ECE 5670/6670 – Software Labs
Lab 1 – Introduction to Simulink

Objectives

The objective of the lab is to become familiar with *Matlab/Simulink* by implementing a simple DC motor simulation with a proportional/integral control law. The lab also provides an opportunity to learn about saving figures for reports.

1 Introduction

Videos available on the main web page show the steps needed to perform the experiments of this lab. Other tutorials may also be consulted on-line.

2 Experiments

(a) Create a *Simulink* block diagram to simulate the response of a brush DC motor described by

$$\frac{d\omega(t)}{dt} = -100\omega(t) + 1000v(t), \tag{1}$$

where ω (rad/s) is the angular velocity and v (V) is the voltage applied. Fig. 1 shows the *Simulink* diagram of a possible implementation.

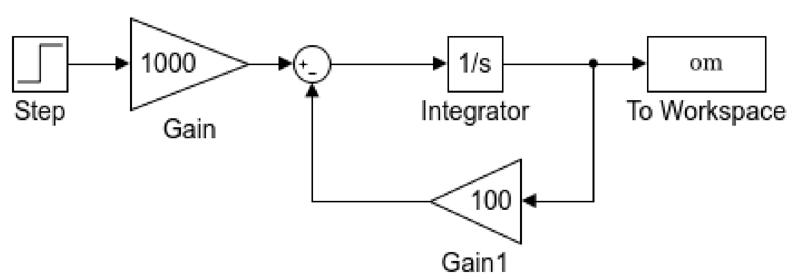

Figure 1: Diagram for the simulation of a first-order system

Simulate the response for 0.2 s, applying a step of voltage of 10 V at $t = 0.1$ s. Plot the speed (in rad/s) as a function of time.

(b) Augment the previous simulation with a proportional-integral controller

$$v(t) = k_P\,(\omega_{REF}(t) - \omega(t)) + k_I \int (\omega_{REF}(\tau) - \omega(\tau))\,d\tau. \tag{2}$$

Let $k_P = 0.02$ and $k_I = 4$. Fig. 2 shows the *Simulink* diagram of a possible implementation.

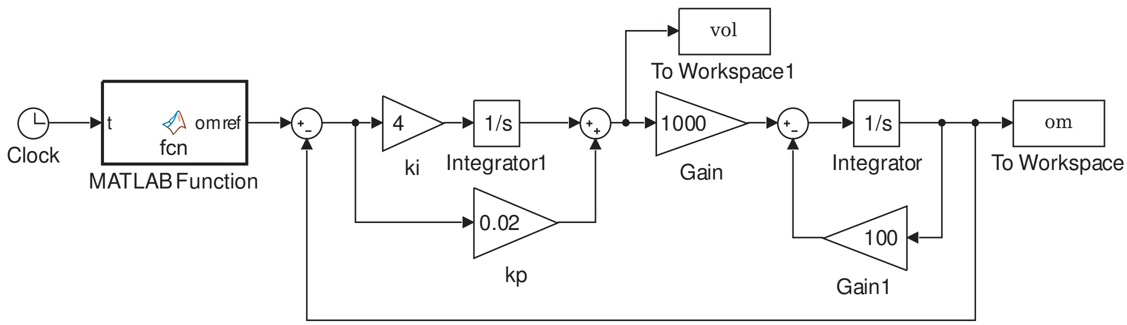

Figure 2: Diagram for the simulation of a first-order system with a PI controller

Simulate the response for 0.5 s, applying a step of reference velocity to 100 rad/s at $t = 0.1$ s and another step to 200 rad/s at $t = 0.3$ s. Plot the speed and the voltage as functions of time. Also include a picture of your final *Simulink* diagram and of any sub-blocks or embedded m-files.

3 Report at a glance

Be sure to include:

- Plot of velocity response to a step of voltage.
- Plot of closed-loop velocity response.
- Plot of closed-loop voltage response.
- Picture of final *Simulink* diagram and of any sub-blocks or embedded m-files.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 2 – Open and closed-loop control of a single-phase motor

Objectives

The objective of this lab is to observe the responses of a single-phase motor controlled using a Hall effect sensor and supplied by a DC source. Speed and torque ripples are estimated. The experiments also show that the motor can be operated under very limited conditions using an open-loop controller.

1 Introduction

A brushless DC fan motor is considered. The model of Example 1 in Chapter 1 is modified for an arbitrary number of pole pairs n_P. The model remains the same, but with the substitutions

$$\theta \rightarrow n_P\theta, \;\; \psi_0 \rightarrow n_P\psi_0 \triangleq K. \tag{3}$$

Overall, the model becomes

$$\begin{aligned} L\frac{di_A}{dt} &= v_A - Ri_A + K\omega\sin(n_P\theta) \\ J\frac{d\omega}{dt} &= -K\sin(n_P\theta)i_A - \tau_{LF} \\ \frac{d\theta}{dt} &= \omega, \end{aligned} \tag{4}$$

where v_A (V) and i_A (A) are the voltage and the current applied to the motor winding, ω (rad/s) is the angular velocity of the motor, θ (rad) is the angular position, and τ_{LF} (N m) is the load and friction torque. The motor torque is

$$\tau_M = -K\sin(n_P\theta)i_A. \tag{5}$$

The load and friction torque is assumed to be due to aerodynamic friction and be of the form

$$\tau_{LF} = B_2\,\omega^2\text{sign}(\omega). \tag{6}$$

The parameters are:

- R (Ω), the resistance of the phase winding.
- L (H), the inductance of the phase winding.

- K (N m/A or V s), the torque constant, or back-emf constant.
- J (kg m^2), the rotor's moment of inertia.
- B_2 (N m s^2), the coefficient of the friction torque.

The motor is controlled electronically using a single Hall effect sensor and a DC supply, so that

$$v_A = -V_{DC}\ \text{sign}\left(\sin(n_P\theta)\right), \tag{7}$$

where V_{DC} is the DC supply voltage.

2 Experiments

(a) Implement the model of the motor and control law in *Simulink*. Fig. 3 shows the *Simulink* diagram of a possible implementation. The *Control* block implements (7) and the *Single-phase motor model* block implements (4) and (6).

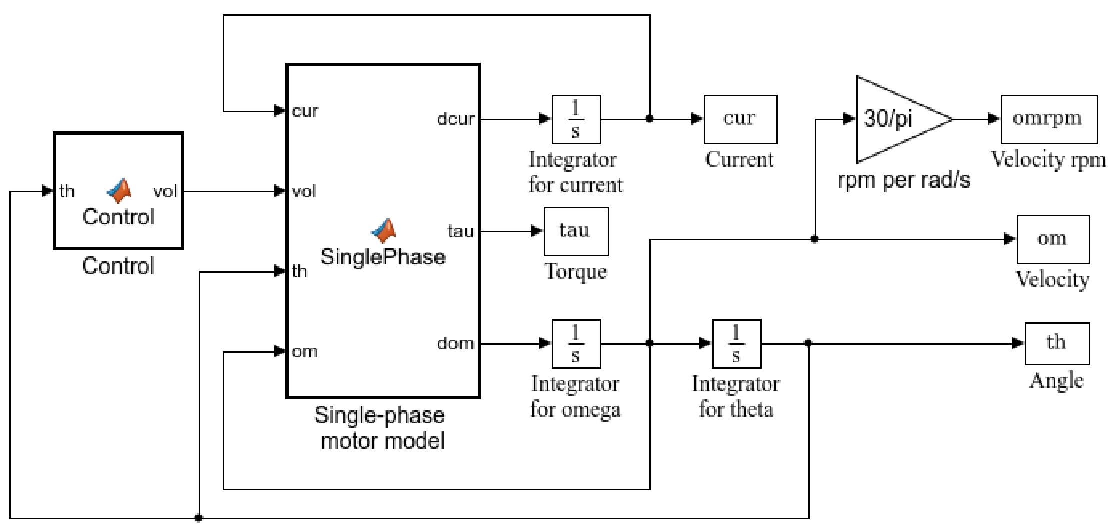

Figure 3: Diagram for the simulation of a single-phase motor

Let $R = 26\ \Omega$, $L = 5$ mH, $K = 0.024$ N m/A, $J = 2.8\ 10^{-5}$ kg m^2, $B_2 = 4.6\ 10^{-8}$ N m s^2, $n_P = 2$, and $V_{DC} = 12$ V. Let all initial conditions be zero, except for $\theta(0) = \pi/6$. Note that the initial condition can be set in the integrator for θ. Plot the speed (ω) in rpm and the motor torque (τ_M) in N m as functions of time for 10 s. Also plot a close-up from 9.5 to 10 s. Estimate the average speed and the average torque in steady-state, and the percent ripples (approximated as half of the peak-to-peak value divided by the average value). To increase the precision of the results, it is recommended to reduce the relative tolerance in *Simulink*'s *Configuration Parameters* to $1e-6$.

(b) Repeat the simulations for $V_{DC} = 3$ V, 6 V, 9 V, and 12 V and plot the average steady-state speed as a function of the DC voltage.
(c) Replace the control law by the open-loop control law

$$v_A = -V_{DC}\ \mathrm{sign}(\sin(n_P \theta_{REF})). \tag{8}$$

The angle θ of (7) was replaced by a desired angle

$$\theta_{REF} = \omega_{REF}\ t, \tag{9}$$

and ω_{REF} is some desired rotational speed. In *Simulink*, θ_{REF} can be obtained by applying ω_{REF} to an integrator block. Simulate the responses of the motor with this open-loop control law. The objective is to show that the motor will rotate at the speed ω_{REF}, but only for a very restricted set of frequencies. Find a constant speed reference ω_{REF} for which the motor position θ follows the reference position θ_{REF} (search in the range from 10 to 50 rad/s). Plot θ and θ_{REF} as functions of time. Comment on the speed ripple.
(d) Repeat the simulation of part (c) for $\theta(0) = -\pi/6$ and comment on the results.

3 Report at a glance

Be sure to include:

- Plot of the speed and torque over 10 s and over the last 0.5 s.
- Give estimates of average speed and motor torque, and values of ripple.
- Plot of the average steady-state speed as a function of the DC voltage.
- Plot θ and θ_{REF} for a speed ω_{REF} such that θ follows θ_{REF} approximately, and value of ω_{REF}.
- Plot of the response for $\theta(0) = -\pi/6$.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 3 – Modeling of the brush DC motor

Objectives

The objective of the lab is to observe the characteristics of a brush DC motor. The responses of the DC motor to steps of voltage are measured, and the data is used to estimate its parameters.

1 Introduction

A model for the brush DC motor with Coulomb and viscous friction is

$$\begin{aligned} L\frac{di}{dt} &= v - Ri - K\omega \\ J\frac{d\omega}{dt} &= Ki - C\ \mathrm{sign}(\omega) - B\omega, \end{aligned} \tag{10}$$

where v (V) is the voltage applied to the motor, i (A) is the current, and ω (rad/s) is the angular velocity of the motor. The parameters are:

- R (Ω), the armature resistance.
- L (H), the armature inductance.
- K (N m/A or V s), the torque constant, also called the back-emf constant.
- J (kg m^2), the rotor's moment of inertia.
- C (N m), the coefficient of Coulomb friction.
- B (N m s), the coefficient of viscous friction.

Assuming that the inductance is negligible, $v = Ri + K\omega$ and the following approximate model is obtained

$$J\frac{d\omega}{dt} = -(\frac{K^2}{R} + B)\ \omega + \frac{K}{R}v - C\ \mathrm{sign}(\omega). \tag{11}$$

With positive direction of motion, the Coulomb friction can be viewed as a constant disturbance, and the transfer function from the voltage to the angular velocity is a first-order linear time-invariant system.

2 Experiments

2.1 Parameter estimation

The simulation model *Lab3.mdl* applies a sequence of voltages to the motor at specified times. Fig. 4 shows the diagram of the simulation. The *Command generator* applies six increasing steps of voltages to the motor.

