Speed Control of an Electrical Drive

Vitesse - Track A - Trinome 32

JIN DING

CARLOS BENÍTEZ ROSETY

MARCOS ICHIRO SASAKI

TP Report
Control Theory

19/09/2025 - 22/10/2025

CentraleSupélec

Abstract

This practical work focuses on the modeling, control, and experimental validation of a DC electrical drive system. The objective was to design and implement cascaded control strategies capable of regulating the motors speed and current with high precision. The study began with the identification of the motors electrical and mechanical parameters including resistance, inductance, torque and voltage constants, friction coefficients, and moment of inertia which provided the foundation for accurate system modeling.

A two-level control structure was developed: an inner current loop designed via pole placement to achieve fast and stable current regulation, and an outer speed loop tuned using both linear quadratic (LQ) and proportional control strategies. An observer was then implemented to estimate unmeasured variables such as speed and load torque, improving disturbance rejection and enabling feed-forward compensation.

Simulations were performed in MATLAB/Simulink to validate the theoretical designs under various operating conditions, including load disturbances and setpoint variations. The experimental implementation confirmed the robustness of the controllers, demonstrating low overshoot, minimal steady-state error, and good dynamic performance. The observer-based compensation was also evaluated, showing effective convergence in simulation and practical limitations due to measurement noise.

Overall, the project successfully bridged theoretical control design with real-world application, illustrating the integration of modeling, simulation, and experimental testing in modern electromechanical systems.

Contents

1   Identification of the System Parameters

1.1   Introduction

The objective of this practical work is to design and implement control strategies for the speed regulation of a direct current (DC) electrical drive system. Such systems are widely used in transportation, robotics, and industrial automation, where precise speed control ensures efficiency and reliability.

The study combines modeling, simulation, and experimentation. First, the electrical and mechanical parameters of the DC motor are identified using a platform composed of two coupled DC machines, sensors, and a DC/DC converter. Then, controllers are designed and validated in Matlab/Simulinkan inner current loop by pole placement and an outer speed loop using a linear quadratic (LQ) approach. Finally, the controllers are tested on the physical setup to compare real and simulated performances.

Part 1 focuses on parameter identification, determining key characteristics such as resistance, inductance, torque and voltage constants, friction coefficients, and inertia. These parameters provide the foundation for accurate modeling and effective control design in the subsequent stages.

The equations governing the DC motor system are as follows:

$$U_m(t) = R \cdot I(t) + L\frac{dI(t)}{dt} + E(t) \tag{1}$$

$$E(t) = K_e \cdot \Omega(t) \tag{2}$$

$$C_e(t) = K_c \cdot I(t) \tag{3}$$

$$C_e(t) = C_r(t) + C_s + f \cdot \Omega(t) + J\frac{d\Omega(t)}{dt} \tag{4}$$

where:

- R: armature resistance (Ω)

- L: armature inductance (H)

- E: counter-electromotive force of the motor (V)

- Ω: rotational speed of the rotor (rad $\cdot$ s^{-1})

- C_e: electromagnetic torque of the motor (N $\cdot$ m)

- K_e: voltage constant (V $\cdot$ s)

- K_c: torque constant (N $\cdot$ m $\cdot$ A^{-1})

- C_r: resistant torque (N $\cdot$ m)

- C_s: Coulomb friction torque (N $\cdot$ m)

- f: viscous friction coefficient (N $\cdot$ m $\cdot$ s)

- J: moment of inertia of the shafts of the two machines (kg $\cdot$ m^2)

1.2   Armature Resistance and Inductance: R and L

The armature resistance (R) and inductance (L) characterize the electrical behavior of the DC motors armature circuit. The resistance R represents the opposition to the flow of current in the winding, while the inductance L reflects the ability of the winding to store energy in its magnetic field. These parameters directly influence the transient response of the current when a voltage is applied to the motor.

To determine R and L, the rotor is kept stationary so that no back electromotive force (EMF) is generated. A step voltage $u_c(t)$ of $2.5\,\mathrm{V}$ is applied through the DC/DC converter, and the corresponding current response is observed in the Simulink oscilloscope. The experiment can be modeled as a simple first-order RL circuit governed by:

$$U_m(t) = RI(t) + L\frac{dI(t)}{dt}.$$

At steady state, the current reaches a constant value I_∞, allowing the armature resistance to be calculated as:

$$R = \frac{U_m}{I_\infty}.$$

The inductance L is obtained from the electrical time constant τ_e of the current response, given by:

$$\tau_e = \frac{L}{R} \quad \implies \quad L = R \cdot \tau_e.$$

The measured quantities during the experiment are summarized in Table 1.

Table 1: Measured and calculated values for R and L

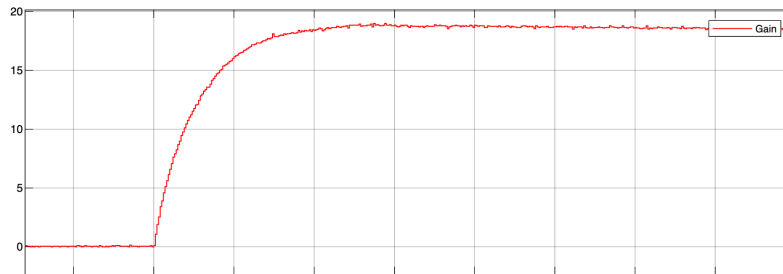| Quantity	Symbol	Measured Value	Unit
Applied voltage	V	6.5	V
Steady-state current	I	18.55	A
Electrical time constant	τ_e	0.025	s
Calculated resistance	R	0.3504	Ω
Calculated inductance	L	0.00876	H

Figure 2: Current response $I(t)$ to step voltage input, showing the transient behavior and steady-state value.

Speed Control of an Electrical Drive   4

Thus, from the steady-state and transient measurements, the armature resistance and inductance of the DC machine were found to be:

$$R = 0.3504 \ \Omega, \qquad L = 0.00876 \ \text{H}.$$

These values will be used in subsequent stages for the modeling and control design of the electrical drive system.

1.3   Voltage Constant and Torque Constant: $K_e = K_c$

In steady state, the armature voltage satisfies

$$U_m = E + R\,I = K_e\,\Omega + R\,I,$$

so the (backEMF) voltage constant K_e (equal to the torque constant K_c in SI units) can be obtained directly from averaged steady-state measurements as

$$K_e \ = \ \frac{\overline{U}_m - R\,\overline{I}}{\overline{\Omega}} \qquad (\text{with } \overline{\Omega} \text{ in rad/s}).$$

With the rotor turning (stator energized), we applied a constant control u_c at the converter input and increased it in steps $u_c = 0, 2, 3, 4$ V, waiting for steady state each time. At steady state we recorded: (i) the average armature voltage U_m (voltmeter), (ii) the current sensor output V_I (converted to I via $I = V_I/K_{\text{cap}I}$ with $K_{\text{cap}I} = 0.1$ V/A), and (iii) the rotational speed in rpm (converted to Ω via $\Omega = \text{rpm} \cdot 2\pi/60$).

Table 2: Steady-state measurements and converted quantities.

u_c (V)	V_I (V)	rpm	Ω (rad/s)	$I = V_I/0.1$ (A)	U_m (V)
0	0.0374	3.0184	0.316086	0.374	0
2	0.9976	57.1736	5.987205	9.976	5
3	1.1536	184.3761	19.307820	11.536	16
4	1.2517	317.6292	33.262052	12.517	27

Using the previously identified armature resistance

$$R = 0.350404313 \ \Omega,$$

and the averaged values from our measurements (already converted to SI units):

$$\overline{U}_m = 12 \ \text{V}, \quad \overline{I} = 0.860075 \ \text{A}, \quad \overline{\Omega} = 14.7182909 \ \text{rad/s} \ (\text{from } 140.549325 \ \text{rpm}),$$

we obtain

$$K_e = \frac{12 - 0.350404313 \times 0.860075}{14.7182909} = 0.794835901 \ \text{V} \cdot \text{s/rad}.$$

Hence, in SI units,

$$\boxed{K_e = K_c \approx 0.794835901 \ \text{V} \cdot \text{s/rad}}.$$

Speed Control of an Electrical Drive   5

1.4   Coulomb Friction Torque and Viscous Friction Coefficient: C_s and f

In steady state (constant speed, $\dot{\Omega} = 0$) and with zero external load torque ($C_r = 0$), the DC motor torque balance reduces to

$$K_c\,I \;=\; f\,\Omega \;+\; C_s,$$

where K_c is the torque constant (equal to the backEMF constant in SI), f is the viscous friction coefficient, and C_s is the Coulomb (dry) friction torque.

Hence, plotting the measured current I versus angular speed Ω and fitting the affine model

$$I \;=\; a\,\Omega \;+\; b$$

yields $f = K_c\,a$ and $C_s = K_c\,b$.

Table 3: Steady-state points used for $I\Omega$ identification.

u_c (V)	U_m (V)	rpm	Ω (rad/s)	I (A)
2	5	57.1736	5.987205	0.9976
3	16	184.3761	19.307820	1.1536
4	27	317.6292	33.262052	1.2517
6	49	593.3711	62.137676	1.6074

A linear regression of I vs. Ω gives (see plotted trendline):

$$I \;=\; \underbrace{0.0107}_{a}\,\Omega \;+\; \underbrace{0.9293}_{b} \quad [\text{A}],$$
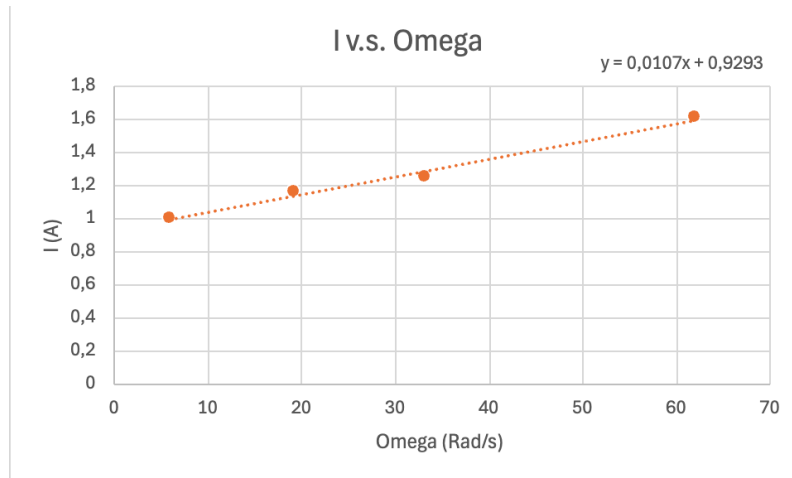
with Ω in rad/s.

Figure 3: Linear regression of steady-state current I versus angular speed Ω.

Using the previously identified $K_c = K_e = 0.794835901$ V s/rad $=$ N m/A, we obtain

$$f = K_c a = 0.794835901 \times 0.0107 = \mathbf{0.008504744} \text{ N m s},$$

$$C_s = K_c b = 0.794835901 \times 0.9293 = \mathbf{0.738641003} \text{ N m}.$$

These friction parameters will be used in the motor model and subsequent controller design/validation.

1.5 Moment of Inertia of the System: J

When the control signal is removed, the motor supply is cut and the rotor slows down under the effect of friction only. The deceleration obeys

$$J \dot{\Omega}(t) + f \Omega(t) + C_s = 0,$$

or, with $\tau = \frac{J}{f}$,

$$\dot{\Omega}(t) + \frac{1}{\tau} \Omega(t) = -\frac{C_s}{J}.$$

Its solution for $t \geq 0$ (with $\Omega_0 = \Omega(0)$ at switch–off) is

$$\Omega(t) = \left(\Omega_0 + \frac{C_s}{f} \right) e^{-t/\tau} - \frac{C_s}{f}. \qquad (*)$$
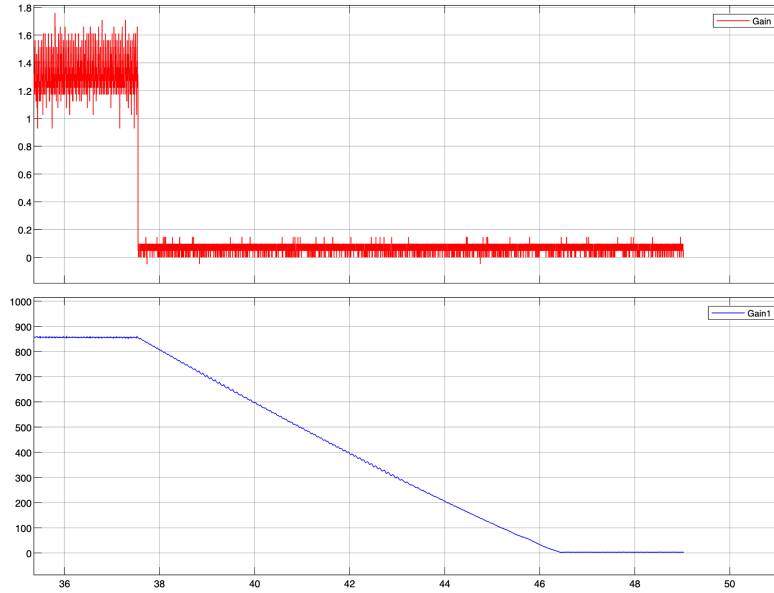
Figure 4: Angular velocity $\Omega(t)$ decay (blue curve) after supply cut (Figure 4).

We used MATLAB code to fit a function that approximates the experimental speed decay curve shown in Figure 4. The fitting process involved nonlinear least squares optimization to match the analytical model (4) (equation (*)) to the measured data, with τ and Ω_0 as fitting parameters while f and C_s were held constant at their

previously identified values. The fitted function closely matches the experimental decay, and from this fitted model we directly obtained the moment of inertia as Using the fitted function parameters, we obtain

$$J = 0.1213266 \text{ kg} \cdot \text{m}^2.$$

This J is used in the subsequent modeling and control design stages.

