Laboratory activity 0: Position PID–control of a DC servomotor

(Laboratory Assignments)

Riccardo Antonello* Francesco Ticozzi*

March 19, 2025

1 Activity goal

The goal of this laboratory activity is to design a PID position controller for the DC servomotor available in the laboratory. The design is carried out in the frequency domain (i.e., using the *Bode's method*) with the adoption of a nominal model of the motor, whose parameters are deduced from the datasheets. In the second part of the activity, the motor parameters, including static and viscous friction, are estimated through simple experimental tests. These parameters will be used in the next laboratory activity to improve control performance, particularly by implementing a feedforward inertia and friction compensation scheme.

2 Laboratory assignments: numerical simulations

2.1 Simulink model of the DC servomotor with inertial load

1) Implement a Simulink model of the DC gearmotor available in laboratory, according to the block diagram and the the model parameters shown in the Handout. Since the nominal values of the friction parameters are still unknown (they will be estimated in a following laboratory activity), use the following tentative values to test the Simulink model:

$$B_{eq} = 2.0 \times 10^{-6}\,\mathrm{Nm/(rad/s)}\,, \qquad \tau_{sf} = 1.0 \times 10^{-2}\,\mathrm{N\,m} \tag{1}$$

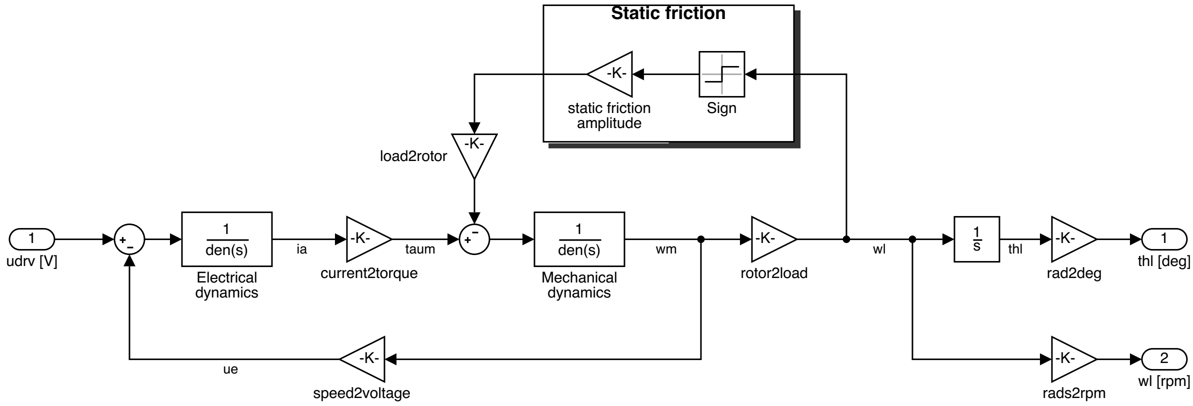
Moreover, remind that $J_l = J_d + 3\,J_{72}$. The position and speed variables of the model are measured in rad and $[\mathrm{rad/s}]$, respectively. However, for analysis purposes, it is preferred to measure the position in $[\mathrm{deg}]$ units, and the speed in $[\mathrm{rpm}]$ (rotations per minute). Therefore, consider to insert the following conversion gains in the Simulink model:

- conversion gain `rad2deg` from $[\mathrm{rad}]$ to $[\mathrm{deg}]$ units at the motor position output (`rad2deg=180/pi`).
- conversion gain `rads2rpm` from $[\mathrm{rad/s}]$ to $[\mathrm{rpm}]$ units at the motor speed output (`rads2rpm=60/2/pi`).

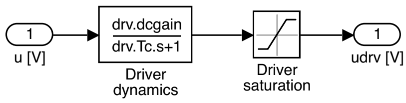
A possible implementation of the Simulink model is shown in Fig. 1a. The parameters used in the Simulink model are defined in the Listing 1.

2) Implement the model of the voltage driver as shown in Fig. 1b. The model includes a *Continuous*

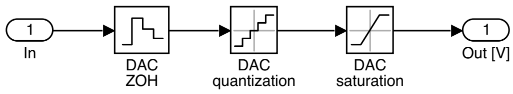
*Dept. of Information Engineering (DEI), University of Padova; email: `{antonello, ticozzi}@dei.unipd.it`