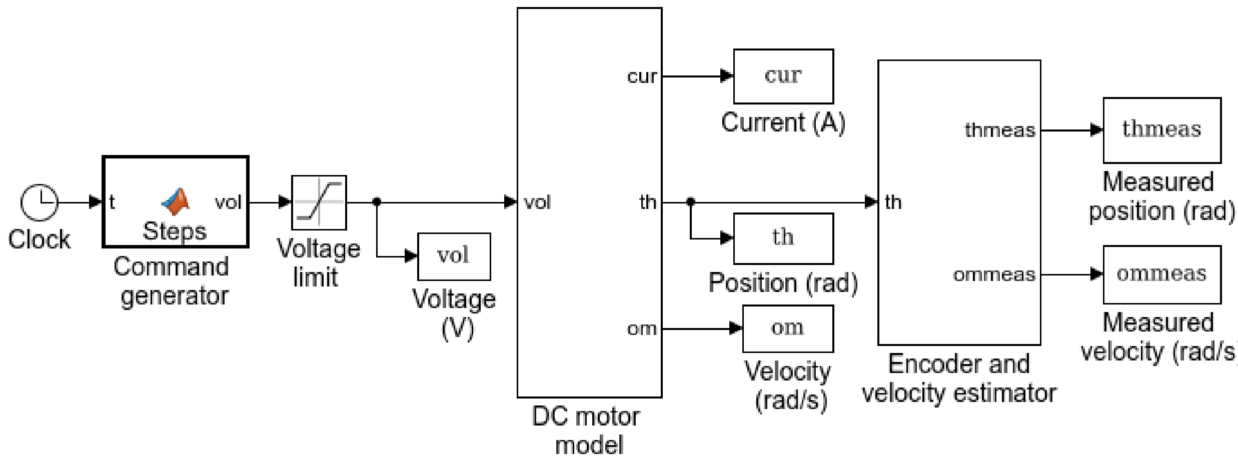

Figure 4: Diagram of the brush DC motor simulation

Make yourself familiar with the model, then run the simulation. In *Matla*b's workspace, plot $v(t)$, $\omega(t)$ and $i(t)$. Note that, after a transient period starting when $v(t)$ changes, $\omega(t)$ and $i(t)$ reach constant values. These values are such that

$$\begin{aligned} v &= R\,i + K\omega \\ i &= \frac{C}{K} + \frac{B}{K}\,\omega. \end{aligned} \tag{12}$$

Proceed with the following tasks:

(a) Using *Matlab*'s data cursor, measure the steady-state values (choosing a middle value when there is noise), and create a table with the six values of voltage, velocity, and current.

(b) Plot $i(\omega)$ as well as your best fit of the line $i = a + b\omega$. The parameters a and b are your estimates of C/K and B/K.

(c) Repeat part (b) with a plot of v/i as a function of ω/i, and deduce estimates of R and K. Using the results of part (b), deduce values of C and B.

(d) Plot $\omega(t)$ and, from the response to the first step of voltage, estimate the time constant of the system. Based on the first-order model (11), the time constant is equal to $RJ/(K^2 + BR)$. Deduce an estimate of J.

(e) Plot $i(t)$ and, from the response to the first step of voltage, estimate di/dt at the initial time of transition, where $L\,di/dt = v$. Deduce an estimate of L.

(f) Observe that the variable defining the simulation step size is T_S using *Simulation/Model Configuration Parameters*. Find the value of T_S in the initialization function by looking for *Model Explorer* and clicking *Lab3*, *Callbacks*, *InitFcn**. The initialization function is a useful technique to initialize parameters that appear in several places in the simulation model. Verify that the delay observed in the response of part (e) matches the simulation step size.

(g) Look inside the simulation of the brush DC motor and compare the estimated values of parts (b), (c), (d), and (e) to the values used in the model. Observe that some parameters are estimated more reliably than others with the method proposed.

2.2 Sensor noise

The simulation model includes a model of the encoder that is used in practice to measure the angular position. The model is a *quantizer*, performing an operation similar to the one performed in A/D converters. The measured velocity is obtained from the measured position using

$$\omega_{meas}(t) = \frac{\theta_{meas}(t) - \theta_{meas}(t - T_S)}{T_S}, \tag{13}$$

where T_S is the sampling period. Differentiation of the quantized position produces significant noise on the velocity estimate. Averaging of the velocity over N_{avg} samples reduces the noise, and is equivalent to replacing T_S by $N_{AVG}T_S$ in (13).

Proceed with the following tasks:

(a) Find the value of N_{AVG} in the *Encoder and velocity estimator* block.

(b) Find the quantization level $\Delta\theta$ from the quantizer block of the model. Also observe the effect of quantization by plotting and zooming on $\theta_{meas}(t)$. Because the data includes periods of low velocity, the quantization level can be determined from the minimum non-zero value of the signal obtained by applying the *Matlab* function *diff* to $\theta_{meas}(t)$.

(c) Plot $\omega_{meas}(t)$ and zoom on the steady-state response after the first step of speed reference to obtain an estimate of the peak-to-peak velocity noise on the measurement. Repeat for the last step of velocity. Observe that the level of noise $\Delta\omega$ on the velocity is the same at low and high velocities and can be related to the level of quantization noise $\Delta\theta$ using

$$\Delta\omega = \frac{\Delta\theta}{N_{AVG}T_S}. \tag{14}$$

Note that averaging of the velocity estimate reduces the noise, but also delays the velocity estimate.

3 Report at a glance

Be sure to include:

- Table of values of ω, v, i.
- Plots of $i(\omega)$, v/i (ω/i), $\omega(t)$, $i(t)$ and values of the estimated parameters.
- Value of T_S.
- Comparison of the estimated parameters to the values of the model.
- Values of N_{AVG}, $\Delta\theta$, and $\Delta\omega$ and calculation for $\Delta\omega$.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 4 – Control of a brush DC motor

Objectives

The objective of the lab is to experiment with a proportional-integral (PI) control law for the tracking of velocity references by a brush DC motor. A proportional-integral-derivative (PID) control law is also tested for positioning. Controller parameters are selected using a pole placement procedure, and modifications are applied to the control law to avoid overshoot in the responses.

1 Introduction

An approximate model of a brush DC motor under voltage command is

$$\frac{d\omega}{dt} = -a\ \omega + k_1\ v, \tag{15}$$

where

$$a = \frac{K^2}{RJ},\ k_1 = \frac{K}{RJ}. \tag{16}$$

Friction is neglected and treated as a disturbance for the purpose of control. The transfer function of the system is

$$\frac{\omega(s)}{v(s)} = \frac{k_1}{s+a}. \tag{17}$$

In this lab, the first-order system representation of the motor is used to design control laws for velocity and position.

2 PI control of velocity

A proportional-integral (PI) control law for the velocity consists in choosing a voltage

$$v = k_P(\omega_{REF} - \omega) + k_I \int (\omega_{REF} - \omega) dt, \tag{18}$$

where ω_{REF} is the reference velocity, and k_P and k_I are the proportional and integral gains. The transfer function of the compensator is

$$C(s) = k_P + \frac{k_I}{s}, \tag{19}$$

so that the closed-loop transfer function is given by

$$\frac{\omega(s)}{\omega_{REF}(s)} = \frac{k_1(k_P s + k_I)}{s^2 + (a + k_1 k_P)s + k_1 k_I}. \tag{20}$$

A possible option consists in choosing the PI gains so that the zero of the compensator matches the pole of the plant. Then,

$$z = -\frac{k_I}{k_P} = -a. \tag{21}$$

This choice results in a pole/zero cancellation, which is acceptable if $a > 0$ and is well-known. The closed-loop transfer function becomes

$$\frac{\omega(s)}{\omega_{REF}(s)} = \frac{k_1 k_P}{s + k_1 k_P}, \tag{22}$$

which is a first-order system with a pole at $s = -k_1 k_P$. For a desired pole at $s = -a_D$, the PI gains are given by

$$k_P = \frac{a_D}{k_1},\ \ k_I = \frac{a a_D}{k_1}. \tag{23}$$

Note that, with this design choice, a feedforward gain k_F is not needed in the PI controller.

3 PID control of position

For the control of position, the transfer function of the system becomes

$$\frac{\theta(s)}{v(s)} = \frac{k_1}{s(s+a)}. \tag{24}$$

A proportional-integral-derivative (PID) control law is used and is given by

$$v = k_P(\theta_{REF} - \theta) + k_I \int (\theta_{REF} - \theta)dt + k_D \frac{d}{dt}(\theta_{REF} - \theta), \tag{25}$$

where θ_{REF} is the reference position of the motor, and k_P, k_I, and k_D are the proportional, integral, and derivative gains, respectively. The response of the closed-loop system depends on all three parameters in a complex manner, making manual tuning difficult. To simplify the selection of the controller parameters, a method is used to compute the gains as functions of a single parameter, namely the desired location of the closed-loop poles.

With

$$C(s) = k_P + \frac{k_I}{s} + k_D s, \tag{26}$$

the closed-loop transfer function becomes

$$\frac{\theta(s)}{\theta_{REF}(s)} = \frac{(k_D s^2 + k_P s + k_I)k_1}{s^3 + (a + k_D k_1)s^2 + k_P k_1 s + k_I k_1}. \tag{27}$$

The closed-loop poles are all placed at some desired value $s = -a_D$ if the denominator of the transfer function is equal to $s^3 + 3a_D s^2 + 3a_D^2 s + a_D^3$. This property leads to the values of the PID gains

$$k_P = \frac{3a_D^2}{k_1},\ k_I = \frac{a_D^3}{k_1},\ k_D = \frac{3a_D - a}{k_1}. \tag{28}$$

The zero of the compensator is equal to

$$z = -\frac{k_I}{k_P} = -\frac{a_D}{3}. \tag{29}$$

4 Improvements

Some modifications are brought to the control law to improve the responses:

(a) **Anti-windup:** for large steps of the reference input, the response of the system exhibits an overshoot, even if no overshoot is observed for small steps. The problem occurs because the control input reaches a limit and the response of the system deviates from the linear response. The integrator value becomes large, but the output does not produce the negative feedback needed to limit the integrator state. To fix the problem, integration may be stopped when a limit is reached.

(b) **Modified gains:** in the PID control law, the derivative action is not applied to θ_{REF}, so that large control values do not result from step changes in the reference input. A feedforward gain k_F is also applied to the reference angle in the proportional path to move the zero of the closed-loop transfer function. The control law becomes

$$v = k_F k_P \theta_{REF} - k_P \theta + k_I \int (\theta_{REF} - \theta) dt - k_D \frac{d\theta}{dt}, \tag{30}$$

With the modifications, the transfer function becomes

$$\frac{\theta(s)}{\theta_{REF}(s)} = \frac{(k_F k_P s + k_I)k_1}{s^3 + (a + k_D k_1)s^2 + k_P k_1 s + k_I k_1}. \tag{31}$$

The controller zero is moved to

$$z_F = -\frac{k_I}{k_F k_P} = -\frac{a_D}{3k_F}. \tag{32}$$

Possible choices include $k_F = 2/3$, which gives a fast response with very small overshoot, $k_F = 1/3$, which results in the cancellation of a closed-loop pole, and $k_F = 0$, which removes the controller zero altogether.

5 Experiments

5.1 PI control of velocity

(a) Implement the PI control law (18) using the brush DC motor simulation file *Lab3.mdl*. Use the ω_{meas} output from the velocity estimator as the feedback to the controller. For a_D, use 120 rad/s (this value is approximately equal to $2a$, so that this choice represents an acceleration of the open-loop response by a factor of 2). Add anti-windup protection to account for the voltage limit of 25 V (note that an implementation of anti-windup is given on p. 94 of the textbook).

Simulate the response of the system for a sequence of reference values equal to 0 rpm, 1000 rpm, 2000 rpm, 3000 rpm, 2000 rpm, 1000 rpm, and 0 rpm, all separated by 1 s from each other. Remember, though, that the speed in the simulation is expressed in rad/s. Plot ω_{meas} and ω_{REF} (in rpm) as functions of time and discuss the results. In particular, explain why the motor is not capable of reaching the highest speed.

Fig. 5 shows a possible organization of the overall simulation, including the PI controller as a sub-block. A suggestion for the PI controller is to implement the algorithm as the combination of an integrator and a *Matlab* m-function, as shown on Fig. 6. All the equations of the algorithm can then be coded rapidly in the PI controller block. Using motor parameters obtained from the simulation model, compute exact values of the controller gains. Check that the values rounded to the first significant digit are $k_P = 0.2$ and $k_I = 10$.