2 Controller Design and Validation by Simulations

2.1 Current Regulation

2.1.1 State-Space Model of the Augmented System

The armature circuit of the DC motor is governed by:

$$U_m(t) = R\,I(t) + L\,\frac{dI(t)}{dt} + K_e\,\Omega(t)$$

and the converter is modeled as:

$$U_m(t) = k_h\,u_c(t), \qquad k_h = 10$$

Hence, the current dynamics can be written as:

$$\frac{dI(t)}{dt} = -\frac{R}{L}\,I(t) + \frac{k_h}{L}\,u_c(t) - \frac{K_e}{L}\,\Omega(t)$$

To ensure zero steady-state error, an integral action is introduced through the integral of the current error:

$$x_{\varepsilon I}(t) = \int_0^t \left(V_{I^*}(\tau) - V_I(\tau) \right) d\tau$$

where the current sensor gives:

$$V_I = K_{\mathrm{conv},I}\,I(t), \qquad K_{\mathrm{conv},I} = 0.1 \text{ V/A}$$

Thus:

$$\frac{dx_{\varepsilon I}(t)}{dt} = V_{I^*}(t) - V_I(t)$$

Define the augmented state vector and the output as:

$$x_I(t) = \begin{bmatrix} I(t) \\ x_{\varepsilon I}(t) \end{bmatrix}, \qquad y_I(t) = V_I(t) = K_{\mathrm{conv},I}\,I(t)$$

The augmented state-space model becomes:

$$\frac{d}{dt}\begin{bmatrix} I \\ x_{\varepsilon I} \end{bmatrix} = \underbrace{\begin{bmatrix} -\dfrac{R}{L} & 0 \\ -K_{\mathrm{conv},I} & 0 \end{bmatrix}}_{A_I} \begin{bmatrix} I \\ x_{\varepsilon I} \end{bmatrix} + \underbrace{\begin{bmatrix} \dfrac{k_h}{L} \\ 0 \end{bmatrix}}_{B_I} u_c(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_I'} V_{I^*}(t) + \underbrace{\begin{bmatrix} -\dfrac{K_e}{L} \\ 0 \end{bmatrix}}_{B_I''} \Omega(t)$$

The output equation is given by:

$$y_I(t) = \underbrace{\begin{bmatrix} K_{\mathrm{conv},I} & 0 \end{bmatrix}}_{C_I} x_I(t)$$

Speed Control of an Electrical Drive   9

Summary of matrices:

$$A_I = \begin{bmatrix} -\dfrac{R}{L} & 0 \\ -K_{\text{conv},I} & 0 \end{bmatrix}, \qquad\qquad B_I = \begin{bmatrix} \dfrac{k_h}{L} \\ 0 \end{bmatrix},$$

$$B_I' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad\qquad B_I'' = \begin{bmatrix} -\dfrac{K_e}{L} \\ 0 \end{bmatrix},$$

$$C_I = \begin{bmatrix} K_{\text{conv},I} & 0 \end{bmatrix}$$

Controllability: The controllability matrix is:

$$\mathcal{C} = [B_I \ \ A_I B_I] = \begin{bmatrix} \dfrac{k_h}{L} & -\dfrac{R\,k_h}{L^2} \\ 0 & -\dfrac{K_{\text{conv},I}\,k_h}{L} \end{bmatrix}$$

The determinant is:

$$\det(\mathcal{C}) = -\frac{K_{\text{conv},I}\,k_h^2}{L^2} \neq 0$$

Therefore, the augmented system is **controllable**.

2.1.2 Characteristic Polynomial and Desired Poles

The design requirements for the current control loop are:

- Settling time: $T_s = 0.1$ s

- Overshoot: $M_p < 10\%$

A damping ratio of $\zeta_I = 0.7$ is chosen to satisfy the overshoot specification. From the second-order system relationship:

$$T_s \approx \frac{4}{\zeta_I \omega_{nI}}$$

the natural frequency is obtained as:

$$\omega_{nI} = \frac{4}{\zeta_I T_s} = \frac{4}{0.7 \times 0.1} \approx 57.1 \text{ rad/s}$$

The desired characteristic polynomial for the current loop is:

$$\lambda^2 + 2\zeta_I \omega_{nI} \lambda + \omega_{nI}^2 = 0$$

$$\lambda^2 + 80\,\lambda + 3267 = 0$$

The corresponding desired eigenvalues (poles) are:

$$\lambda_{1,2} = -\zeta_I \omega_{nI} \pm j\,\omega_{nI}\sqrt{1 - \zeta_I^2} \approx -40.0 \pm j\,40.8$$

In MATLAB, these values are obtained using:

```
zeta_I = 0.7;                    % Damping coefficient of the current loop
omega_nI = 4 / (0.7 * 0.1);      % Natural frequency of the current loop
lambda_I = roots([1 2*zeta_I*omega_nI omega_nI^2]);
```

Thus, the desired complex conjugate poles ensuring the specified dynamic response are located at:

$$\boxed{\lambda_{1,2} = -40.0 \pm j\,40.8}$$

2.1.3   Control Law by State Feedback

The control law for the current loop is defined as:

$$u_c(t) = -K_I\,x_I(t) = -[K_{I1}\ K_{I2}] \begin{bmatrix} I(t) \\ x_{\varepsilon I}(t) \end{bmatrix}$$

The gain vector K_I is determined by pole placement in order to obtain the desired dynamic performance, corresponding to the eigenvalues found in Question 2.1.b.
Using MATLAB, the computation is performed as:

```
K_I = place(A_I, B_I, lambda_I);  % Find the controller gain for state feedback
e = eig(A_I - B_I*K_I);           % Verify the closed-loop poles
```

This function determines the feedback gain matrix $K_I = [K_{I1}\ K_{I2}]$ such that the closed-loop system matrix

$$A_{cl} = A_I - B_I K_I$$

has the desired eigenvalues

$$\lambda_{1,2} = -40 \pm j\,40.8$$

The resulting control law ensures that the current loop exhibits a fast and well-damped response, in accordance with the specified settling time ($T_s = 0.1$ s) and overshoot ($M_p < 10\%$).

2.1.4   Validation with the Synthesis Model

The closed-loop model of the current loop is validated using the state-space representation obtained in the previous steps. The closed-loop system matrix is given by:

$$A_{\text{bf\_I}} = A_I - B_I K_I$$

and the corresponding state-space model is defined in MATLAB as:

```
Abf_I = A_I - B_I*K_I;              % Closed-loop state matrix
Bbf_I = [0; 1];                    % Input matrix for current reference
Cbf_I = [1 0];                     % Output matrix (current measurement)
Dbf_I = 0;                         % Direct transmission term
sys_bf_I = ss(Abf_I, Bbf_I, Cbf_I, Dbf_I);  % Closed-loop system

Iref = 20;                         % Current reference (A)
VRef = Iref * k_capI;              % Convert Iref to Vref using current sensor gain
```

```
opt = stepDataOptions('StepAmplitude', VRef);
[y_step_I, t_step_I, x_step_I] = step(sys_bf_I, opt);    % Step response
u_step_I = -K_I * x_step_I';       % Control voltage (state feedback law)

% Plotting the current and voltage responses
figure(1)
subplot(2,1,1)
plot(t_step_I, y_step_I, 'b', 'LineWidth', 2)
grid on
ylabel('Current (A)')
title('Response of the current loop')

subplot(2,1,2)
plot(t_step_I, k_h*u_step_I, 'r', 'LineWidth', 2)
grid on
ylabel('Voltage (V)')
xlabel('Time (s)')
```

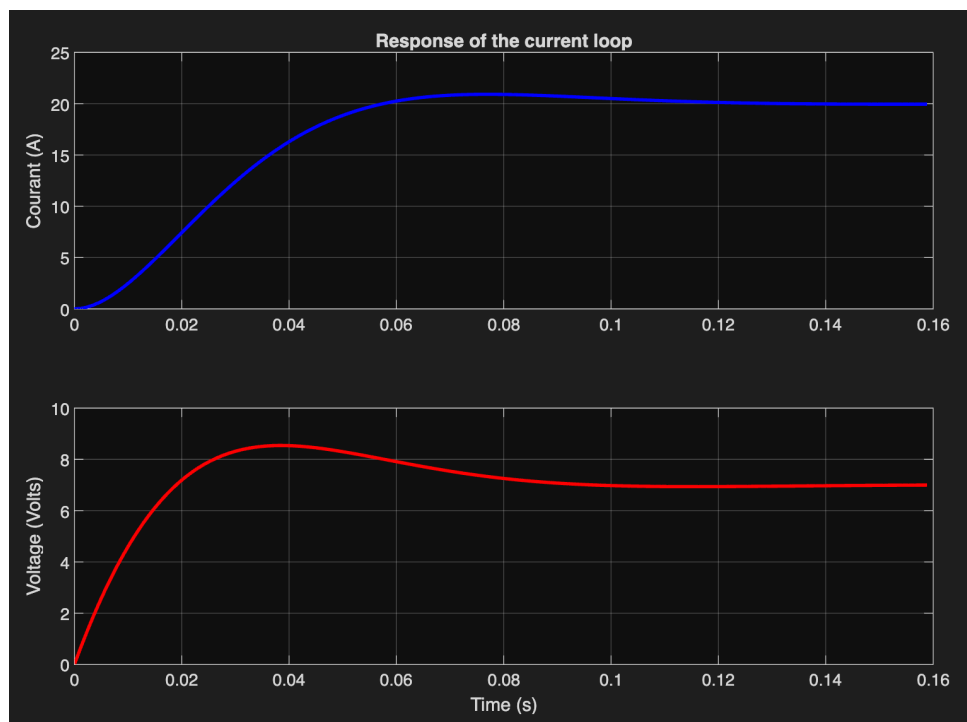The resulting simulation is shown in Figure 5.

Figure 5: Step response of the current loop to a setpoint of 20 A.

Analysis of the results:
   The step response of the current loop shows that:

- The current reaches its steady-state value of 20 A in approximately **0.1 s**.

- The overshoot is around **10%**, corresponding to the damping ratio $\zeta_I = 0.7$.

- The system exhibits a well-damped second-order behavior with no steady-state error.

The control voltage (bottom plot) shows a smooth transient reaching about **89 V** before settling near 6 V once the current reaches 20 A. This confirms that the designed state-feedback controller ensures a fast and stable current loop, satisfying the required specifications:

$$T_s \approx 0.1 \text{ s}, \quad M_p < 10\%.$$

Hence, the performance objectives for the current loop are successfully met.

2.1.5   Validation with the engine simulator (TPVitessePartie21.slx)

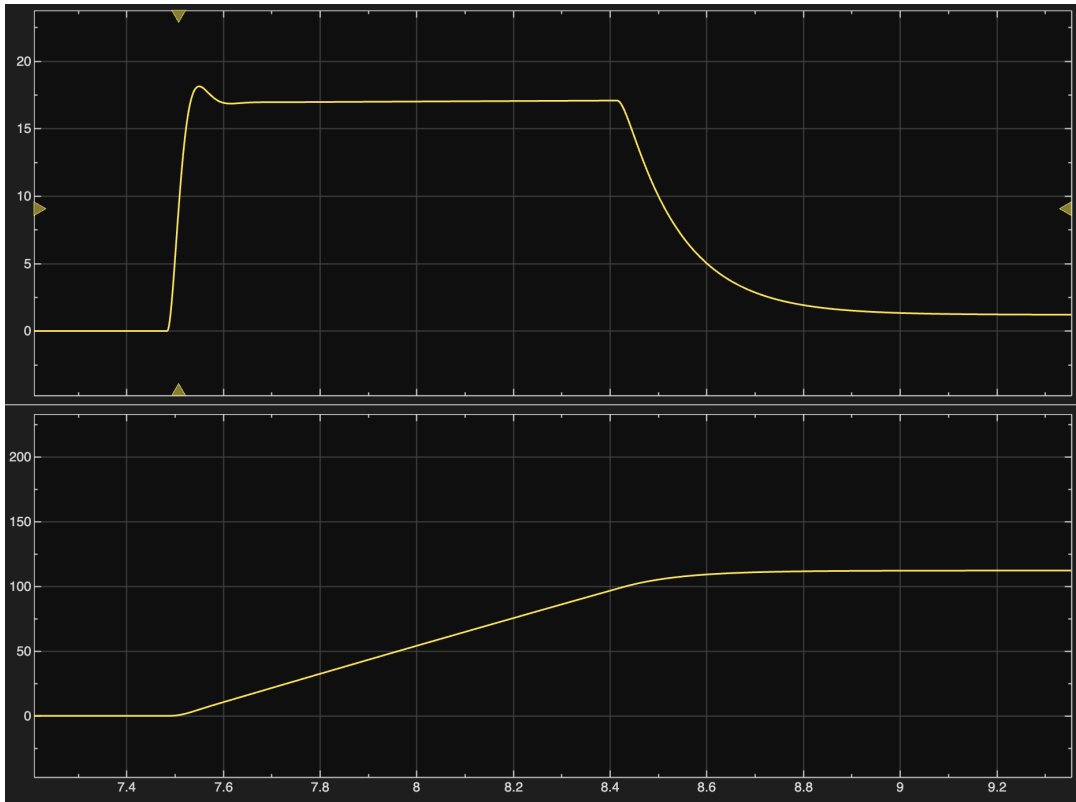**Scenario A:** $C_r = 0$ and $I^\star(t)$ steps from 0 to 20 A at $t = 1$ s.

Figure 6: Scenario A step $I^\star : 0 \to 20$ A. Top: measured current $I(t)$; bottom: speed $\Omega(t)$.

Performance check. From Fig. 6, $I(t)$ reaches the reference with a small transient overshoot ($< 10\%$), settles rapidly (order 10^{-1} s), and exhibits negligible steadystate error due to the integral action in the current loop. **Why $I(t)$ decreases at high speed.** The electrical and mechanical dynamics are

$$U_m = RI + L\dot{I} + K_e\Omega, \qquad J\dot{\Omega} = K_cI - f\Omega - C_s.$$

As Ω grows, the backEMF term $K_e\Omega$ increases, reducing the effective voltage available to sustain $I = 20$ A. When the actuator saturates ($U_m = \pm90$ V), the controller cannot compensate, so $I(t)$ droops slightly while Ω is high.

