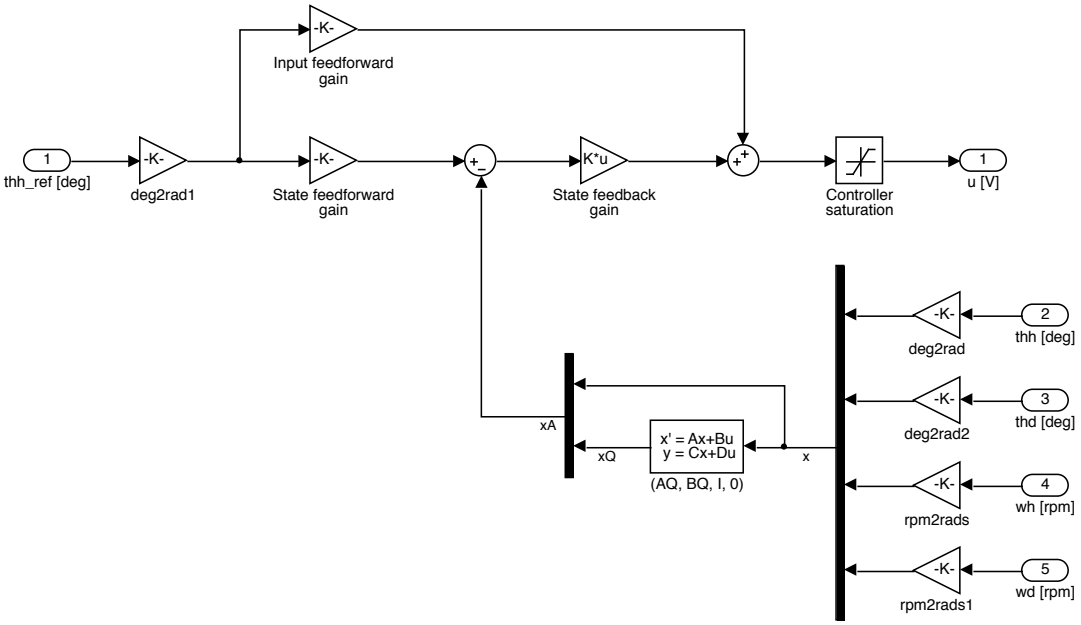


Figure 6: Possible Simulink implementation of the FS–LQR of point 9 in Sec. 2.4 for nominal perfect tracking of constant set–points.

As discussed in class, the FS–LQR can be implemented as a conventional LQR operating on the augmented state $\boldsymbol{x}_A$. A possible Simulink implementation of the FS–LQR is shown in Fig. 6.

10. (**optional**) Validate the design of point 9 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1.

    Consider the following choices for the gain q_{22} in the weight (18):

    $$q_{22} \in \{0.01,\, 1,\, 100\} \tag{29}$$

    For each option, evaluate how much the resonant mode on the beam deflection ϑ_d is attenuated by the FS–LQR, in response to a hub step reference input with amplitude equal to 50°.

11. (**optional**) Repeat the design of point 9 by introducing the integral action in the controller, in order to achieve *robust* tracking of constant position set–points (and perfect rejection of constant load disturbances).

    The new design procedure differs from the previous one only for few details, that are reported below:

    – after defining the augmented–state model Σ_A with matrices (26), it is necessary to further extend the state vector $\boldsymbol{x}_A$ by adding the integrator state x_I. The extended–state model $\Sigma_e = (\boldsymbol{A}_e,\, \boldsymbol{B}_e,\, \boldsymbol{C}_e,\, 0)$ with state vector $\boldsymbol{x}_e = [x_I,\, \boldsymbol{x}_A]^T$ is (see also the handout of laboratory activity 1)

    $$\boldsymbol{A}_e = \begin{bmatrix} 0 & \boldsymbol{C}_A \\ \boldsymbol{0}_{6\times 1} & \boldsymbol{A}_A \end{bmatrix}, \qquad \boldsymbol{B}_e = \begin{bmatrix} 0 \\ \boldsymbol{B}_A \end{bmatrix}, \qquad \boldsymbol{C}_e = \begin{bmatrix} 0, & \boldsymbol{C}_A \end{bmatrix} \tag{30}$$

    – the cost matrix $\boldsymbol{Q}_A$ in (27) must be further extended to include the weight for the