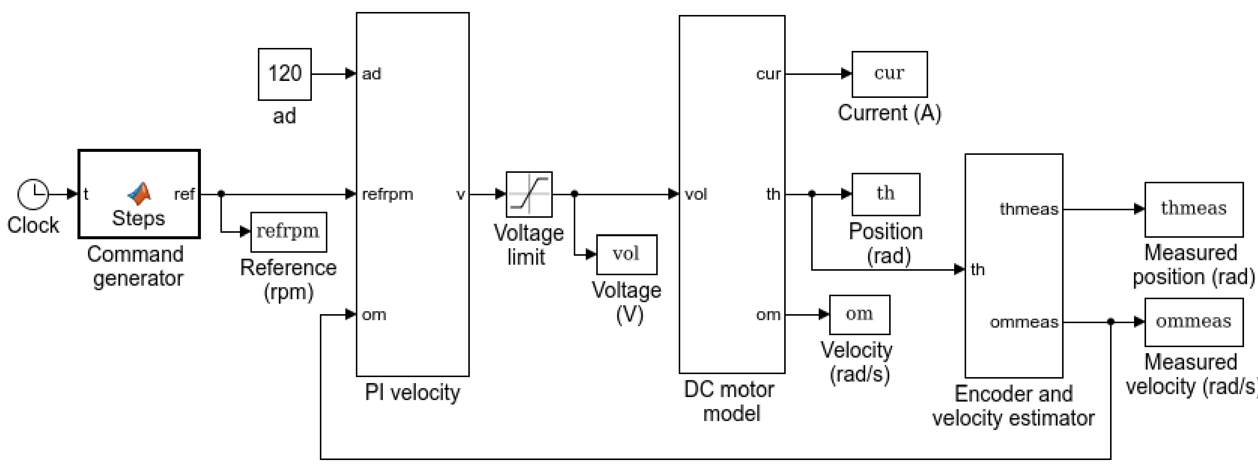

Figure 5: Diagram for the simulation of a PI velocity controller

(b) Repeat part (a) with a larger value of a_D (800 rad/s), and discuss the results.

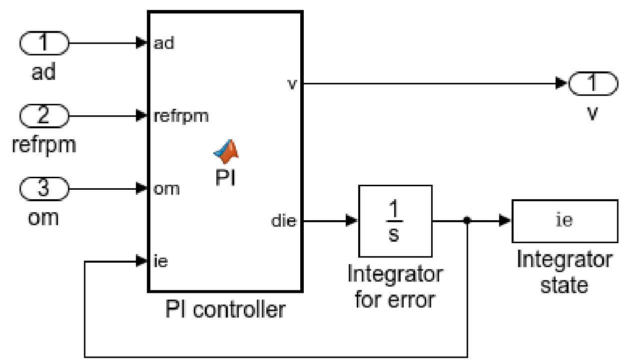

Figure 6: Suggested implementation of the PI controller

5.2 PID control of position

(a) Implement the PID control law (30) with $a_D = 60$ rad/s and $k_F = 0$. Fig. 5 shows a possible organization of the new simulation. Check that your values of the controller gains rounded to the first significant digit are $k_P = 20$, $k_I = 400$, and $k_D = 0.2$. Use ω_{meas} for $d\theta/dt$ and θ_{meas} for θ. Include anti-windup protection to account for the voltage limit of 25 V. For debugging, however, it may be convenient to first remove the voltage limit in the simulation and test the control algorithm without anti-windup.

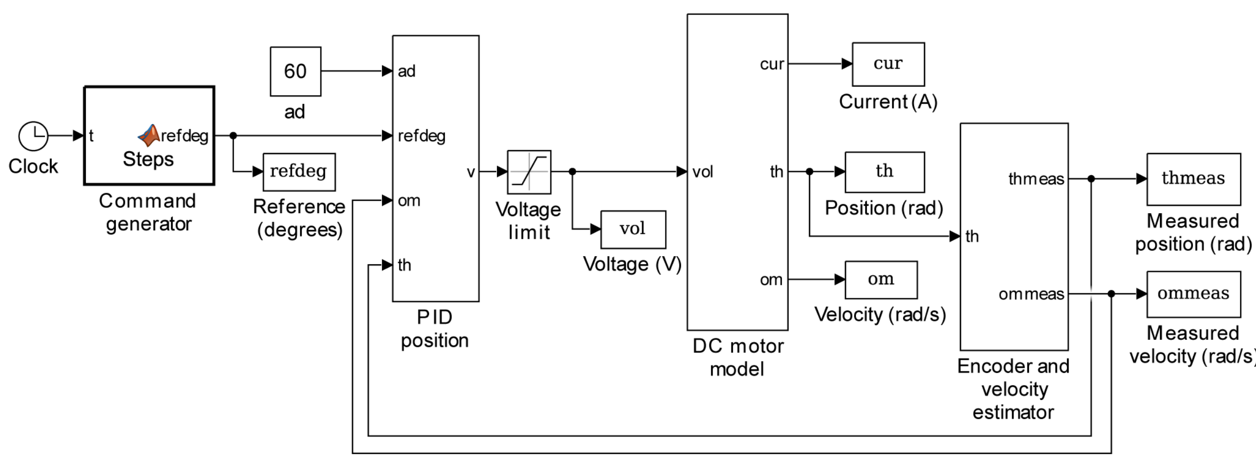

Figure 7: Diagram for the simulation of a PID position controller

Simulate the response of the system for a sequence of reference values equal to 0 deg, 90 deg, 3600 deg, and 0 deg, all separated by 1 s from each other. Remember, though, that the position in the simulation is expressed in rad. Plot θ_{meas} and θ_{REF} (in degrees) as functions of time and discuss the results. Also plot the responses focusing on the first 90 degree step response.

(b) Test the system of part (a) without the anti-windup protection (keeping $k_F = 0$), and observe the windup phenomenon.

(c) Test the system of part (a) with anti-windup protection and with $k_F = 1$. Observe the overshoot on the response to the 90° step. As the voltage limit is not reached, the behavior is linear so that this overshoot is different from the one observed in part (b).

6 Report at a glance

Be sure to include:

- Values of the PI gains for the velocity controller.
- Plot of ω_{meas} and ω_{REF} as functions of time for $a_D = 120$ rad/s.
- Plot of ω_{meas} and ω_{REF} as functions of time for $a_D = 800$ rad/s.
- Values of the PID gains for the position controller.
- Plot of θ_{meas} and θ_{REF} as functions of time.
  Additional plot focused on the 90° step.
- Plot of θ_{meas} and θ_{REF} with anti-windup protection disabled (integral action should remain).
- Plot of θ_{meas} and θ_{REF} with low magnitude zero ($k_F = 1$) focused on the 90° step.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 5 – Open-loop control of a two-phase PM synchronous motor

Objectives

The objective of the lab is to experiment with different methods of control for two-phase PM synchronous motors. Sinusoidal and four-step commutation methods are implemented. The maximum speed attainable with these methods is investigated. The oscillatory response to a step in four-step commutation mode is also observed. A DQ method of control is implemented, and the significant increase of speed attainable using field weakening is observed.

1 Introduction

1.1 Two-phase PM synchronous motor model

The model of the motor is given by

$$\begin{aligned}
L\frac{di_A}{dt} &= v_A - Ri_A + K\omega\sin(n_P\theta) \\
L\frac{di_B}{dt} &= v_B - Ri_B - K\omega\cos(n_P\theta) \\
J\frac{d\omega}{dt} &= -Ki_A\sin(n_P\theta) + Ki_B\cos(n_P\theta) - \tau_{LF} \\
\frac{d\theta}{dt} &= \omega,
\end{aligned} \tag{33}$$

where v_A and v_B (V) are the voltages applied to the two phases of the motor, i_A and i_B (A) are the currents, ω (rad/s) is the angular velocity of the motor, θ (rad) is the angular position, and τ_{LF} (N m) is the load and friction torque. The parameters are:

- R (Ω), the resistance of a winding.
- L (H), the inductance of a winding.
- K (N m/A or V s), the torque constant, as well as the back-emf constant.
- J (kg m^2), the rotor's moment of inertia.
- n_P, the number of pole pairs.

1.2 Open-loop control

Methods of open-loop control include:

- sinusoidal commutation where the stator voltages are two-phase sinusoidal voltages

$$v_A = V_{pk}\cos(\theta_S),\;\; v_B = V_{pk}\sin(\theta_S), \tag{34}$$

  with

$$\frac{d\theta_S}{dt} = \omega_S. \tag{35}$$

  The variable $\omega_S = 2\pi\, f_S$ is the angular frequency (in rad/s) and f_S is the stator electrical frequency (in Hz).

- four-step commutation methods (one-phase-on and two-phase-on) where the sinusoidal voltages are replaced by square wave approximations. The two-phase-on method corresponds to

$$v_A = V_{pk}\ \mathrm{sign}(\cos(\theta_S)),\;\; v_B = V_{pk}\ \mathrm{sign}(\sin(\theta_S)), \tag{36}$$

  where the sign function is such that $\mathrm{sign}(x) = 1$ for $x > 0$, $\mathrm{sign}(0) = 0$, and $\mathrm{sign}(x) = -1$ for $x < 0$.

The reference speed (equal to the synchronous speed) and the reference angle corresponding to the two-phase voltages are

$$\omega_{REF} = \frac{\omega_S}{n_P},\;\; \theta_{REF} = \frac{\theta_S}{n_P}. \tag{37}$$

In general, the motor speed ω will converge to the reference speed *if the motor stays synchronized.*

1.3 Response to a single step

The dynamics of the system (33) linearized around $n_P\theta = 0°$, $\omega = 0$, $i_A = I$, and $i_B = 0$ are given by

$$\frac{d}{dt}\begin{pmatrix} \Delta i_A \\ \Delta i_B \\ \Delta\omega \\ \Delta\theta \end{pmatrix} = \underbrace{\begin{pmatrix} -R/L & 0 & 0 & 0 \\ 0 & -R/L & -K/L & 0 \\ 0 & K/J & 0 & -KIn_P/J \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{A}\underbrace{\begin{pmatrix} \Delta i_A \\ \Delta i_B \\ \Delta\omega \\ \Delta\theta \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} 1/L & 0 \\ 0 & 1/L \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{B}\underbrace{\begin{pmatrix} \Delta v_A \\ \Delta v_B \end{pmatrix}}_{u}. \tag{38}$$

The load and friction torque was assumed to be zero. The eigenvalues of the matrix A specify the poles of the system and are the same for other stable equilibrium positions (e.g., $n_P\theta = 90°$, $\omega = 0$, $i_A = 0$, and $i_B = I$). Two eigenvalues of the system matrix are lightly damped for typical values of motor parameters and the imaginary parts of the eigenvalues are between ω_a and ω_b, where

$$\omega_a = \sqrt{\frac{KIn_P}{J}}, \ \omega_b = \sqrt{\frac{KIn_P}{J} + \frac{K^2}{LJ}}. \tag{39}$$

In one-phase-on mode, $I = V/R$, where V is the amplitude of the voltages. In two-phase-on mode, the equilibrium positions are shifted by 45° and the eigenvalues satisfy the same formulas, but with the magnitude of the current vector replaced by $I = \sqrt{2}V/R$.

1.4 DQ transformation and DQ model

A useful representation of the motor is obtained through the DQ transformation

$$\begin{aligned} v_d &= \cos(n_P\theta)\ v_A + \sin(n_P\theta)\ v_B \\ v_q &= -\sin(n_P\theta)\ v_A + \cos(n_P\theta)\ v_B, \end{aligned} \tag{40}$$

whose inverse is given by

$$\begin{aligned} v_A &= \cos(n_P\theta)\ v_d - \sin(n_P\theta)\ v_q \\ v_B &= \sin(n_P\theta)\ v_d + \cos(n_P\theta)\ v_q. \end{aligned} \tag{41}$$

In the DQ coordinates, the model of the motor becomes

$$\begin{aligned} L\frac{di_d}{dt} &= v_d - Ri_d + n_P\omega Li_q \\ L\frac{di_q}{dt} &= v_q - Ri_q - n_P\omega Li_d - K\omega \\ J\frac{d\omega}{dt} &= Ki_q - \tau_{LF} \\ \frac{d\theta}{dt} &= \omega. \end{aligned} \tag{42}$$

For negligible inductance L,

$$v_q = Ri_q + K\omega, \tag{43}$$

and the response from v_q to ω is similar to the response of a voltage-controlled brush DC motor with negligible inductance, namely

$$J\frac{d\omega}{dt} = -\frac{K^2}{R}\omega + \frac{K}{R}v_q - \tau_{LF}. \tag{44}$$

1.5 Field weakening

At low speeds, it is reasonable to set $v_d = 0$, resulting in $i_q \simeq v_q/R$. At high speeds, however, the current i_q rapidly decreases for a given voltage v_q because of inductive effects and due to the back-emf $K\omega$. In order to increase the torque available, the current i_q may be maximized for a bounded voltage by setting a relationship between the DQ voltages, such as

$$v_d = -\frac{n_P \omega L}{R} v_q. \tag{45}$$

Equation (45) is an optimal solution to the problem of torque maximization under voltage constraints and is called *field weakening*.