Speed Control of an Electrical Drive   13

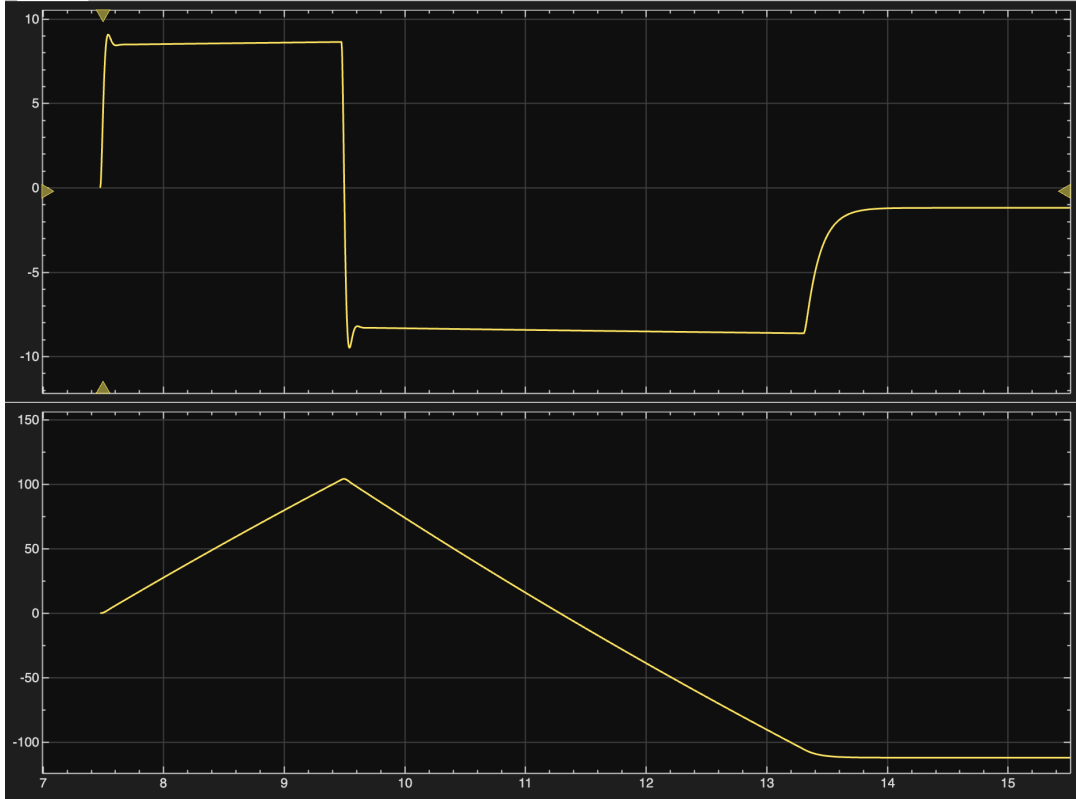**Scenario B:** $C_r = 0$ and $I^\star(t)$ steps from $+10$ A to -10 A at $t = 2$ s.

Figure 7: Scenario B step $I^\star : +10 \to -10$ A. Top: measured current $I(t)$; bottom: $\Omega(t)$.

Reason for the delay in sign reversal. Prior to $t = 2$ s, the integrator has accumulated a positive state to track $+10$ A. At the negative step, the demanded control drives u_c to its limit; the integrator continues to wind up while U_m is saturated, and the nonzero backEMF $K_e\Omega$ further increases the voltage needed to reverse the current. The combination of actuator saturation and integrator windup delays the zero crossing of $I(t)$.

Remedy. Implement antiwindup in the current controller, e.g. (i) *integrator clamping/conditional integration* when u_c or U_m saturates, or (ii) a *tracking (back-calculation) loop* that drives the integrator toward the saturated actuator output. Optionally add an EMF feedforward term $U_{m,\text{ff}} = K_e\Omega$ so the controller primarily regulates $RI + L\dot{I}$. With antiwindup, the delay is effectively removed while preserving the settling-time and overshoot specifications.

2.2 Speed Regulation by a Controller with Integrator

2.2.1 State-Space Model of the Augmented System

Assuming an ideal current loop, the speed dynamics of the DC motor are given by:

$$J\frac{d\Omega(t)}{dt} = K_c V_I^*(t) - C_r(t) - f\,\Omega(t)$$

The speed measurement is expressed as:

$$V_\Omega(t) = K_{\text{cap},\Omega}\,\Omega(t)$$

To ensure zero steady-state error and disturbance rejection, an integral of the speed error is introduced:

$$x_{\varepsilon\Omega}(t) = \int_0^t \left(V_{\Omega^*}(\tau) - V_\Omega(\tau)\right) d\tau$$

The state vector and output are defined as:

$$x_\Omega(t) = \begin{bmatrix} \Omega(t) \\ x_{\varepsilon\Omega}(t) \end{bmatrix}, \qquad y_\Omega(t) = V_\Omega(t) = K_{\text{cap},\Omega}\,\Omega(t)$$

The augmented state-space model becomes:

$$\frac{d}{dt}\begin{bmatrix} \Omega \\ x_{\varepsilon\Omega} \end{bmatrix} = \underbrace{\begin{bmatrix} -\dfrac{f}{J} & 0 \\ -K_{\text{cap},\Omega} & 0 \end{bmatrix}}_{A_\Omega}\begin{bmatrix} \Omega \\ x_{\varepsilon\Omega} \end{bmatrix} + \underbrace{\begin{bmatrix} \dfrac{K_c}{J} \\ 0 \end{bmatrix}}_{B_\Omega} V_I^*(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B'_\Omega} V_{\Omega^*}(t) + \underbrace{\begin{bmatrix} -\dfrac{1}{J} \\ 0 \end{bmatrix}}_{B''_\Omega} C_r(t)$$

and

$$y_\Omega(t) = \underbrace{\begin{bmatrix} K_{\text{cap},\Omega} & 0 \end{bmatrix}}_{C_\Omega} x_\Omega(t)$$

Controllability:

$$\mathcal{C}_\Omega = [B_\Omega \ \ A_\Omega B_\Omega] = \begin{bmatrix} \dfrac{K_c}{J} & -\dfrac{fK_c}{J^2} \\ 0 & -\dfrac{K_{\text{cap},\Omega}K_c}{J} \end{bmatrix}$$

with

$$\det(\mathcal{C}_\Omega) = -\frac{K_{\text{cap},\Omega}K_c^2}{J^2} \neq 0$$

Therefore, the augmented system is **controllable**.

2.2.2 LQ Control Design

The control law is defined as:

$$V_I^*(t) = -K_\Omega\,x_\Omega(t) = -[K_{\Omega1} \ \ K_{\Omega2}]\begin{bmatrix} \Omega(t) \\ x_{\varepsilon\Omega}(t) \end{bmatrix}$$

The feedback gain K_Ω is obtained using the Linear Quadratic (LQ) approach, which minimizes the cost function:

$$J(t) = \int_0^\infty \left(x_\Omega^T Q x_\Omega + r\,(V_I^*(t))^2\right) dt$$

where $Q = \text{diag}(q_1, q_2)$ and r are the weighting factors.

A stabilizing solution exists if (A_Ω, B_Ω) is controllable (verified in Question 2.2.a) and if Q is positive semi-definite and $r > 0$.

Normalization of the state variables:

- $\Omega_{\max} = 1500 \text{ rpm} = 1500 \times 2\pi/60$

- $V_{W_{\max}} = \Omega_{\max} \times K_{\text{cap},\Omega}$

- $x_{\varepsilon\Omega,\max} = 0.5\,V_{W_{\max}} \times 0.2$

- $U_{\max} = 4 \text{ V}$

These limits are used to normalize the state and control variables prior to tuning the weighting matrices.

MATLAB implementation:

```
%% Question 2.2.b : LQ approach

% Normalization of the state variables
omega_max = 1500*2*pi/60;          % Maximum speed (rad/s)
VW_max = omega_max*K_capW;         % Max speed sensor voltage
q_wmax = 0.5*VW_max*0.2;           % Max integral of the error
U_max = 4;                         % Max control voltage (V)

% Scaling matrices
Qn_lq = [1/omega_max 0; 0 1/q_wmax];
Rn_lq = 1/U_max;

% Base weighting matrices
Q_tilda_lq = [100 0; 0 200];
R_tilda_lq = 1;

% Normalized LQ matrices
Q_lq = Qn_lq * Q_tilda_lq * Qn_lq;
R_lq = Rn_lq * R_tilda_lq * Rn_lq;

% Compute LQ gain
K_W = lqr(A_W, B_W, Q_lq, R_lq);
```

The function `lqr()` returns the optimal feedback gain vector $K_\Omega = [0.417 \; -69.13]$ that minimizes $J(t)$ and ensures a stable closed-loop behavior.

Effect of weighting factors:

- Increasing q_1 emphasizes faster speed tracking but may cause larger control efforts.

- Increasing q_2 improves disturbance rejection (faster integral action).

- Increasing r reduces the control voltage amplitude but slows down the response.

A suitable balance between Q and r results in a settling time close to 2 s and an overshoot below 10%, as required by the design specifications.

2.2.3   Validation of the speed controller

The objective of this part is to evaluate the performance of the speed control loop implemented in MATLAB/SIMULINK under different operating conditions. The control law is tested with various speed setpoints $\Omega^\star(t)$ and with/without a load torque $C_r(t)$.

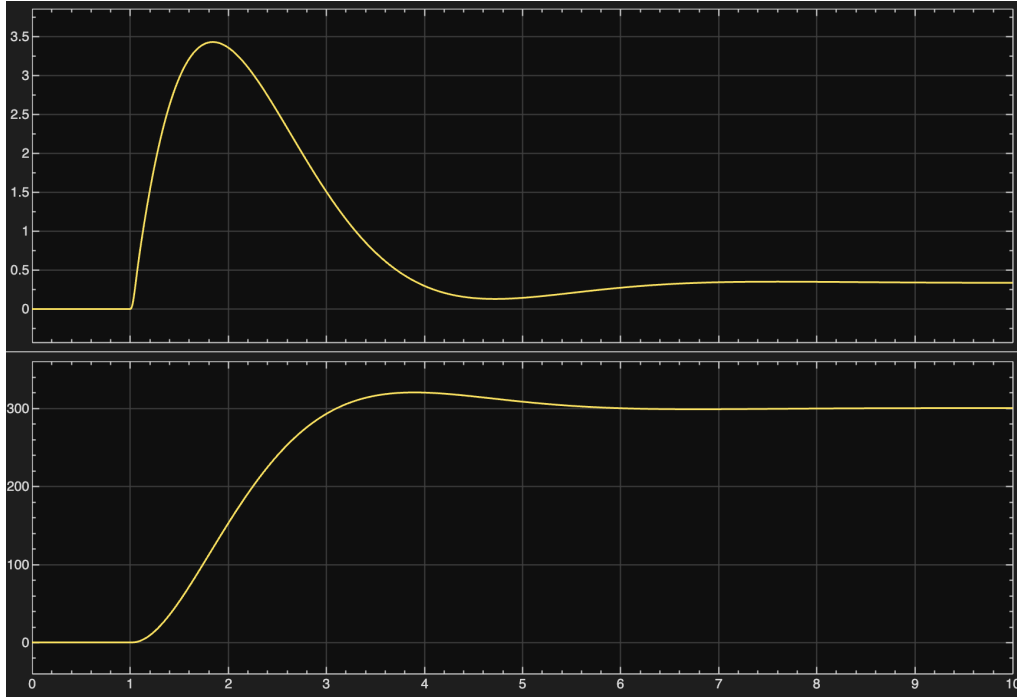**Scenario 1:** $C_r(t) = 0$, the setpoint $\Omega^\star(t)$ changes from 0 to 300 rpm at $t = 1$ s.

Figure 8: Scenario 1 $\Omega^\star(t)$ step from 0 to 300 rpm at $t = 1$ s.

Scenario 2: $C_r(t) = 0$, the setpoint $\Omega^\star(t)$ changes from 0 to 500 rpm at $t = 1$ s.
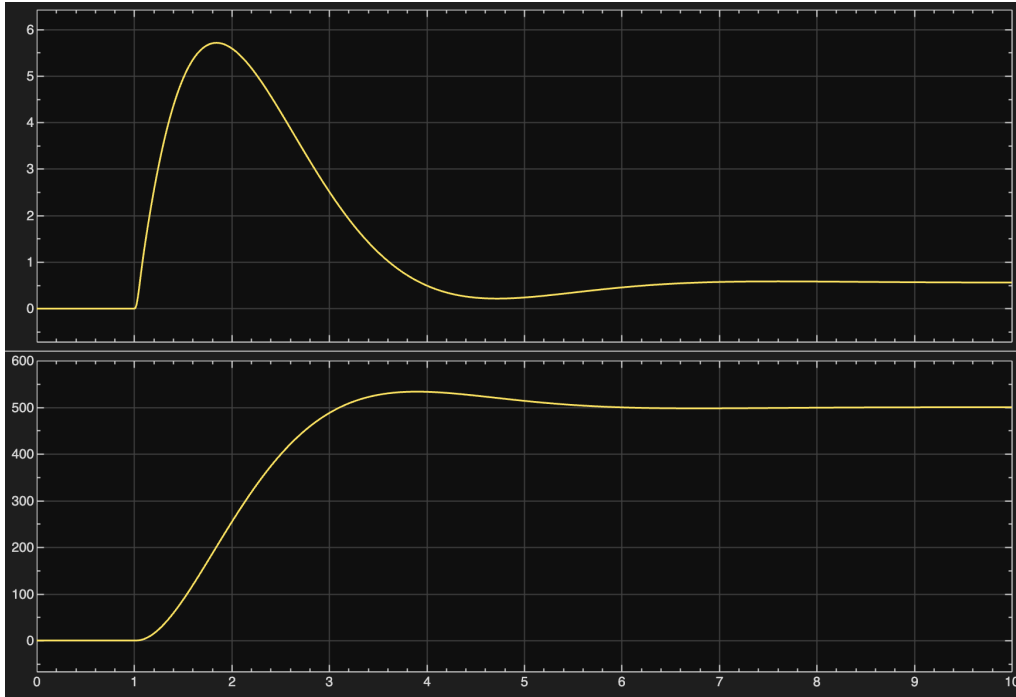
Figure 9: Scenario 2 $\Omega^\star(t)$ step from 0 to 500 rpm at $t = 1$ s.