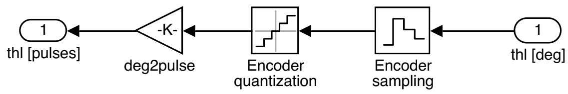
(a) DC motor + gearbox



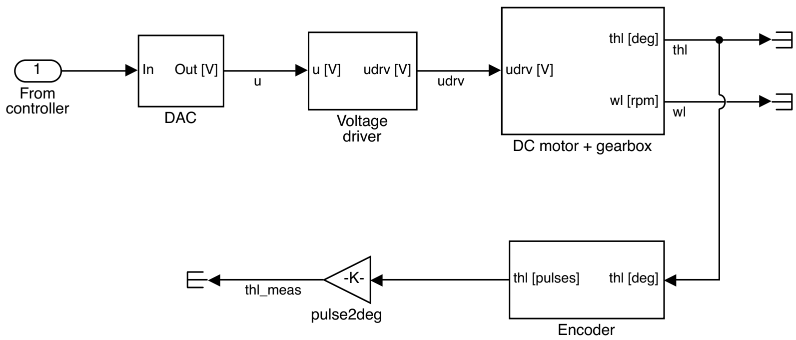
(b) Voltage driver



(c) DAC



(d) Encoder



(e) DC servomotor (top model)

Figure 1: Detailed Simulink model of the DC servomotor.

→ *Transfer Function* block specifying the transfer function

$$H_{drv}(s) \;=\; \frac{k_{drv}}{T_{drv}\, s \,+\, 1} \tag{2}$$

which defines the driver lowpass dynamics, and a *Discontinuities* → *Saturation* block that accounts for the voltage limitation of the power amplifier. Since the amplifier output cannot exceed the voltage supply levels of $\pm 12\,\mathrm{V}$, set the saturation levels to those values.

3) Implement the model of the digital–to–analog (DAC) converter, which provides the reference voltage u to the voltage driver, as shown in Fig. 1c. The model includes a *Discrete* → *Zero–Order Hold* block to simulate the holding process of the DAC output within each sampling period, and a *Discontinuities* → *Quantizer* block to simulate the quantisation of the DAC output due to the finite resolution of the converter. Finally, a *Discontinuities* → *Saturation* block accounts for the finite DAC output range. Set the sampling time of the ZOH block to $T_s \;=\; 1\,\mathrm{ms}$, which is the sampling time that will be used to run all the experiments. Set the levels of the saturation block to $\pm 10\,\mathrm{V}$, which correspond to the positive/negative full–scale values of the DAC. Since the DAC resolution is equal to $16\,\mathrm{bits}$, set the quantisation interval of the quantiser block equal to $q = 20/(2^{16} - 1)\mathrm{V}$.

4) Implement the model of the incremental optical encoder (used to measure the load position) as shown in Fig. 1d. The model includes a *Discrete* → *Zero–Order Hold* block to simulate the ideal sampling of the *load–side* position measurement[1], and a *Discontinuities* → *Quantizer* block to account for the finite resolution of the position transducer. Set the sampling time of the ZOH block to $T_s \;=\; 1\,\mathrm{ms}$, which is the sampling time that will be used to run all the experiments. Set the quantisation interval q of the quantiser block according to the resolution of the encoder installed on the Quanser SRV–02 unit used for the experiments:

- for all the units except “MOTORE 8” and “MOTORE 10”: $q = 360/(500 \times 4)\,\mathrm{deg}$
- for the “MOTORE 8” and “MOTORE 10” units: $q = 360/(1024 \times 4)\,\mathrm{deg}$

Since the encoder output is the “pulse count”, rather than the position measurement in $[\mathrm{deg}]$ units, insert a *Math* → *Gain* block at the quantiser output to perform the conversion from degrees units to pulse count. For such purpose, set the gain equal to $1/q$.

5) Combine all the models prepared in points 1)–4), to form a Simulink model of the whole DC servomotor, as shown in Fig. 1e. It is desired to have a position measurement in $[\mathrm{deg}]$ units. Therefore, consider to insert a `pulse2deg` conversion gain at the encoder output, to perform the conversion from “pulse count” to $[\mathrm{deg}]$ units. The gain is equal to `pulse2deg=360/ppr`, where `ppr` is the number of encoder pulses per rotation (i.e. 500×4 on all the motor units except “MOTORE 8” and “MOTORE 10”; 1024×4 on “MOTORE 8” and “MOTORE 10” units).

[1] Note that there is no specific block for the ideal sampler in Simulink. Although counterintuitive, the conventional way of implementing an ideal sampler in Simulink consists of using a *Discrete* → *Zero–Order Hold* block, which is also used (as its name states) to implement the zero–order discrete–to–continuous time interpolator.