2 Experiments

2.1 Open-loop response with sinusoidal commutation

The file *Lab5.mdl* contains a simulation of a stepper motor, which is an example of two-phase PM synchronous motor. Fig. 8 shows the Simulink diagram of the simulation. The *Command generator* applies two-phase sinusoidal voltages of constant magnitude and with frequency increasing in steps, so that the motor progressively accelerates.

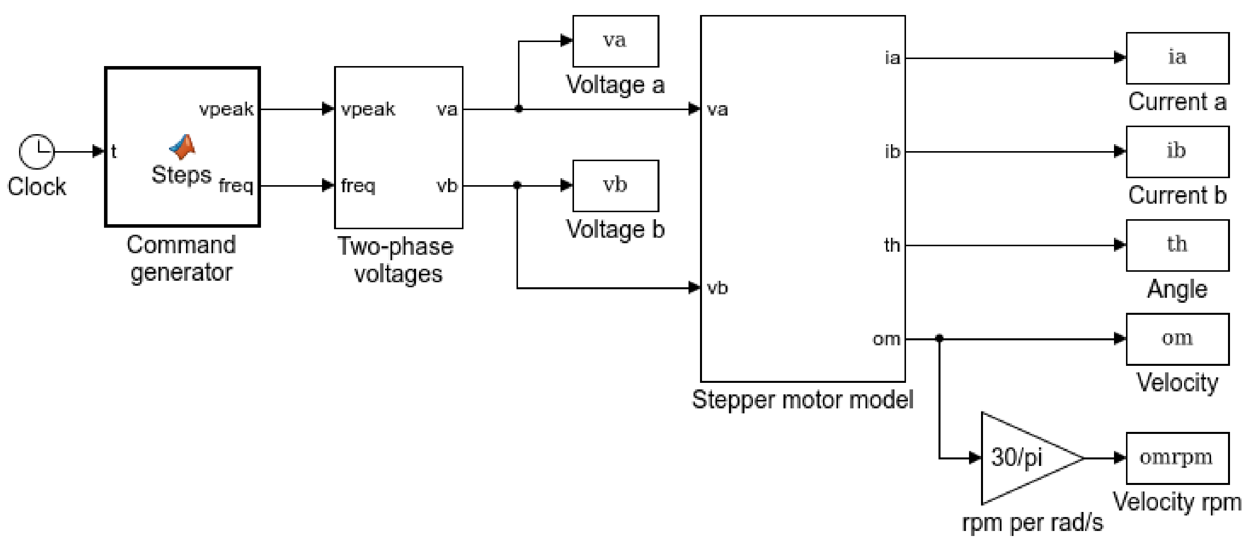

Figure 8: Diagram of the simulation of a stepper motor

Run the simulation and plot the speed (in rpm) as a function of time, together with the reference speed using (37) (the number of pole pairs of the motor is $n_P = 12$). Observe that the response becomes increasingly oscillatory as the speed increases.

2.2 Open-loop response with increasing frequency

Replace the contents of the command generator so that the frequency f_S (variable *freq*) increases linearly from $f_S = 0$ to $f_S = f_{MAX}$, where f_{MAX} is a variable set in the command generator. The frequency should increase over 20 s, then stay at f_{MAX} for the remainder of the simulation time, which is 40 s. Set the value of the peak voltage to a constant 25 V.

Simulations of the motor show that:

- for $f_{MAX} = 0 \rightarrow 54$ Hz, the motor velocity and the angle track their references given by (37).
- for $f_{MAX} = 55 \rightarrow 64$ Hz, the motor velocity oscillates significantly, but the angle continues to track the reference (at least approximately).
- for $f_{MAX} \geqslant 65$ Hz, the motor velocity drops and the motor either stops or rotates at a slower speed than the reference.

Verify these claims by plotting the speed together with the reference speed (in rpm) for $f_{MAX} =$ 54 Hz, 55 Hz, and 65 Hz. Also plot the angular position θ (in rad) together with the reference $\theta_{REF} = \theta_S/n_P$ (in rad), where θ_S is the variable *alpha* found inside the *Two-phase voltages* block. Finally, plot the stator voltages v_A and v_B over a time period corresponding to about three cycles of the sinusoidal commutation sequence, choosing a period at the end of the experiment where the frequency is constant at 65 Hz.

2.3 Open-loop response with two-phase-on four-step commutation

Replace the two-phase voltages block by a commutation block as shown on Fig. 9. In the code, add an option for two-phase-on four-step commutation using (36). An input of the block (labelled *com* on Fig. 9) gives the option of choosing between different commutation methods, but is not necessary for the lab.

It turns out that the behavior with four-step commutation is similar to the behavior with sinusoidal commutation, but with the frequency ranges: $0 \rightarrow 58$, $59 \rightarrow 72$, and above 73 Hz. In this case, simulate the responses for $f_{MAX} = 72$ Hz and $f_{MAX} = 73$ Hz, Plot the speed together with the reference speed (in rpm), and plot the angular position θ together with the reference θ_{REF}. Also plot the voltages over a time period including about three cycles of the four-step commutation sequence at 73 Hz.

Observe that the motor exhibits complicated resonant behavior, in addition to loss of synchronism. Peaks are visible in the velocity plots at low and high frequencies. Give the reference speeds at which these peaks occur and the corresponding frequencies of the stator voltages (in Hz).

2.4 Stepping mode

Replace the linearly increasing frequency command by a fixed frequency of 6 Hz and let the simulation time be 0.15 s. Plot the angular position of the motor (in rad) for the two-phase-on mode. Observe the oscillatory response in stepping mode, and verify that the equilibrium

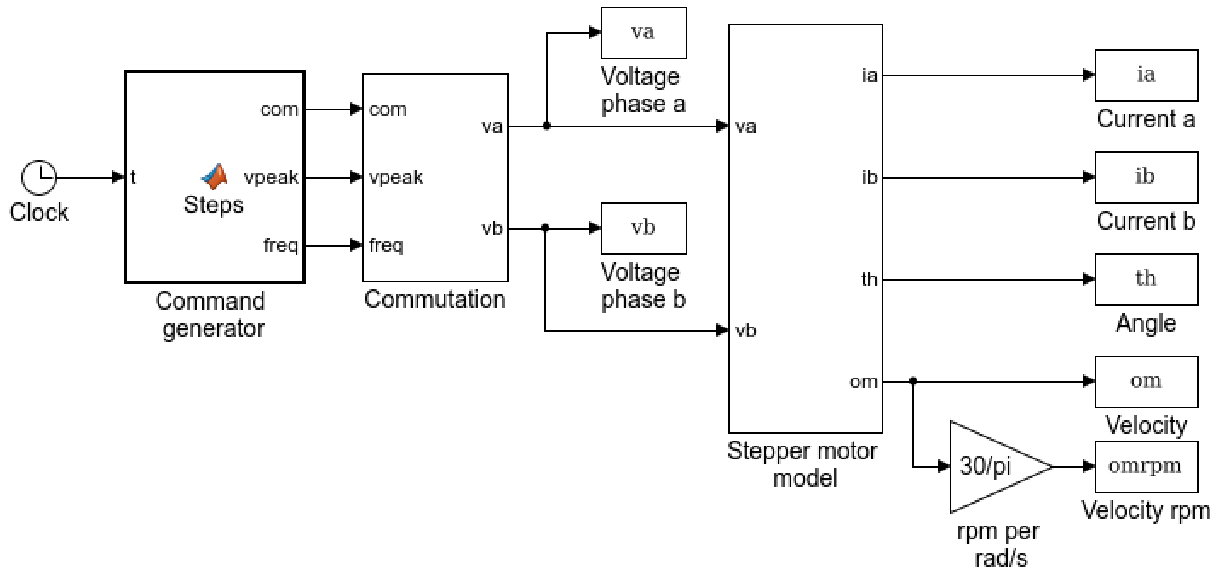

Figure 9: Diagram for the simulation of four-step commutation

positions are at $n_P\theta^* = 45°$, $135°$, ... Note that the first step is smaller because the motor is initialized at the zero position.

Measure the magnitude of a step from the plot (in rad and deg), and compare to the expected value of $90°/n_P$ using $n_P = 12$. From the response, estimate the frequency of oscillations using the time separating the first two peaks of the oscillatory response. Compare the value of the frequency to those predicted by ω_a and ω_b in (39) (convert from rad/s to Hz) using the parameters found inside the simulation model and $I = \sqrt{2}V/R$.

2.5 Open-loop DQ control

Replace the commutation block in the simulation by a DQ control block with V_{pk} as an input, as shown in Fig. 10. The variable *fw* is used to choose between no field weakening ($fw = 0$) and field weakening ($fw = 1$). For $fw = 0$, let $v_d = 0$ and $v_q = V_{pk}$, where V_{pk} increases linearly from 0 to 25 V over 20 s, then stays at 25 V. Compute v_A and v_B using the inverse DQ transformation (41). Simulate the responses for 30 s and plot the speed in rpm.

Next, set $fw = 1$ and, for that value, replace v_d by the value specified by the field weakening formula (45), and adjust v_q such that $v_d^2 + v_q^2 = V_{pk}^2$. Plot the speed in rpm and compare the maximum speed reached with both options of DQ control. Compare the results to those obtained with the sinusoidal and four-step commutation modes. Note that the solution should be such that $v_d < 0$ and $v_q > 0$.

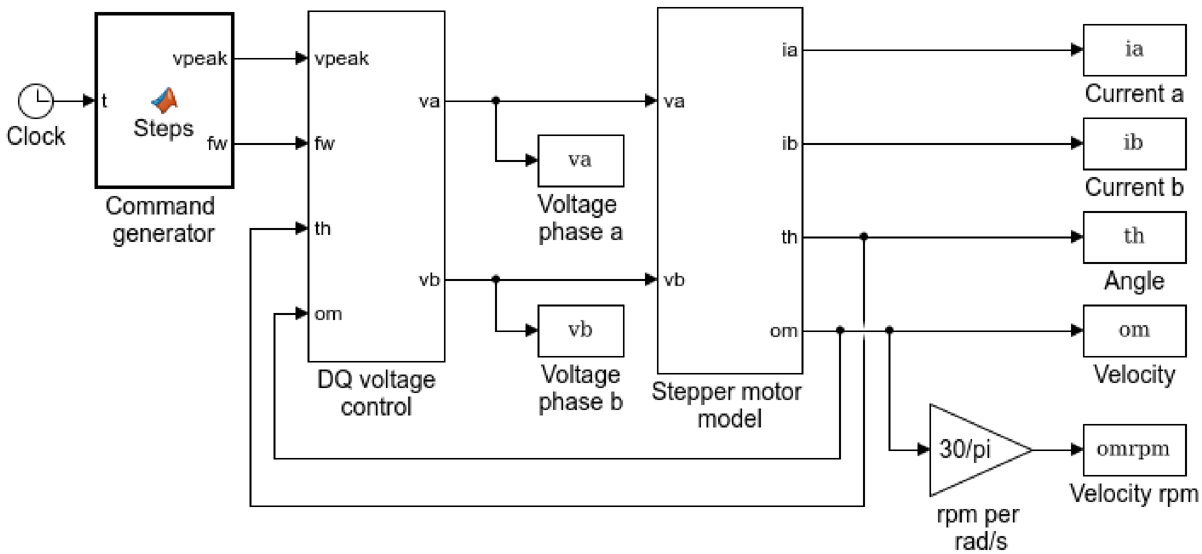

Figure 10: Diagram for the simulation of open-loop DQ control

3 Report at a glance

Be sure to include:

- Plot of open-loop speed response and reference speed (in rpm) with increasing steps of frequency.
- Plot of open-loop speed response and reference speed (in rpm) with sinusoidal commutation and linearly increasing frequency for the three frequencies f_{MAX} specified. Plots of angular position and reference (in rad) and plot of voltages over about three cycles.
- Plot of open-loop speed response and reference speed (in rpm) with two-phase-on commutation and linearly increasing frequency for the two frequencies f_{MAX} specified. Plots of angular position and reference (in rad) and estimates of the references where the low speed resonances occur. Plot of voltages over about three cycles.
- Plot of step response with two-phase-on commutation. Verification of the step size and estimation of the frequency of oscillation. Comparison with the formulas.
- Plot of velocity (in rpm) with DQ control and $v_d = 0$. Value of the maximum speed achieved.
- Plot of velocity (in rpm) with DQ control and field weakening. Value of the maximum speed achieved.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 6 – Brushless DC motor control with six-step commutation

Objectives

The objective of the lab is to control the speed of a brushless DC motor using a six-step commutation scheme. The commutation method is first tested, followed by a current control loop, and then by a velocity control loop.