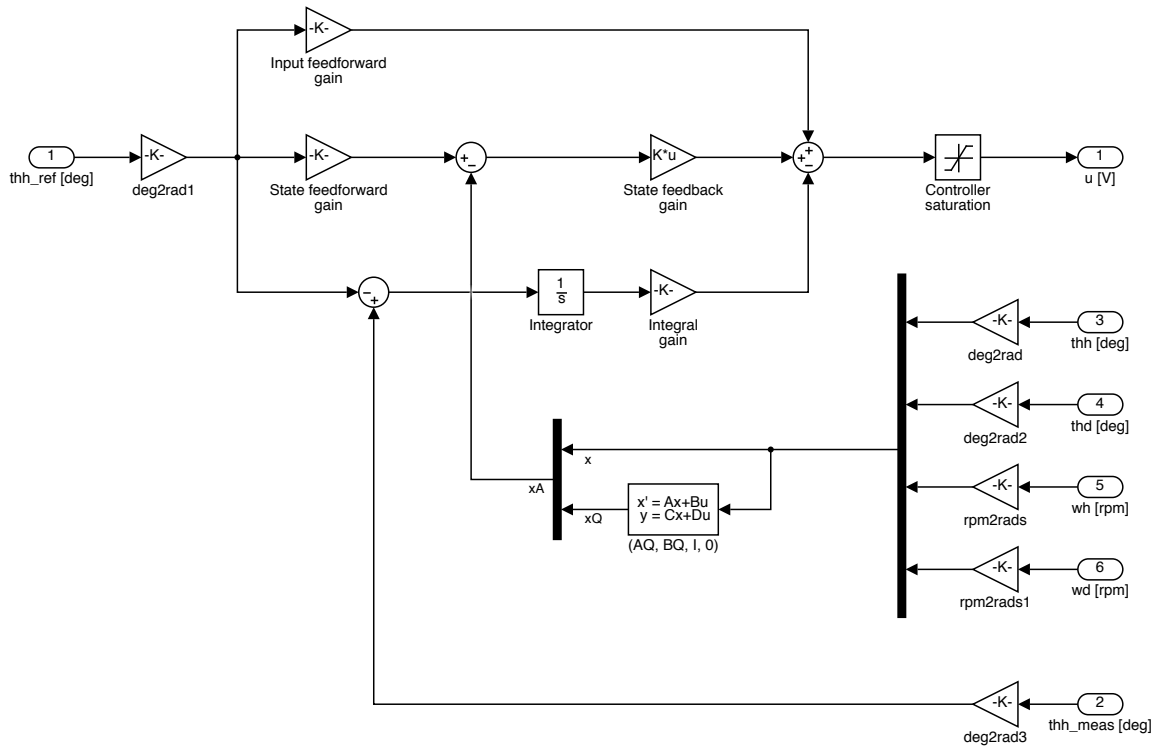


Figure 7: Possible Simulink implementation of the FS–LQR of point 11 in Sec. 2.4 for robust perfect tracking of constant set–points, using the integral action approach.

integrator state, i.e. define the new cost weights

$$\boldsymbol{Q}_e = \begin{bmatrix} q_I & \boldsymbol{0}_{1\times 6} \\ \boldsymbol{0}_{6\times 1} & \boldsymbol{Q}_A \end{bmatrix}, \qquad R_e = R_A \tag{31}$$

where $q_I > 0$ is the weight associated with the integrator state x_I.

– obtain the state feedback matrix $\boldsymbol{K}_e = [K_I, \boldsymbol{K}_A] \in \mathbb{R}^{1\times 7}$ by solving the LQR design problem for the extended–state model Σ_e, with cost weights (31).

The feedforward gains $\boldsymbol{N}_{x_A} \in \mathbb{R}^{6\times 1}$ and $N_u \in \mathbb{R}$ are instead computed as in the previous point 9.

A possible Simulink implementation of the FS–LQR with integral action is shown in Fig. 7.

12. (**optional**) Validate the design of point 11 in simulation, using the Simulink model of the DC gearmotor with resonant load developed in Sec. 2.1.

For the gain q_{22} in the weight (18), consider the options reported in (29). For the cost q_I associated with the integrator state, use $q_I = 10$ as a first trial.

For each alternative of the q_{22} gain, evaluate how much the resonant mode on the beam deflection ϑ_d is attenuated by the FS–LQR, in response to a step hub–position reference with amplitude equal to $50°$.