Listing 1: `load_params_inertial_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 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)
```

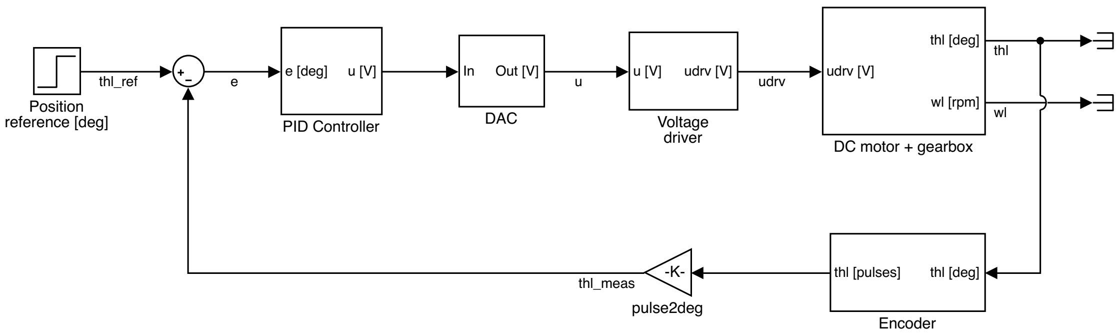
```
drv.C1 = 100e-9;                  %   op-amp input capacitor
drv.outmax = 12;                  %   op-amp max output voltage