1 Introduction

A standard circuit for the control of a brushless DC motor is the three-phase inverter shown on Fig. 11. A resistor with value R_{SENSE} is placed at the bottom of the circuit as a means of sensing the motor current, with $I_{SENSE} = V_{SENSE}/R_{SENSE}$.

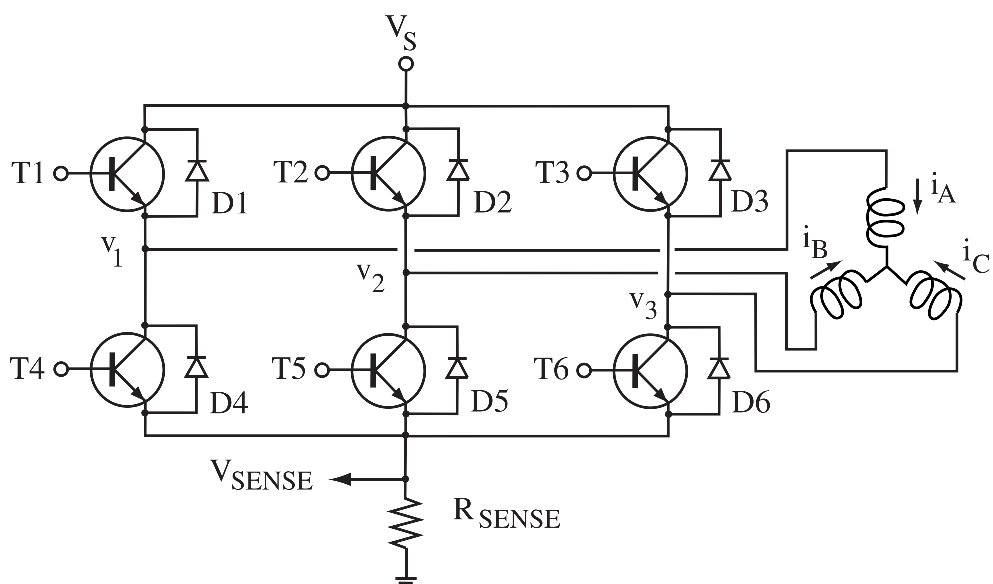

Figure 11: Three-phase inverter and brushless DC motor

Transistors are turned on based on the following commutation table.

Sector	$n_P\theta$	Transistors on
S1	$-30° \to 30°$	T2, T6
S2	$30° \to 90°$	T2, T4
S3	$90° \to 150°$	T3, T4
S4	$150° \to 210°$	T3, T5
S5	$210° \to 270°$	T1, T5
S6	$270° \to 330°$	T1, T6

2 Experiments

2.1 Open-loop stepping

The file *Lab6.mdl* contains a simulation of the brushless DC motor. Fig. 12 shows the diagram of the simulation. If the commands of six-step commutation are applied in sequence at regular intervals, the motor moves in a stepping mode where the steps are at the equilibrium positions $n_P\theta^* = 90°, 150°, 210°, \cdots$.

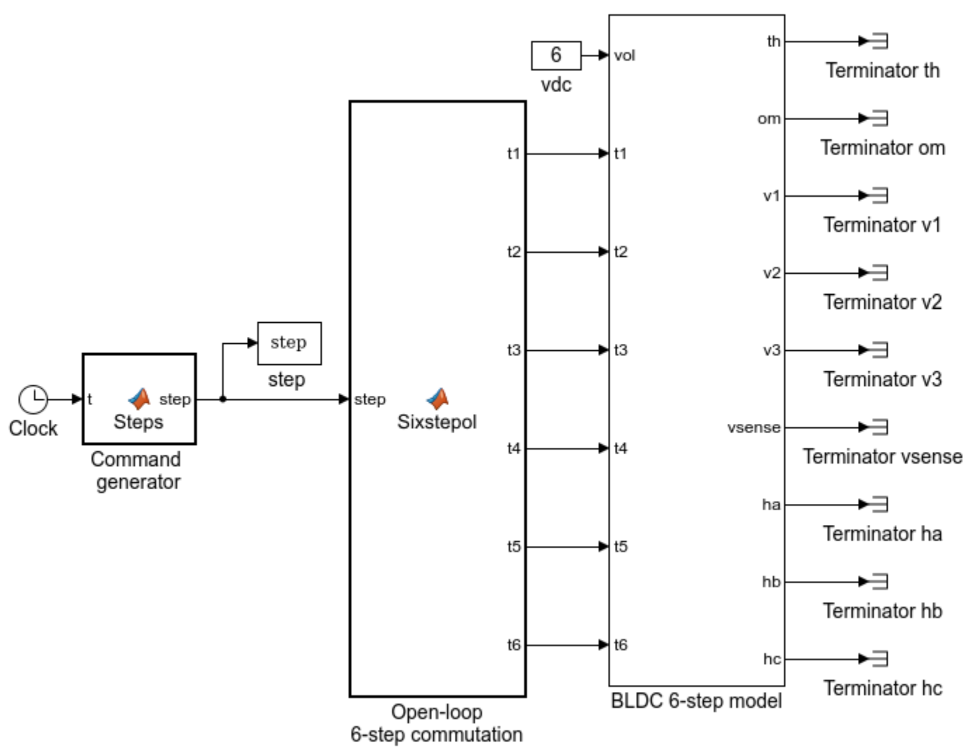

Figure 12: Diagram of the simulation of a brushless DC motor with six-step commutation

The inputs and outputs of the BLDC model are:

- *vol*, the DC supply voltage (in V).
- *t1* to *t6*, the six transistor commands (0 for off, 1 for on).
- *th*, the position of the motor (in rad).
- *om*, the velocity of the motor (in rad/s).
- *v1*, *v2*, *v3*, the line voltages applied to the motor windings (in V).
- *vsense*, the voltage on the sensing resistor (in V).
- *ha*, *hb*, *hc*, the Hall effect sensor outputs (0 for off, 1 for on).

The simulation also includes a block that implements the six-steps of the commutation logic and a command generator that moves from step to step twice across the table. Run the simulation and observe the response of the motor, which is reminiscent of the response of the stepper motor. Note that the step size is $60°/n_P$, where $n_P = 2$, but the first step is larger than the others because the motor is initialized at 0°

2.2 Quadrature control using six-step commutation

Modify the open-loop commutation logic to implement six-step commutation using the Hall effect sensors. Fig. 13 shows the diagram of a possible implementation.

The decoding of the Hall effect sensors can be performed using the table below.

h_A	h_B	h_C	dir $\geqslant 0$	dir < 0
1	1	0	T2, T6	T3, T5
0	1	0	T2, T4	T1, T5
0	1	1	T3, T4	T1, T6
0	0	1	T3, T5	T2, T6
1	0	1	T1, T5	T2, T4
1	0	0	T1, T6	T3, T4
0	0	0	-	-
1	1	1	-	-

The new commutation block has as inputs the three Hall effect sensors, a variable *dir* defining the sign of the torque (positive torque for $dir = 1$, negative torque otherwise), and a variable *on* defining whether to turn on the transistors. Transistors that are not on are off, and all transistors are off for $on = 0$ (discontinuous conduction mode). Let the DC supply voltage be 6 V. Blocks should also be added in *Simulink* to process the voltage from a 0.5 Ω sensing resistor as follows:

- scale the voltage measured on the resistor to read a value in A.
- take the absolute value of the current.
- filter the signal with a first-order filter having unity gain and a pole at $s = -5000$ rad/s.

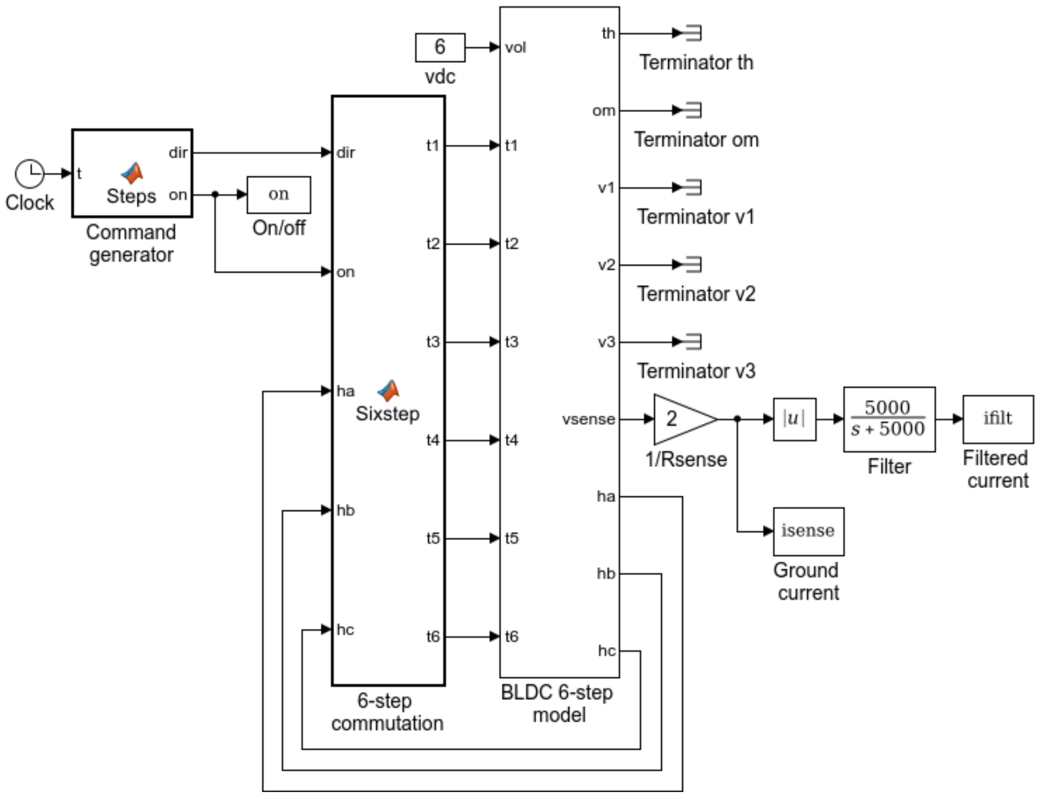

Figure 13: Implementation of six-step commutation

Generate commands so that

$$\begin{aligned}
dir &= 1 \text{ with } on = 1 \quad \text{(for 0.03 s)} \\
dir &= 1 \text{ with } on = 0 \quad \text{(for 0.03 s)} \\
dir &= -1 \text{ with } on = 1 \quad \text{(for 0.03 s)} \\
dir &= -1 \text{ with } on = 0 \quad \text{(for 0.02 s).} \qquad (46)
\end{aligned}$$

Run the simulation for 0.11 s. Plot the responses of the velocity and of the filtered current. Next, run another simulation with $dir = 1$ and $on = 1$ for the whole period, and plot the responses of the velocity and of the filtered current, as well as the variables *t1* to *t6*, *ha*, *hb*, *hc*, and *v1*, *v2*, *v3*. Keep the presentation compact without sacrificing clarity by plotting 2 to 3 variables in a row using the function *subplot* in Matlab. Also plot the complementary transistors (e.g., *t1* and *t4*) on the same graph. Comment on the responses, in particular the different segments of the voltages *v1*, *v2*, *v3* (at the source voltage, at the ground voltage, with an open winding, and the spike of voltage when the current is interrupted).

2.3 Current control

Having completed the six-step commutation block, augment the controller to achieve current regulation. Fig. 14 shows the block diagram of a possible implementation. Start by raising the supply voltage *vdc* to 12 V.