Scenario 3: $C_r(t) = 0$, the setpoint $\Omega^\star(t)$ changes from 0 to 1200 rpm at $t = 1$ s.
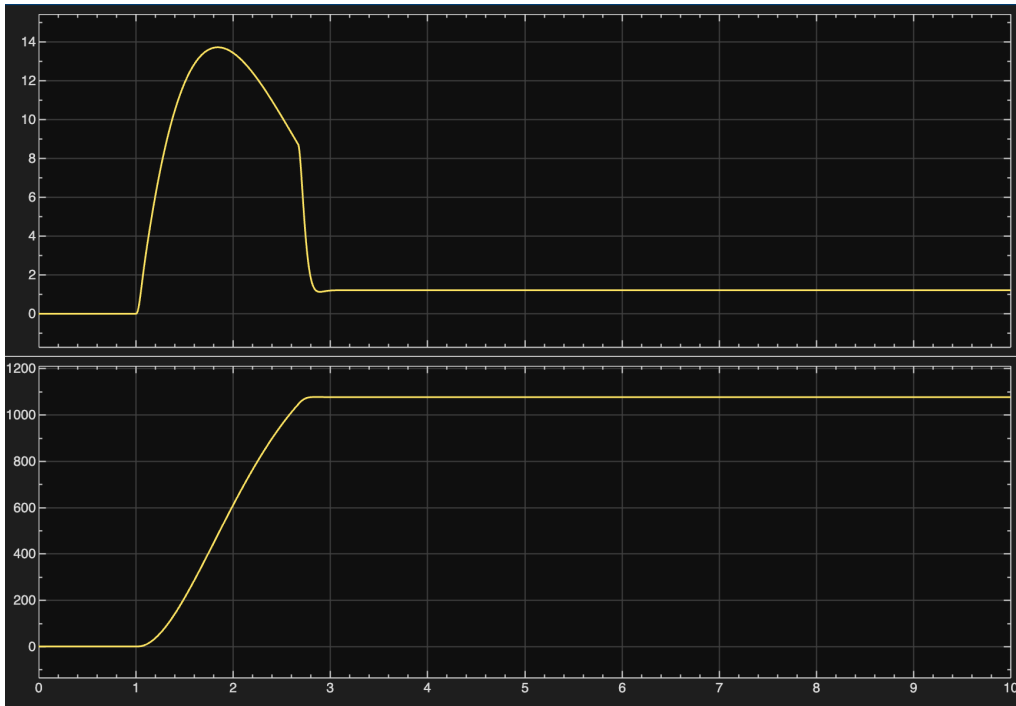

Figure 10: Scenario 3 $\Omega^\star(t)$ step from 0 to 1200 rpm at $t = 1$ s.

Scenario 4:   The load torque $C_r(t)$ changes from 0 to 5 Nm at $t = 10$ s, while the setpoint $\Omega^\star(t)$ changes from 0 to 300 rpm at $t = 1$ s.
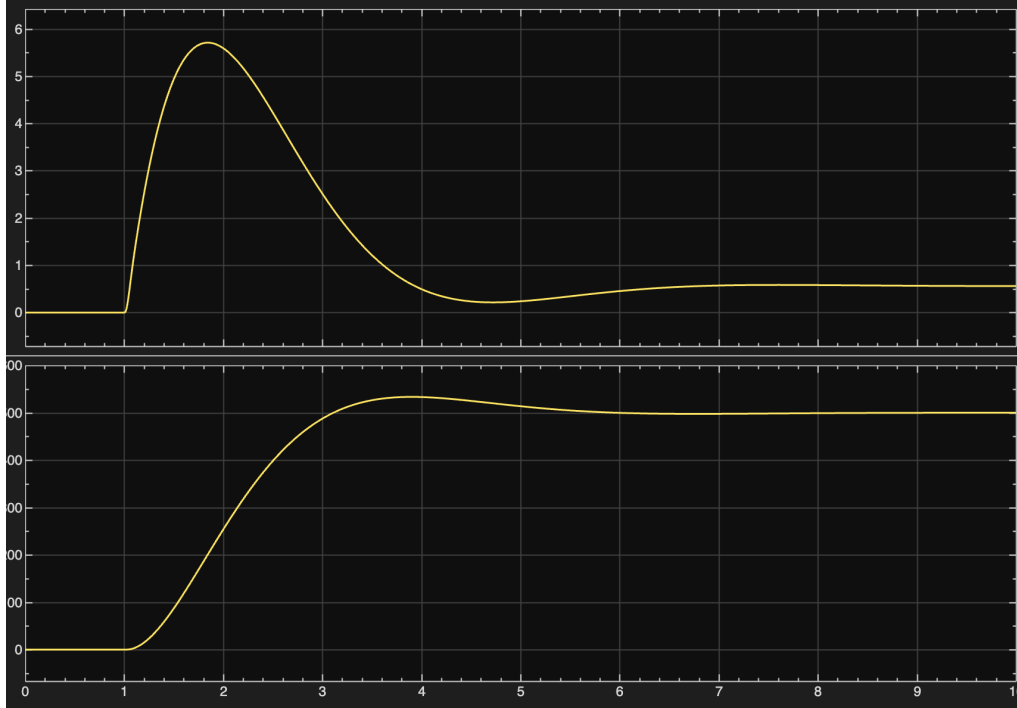
Figure 11: Scenario 4 $\Omega^\star(t)$ step from 0 to 300 rpm at $t = 1$ s with $C_r(t)$ disturbance at $t = 10$ s.

Analysis.   Each test allows evaluation of the systems dynamic behavior in terms of rise time, overshoot, and steadystate accuracy. Scenarios 13 demonstrate the controllers response to increasing speed references, highlighting its bandwidth and saturation limits. Scenario 4 evaluates the robustness of the controller to load disturbances, verifying that the integral component ensures steadystate compensation of C_r while maintaining the desired speed.

2.2.4   Stability Margin Analysis

The stability of the designed LQ speed controller can be analyzed by examining the open-loop transfer function of the system defined by:

$$\dot{x}_\Omega(t) = A_\Omega x_\Omega(t) + B_\Omega u(t), \qquad y(t) = K_\Omega x_\Omega(t)$$

The open-loop system is implemented in MATLAB and converted to a transfer function form to determine the gain and phase margins using the `margin()` function. **MATLAB implementation:**

```
% Question 2.2.d : Stability margin analysis

C_bo = K_W;                % Controller acts as output matrix
D_bo = 0;                  % No direct transmission term
sys_bo = ss(A_W, B_W, C_bo, D_bo);  % Open-loop state-space model
G_bo = tf(sys_bo);         % Convert to transfer function
```

```
figure(3)
margin(G_bo);              % Bode plot with margins
grid on

[GM, PM, Wcg, Wcp] = margin(G_bo);
fprintf('\n--- Stability margins ---\n');
fprintf('Gain margin = %.2f dB\n', 20*log10(GM));
fprintf('Phase margin = %.2fř\n', PM);
fprintf('Gain crossover frequency = %.2f rad/s\n', Wcg);
fprintf('Phase crossover frequency = %.2f rad/s\n', Wcp);
```

Obtained results:

- Gain margin: ∞ dB

- Phase margin: 73.31°

- Gain crossover frequency: not defined (system always stable)

- Phase crossover frequency: 28.54 rad/s
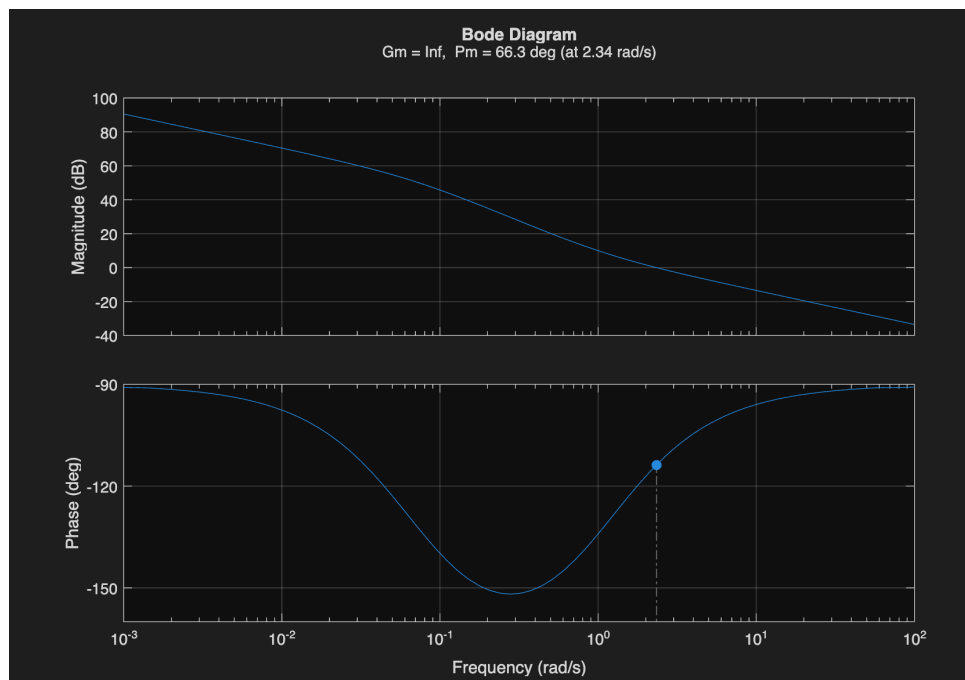

Figure 12: Bode diagram of the open-loop speed control system.

Interpretation of results:

- The infinite gain margin indicates that the system remains stable for all practical gain variations the open-loop gain does not cross 0 dB.

- The phase margin of 73.3° demonstrates a high robustness and well-damped transient response.

- The crossover frequency of 28.5 rad/s defines the approximate bandwidth of the speed loop, ensuring sufficient speed-tracking capability without excessive oscillations.

In conclusion, the LQ-designed controller provides a highly stable and robust speed control system, satisfying the required performance criteria for both stability and dynamic behavior.

2.3   Design of an Observer Capable of Reconstructing the Resistant Torque

2.3.1   Augmented System and Observability

The augmented state-space model of the DC machine used for the observer design is given by:

$$\frac{dx_a(t)}{dt} = A_a x_a(t) + B_a u_c(t), \qquad y_a(t) = C_a x_a(t)$$

where the state vector and output are:

$$x_a = \begin{bmatrix} I \\ \Omega \\ C_r \end{bmatrix}, \quad y_a = V_I$$

The matrices of the augmented system are defined as:

$$A_a = \begin{bmatrix} -\dfrac{R}{L} & -\dfrac{K_c}{L} & 0 \\ \dfrac{K_c}{J} & -\dfrac{f}{J} & -\dfrac{1}{J} \\ 0 & 0 & 0 \end{bmatrix}, \quad B_a = \begin{bmatrix} \dfrac{k_h}{L} \\ 0 \\ 0 \end{bmatrix}, \quad C_a = \begin{bmatrix} K_{\text{cap}I} & 0 & 0 \end{bmatrix}, \quad D_a = \mathbf{0}_{3\times1}$$

MATLAB implementation:

```
A_a = [-R/L  -Kc/L  0;
        Kc/J  -f/J  -1/J;
        0      0     0];
B_a = [kh/L; 0; 0];
C_a = [K_capI 0 0];
D_a = zeros(3,1);

QO_a = obsv(A_a, C_a);    % Observability matrix
rank(QO_a)                % Check the observability rank
```

The observability matrix has full rank ($\text{rank}(QO_a) = 3$), confirming that the system is **observable**.

2.3.2   Selection of Observer Dynamics

According to the design requirements, the observer must be approximately three times faster than the speed control loop. The damping ratio and crossover frequency are set as:

$$\zeta_{\mathrm{n\Omega}} = 0.7, \qquad \omega_{\mathrm{n\Omega}} = \frac{3 \times 4}{0.7 \times 0.1}$$

The desired observer eigenvalues are then computed from the characteristic polynomial:

$$\lambda^2 + 2\zeta_{\mathrm{n\Omega}}\omega_{\mathrm{n\Omega}}\lambda + \omega_{\mathrm{n\Omega}}^2 = 0$$

and an additional third eigenvalue is added, chosen to be twice as fast as the previous ones.

MATLAB implementation:

```
zeta_nW = 0.7;                           % Damping coefficient
omega_nW = 3*4/(0.7*0.1);                % Crossover frequency
lambda_W = roots([1, 2*zeta_nW*omega_nW, omega_nW^2]);
lambda_W = [lambda_W; sum(lambda_W)];    % Add a faster 3rd pole
lambda_obs = lambda_W;                   % Observer eigenvalues
```

These eigenvalues ensure that the observer converges rapidly to the actual system states while maintaining numerical stability.