%   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
```

2.2 Design and numerical validation of the position PID controller

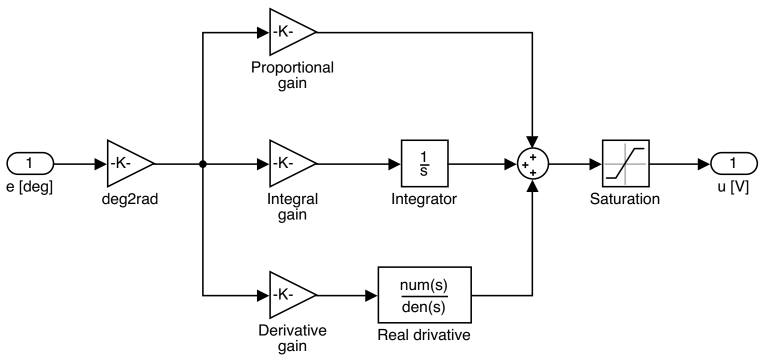
1) Design a position control system for the DC servomotor based on a standard PID controller. The control design specifications are:

- 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 $t_{s,5\%} \leq 0.15\,\mathrm{s}$ and overshoot $M_p \leq 10\%$

Use the Bode's method for the determination of the three controller gains. For the design, use the simplified plant model. Note that such model is specified with inputs in $[\mathrm{V}]$ units, and output in $[\mathrm{rad}]$ units. Therefore, the PID controller will be designed to accept a tracking error with $[\mathrm{rad}]$ units, and to generate a control signal in $[\mathrm{V}]$ units.



(a) Position PID control loop (top model)



(b) PID controller

Figure 2: Detailed Simulink model of the position PID control system.

Since the total viscous friction is unknown and will be estimated later, set $B_{eq} = 0$ in the plant transfer function. For the ratio $\alpha = T_I/T_D$, choose $\alpha = 4$ as a first trial.

Consider to implement a Matlab function to design the PID controller. This function should determine the most suitable controller structure (P, PI, PD, or PID) based on the design specifications and return the corresponding controller gains K_P, K_I and K_D.

2) Implement a Simulink model of the position PID control loop, that includes the accurate model of the DC servomotor developed in the previous Sec. 2.1. For the PID implementation, use a "real derivative" block based on a first order high–pass filter with T_L selected according to the guideline reported in the handout. A possible implementation of the Simulink model is shown in Fig. 2.

3) Simulate the response of the position control loop implemented in point 2) to different step reference inputs. Consider the following values for the step amplitude: $10°$, $30°$, $50°$, $90°$, $180°$, $360°$.

For the simulations, use the *Variable–step ode45 (Dormand–Prince)* solver, which can be set in *Configuration Parameters* $\rightarrow$ *Solver* $\rightarrow$ *Type* and *Solver*.

Since a variable-step solver adapts the step size based on system dynamics, the simulation may hang when the speed becomes very small. This occurs because the solver struggles to find an adequately small step size where the speed does not change sign, potentially triggering a "too many zero crossings" error in Simulink. To avoid this issue, deselect the *Enable zero–crossing detection* in the *Block Parameters* window of the *Sign* block used to model the static friction torque.

3 Laboratory assignments: experimental tests

3.1 Validation of the position PID controller with the motor black-box model

1) To test the position PID controller on the motor on the real DC servomotor, follow these steps:

- make a copy of the Simulink model prepared in point 2) of Sec. 2.2 for the numerical simulations. Refer to Fig. 2 and Fig. 3a for a possible implementation.