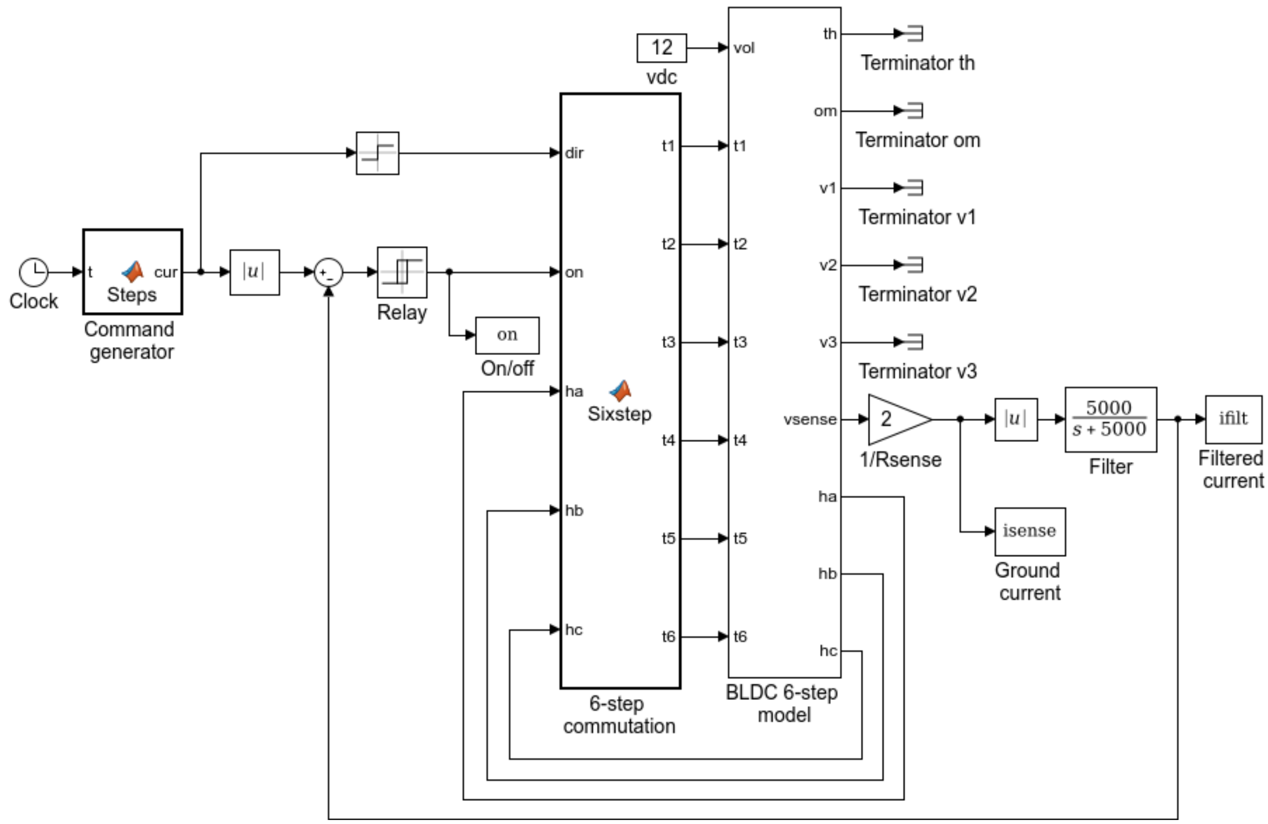

Figure 14: Implementation of current control

The reference current I_{REF} is labelled *cur* in the diagram The sign of the current determines the *dir* variable of the six-step commutation block (a positive reference current implies a positive torque command, and a negative reference current implies a negative torque command). The *signum* block from the *Simulink* library implements the sign function. The current loop then compares the absolute value of the current reference to the filtered measurement of the ground current I_{SENSE} (also taken in absolute value). The error drives the on/off switch of the commutation block through a hysteresis controller. The hysteresis block, called *Relay*, can be found in the *Simulink* library. Set its parameters so that the switch on point is 0.001, the switch off point is -0.001, the output when *on* is 1, and the output when *off* is 0.

Test the current controller with a reference

$$\begin{aligned} I_{REF} &= 0\ A,\ 0.5\ A,\ 1\ A,\ 0.5\ A,\ 0\ A,\ -0.5\ A,\ -1\ A,\ -0.5\ A \quad \text{(each for 0.01s)} \\ &= 0\ A \quad \text{(for 0.03s).} \end{aligned} \tag{47}$$

Simulate the responses for 0.11 s. Plot the velocity and the filtered current as functions of time. Comment on the responses.

Observe that the acceleration is approximately proportional to the current, so that a model

$$\frac{\omega(s)}{I_{REF}(s)} = \frac{k_0}{s} \tag{48}$$

can be assumed for control design. Give the value of k_0 (with units) estimated from the responses using

$$k_0 = \frac{\Delta\omega/\Delta t}{ifilt}, \tag{49}$$

and with values of $\Delta\omega/\Delta t$ and I_{REF} obtained between 0.02 s and 0.03 s in the simulation.

2.4 Velocity control

Based on the result of the previous section, a proportional control law with gain k_P for the velocity using I_{REF} as a control input yields a stable first-order system with pole at $s = -k_0 k_P$. Fig. 15 shows the diagram of a possible implementation.

Compute the gain k_P needed to obtain a response with a time constant equal to 3 ms using the estimated value of k_0. Implement the proportional velocity control loop in the simulation and test it with a reference velocity equal to 0 rad/s for 0.01 s, followed by 40 rad/s, 80 rad/s, 40 rad/s, 0 rad/s, -40 rad/s (each for 0.02 s), followed by 0 rad/s (for 0.03 s). Run the simulation for 0.14 s and plot the velocity and the filtered current as functions of time. Comment on the responses.

3 Report at a glance

Be sure to include:

- Six-step commutation
  - Plot of velocity and filtered current for varying commands.
  - Plot of velocity and filtered current for fixed commands.
  - Plot of transistor commands, Hall effect sensors, and line voltages.
- Current control
  - Plot of velocity and filtered current for the profile with varying current.
  - Computation of the constant k_0.

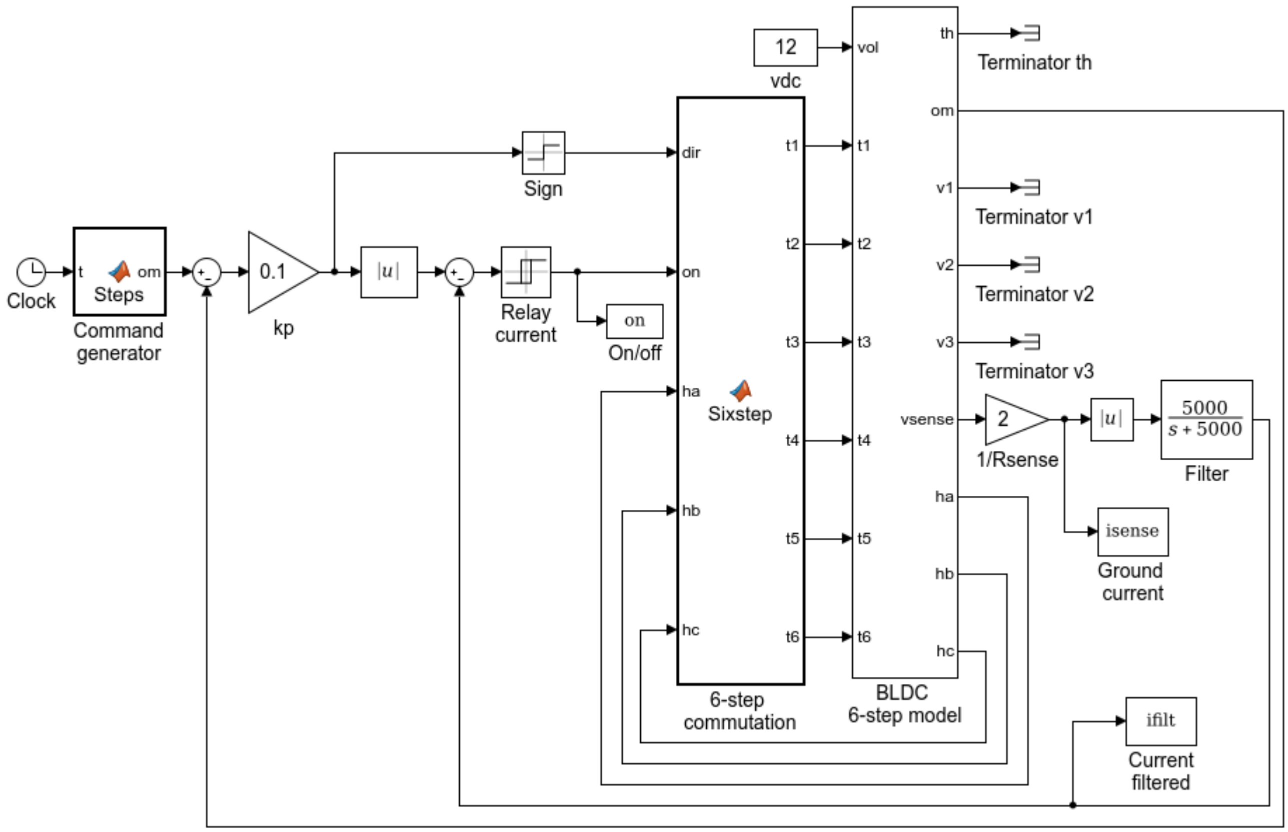

Figure 15: Implementation of velocity control

- Velocity control
  - Computation of the proportional gain.
  - Plot of velocity and filtered current for the profile with varying velocity.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 7 – Position control of a three-phase PM synchronous motor with sinusoidal commutation

Objectives

The objective of this lab is to implement a position control law for a brushless DC motor. A three-phase DQ transformation is used. An inner PI control loop regulates the currents, and an outer PID control loop regulates the position. Modifications are applied to the control laws to ensure good performance while enforcing limits on the voltages, currents, and velocity.

1 Introduction

The model of a $Y-$connected three-phase PM synchronous motor (including sine wave BLDC motors) was found to be given by

$$\begin{aligned}
L\frac{di_A}{dt} &= v_A - Ri_A + K\omega\sin(n_P\theta) \\
L\frac{di_B}{dt} &= v_B - Ri_B + K\omega\sin(n_P\theta - 120^\circ) \\
L\frac{di_C}{dt} &= v_C - Ri_C + K\omega\sin(n_P\theta - 240^\circ) \\
J\frac{d\omega}{dt} &= -K\left(i_A\sin(n_P\theta) + i_B\sin(n_P\theta - 120^\circ) + i_C\sin(n_P\theta - 240^\circ)\right) - \tau_{LF},
\end{aligned} \tag{50}$$

where v_A, v_B, v_C (V) are the line-to-neutral voltages applied to the three phases of the motor, i_A, i_B, i_C (A) are the line currents, ω (rad/s) is the angular velocity of the motor, θ (rad) is the angular position, and τ_{LF} (N m) is the load and friction torque. *Note that, to be precise or to implement equations in Matlab,* 120° *and* 240° *should be replaced by* $2\pi/3$ *and* $4\pi/3$*, respectively.*

The parameters are:

- R (Ω), the resistance of a phase winding.
- L (H), the inductance of a phase winding.
- K (N m/A or V s), the torque constant as well as the back-emf constant.
- J (kg m^2), the rotor's moment of inertia.
- n_P, the number of pole pairs.

Neglecting homopolar variables, the *equal magnitude* 3-phase DQ transformation is given by

$$\begin{pmatrix} i_d \\ i_q \end{pmatrix} = \frac{2}{3} \begin{pmatrix} \cos(n_P\theta) & \cos(n_P\theta - 120^\circ) & \cos(n_P\theta - 240^\circ) \\ -\sin(n_P\theta) & -\sin(n_P\theta - 120^\circ) & -\sin(n_P\theta - 240^\circ) \end{pmatrix} \begin{pmatrix} i_A \\ i_B \\ i_C \end{pmatrix}. \tag{51}$$

The inverse transformation is

$$\begin{pmatrix} i_A \\ i_B \\ i_C \end{pmatrix} = \begin{pmatrix} \cos(n_P\theta) & -\sin(n_P\theta) \\ \cos(n_P\theta - 120^\circ) & -\sin(n_P\theta - 120^\circ) \\ \cos(n_P\theta - 240^\circ) & -\sin(n_P\theta - 240^\circ) \end{pmatrix} \begin{pmatrix} i_d \\ i_q \end{pmatrix}. \tag{52}$$

The following DQ model results

$$\begin{aligned} L\frac{di_d}{dt} &= v_d - Ri_d + n_P\omega L i_q \\ L\frac{di_q}{dt} &= v_q - Ri_q - n_P\omega L i_d - K\omega \\ J\frac{d\omega}{dt} &= \frac{3}{2}Ki_q - \tau_{LF} \\ \frac{d\theta}{dt} &= \omega. \end{aligned} \tag{53}$$

Note that the transfer function from i_q to θ is the generic model

$$P(s) = \frac{k_0}{s^2}, \ \text{with} \ \ k_0 = \frac{3}{2}\frac{K}{J}. \tag{54}$$

DQ control consists in implementing current control loops for the currents i_d and i_q, and a position/velocity control loop using the current reference $i_{q,REF}$ assuming that the current i_q tracks the reference. In quadrature control, the reference value for i_d is set to zero. The overall organization of the components of the controller is shown on Fig. 16.