2.3.3   Observer design from current measurement and control input

We reconstruct the augmented state

$$x_a = \begin{bmatrix} I & \Omega & C_r \end{bmatrix}^\top, \qquad y_a = V_I = K_{\mathrm{cap}I}\, I,$$

using a Luenberger observer driven by the measured current y_a and the control u_c. With the augmented plant

$$\dot{x}_a = A_a x_a + B_a u_c, \qquad y_a = C_a x_a,$$

the observer is chosen as

$$\boxed{\dot{\hat{x}}_a = \left(A_a - K_{\mathrm{obs}} C_a \right) \hat{x}_a + B_a u_c + K_{\mathrm{obs}}\, y_a, \qquad y_{\mathrm{obs}} = C_{\mathrm{obs}}\, \hat{x}_a,}$$

where $y_{\mathrm{obs}} = \begin{bmatrix} \hat{I} & \hat{\Omega} & \hat{C}_r \end{bmatrix}^\top$ and typically $C_{\mathrm{obs}} = I_3$. The gain K_{obs} is computed by pole placement to make the observer dynamics significantly faster than the plant (e.g., $3-5\times$ the closed-loop bandwidth).

Equivalently, grouping u_c and y_a as inputs,

$$\dot{\hat{x}}_a = A_{\mathrm{obs}}\hat{x}_a + B_{\mathrm{obs}} \begin{bmatrix} u_c \\ y_a \end{bmatrix}, \quad A_{\mathrm{obs}} = A_a - K_{\mathrm{obs}}C_a, \quad B_{\mathrm{obs}} = \begin{bmatrix} B_a & K_{\mathrm{obs}} \end{bmatrix}.$$

```
% Given: A_a, B_a, C_a, desired observer poles lambda_obs (1x3)
K_obs = place(A_a', C_a', lambda_obs')';   % Luenberger gain

A_obs = A_a - K_obs*C_a;                    % Observer A
B_obs = [B_a, K_obs];                       % Inputs [u_c; y_a]
C_obs = eye(3);                             % Outputs [Î; ; Cr]
D_obs = zeros(3,2);
```

Notes. (i) Ensure (A_a, C_a) is observable. (ii) Choose λ_{obs} real or complex with negative real parts and magnitude $3-5$ times larger than the controller poles to obtain fast, well-damped convergence of $[\hat{I}, \hat{\Omega}, \hat{C}_r]$ to their true values.

2.3.4 Observer implementation and convergence check

We implement a Luenberger observer that reconstructs the state

$$x_c = \begin{bmatrix} I & \Omega & C_r \end{bmatrix}^\top$$

from the measured armature current $V_I = K_{\text{cap}I}\, I$ and the available control input u_c. With the augmented plant

$$\dot{x}_c = A_c\, x_c + B_c\, u_c, \qquad y_c = C_c\, x_c \;\; (y_c = V_I),$$

the observer is

$$\dot{\hat{x}}_c = A_c\, \hat{x}_c + B_c\, u_c + K_{\text{obs}}\, (y_c - \hat{y}_c), \qquad \hat{y}_c = C_c\, \hat{x}_c,$$

where K_{obs} is chosen so that the observer poles are about three times faster than the current-loop poles (as required), ensuring rapid convergence.

Convergence results. Figures 13–15 compare the estimated variables $\begin{bmatrix} \hat{I}, & \hat{\Omega}, & \hat{C}_r \end{bmatrix}$ with their true values obtained from the simulator.
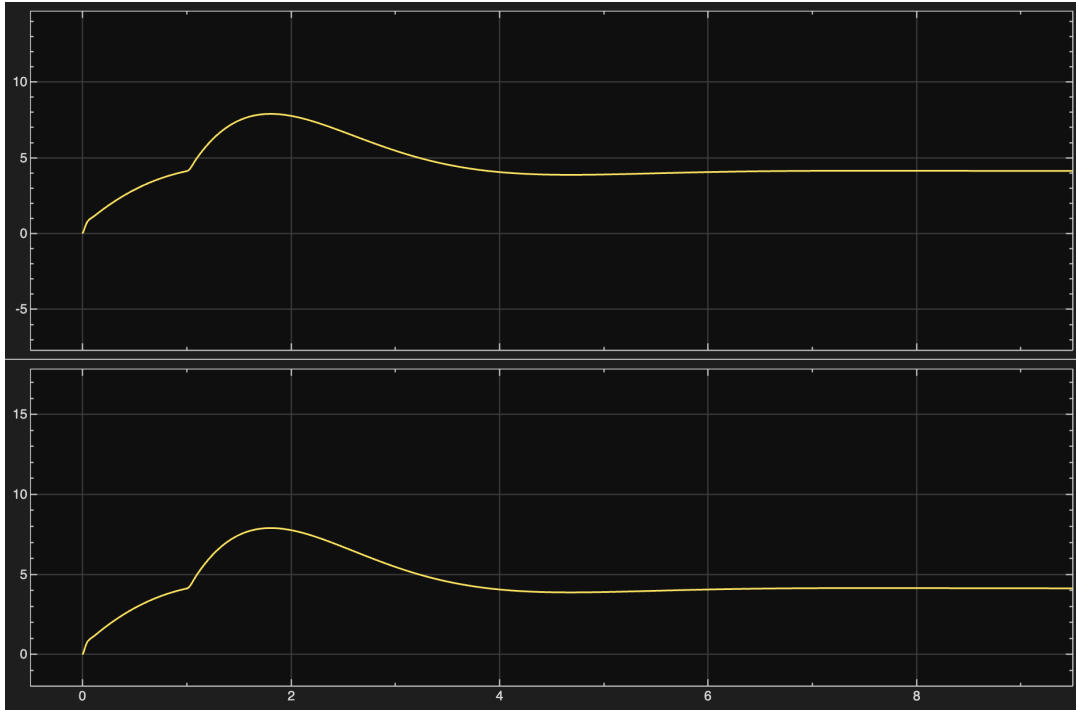
Figure 13: Current estimation: measured/true $I(t)$ vs. $\hat{I}(t)$. The estimate rapidly converges after a short transient.
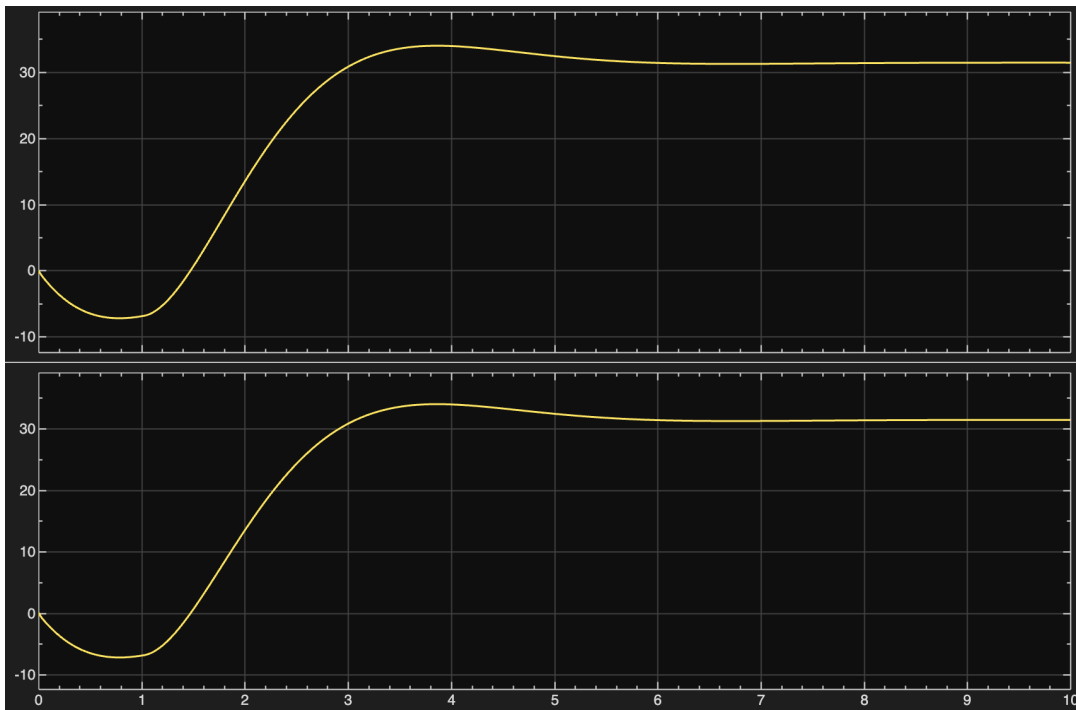

Figure 14: Speed estimation: true $\Omega(t)$ vs. $\hat{\Omega}(t)$. The reconstruction tracks the reference dynamics with small transient error.
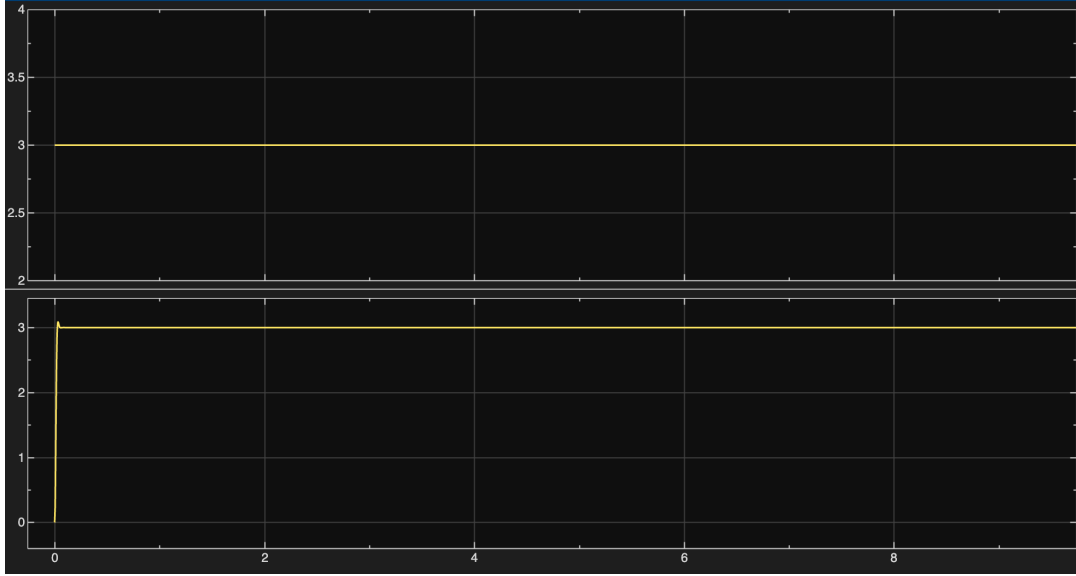
Figure 15: Disturbance torque estimation: true $C_r(t)$ vs. $\hat{C}_r(t)$. The observer converges to the (piecewise constant) applied disturbance.

Comment.   Across all tests the estimated variables converge quickly to their true values, with only a brief initial transient and negligible steady-state bias, which validates the observer design and the choice of K_{obs} (fast poles).

2.3.5   Determination of the Observer Gain

The observer gain K_{obs} is computed by pole placement, ensuring the desired dynamic performance. Using the transposed system matrices, we find:

$$K_{\text{obs}} = \texttt{place}(A_a', C_a', \lambda_{\text{obs}})'$$

MATLAB implementation:

```
K_obs = place(A_a', C_a', lambda_obs);    % Observer gain (transpose form)
K_obs = K_obs';                           % Convert to column vector
```

The resulting observer equations are:

$$\frac{d\hat{x}_a(t)}{dt} = (A_a - K_{\text{obs}}C_a)\hat{x}_a(t) + B_a u_c(t) + K_{\text{obs}}y_a(t)$$

The corresponding matrices of the observer are:

$$A_{\text{obs}} = A_a - K_{\text{obs}}C_a$$
$$B_{\text{obs}} = [\, B_a \;\; K_{\text{obs}} \,]$$
$$C_{\text{obs}} = I_{3\times3}$$
$$D_{\text{obs}} = \mathbf{0}_{3\times2}$$

MATLAB implementation:

Speed Control of an Electrical Drive   25

```
A_obs = A_a - K_obs*C_a;
B_obs = [B_a, K_obs];
C_obs = eye(3);
D_obs = zeros(3,2);
```

2.4   Speed Regulation with Only Proportional Control and Static Compensation of the Resistant Torque with the Value Estimated by the Observer

2.4.1   Choice of eigenvalue and computation of $K_{\Omega p}$

The speed dynamics under proportional control can be written as:

$$\dot{\Omega}(t) = -\frac{f}{J}\Omega(t) + \frac{K_c}{K_{\text{cap}I}J}V_I^*(t) - \frac{1}{J}C_r(t),$$

where $V_I^*(t)$ is the control variable of the outer loop. Assuming $V_I^*(t) = -K_{\Omega p}\Omega(t)$ and neglecting disturbances, the closed-loop pole is:

$$\lambda_{\text{cl}} = -\frac{f}{J} - \frac{K_c}{K_{\text{cap}I}J}K_{\Omega p}.$$

To ensure a stable yet responsive system while keeping the observer faster, the pole was experimentally chosen as:

$$\lambda_{Wp} = -160.$$

Using the system parameters from the identification phase and the MATLAB command:

```
lambda_Wp = -160;
K_Wp = place(-f/J, Kc/(K_capI*J), lambda_Wp);
```

we obtain:
$$\boxed{K_{\Omega p} = 2.48.}$$

This gain provides a good compromise between speed of response and system stability.