- replace the model of the DC servomotor with the blocks of the Simulink Desktop Real–Time (SLDRT) toolbox that enable the interaction with the experimental device, as shown in Fig. 3b. In addition to send the voltage command to the DAC (using the *Analog Output* block) and read the pulse count from the encoder (using the *Encoder Input* block), implement a block for sensing the motor current. Follow the instructions provided in the introductory guide to the experimental setup to implement such block (see also the implementation shown in Fig. 3b).
- configure the simulation parameters of the new Simulink model to perform a "real–time simulation". For the purpose, consult the instructions provided in the introductory guide to the experimental setup. In particular, choose a *fixed–step ode3* solver, with step size equal to $T_s = 1\,\mathrm{ms}$ (controller sampling time).

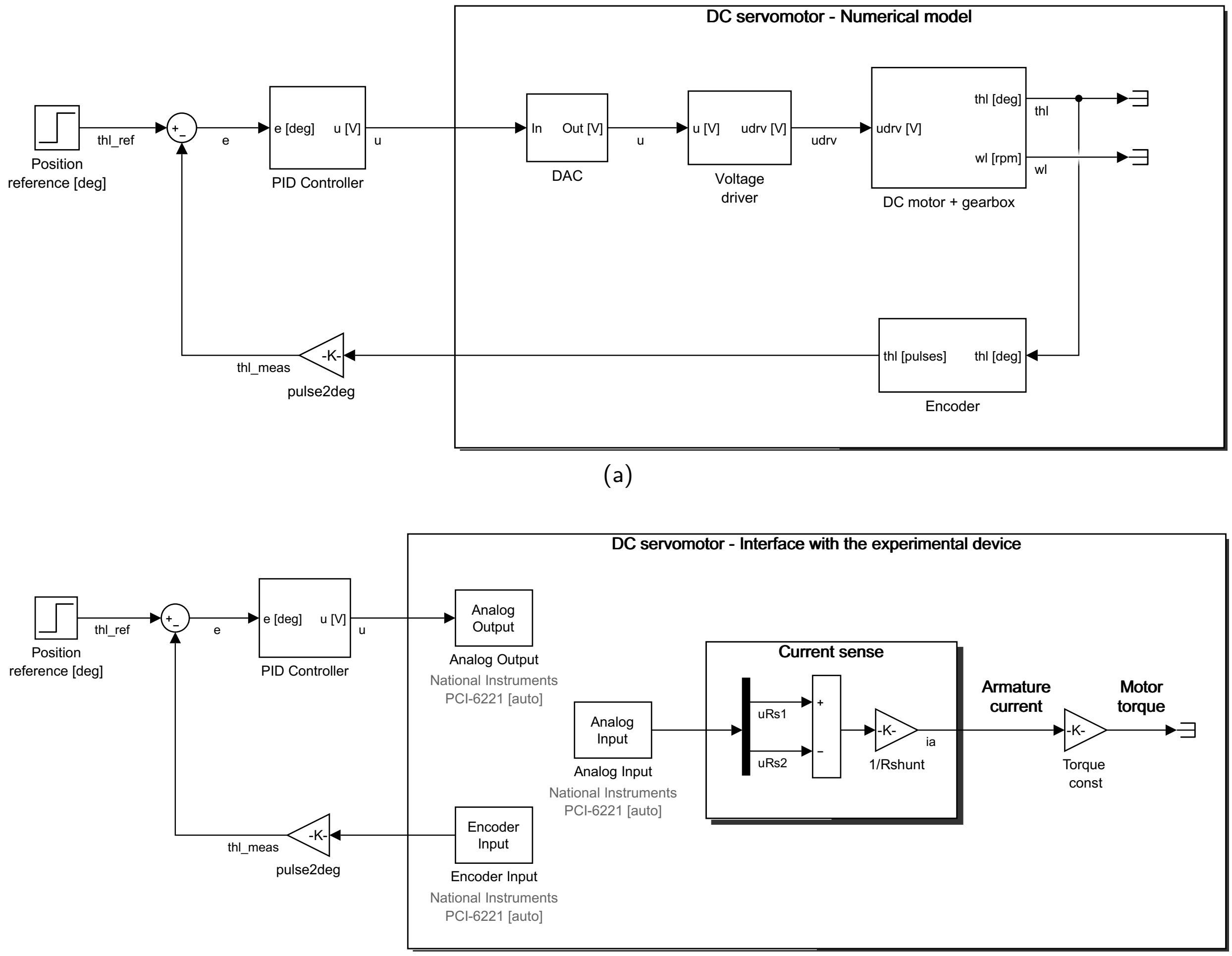


Figure 3: Position PID control loop: (a) Simulink model for numerical simulations; (b) Simulink model for experiments ("real–time simulations").

2) Repeat the test described in point 3) of Sec. 2.2 on the experimental device, by using the Simuilnk model prepared in the previous point.

3.2 Estimation of the friction parameters

1) Apply the procedure outlined in the Handout for estimating the parameters of the static and viscous friction. Since the estimation procedure requires to measure the motor torque (or current, provided that the motor torque constant K_t is known) at *constant* speed, use the position PID controller designed at point 1) to impose constant speed levels to the motor (indeed, verify that the position PID controller can be used for such purpose). Note that if ω_l^* is a desired speed reference to be imposed to the motor, then the corresponding position reference ϑ_l^* to be provided to the position control loop is obtained by integration, i.e.

$$\vartheta_l^*(t) \; = \; \int_0^t \omega_l^*(\tau)\, d\tau \tag{3}$$

For collecting multiple pairs of steady state torque vs speed measurements in a single experiment, consider to use a staircase speed reference of the type:

$$\omega_l^*(t) \; = \; k\,\Delta\omega \qquad \text{with} \qquad t \in [\,(k-1)\,\Delta T,\, k\Delta T\,)\,, \quad k = 1, \cdots, N^* \tag{4}$$

where $\Delta\omega$ and ΔT are the height and width of each speed step. Consider to perform a $45\,\mathrm{s}$ experiment with $\Delta\omega = 50\,\mathrm{rpm}$ and $\Delta T = 5\,\mathrm{s}$, so that the final load side speed is $450\,\mathrm{rpm}$, just a little below the motor nominal speed, and collect both the motor current and speed[2] (see Fig. 4a). After completing the experiment with positive rotations, repeat it by reversing speed direction, i.e. by setting $\Delta\omega = -50\,\mathrm{rpm}$.

The motor speed can be obtained by differentiating the measured position with a suitable high–pass filter ("real derivative"). The filter cut–off frequency and roll–off must be chosen to avoid excessive amplification of the encoder quantisation noise at high frequency. A possible choice is the following second order Butterworth high–pass filter

$$H_\omega(s) \; = \; \frac{\omega_c^2\, s}{s^2 + 2\delta\omega_c s + \omega_c^2} \qquad \text{with} \qquad \omega_c = 2\pi\, 20\,, \quad \delta = 1/\sqrt{2} \tag{5}$$