3 Laboratory assignments: experimental tests

3.1 Preliminary configurations for enabling the beam deflection measurement

The beam deflection measurement ϑ_d required for the experimental tests can be retrieved as follows:

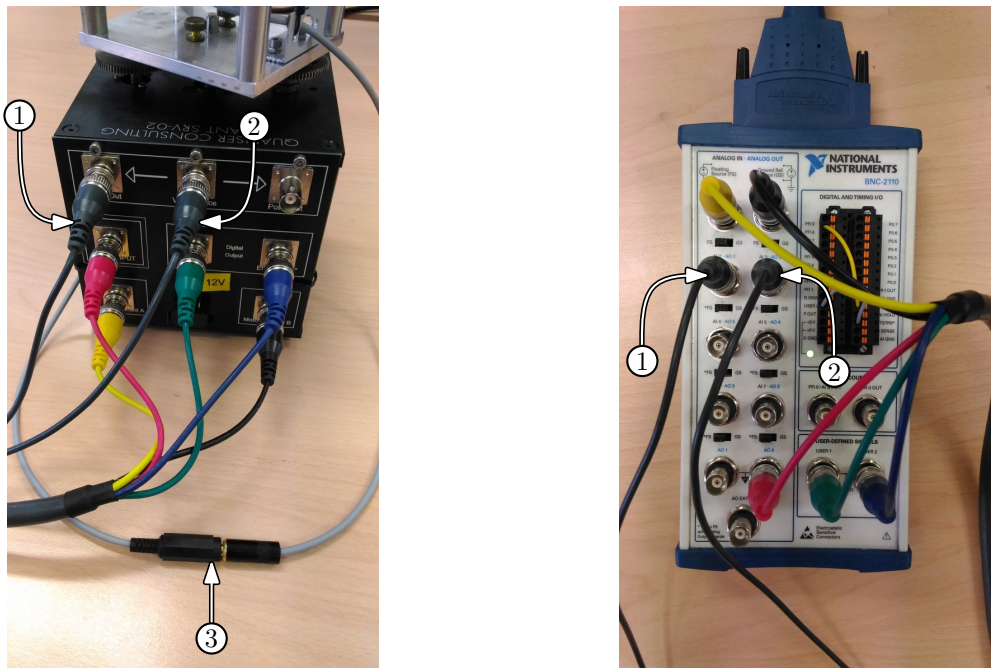
1. After installing the hub on the output shaft of the DC servomotor unit, connect the "potentiometer 2" of the hub with the base unit, using the ad–hoc audio jack connector (see ③ in Fig. 8a).

2. Connect the "Pot 2 Out" and "V Reference" outputs of the SRV–02 unit to the "AI 2" and "AI 3" analog input channels of the BNC–2110 terminal board (see ① and ② in Fig. 8a–8b).

3. Use a *Simulink Desktop Real-Time* $\rightarrow$ *Analog Input* block to read the two analog input channels (see Fig. 9a). For such purpose, configure the block by setting `[3,4]` in the *Input channels* field of the *Block parameters* dialog window; moreover, set the *Input range* field equal to "-5 to $5\,\mathrm{V}$".

4. The beam deflection ϑ_d can be obtained by computing the difference between the "Pot 2 Out" and "V Reference" output voltages, and then multiplying the voltage difference by the "voltage to angular displacement" sensitivity of the potentiometer 2, e.g. $340\,\mathrm{deg}/5\mathrm{V}$ for an angular displacement measured in $[\mathrm{deg}]$ units (see Fig. 9c).

5. It could happen that at the beam equilibrium position, the measured displacement is not exactly zero (measurement *bias*). This happens because of a non–perfect alignment of the potentiometer central position with the beam equilibrium position. To remove the measurement bias, consider to implement the bias removal scheme shown in Fig. 9c–9e. The scheme consists



(a) (b)

Figure 8: Experimental setup configuration: (a) front panel connections; (b) terminal board connections.