2.4.2   Closed-Loop Response of the Proportional Speed Controller

The closed-loop dynamics of the speed loop without integral action are expressed as:

$$\frac{d\Omega(t)}{dt} = \left(-\frac{f}{J} - \frac{K_{\Omega p}K_{\text{cap},\Omega}K_c}{JK_{\text{cap},I}}\right)\Omega(t) + \frac{K_{\Omega p}K_c}{JK_{\text{cap},I}}V_\Omega^*(t)$$

MATLAB implementation:

```
% Question 2.4.b : Closed-loop simulation
A_bfW = -f/J - K_Wp * K_capW * Kc / (K_capI * J);
B_bfW = K_Wp * Kc / (K_capI * J);
C_bfW = 1;
D_bfW = 0;
sys_bf_KW = ss(A_bfW, B_bfW, C_bfW, D_bfW);

VWref_amp = K_capW * 300 * pi / 30;        % Speed reference amplitude (300 rpm)
opt = stepDataOptions('StepAmplitude', VWref_amp);
[y_step_KW, t_step_KW, x_step_KW] = step(sys_bf_KW, opt);
u_step_KW = K_Wp * (VWref_amp - x_step_KW * K_capW); % Control voltage

figure(4)
subplot(2,1,1)
plot(t_step_KW, y_step_KW * 30/pi, 'b', 'LineWidth', 2)
ylabel('Speed (r/min)')
title('Response of the speed loop without integral action')
grid on

subplot(2,1,2)
plot(t_step_KW, 10*u_step_KW, 'r', 'LineWidth', 2)
ylabel('Current (A)')
xlabel('Time (s)')
grid on
```
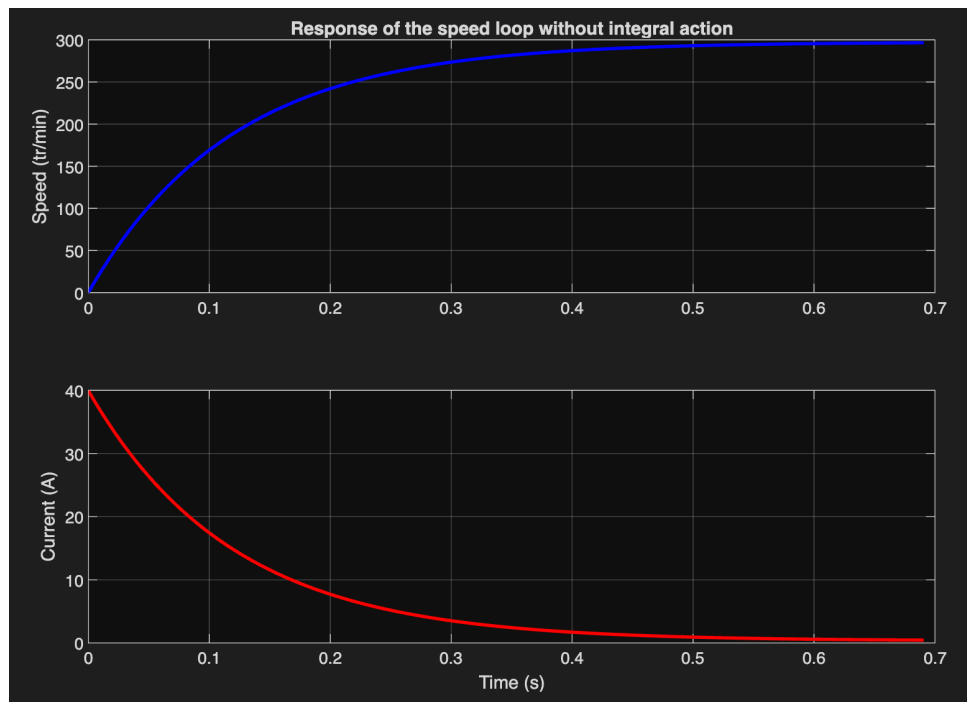

Figure 16: Step response of the speed loop with proportional control only.

Speed Control of an Electrical Drive

Interpretation of results:

- The speed increases smoothly to the reference of 300 r/min with no overshoot.

- The current decreases exponentially as the speed rises, consistent with motor dynamics.

- The system behaves as a first-order response with a short settling time (less than 0.7 s).

The proportional control ensures a stable response, but a small steady-state error remains due to the lack of integral action.

2.4.3   Steady-state error with proportional control only

With proportional speed control and the compensation of C_r disabled, the closed-loop is first order and exhibits a non-zero steady-state error composed of: (i) a term *independent* of C_r (due to finite loop gain), which decreases as $K_{\Omega p}$ increases but cannot be fully cancelled; and (ii) a term *proportional* to C_r, which requires explicit compensation (see 2.4.d).

Numerical evaluation (no disturbance, $C_r = 0$).   For a speed reference of 300 rpm, the MATLAB computation

```
Omega_ss = dcgain(sys_bf_KW) * VWref_amp;   % rad/s
Val_finale = Omega_ss * 30/pi;              % rpm
erreur = 300 - Val_finale;                  % rpm
```

gives $\Omega_{\mathrm{ss}} = 31.1538$ rad/s, i.e.

$$\Omega_{\mathrm{ss}} = 31.1538 \cdot \frac{30}{\pi} \approx 297.50 \text{ rpm}, \quad e_{\mathrm{ss}} = 300 - 297.50 \approx \boxed{2.50 \text{ rpm}}.$$

Simulink observations (with C_r compensation *deactivated*).   Using `TPVitessePartie24.slx`, remove the link from the `K_capI/Kc` block output (disables C_r feedforward):

- **Nominal gain** $K_{\Omega p}$: the response tracks 300 rpm with a small residual offset ≈ 2.5 rpm; see Fig. 17.

- **Halved gain** $K_{\Omega p}/2$: the steady-state error increases and the response slows down (error scales roughly inversely with the proportional gain); see Fig. 18.

- When C_r steps from 0 to 5 N m at $t = 10$ s, a permanent offset appears (the second error term), confirming the need for the compensation in 2.4.d.

2.4.4   Feed-forward compensation of $C_r(t)$ from $\hat{C}_r$

Proposed structure.   Keep the proportional speed loop and add a feed-forward from the estimated resistant torque:

$$V_I^*(t) \;=\; K_{\Omega p}\big(V_\Omega^*(t) - V_\Omega(t)\big) \;+\; K_{\hat{C}_r}\,\hat{C}_r(t),$$

i.e. insert a summing junction at the input of the current loop that adds the signal $\hat{C}_r$ after a static gain $K_{\hat{C}_r}$. This is exactly the structure hinted in the subject (Fig. 16).
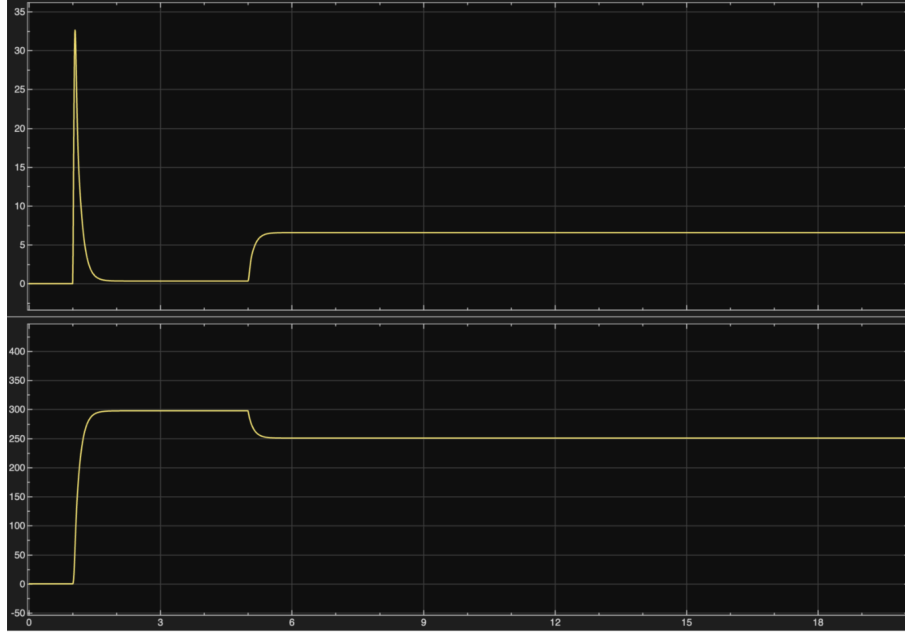
Figure 17: Speed response with $K_{\Omega p}$ (compensation disabled).

Proof of disturbance cancellation. Under the ideal current loop assumption $(I^* = V_I^*/K_{\mathrm{cap}I})$, the speed dynamics are

$$J\dot{\Omega} \;=\; K_c I^* - C_r - f\Omega \;=\; \frac{K_c}{K_{\mathrm{cap}I}}\, V_I^* - C_r - f\Omega.$$

Substituting the proposed V_I^*:

$$J\dot{\Omega} = \frac{K_c}{K_{\mathrm{cap}I}}\, K_{\Omega p}(V_\Omega^* - V_\Omega) \;+\; \frac{K_c}{K_{\mathrm{cap}I}}\, K_{\hat{C}_r}\, \hat{C}_r \;-\; C_r \;-\; f\Omega.$$

Group the terms driven by the (true) disturbance:

$$J\dot{\Omega} = \cdots \;+\; \left(\frac{K_c}{K_{\mathrm{cap}I}}\, K_{\hat{C}_r}\, \hat{C}_r - C_r \right) \;-\; f\Omega.$$

If the observer is exact $(\hat{C}_r = C_r)$, the disturbance channel disappears when

$$\frac{K_c}{K_{\mathrm{cap}I}}\, K_{\hat{C}_r} = 1 \quad \Longleftrightarrow \quad \boxed{K_{\hat{C}_r} = \frac{K_{\mathrm{cap}I}}{K_c}}.$$

Therefore the influence of $C_r(t)$ on $\Omega(t)$ is cancelled with this choice, as required.

Remark on estimation error. If $\hat{C}_r = C_r - e_{C_r}$, the residual input is $-(1/J)\, e_{C_r}$, so perfect cancellation occurs when the observer is accurate; otherwise the steady-state error scales with the estimation error.

Implementation note (Simulink). Enable the feed-forward path from $\hat{C}_r$ to V_I^* through a gain block fixed at $K_{\mathrm{cap}I}/K_c$ (the K_capI/Kc block in TPVitessePartie24.slx). This addition is static feed-forward and does not change the closed-loop poles; it only removes the steady-state disturbance channel (subject to saturation limits).
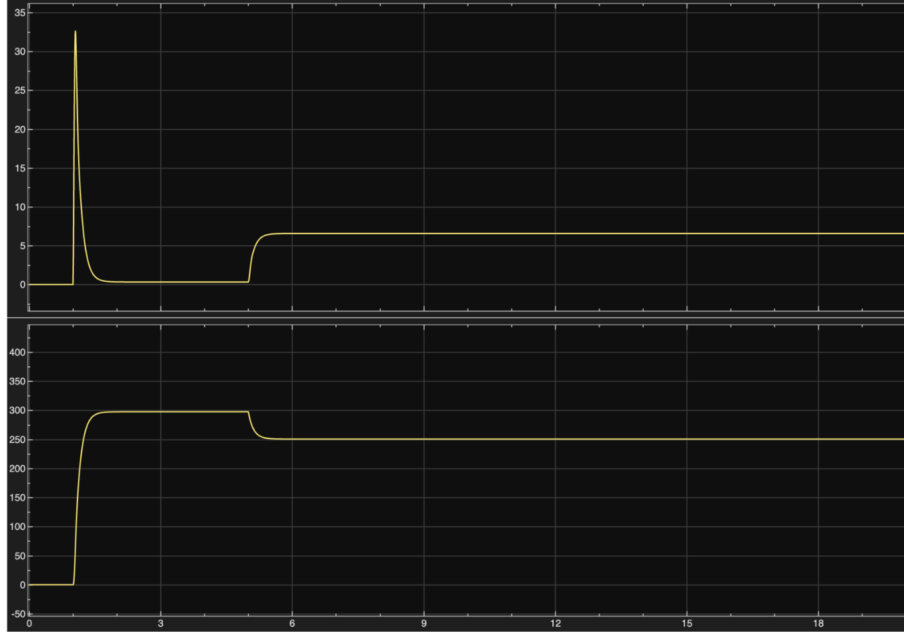
Figure 18: Speed response with $K_{\Omega p}/2$ (compensation disabled).

2.4.5   Validation of steady-state error cancellation (with $\hat{C}_r$ feed-forward)

With the compensation enabled (connecting the block `K_capI/Kc` so that $K_{\hat{C}_r} = K_{\mathrm{cap}I}/K_c$), the command is

$$V_I^*(t) = K_{\Omega p}\big(V_\Omega^*(t) - V_\Omega(t)\big) + \frac{K_{\mathrm{cap}I}}{K_c}\,\hat{C}_r(t),$$

which cancels the disturbance channel under the ideal current loop assumption.

Simulation scenario.   In `TPVitessePartie24.slx`, we apply a speed step Ω^* : $0 \to 300$ rpm at $t = 1\,\mathrm{s}$ and a load step $C_r : 0 \to 5\,\mathrm{N\,m}$ at $t = 5\,\mathrm{s}$. The current and speed responses are shown in Fig. 19.

Results.

- Before the disturbance, $\Omega(t)$ reaches 300 rpm with negligible offset.

- At $t = 5\,\mathrm{s}$, the load step produces a short transient (small dip), after which $\Omega(t)$ returns to the reference with *zero* steady-state error (numerically $< 0.1\%$).

- The current settles to the value needed to balance the load, $I_\infty \approx C_r/K_c$; for $C_r = 5\,\mathrm{N\,m}$ and $K_c \simeq 0.795$, this gives $I_\infty \approx 6.3\,\mathrm{A}$, consistent with the plot.

Conclusion.   Activating the feed-forward $\hat{C}_r$ with $K_{\hat{C}_r} = K_{\mathrm{cap}I}/K_c$ removes the steady-state error of $\Omega(t)$ with respect to $\Omega^*(t)$ despite the load step, as predicted by the analysis in 2.4.d.

Figure 19: Effect of enabling the static compensation $K_{\hat{C}_r} = K_{\mathrm{cap}I}/K_c$ using $\hat{C}_r(t)$ from the observer: the steady-state error due to a C_r step is eliminated while preserving the transient performance.

3   Experimental Validation of the Controllers

This section reports the laboratory validation of the controllers designed in Part 2. Several Simulink configurations were used successively (current loop, speed loop, and finally the observer). Due to limited laboratory time, **only the final implementation corresponding to the observer (Part 3.3) was saved as a representative Simulink diagram**. However, **all measurement results** obtained at each step were recorded and are presented below to assess the performance of every loop. Earlier configurations for the current and speed control followed the same structure as the final one (inner current loop, outer speed loop, voltage saturation, and acquisition blocks).