The motor current can be indirectly obtained by measuring the voltage drop across the shunt resistor, as explained in the introductory guide to the experimental setup. Note that the measured current is very noisy, due to the commutations of the brushes contacts occurring while the motor is rotating. To reduce the commutation noise, the current measurement can be filtered with a suitable low–pass filter, before proceeding with the estimation of the friction parameters. A possible choice is the following second order low–pass filter

$$H_i(s) \; = \; \frac{\omega_{c,i}^2}{s^2 + 2\,\delta_i\,\omega_{c,i}\, s + \omega_{c,i}^2} \qquad \text{with} \qquad \omega_{c,i} = 2\pi\, 20\,, \quad \delta_i = 1/\sqrt{2} \tag{6}$$

2) For each time interval

$$t \in [\,(k-1)\,\Delta T,\, k\Delta T\,)\,, \quad k = 1, \cdots, N^* \tag{7}$$

[2] Alternatively, collect the motor position, and then compute the required speed measurement offline, by using the

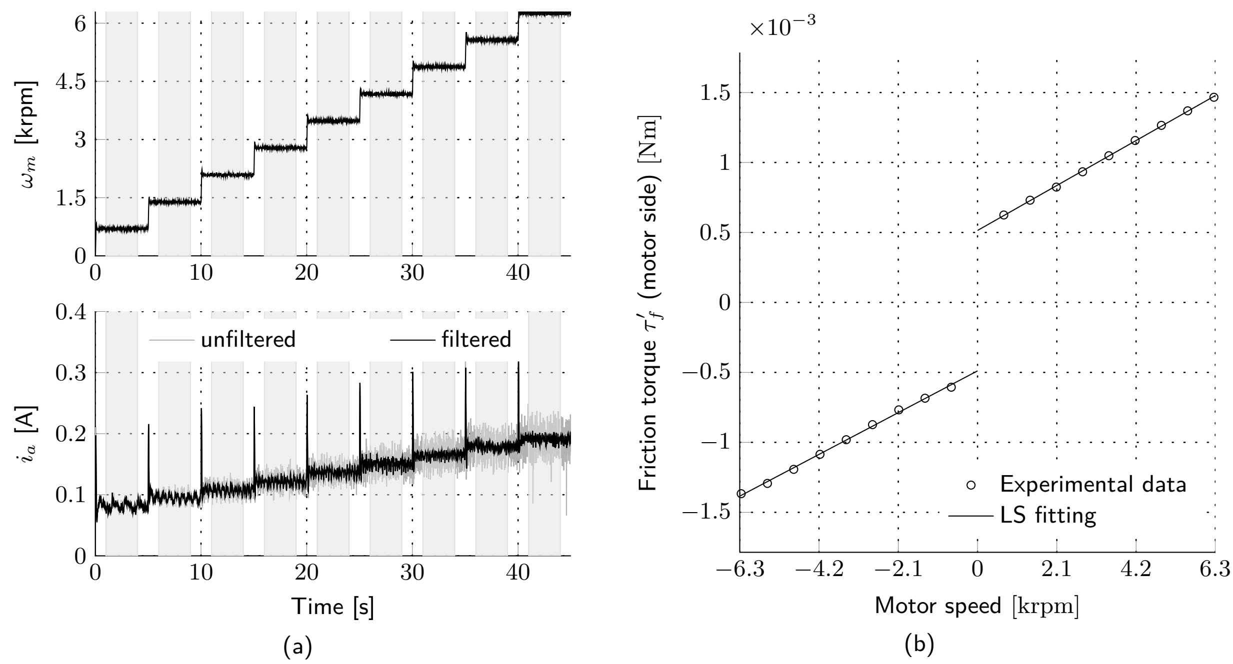


Figure 4: Friction estimation test: (a) experimental data (for positive speed profile); (b) LS estimation of friction parameters. Shaded areas in (a) represent the time interval over which the speed/current averages are computed.

within which the motor speed is constant, compute the average values $\omega_{m,k} = N\,\omega_{l,k}$ and $\tau_{m,k} = K_t\, i_{a,k}$ of the motor speed and torque. In each time interval, avoid to consider the initial speed and torque transients in the computation of the average values (i.e. consider only the portion of data where a constant steady state value is reached – see shaded areas in Fig. 4a).

Remind that two tests were performed, one with a positive staircase speed reference, and the other with a negative reference. Consider to compute the speed/torque averages for the data of both tests. Denote with $Z_+^{N^*} = \{\tau_{m,k},\, \omega_{m,k}\}$ the data set obtained with the positive reference speed, and with $Z_-^{N^*} = \{\tau_{m,k},\, \omega_{m,k}\}$ the data set obtained with the negative reference speed.

3) With the data set $Z_+^{N^*} = \{\tau_{m,k},\, \omega_{m,k}\}$ obtained in point 2), obtain the friction parameters $\hat{B}_{eq+}$ and $\hat{\tau}_{sf+}$ using the estimation procedure outlined in in the Handout.Please pay attention to use the appropriate measurement units when applying the estimation procedure: in particular, the torque should be expressed in $[\mathrm{N\,m}]$, and the speed in $[\mathrm{rad/s}]$. In this way, $\hat{B}_{eq+}$ will be expressed in $[\mathrm{Nm/(rad/s)}]$, and $\hat{\tau}_{sf+}$ in $[\mathrm{N\,m}]$.

Repeat the estimation for the data set $Z_-^{N^*}$, obtaining the estimates $\hat{B}_{eq-}$ and $\hat{\tau}_{sf-}$. Then, obtain the overall estimates of the two friction parameters as follows

$$\hat{B}_{eq} = \frac{\hat{B}_{eq+} + \hat{B}_{eq-}}{2}, \qquad \hat{\tau}_{sf} = \frac{|\hat{\tau}_{sf+}| + |\hat{\tau}_{sf-}|}{2} \tag{8}$$

The final friction torque characteristic (*motor side*, with speed plotted in $[\mathrm{krpm}]$) should look like that reported in Fig. 4b.

4) Suppose that it is desired to assess the uncertainty that affects the estimates (8). This in turn requires to estimate the covariance matrix of the LS solution. The following result can be used for

high–pass filter (5).

such purpose:

Lemma 1 *Suppose that the data used for the estimation of the friction parameters are generated according to the following model:*

$$\tau_{m,k} \,=\, \tau'_{f,k} \,=\, \boldsymbol{\varphi}_k^T \,\boldsymbol{\theta}_0 \,+\, e_k \tag{9}$$