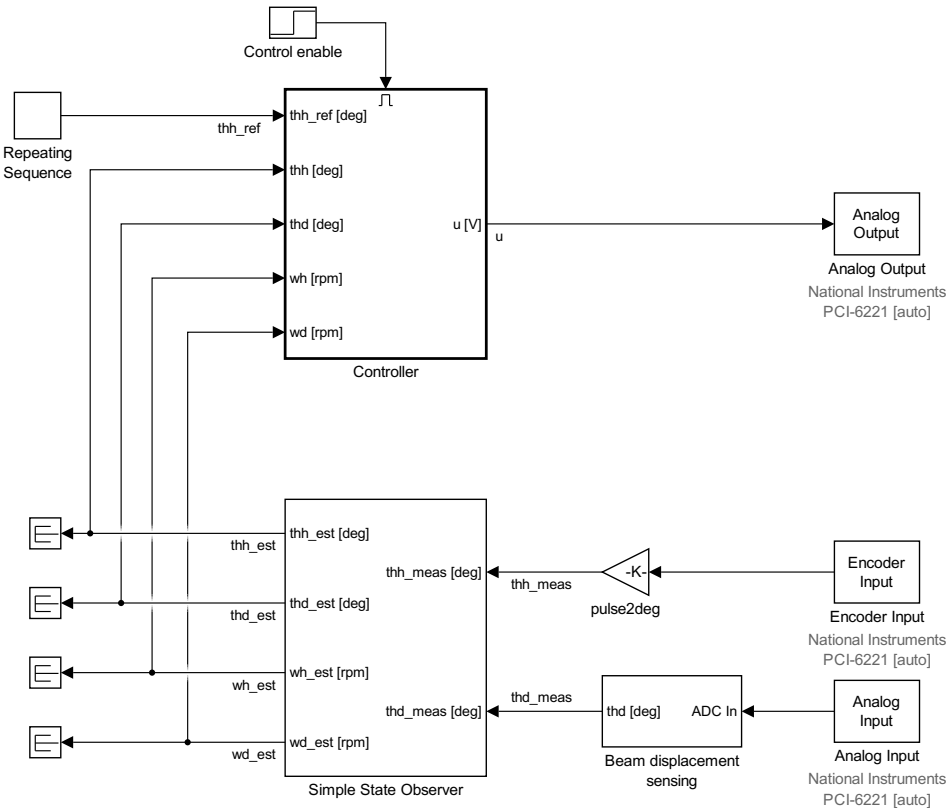
of estimating the bias at the startup, before activating the controller, simply by averaging the measured displacement over a time interval $[t_0, t_1]$, i.e.

$$\vartheta_{d,bias} = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} \vartheta_d(t)\, dt \tag{32}$$

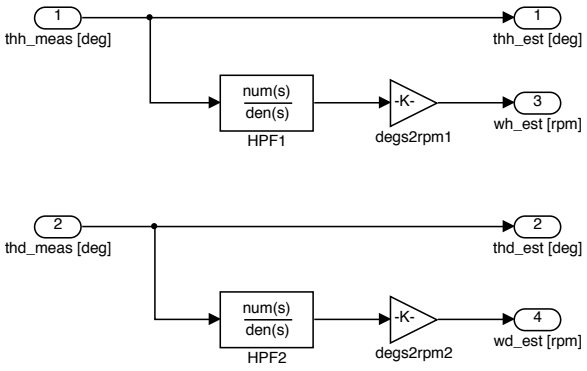
The average can be computed with a *Discrete* $\rightarrow$ *Discrete–Time Integrator* with the *Gain value* field set equal to $1/(t_1 - t_0)$, placed within a subsystem that is enabled only over the time interval $[t_0, t_1]$. For enabling the execution of the subsystem only when the specified condition holds, place a *Ports & Subsystems* $\rightarrow$ *Enable* block within the subsystem, as shown in Fig. 9e. Use a *Logic and Bit Operations* $\rightarrow$ *Interval Test* block to trigger the execution of the subsystem only when the simulation time (provided by a *Sources* $\rightarrow$ *Clock*) is within the time interval $[t_0, t_1]$ (see Fig. 9d). For such purpose, set the *Lower Limit* and *Upper Limit* fields of the *Interval Test* block equal to, respectively, the values t_0 and t_1.

The estimated bias obtained for $t \geq t_1$ can be removed from the beam deflection measurement as shown in Fig. 9c, to obtain a bias free measurement. Note that the controller should be activated only after the removal of the measurement bias, as shown in Fig. 9a (use another *Enable* block within the controller subsystem, to enable the execution only after the time instant t_1).

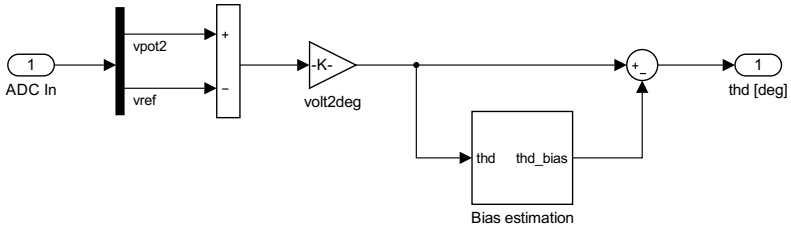
Consider to choose $t_0 = 0.2\,\mathrm{s}$ and $t_1 = 0.7\,\mathrm{s}$ for the experiments. The controller should be activated only after the time instant t_1, i.e. when the bias has been removed from the beam displacement measurement.