2 Current controller

The control loops for i_d and i_q are given by

$$\begin{aligned} v_d &= k_{PC}(i_{d,REF} - i_d) + k_{IC}\int (i_{d,REF} - i_d)d\tau \\ v_q &= k_{PC}(i_{q,REF} - i_q) + k_{IC}\int (i_{qREF} - i_q)d\tau. \end{aligned} \tag{55}$$

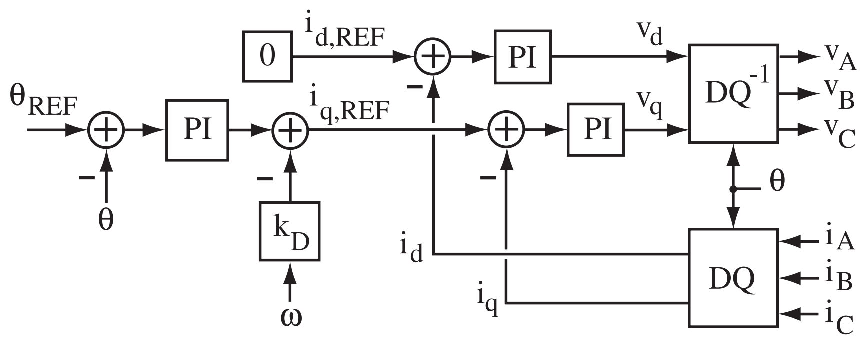

Figure 16: Organization of a position control algorithm

The terms proportional to ω in the electrical equations of (53) are neglected and the controller gains are chosen so that the controller zero cancels a plant pole. The closed-loop transfer functions become

$$\frac{i_d}{i_{d,REF}} = \frac{i_q}{i_{q,REF}} = \frac{a_{DC}}{s + a_{DC}} \tag{56}$$

if one chooses

$$k_{PC} = L\ a_{DC}, \ \ k_{IC} = R\ a_{DC}. \tag{57}$$

3 Position controller

A PID controller with feedforward gain using quadrature control consists in setting

$$\begin{aligned} i_{d,REF} &= 0 \\ i_{q,REF} &= k_P(k_F\theta_{REF} - \theta) + k_I \int (\theta_{REF} - \theta) d\tau - k_D\omega. \end{aligned} \tag{58}$$

Neglecting friction and assuming that the currents track the references, (54) and (58) give the closed-loop transfer function

$$\frac{\theta}{\theta_{REF}} = \frac{k_0\,(k_F k_P s + k_I)}{s^3 + k_D k_0 s^2 + k_P k_0 s + k_I k_0}. \tag{59}$$

A possible design choice is to place the three poles at some desired value $s = -a_D$. This choice corresponds to

$$k_D = \frac{3a_D}{k_0}, \ \ k_P = \frac{3a_D^2}{k_0}, \ \ k_I = \frac{a_D^3}{k_0}. \tag{60}$$

The zero of the closed-loop transfer is located as

$$z_F = -\frac{k_I}{k_F k_P} = -\frac{a_D}{3k_F} \tag{61}$$

and should be adjusted to avoid overshoot in the responses.

4 Command limiting

Limiting of the commands is useful to satisfy hardware constraints. Specifically,

- *voltage limiting:* in steady-state,

$$V_{pk} = \sqrt{v_d^2 + v_q^2},$$

where V_{pk} is the peak line-neutral voltage If $V_{pk} > V_{MAX}$, v_d and v_q should be multiplied by V_{MAX}/V_{pk}.

- *current limiting:* in steady-state

$$I_{pk} = \sqrt{i_d^2 + i_q^2} = i_q,$$

where I_{pk} is the peak line current. The variable $i_{q,REF}$ should be capped to $\pm I_{MAX}$.

- *velocity limiting:* ω_{REF} should be capped to $\pm\omega_{MAX}$. To this effect, the PID position controller can be represented as the combination of a PI position controller and a proportional velocity controller. Specifically, the second equation of (58) is rewritten as

$$\begin{aligned}
\omega_{REF} &= \frac{1}{k_D}\left(k_P(k_F\theta_{REF} - \theta) + k_I \int (\theta_{REF} - \theta)d\tau\right) \\
i_{q,REF} &= k_D(\omega_{REF} - \omega).
\end{aligned} \tag{62}$$

This implementation makes it possible to limit the speed reference by capping ω_{REF} to $\pm\omega_{MAX}$.

Fig. 17 show the current control loop with the voltage limiting. Integration should be stopped in the current control loop (i.e., (55)) when voltage limiting occurs (as represented by AW3). Fig. 18 shows the voltage control loop with current and velocity limiting. Integration should be stopped in the position control loop (i.e., (62)) when velocity, current, or voltage limiting occurs (as represented by AW1, 2, 3).

5 Experiments

5.1 Open-loop response

The file *Lab7.mdl* contains a simulation of a brushless DC motor. Fig. 19 shows the diagram of the simulation. The *Command generator* applies three-phase voltages of increasing magnitude

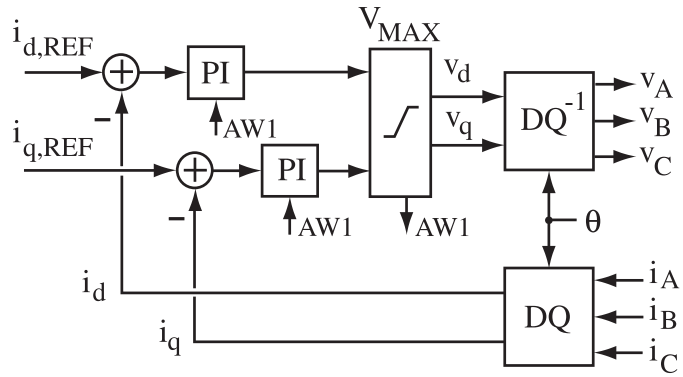

Figure 17: Current control loop with voltage limiting

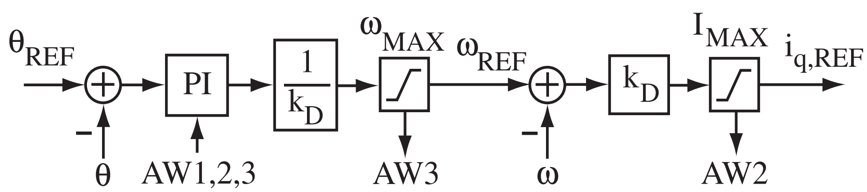

Figure 18: Position control loop with velocity and current limiting

and frequency. Stepwise changes are applied so that the motor progressively accelerates, but the response becomes increasingly oscillatory. Run the simulation and plot the speed (in rpm) as a function of time, together with the synchronous speed (the number of pole pairs is $n_P = 2$).

5.2 Design of the position controller

Find the parameters of the motor in the simulation model, and deduce:

- the PI gains k_{PC} and k_{IC} of the current control loop for $a_{DC} = 2\pi\ 1000$ rad/s.
- the PIDF gains k_P, k_I, k_D, k_F of the position control loop for $a_D = 2\pi\ 20$ rad/s and one closed-loop pole is cancelled by the zero given by (61).

Verify that values of your controller parameters rounded to the first significant digit are $k_{PC} = 10$, $k_{IC} = 6000$, $k_P = 10$, $k_I = 600$, $k_D = 0.1$, and $k_F = 0.3$.

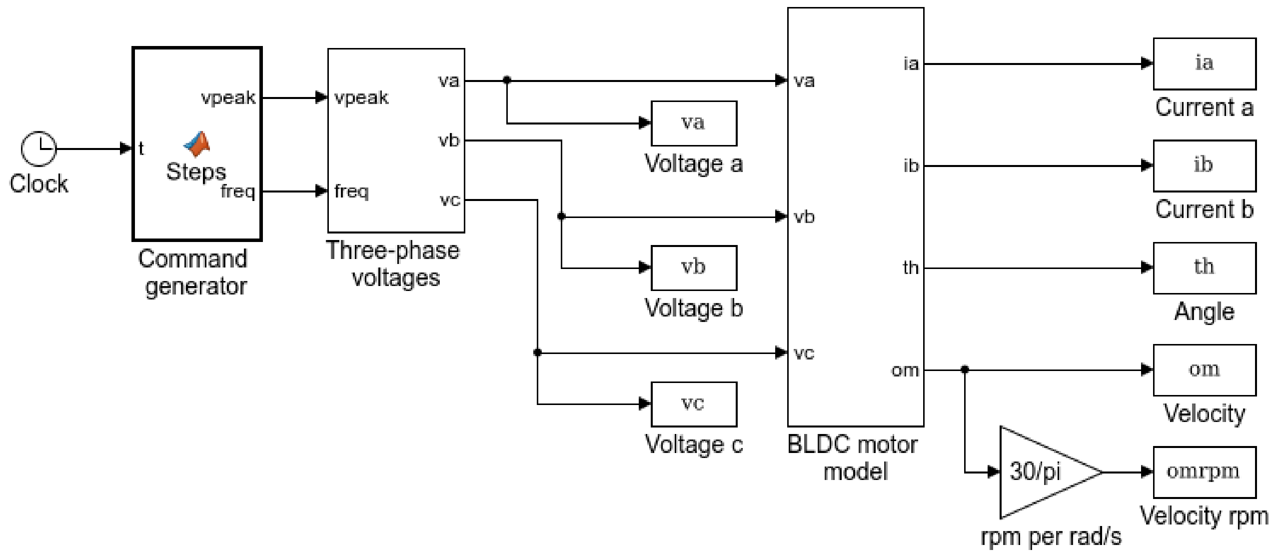

Figure 19: Diagram of the simulation of a BLDC motor

5.3 Testing of the position controller without limiting

A possible implementation of the simulation is shown on Fig. 20. Fig. 21 shows the contents of the *Position controller* block. The inner block named *DQ_control* is an m-file that contains all the equations of the controller. The variable *ie* is the integrator for the velocity controller, while the variables *ieid* and *ieiq* are the integrators for the current controllers.

The DQ control code has the following parts:

- equation (62) to compute $i_{q,REF}$ ($i_{d,REF}$ is set to zero for quadrature control).
- equation (51) to compute i_d and i_q.
- equation (55) to compute v_d and v_q.
- equation (52) with voltages replacing the currents to compute v_A, v_B, and v_C from v_d and v_q.

Section 5.2 should be used for the design of the controller, i.e., the choice of the parameters. The current i_C is obtained from $i_C = -i_A - i_B$.

A possible approach for code development and testing consists in implementing (52) with $v_d = 0$, and $v_q = 5$. The motor should accelerate smoothly. Then, (51) and (55) are added with $i_{q,REF} = 1$ A. Again, the motor should accelerate smoothly. Finally, (62) can be added to control the position. The position control system should be tested with a reference input

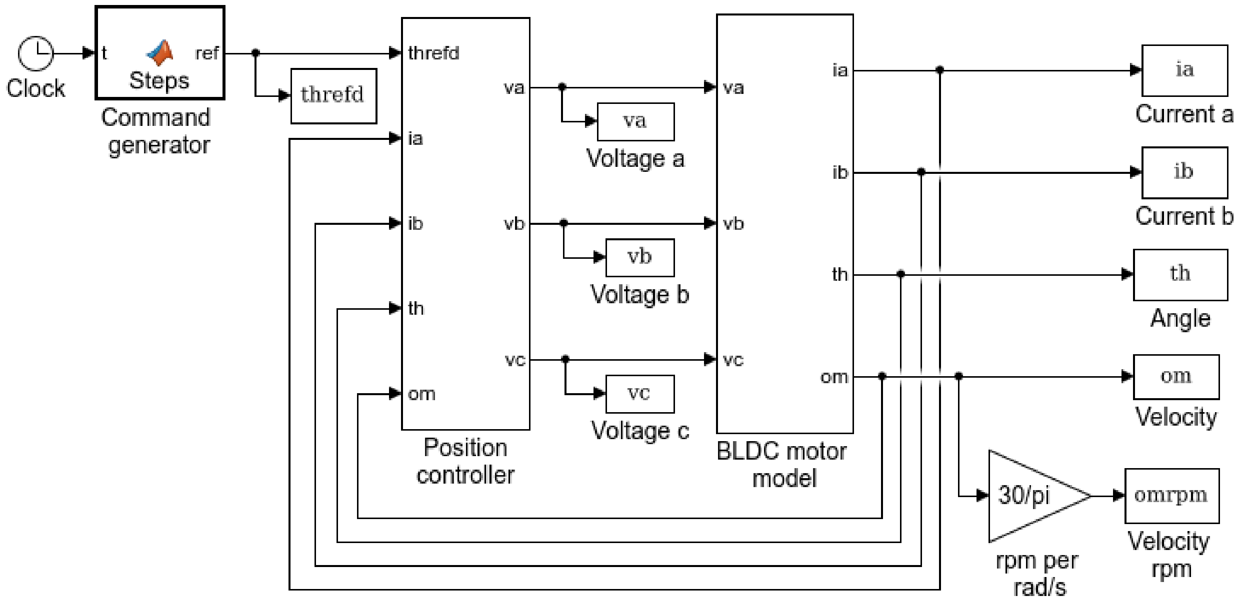

Figure 20: Diagram for the simulation of DQ control

having the following profile.