where $\boldsymbol{\varphi}_k$ is the vector of regressors, $\boldsymbol{\theta}_0$ is the vector of "true" parameters, and e_k is a stochastic white noise with zero mean and variance λ^2. Then, the following properties hold:

(a) $\hat{\boldsymbol{\theta}}_{LS}$, as computed in the Handout, is an unbiased estimate of $\boldsymbol{\theta}_0$.

(b) The covariance matrix of $\hat{\boldsymbol{\theta}}_{LS}$ is given by

$$\mathrm{cov}(\hat{\boldsymbol{\theta}}_{LS}) \,=\, \lambda^2 \,(\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \tag{10}$$

where $\boldsymbol{\Phi}$ is the regressors matrix.

(c) An unbiased estimate of λ^2 is given by

$$s^2 \,=\, \frac{1}{M-p}\, V(\hat{\boldsymbol{\theta}}_{LS}) \tag{11}$$

where $V(\hat{\boldsymbol{\theta}})$ is the quadratic cost function with M the number of measurements, and $p = 2$ the number of estimated parameters.

Use the results of the previous lemma to evaluate the variances $\mathrm{var}(\hat{B}_{eq})$ and $\mathrm{var}(\hat{\tau}_{sf})$ of the estimates (8) (remind that the two variances are the elements on the leading diagonal of the covariance matrix $\mathrm{cov}(\hat{\boldsymbol{\theta}}_{LS})$).

The confidence intervals for the true parameters B_{eq} and τ_{sf} can then be obtained by taking the least–squares estimates $\pm$ a confidence margin, namely

$$\begin{aligned} B_{eq} \,&\in\, \left[\, \hat{B}_{eq} - c\,\sqrt{\mathrm{var}(\hat{B}_{eq})}\,, \quad \hat{B}_{eq} + c\,\sqrt{\mathrm{var}(\hat{B}_{eq})}\,\right] \\ \tau_{sf} \,&\in\, \left[\, \hat{\tau}_{sf} - c\,\sqrt{\mathrm{var}(\hat{\tau}_{sf})}\,, \quad \hat{\tau}_{sf} + c\,\sqrt{\mathrm{var}(\hat{\tau}_{sf})}\,\right] \end{aligned} \tag{12}$$

where c depends on the desired confidence level. For a large number of measurements, the value $c = 1.96$ corresponds to a 95% confidence interval. Instead, when the number of measurements is small, the value of c must be chosen more conservatively, by resorting to the Student distribution with $M - p$ parameters.

3.3 Estimation of the inertia parameters

1) Apply the procedure outlined in the Handout for estimating the total moment of inertia of the DC servomotor (rotor + mechanical load). The estimation procedure requires to measure the motor torque (or current, provided that the motor torque constant K_t is known) during constant acceleration/deceleration phases, which can be imposed again by using the position PID controller designed at point 1). Note that if a_l^* is a desired acceleration reference to be imposed to the motor, then the corresponding position reference ϑ_l^* to be provided to the position control loop is obtained by double

integration, i.e.

$$\vartheta_l^*(t) \;=\; \int_0^t \omega_l^*(\tau)\,d\tau\,, \qquad \omega_l^*(t) \;=\; \int_0^t a_l^*(\tau)\,d\tau \tag{13}$$

For repeating several acceleration/deceleration phases in a single experiment, consider to use a square wave acceleration reference of the type:

$$\omega_l^*(t) \;=\; \int_0^t a_l^*(\tau)\,d\tau \tag{14}$$

with

$$a_l^*(t) \;=\; (-1)^k A \qquad \text{for} \qquad t \in [\,k\,\Delta T,\,(k+1)\Delta T\,)\,, \quad k = 0, \cdots, N^* - 1 \tag{15}$$