Practical constraints. For hardware safety, the armature current was limited to 20 A, voltage commands were saturated at ± 9 V, and step amplitudes were increased gradually to avoid excessive mechanical stress and large current transients.

3.1   Current Regulation Experiment

The experimental results of the current control loop are shown in Figure 20. The upper plot displays the armature current (*courant*) and the lower plot shows the corresponding motor speed (*vitesse*). Several step changes were applied to the current reference, increasing successively from approximately 1 A to 2 A, 3 A and finally 10 A. The current follows the reference with a fast transient and negligible overshoot,

demonstrating a well-tuned inner current loop. Small fluctuations in the measured signal are mainly due to sensor noise. The motor speed evolves consistently with the current variations, rising up to about 1000 rpm and then decreasing when the current is reduced or reversed. This confirms the expected electromechanical coupling between torque and speed. At the end of the experiment, both current and speed return smoothly to zero when the converter command is disabled.
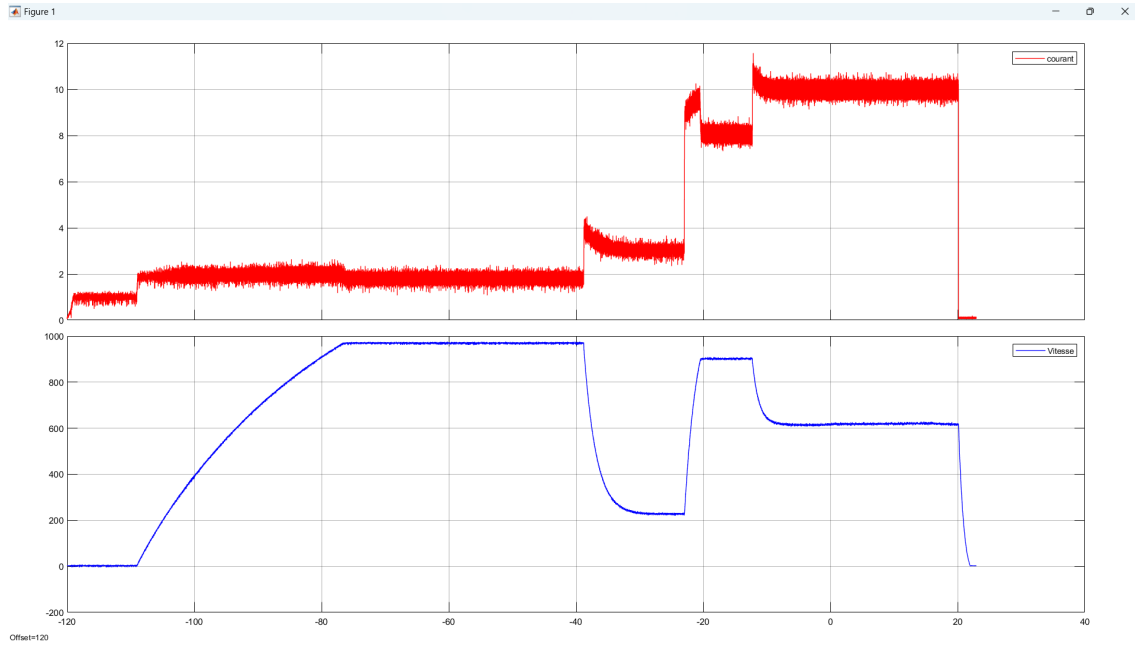

Figure 20: Measured current (top, in A) and speed (bottom, in rpm) during the current regulation test. The current loop shows fast and stable response with negligible overshoot.

3.2   Speed Regulation Experiment

The experimental validation of the speed controller was performed following the four test scenarios defined in Question 2.2.c. For each case, the converter operated in "Automatic" mode, and the speed reference $\Omega^*(t)$ was imposed through the control panel. The measured motor speed (*vitesse*) and armature current (*courant*) were recorded simultaneously.

- **Case 1** : $C_r(t) = 0$, $\Omega^*(t)$ changes from 0 to 300 rpm at $t = 1$ s.

- **Case 2** : $C_r(t) = 0$, $\Omega^*(t)$ changes from 0 to 500 rpm at $t = 1$ s.

- **Case 3** : $C_r(t) = 0$, $\Omega^*(t)$ changes from 0 to 1200 rpm at $t = 1$ s.

- **Case 4** : $\Omega^*(t)$ changes from 0 to 300 rpm at $t = 1$ s, while the resistant torque $C_r(t)$ increases from 0 to 5 Nm at $t = 10$ s.

The measured responses are summarized in Figures 21–24. In all cases, the speed regulation shows a well-damped transient with limited overshoot and negligible steady-state error. For moderate speed commands (300 rpm and 500 rpm),

the current peaks remain within the admissible range (below 20 A), confirming that the current limiter effectively constrains the torque demand. When the reference is increased to 1200 rpm, the system still converges but the transient becomes slower, illustrating the saturation effect of the voltage converter.

When a disturbance torque of 5 Nm is applied at $t = 10$ s, a brief speed drop is observed, immediately compensated by the integral action of the speed controller. The steady-state speed returns to its nominal value within less than one second, demonstrating the robustness of the closed-loop regulation against load disturbances.
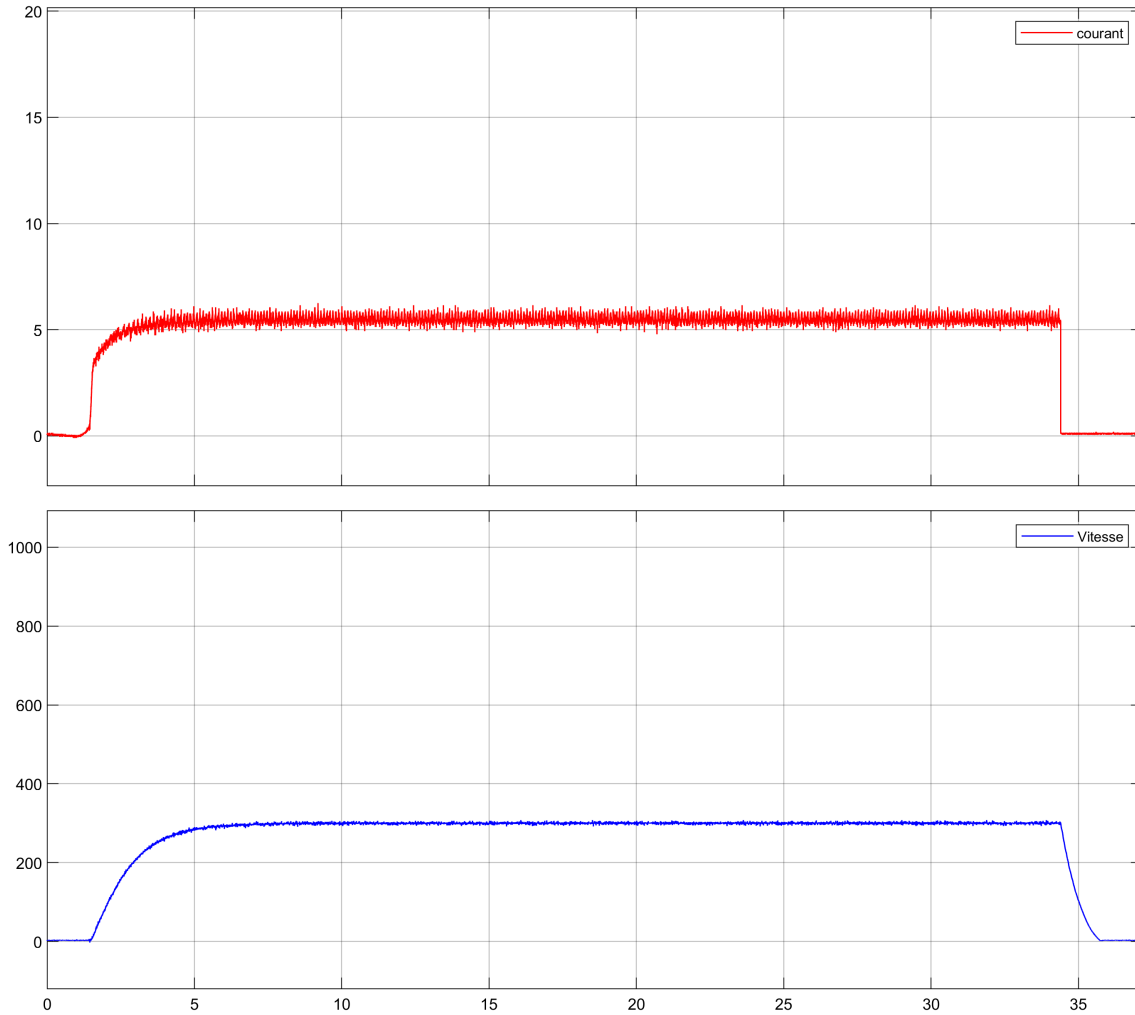
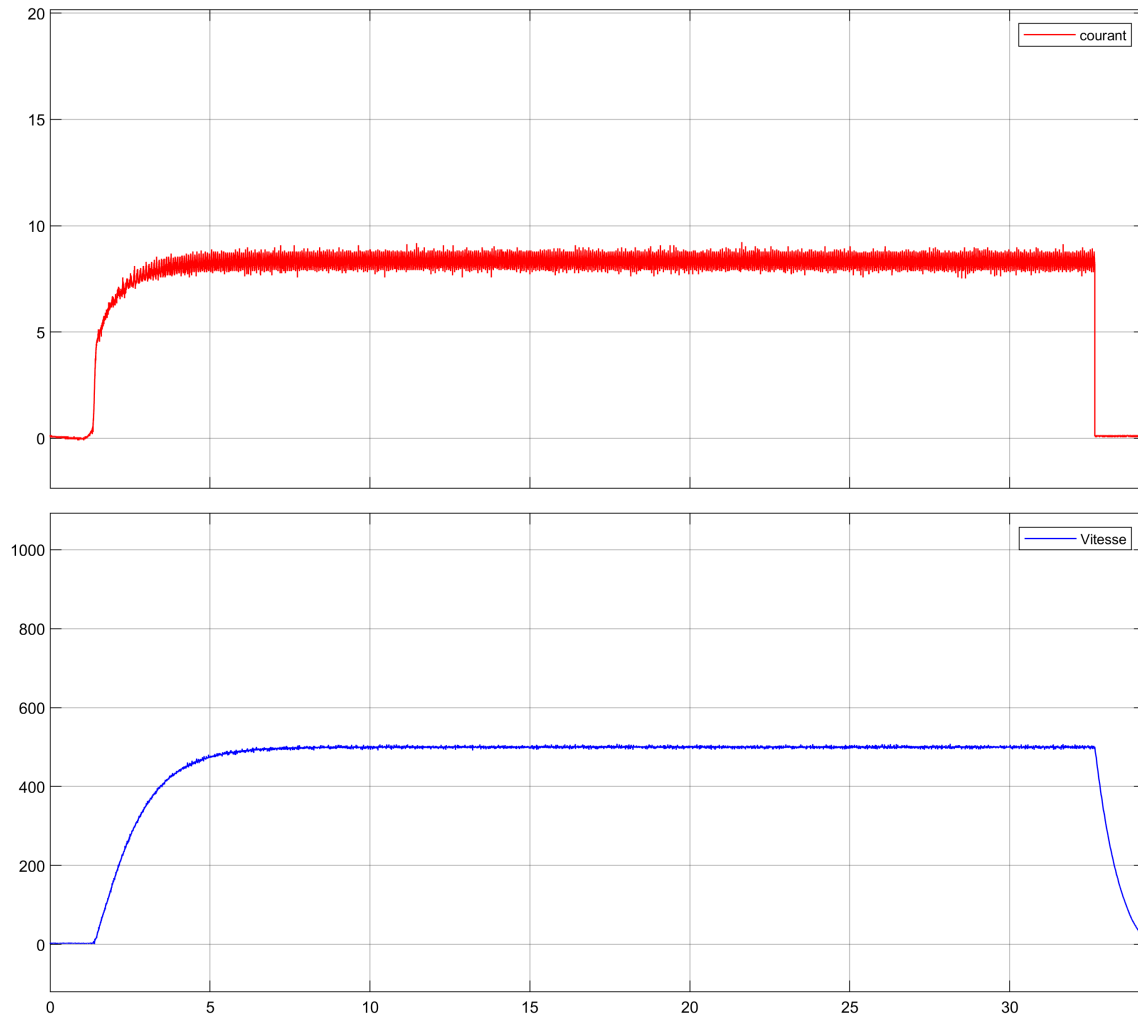Figure 21: Speed loop — measured response for a $0 \rightarrow 300$ rpm reference step.