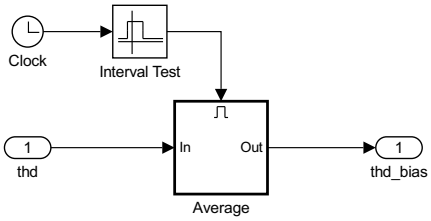
(a) State controller & observer



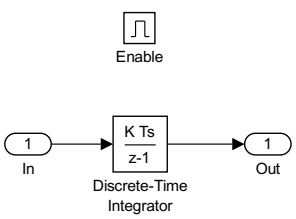
(b) "Simple" state observer



(c) Beam displacement sensing



(d) Bias estimation



(e) Average

Figure 9: Possible Simulink implementation of the beam displacement sensing block with measurement bias removal.

3.2 Estimation of the elastic joint parameters

1. Estimate the natural frequency ω_n and the damping factor δ of the elastic joint with the procedure described in the Handout. Since the behaviour of the elastic joint is linear only for small angles, consider to never exceed an initial deflection of 20° when measuring the natural response required for the estimation procedure. During the tests, make sure to keep the hub position locked.

   Say $\hat{\omega}_n$ and $\hat{\delta}$ the estimated joint parameters; then, by using the nominal value $J_b = 1.4 \times 10^{-3}\,\mathrm{kg\,m^2}$ of the beam inertia, obtain the estimates $\hat{B}_b$ and $\hat{k}$ of, respectively, the beam viscous friction coefficient and the elastic joint stiffness as follows:

$$\hat{B}_b = J_b \times \left(2\,\hat{\delta}\,\hat{\omega}_n\right), \qquad \hat{k} = J_b \times \hat{\omega}_n^2 \tag{33}$$

3.3 Position PID control

1. Repeat the PID control design of point 1 in Sec. 2.2 with the new elastic joint parameters estimated in point 1 of Sec. 3.2.

2. Validate the PID controller redesigned in point 1 on the experimental setup.

   Perform the experimental tests by adopting the same methodology used for the numerical simulations of point 2 in Sec. 2.2.

3.4 Position state–space control using eigenvalues placement design methods

1. Repeat the control designs listed below:

   - the state–space controller for nominal tracking of constant hub position set–points obtained in point 1 of Sec. 2.3;
   - (**optional**) the state–space controller with integral action for robust tracking of constant hub position set–points obtained in point 3 of Sec. 2.3;

   using the new elastic joint parameters estimated in point 1 of Sec. 3.2.

2. Validate the designs of point 1 on the experimental setup.

   Perform each experimental test by adopting the same methodology used for the corresponding numerical simulation.

3.5 Position state–space control using LQR methods

1. Repeat the control designs listed below:

   - the LQR obtained in point 1 of Sec. 2.4, using argumentations based on SRL;
   - the LQR obtained in point 3 of Sec. 2.4, using the Bryson's rule;
   - (**optional**) the LQR with integral action obtained in point 5 of Sec. 2.4, using argumentations based on SRL;
   - (**optional**) the LQR with integral action obtained in point 7 of Sec. 2.4, using the Bryson's rule;

- (**optional**) the FS–LQR obtained in point 9 of Sec. 2.4;
- (**optional**) the FS–LQR with integral action obtained in point 11 of Sec. 2.4;

using the new elastic joint parameters estimated in point 1 of Sec. 3.2.

2. Validate the designs of point 1 on the experimental setup.

   Perform each experimental test by adopting the same methodology used for the corresponding numerical simulation.

4 LAB 3 CHALLENGE

Design, using one or a combinations of the techniques presented in the course, a control system for the motor with resonant load such that:

1. **It ensures asymptotic tracking of step references;**
2. **It ensures an overshoot $Mp \leq 30\%$ for a 50 deg step reference;**
3. **It attains the shortest settling time $t_{s,5\%}$ you are able to achieve for the same reference;**

A pdf file (2 pages max.) including:

1. **the description of the control structure and strategy;**
2. **the parameters and values needed to implement the controller;**
3. **a plot of the response to the 50 degree step;**
4. **the corresponding smallest value of t_s you attained;**

should be uploaded in the repository by May 17, 2022 (before midnight).