where A and ΔT are the amplitude and duration of the acceleration/deceleration phase. Consider to perform a $20\,\mathrm{s}$ experiment with $A = 450\,\mathrm{rpm/s}$ and $\Delta T = 1\,\mathrm{s}$, so that the motor speed at the end of each acceleration phase is $450\,\mathrm{rpm}$, just a little below the motor nominal speed, and collect the motor current, speed and acceleration[3] (see Fig. 5).

The motor speed can be obtained by differentiating the position measurement with the high–pass filter (5). Similarly, the motor acceleration can be obtained by differentiating the speed measurement with the second order Butterworth high–pass filter

$$H_a(s) \;=\; \frac{\omega_{c,a}^2\, s}{s^2 + 2\delta_a\omega_{c,a}s + \omega_{c,a}^2} \qquad \text{with} \qquad \omega_{c,a} = 2\pi\,20\,, \quad \delta_a = 1/\sqrt{2} \tag{16}$$

Instead, the motor current is obtained as explained in point 1) of Sec. 3.2, namely by measuring the voltage drop across the shunt resistor, and then using (6) to reduce the effects of the brushes commutation noise.

2) Use the estimated friction parameters (8) to compute the friction torque signal

$$\hat{\tau}_f'(t_k) \;=\; \hat{B}_{eq}\,\omega_m(t_k) \,+\, \frac{\hat{\tau}_{sf}}{N}\,\mathrm{sign}\,(\omega_m(t_k)) \qquad \text{with} \qquad t_k = k\,T_s\,, \quad k = 0, \ldots, M-1 \tag{17}$$

where $\omega_m(t_k)$ is the motor speed measured at the generic sampling time instant t_k, and M is the number of samples collected during the experiment. Then, use (**??**) to compute the torque signal

$$\hat{\tau}_i(t_k) \;=\; \tau_m(t_k) \,-\, \hat{\tau}_f'(t_k) \tag{18}$$

which represents the fraction of motor torque used to accelerate the total inertia $\hat{J}_{eq}$ (inertial torque component – see bottom plot in Fig. 5b). In (18), $\tau_m(t_k)$ is the motor torque measured at the generic sampling time instant t_k (obtained by multiplying the measured current $i_a(t_k)$ by the nominal torque constant K_t).

3) For each time interval

$$t \in [\,k\,\Delta T,\,(k+1)\Delta T\,)\,, \quad k = 0, \cdots, N^* - 1 \tag{19}$$

within which the motor acceleration is constant, compute the average values $a_{m,k} = N\,a_{l,k}$ and $\hat{\tau}_{i,k}$ of the motor acceleration and inertial torque component. In each time interval, avoid to consider

[3] Alternatively, collect the motor position, and then compute the required speed and acceleration measurements offline, by using the high–pass filters (5) and (16).

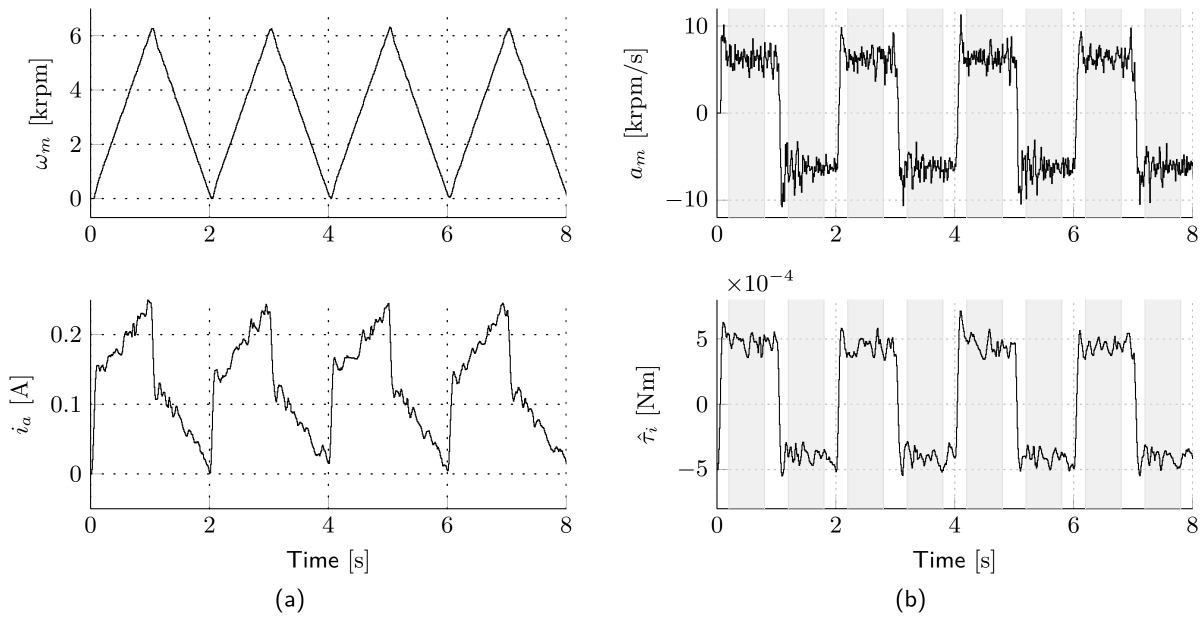


Figure 5: Inertia estimation test: (a) speed and current profiles; (b) acceleration and inertial torque profiles. Shaded areas in (b) represent the time intervals over which the averages of the acceleration and inertial torque measurements are computed.

the initial acceleration and torque transients in the computation of the average values (see shaded areas in Fig. 5b).

Note that $a_{m,k}$ is a positive acceleration when k is even, and negative otherwise.

4) Obtain the estimate of the equivalent inertia using the estimation procedure outlined in the Handout yields

$$\hat{J}_{eq} = \frac{1}{P} \sum_{n=0}^{P-1} \frac{\hat{\tau}_{i,2n} - \hat{\tau}_{i,2n+1}}{a_{m,2n} - a_{m,2n+1}} \tag{20}$$

with $P = N^*/2$ (assuming N^* even). In (20) it has been used the fact that given the acceleration reference (15), the average value $a_{m,k}$ of the motor acceleration obtained in point 3) is positive when k is even, and negative otherwise.