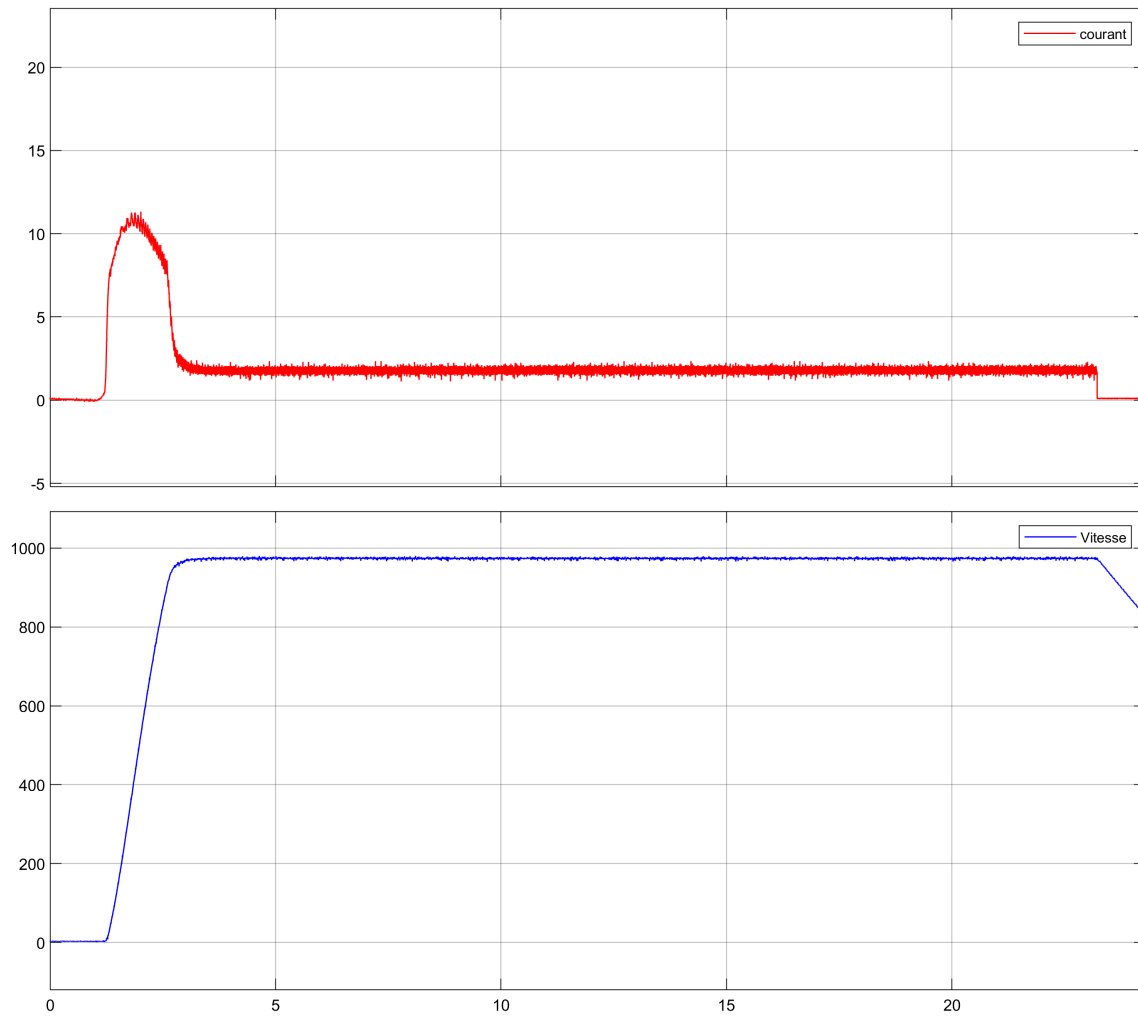Figure 22: Speed loop — measured response for a 0 → 500 rpm reference step.

Figure 23: Speed loop — measured response for a $0 \rightarrow 1200$ rpm reference step, showing converter saturation effects.
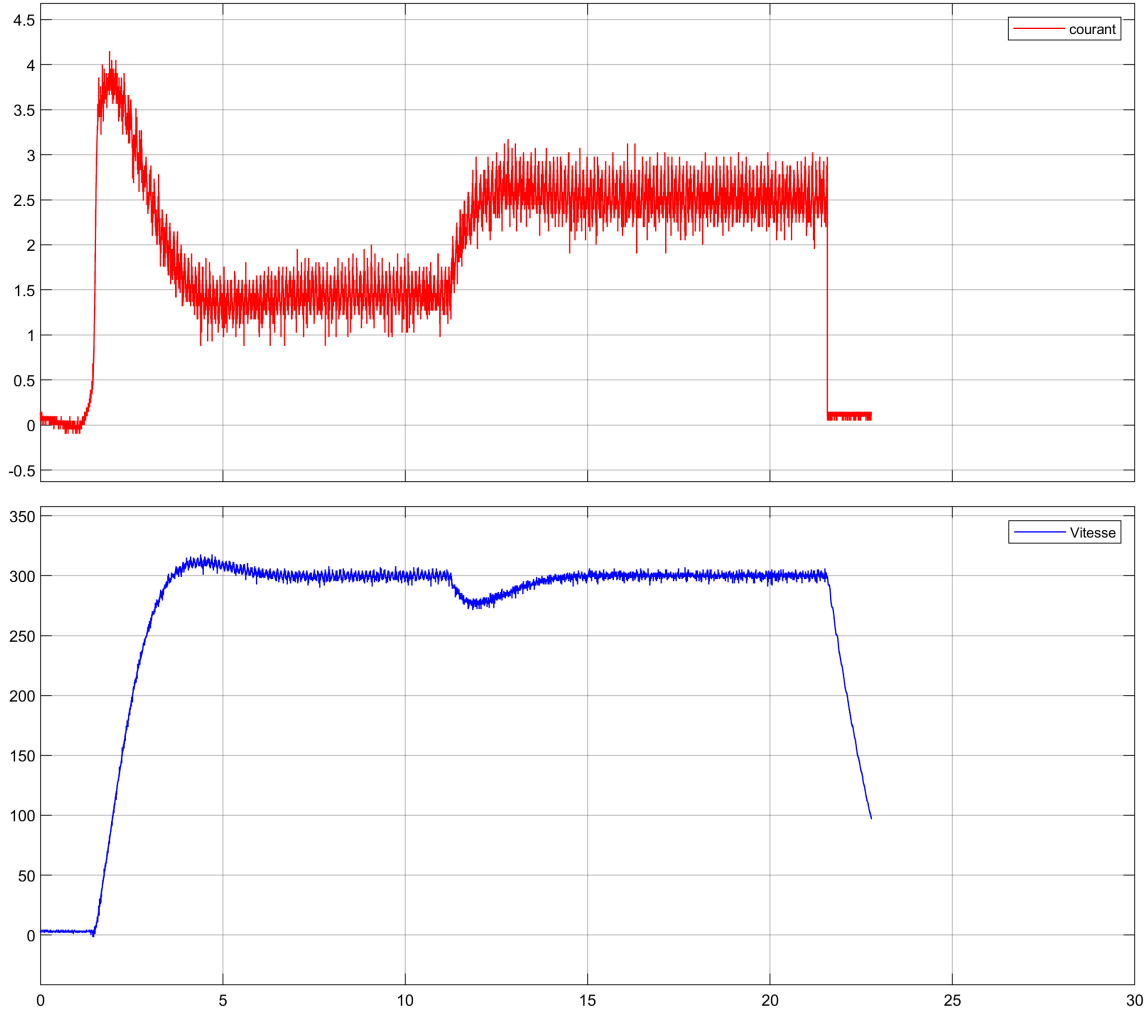
Figure 24: Speed loop — disturbance rejection test: a 5 Nm torque is applied at $t = 10$ s. The controller compensates the perturbation and restores the nominal speed.

3.3   Observer Implementation and Validation

Finally, the observer developed in Section 2.3 was implemented to estimate the states $[\hat{I},\ \hat{\Omega},\ \hat{C}_r]$. The **final Simulink configuration** used for all experiments is shown in Figure 25; it summarizes the complete control architecture (inner current loop, outer speed loop, observer, saturation, and I/O blocks).
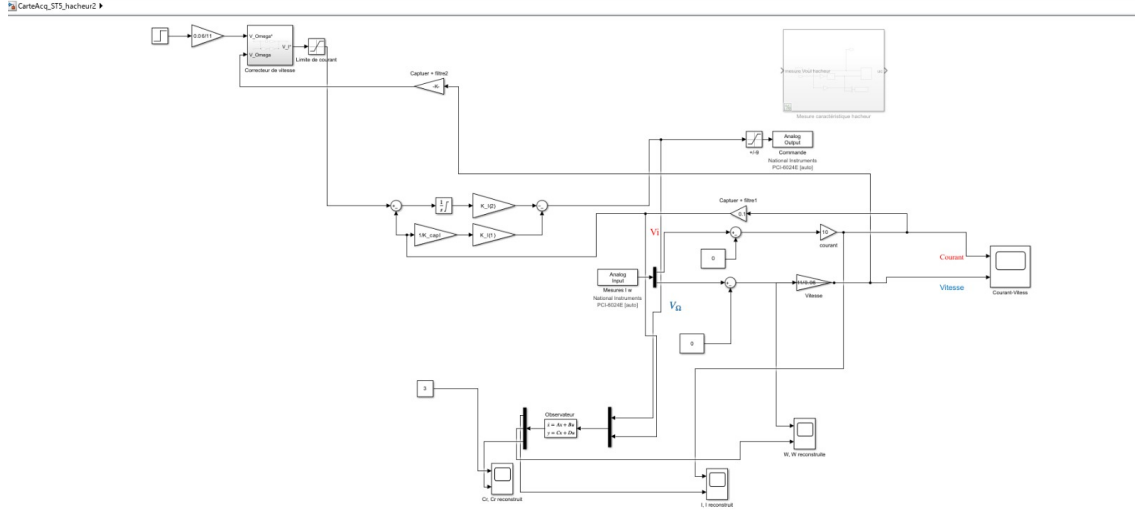
Figure 25: Final Simulink implementation used on the bench (saved during Part 3.3). Earlier configurations for the current and speed loops followed the same structure.

The current estimate converged rapidly and accurately. The speed estimate showed small mismatches and high-frequency noise, mainly due to sensor noise and converter nonlinearities. When using $\hat{C}_r$ for proportional disturbance compensation, the closed loop became sensitive to this noise and could lose stability. Therefore, compensation based on $\hat{C}_r$ was not retained on the bench without additional filtering or gain scheduling. Figure 26 illustrates the convergence of the estimated variables towards their measured counterparts.
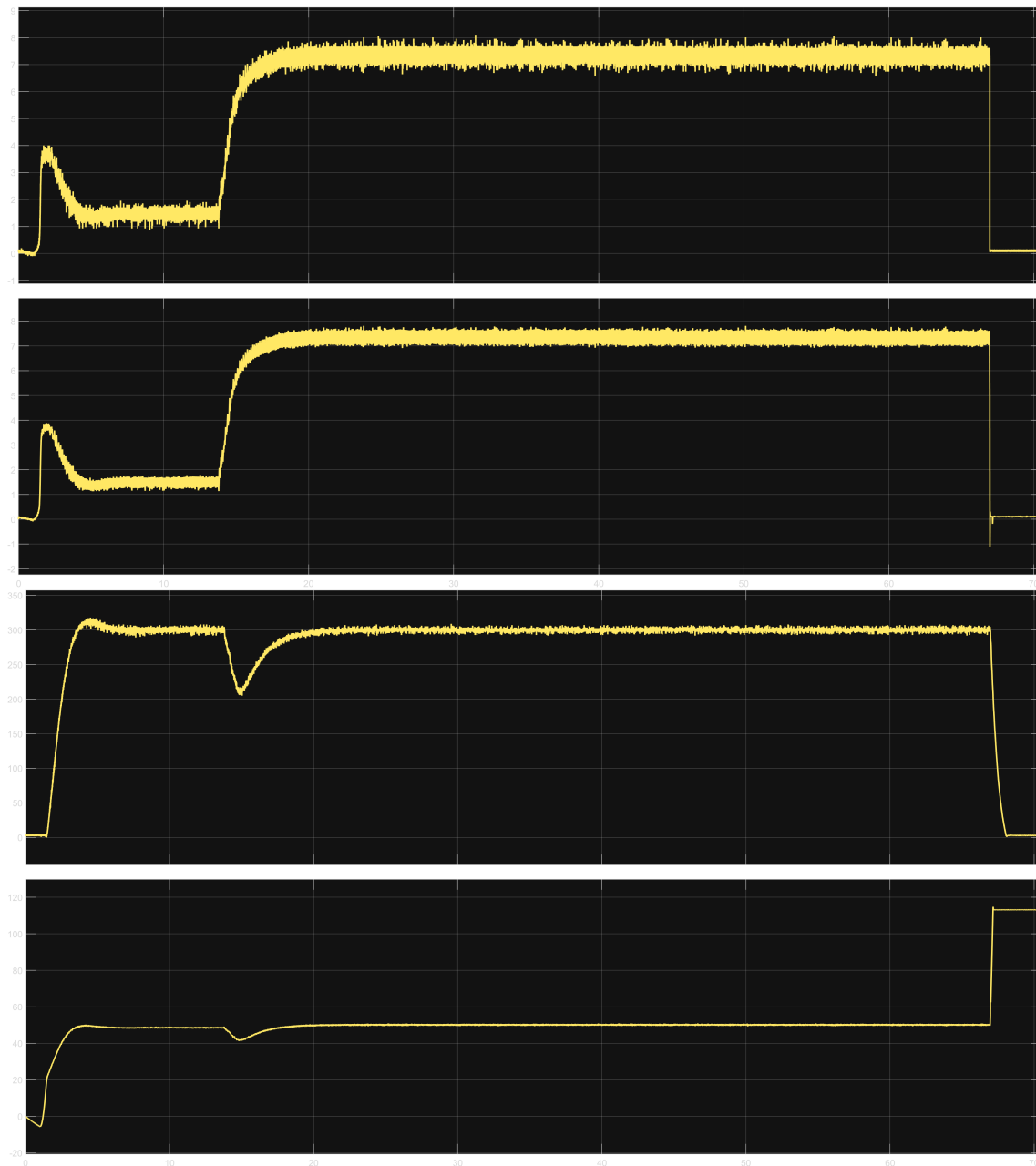
Figure 26: Observer validation — comparison between measured and estimated variables (top: current , bottom: speed).

4   Conclusion

The experimental validation confirmed that the control structures developed in simulation work effectively on the real DC drive, with minor adjustments to account for hardware and safety constraints. The current and speed control loops both met the steady-state and dynamic specifications, demonstrating good robustness against disturbances.

The observer-based approach was successful in simulation and partially in the laboratory. The current estimation was accurate, but the torque estimation was affected by measurement noise, leading to instability when used for disturbance compensation. This highlights the practical challenges of implementing sensorless control on real hardware.

Overall, this practical work successfully connected theoretical modeling, simulation, and experimental implementation. It demonstrated the fundamental principles of cascaded control and observer design, and showed how these techniques can be adapted for real-world electrical drives, where nonlinearity, friction, and converter effects must be carefully handled to ensure system stability and precision.