Time (s)	θ_{REF} (deg)
0-1	0
1-2	10
2-3	0
3-5	90
5-7	0
7-10	3600
10-12	0

Plot the position and reference in degrees as functions of time from 0 to 6 s and separately from 6 to 12 s. Over the full period from 0 to 12 s, plot the line current i_A, the line current i_B, the line-to-line voltage $v_A - v_B$, the line-to-line voltage $v_B - v_C$, and the speed (in rpm). The position should track the reference well without overshoot. However, the voltages, currents, and velocity reach large values for the motor under consideration.

Remember to convert degrees and rpm to rad and rad/s in the code when necessary. The arguments of the cos **and** sin **functions must be in rad.**

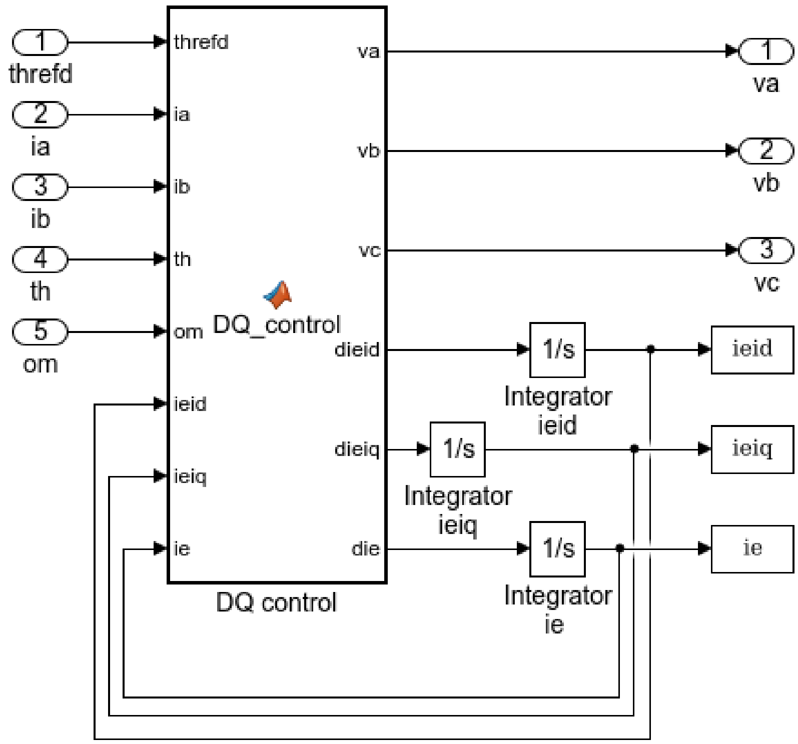

Figure 21: Contents of the position controller block

5.4 Testing of the position controller with limiting

In this section, the objective is to implement the techniques of Section 4 so that the responses stay within limits $V_{MAX} = 30$ V, $I_{MAX} = 7$ A, and $\omega_{MAX} = 4000$ rpm. It should be noted that, without modification to the controllers, these limits are exceeded by wide margins. Repeat the previous plots with limiting and anti-windup, including horizontal lines delineating the positive and negative limits of voltage, current, and speed on the plots. For the voltages, the line-to-neutral limit V_{MAX} should be converted to an equivalent line-to-line limit (multiply by $\sqrt{3}$).

6 Report at a glance

Be sure to include:

- Plot of the open-loop speed response with the synchronous speed.
- Values of the controller parameters.
- Plot of θ and θ_{REF} from 0 to 6 s.
- Plot of θ and θ_{REF} from 6 to 12 s.

- Plots of i_A and i_B with their limits.
- Plots of $v_A - v_B$ and $v_B - v_C$ with their limits.
- Plot of ω (in rpm) with its limits.

First show the plots without command limiting, then repeat the same plots with command limiting and observe the differences.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units for all of your values.

Lab 8 – Slip control of induction motors

Objectives

The objective of the lab is to implement a slip control method for the tracking of speed references by a squirrel-cage induction motor. For simplicity, current command is assumed. The responses of the motor with fixed stator frequency are first observed. Then, a slip control method is implemented and evaluated. An adjustment is made to the algorithm that significantly improves the responses.

1 Introduction

The lab considers a two-phase squirrel-cage induction motor with current command. The stator currents are given by

$$i_{SA} = I_S \cos(\theta_S),\ i_{SB} = I_S \sin(\theta_S), \tag{63}$$

with

$$\frac{d\theta_S}{dt} = \omega_S. \tag{64}$$

The slip is

$$S = \omega_S - n_P \omega. \tag{65}$$

For constant ω and ω_S, and assuming small slip S, the torque in steady-state is equal to

$$\tau_M = \frac{n_P M^2 I_S^2}{R_R}\ S. \tag{66}$$

Even though (66) was derived assuming constant speed and slip, the expression is used to obtain a transfer function from slip S to speed ω. The result is

$$P(s) = \frac{k_0}{s}, \tag{67}$$

with

$$k_0 = \frac{n_P M^2 I_S^2}{J R_R}. \tag{68}$$

The model (67) is valid for slowly-varying signals. When the effect of the rotor dynamics is significant, the transfer function

$$P(s) = \frac{k_0}{s} \frac{1}{T_R s + 1} \tag{69}$$

provides a better approximation than (67).

2 Open-loop control

Open-loop control of speed consists in setting the stator frequency so that

$$\omega_S = n_P \omega_{REF}, \tag{70}$$

where ω_{REF} is a reference velocity. Using (67), the transfer function from the reference speed to the speed is

$$\frac{\omega(s)}{\omega_{REF}(s)} = \frac{k_0 n_P}{s + k_0 n_P}. \tag{71}$$

The system is a stable first-order system with unity gain and a pole at $s = -k_0 n_P$.

The approximation is good if the rotor time constant T_R is small (i.e., $1/T_R \gg k_0 n_P$). Otherwise, a better approximation uses (69) to obtain the second-order system

$$\frac{\omega(s)}{\omega_{REF}(s)} = \frac{k_0 n_P}{T_R s^2 + s + k_0 n_P}. \tag{72}$$

The system is still stable with unity gain, but responses may be oscillatory.

3 Slip control

3.1 Basic slip control

A slip control scheme is shown on Fig. 22. The equations include (63) and (64). S is replaced by S_{REF} in (65), so that

$$\theta_S = \int_0^t (S_{REF} + n_P \omega)\ d\tau. \tag{73}$$

The PI controller is given by

$$\begin{aligned} S_C &= k_P (k_F \omega_{REF} - \omega) + k_I e_I \\ \frac{de_I}{dt} &= \omega_{REF} - \omega. \end{aligned} \tag{74}$$

The anti-windup signal (AW) specifies that integration is stopped ($de_I/dt = 0$) when a limit is reached. The limits are specified through

$$\begin{aligned} &S_{REF} = S_C && \text{for } -1/T_R \leqslant |S_C| \leqslant 1/T_R \\ &S_{REF} = 1/T_R && \text{for } S_C > 1/T_R \\ &S_{REF} = -1/T_R && \text{for } S_C < -1/T_R. \end{aligned} \tag{75}$$

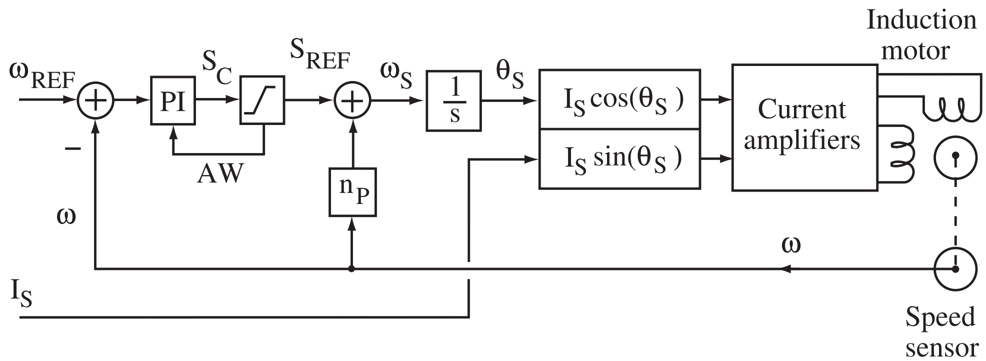

Figure 22: Slip control scheme

For the design of the PI controller, the simplified model with a slip input is used, i.e.,

$$\frac{\omega(s)}{S_{REF}(s)} = \frac{k_0}{s}. \tag{76}$$

The closed-loop transfer function is

$$\frac{\omega(s)}{\omega_{REF}(s)} = \frac{k_0\,(sk_F k_P + k_I)}{s^2 + sk_0 k_P + k_0 k_I}. \tag{77}$$

The two poles are located at $s = -a_D$ if

$$k_P = \frac{2a_D}{k_0}, \quad k_I = \frac{a_D^2}{k_0}. \tag{78}$$

The feedforward gain k_F is used to move the zero and avoid an overshoot in the responses.

3.2 Modified slip control

As for the open-loop response, the responses deviate from the prediction when the time constant T_R is large. The problem can be reduced by multiplying S_{REF} by $T_R s+1$, effectively cancelling the extra pole in the system. It turns out that the modification can be implemented simply by adding a compensation term to the angle θ_S in (73), so that

$$\theta_S = \int_0^t (S_{REF} + n_P\omega)\,d\tau + T_R\;S_{REF}. \tag{79}$$

4 Experiments

4.1 Open-loop control

The file *Lab8.mdl* simulates an induction motor under current command. Fig. 23 shows the diagram of the simulation. A block generates two-phase currents of specified magnitude and frequency. The motor is a 50 Hz motor with two poles pairs, so that the synchronous speed is 1500 rpm (or 50 π rad/s). The reference velocity is set to the synchronous value and a load torque equal to 5 N m is applied at 6 s.

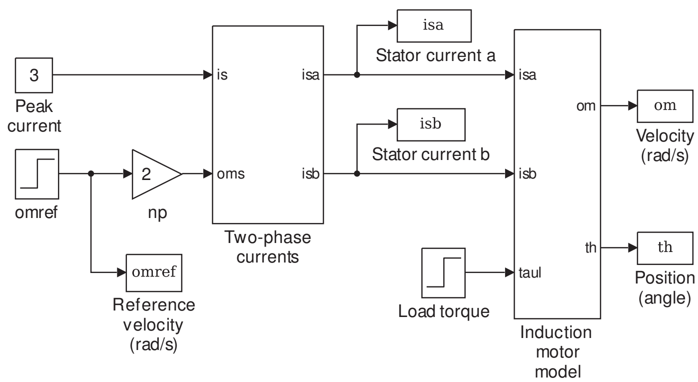

Figure 23: Diagram of the simulation of a squirrel-cage induction motor under current command

Run the simulation and plot the speed and speed reference (in rpm) as functions of time. Observe that the speed increases slowly initially, then faster when the motor gets close the synchronous speed. The response is oscillatory when reaching the synchronous speed. The speed drops slightly when the load torque is applied. Give the final value of the speed (in rpm) and of the normalized slip (in %).

4.2 Basic slip control

Implement the slip controller in the simulation model and increase the peak current to 8.6 A. Fig. 24 shows the *Simulink* diagram of a possible implementation. Fig. 25 shows the contents of the slip control block, which includes a PI controller with limiting and anti-windup.

Code the equations of the PI controller using $a_D = 1$ rad/s, adding a feedforward gain $k_F = 2/3$. The induction motor model is the two-phase equivalent of a three-phase motor, so that that k_0 in (68) should multiplied by 3/2. Let the speed reference be a step command going from zero to 1500 rpm at $t = 2$ s, and the load/friction torque going from zero to 5 N m at $t = 6$ seconds.

Simulate the response of the system for 10 seconds. Plot the speed with the speed reference (in rpm) as functions of time. Observe that the speed response is slow and that the effect of the load torque is very large. Plot the stator currents between 2 and 3 s to see how the frequency of the two-phase voltages increases with the speed. Repeat the simulation for $a_D = 5$ rad/s, and observe that the response becomes very oscillatory.

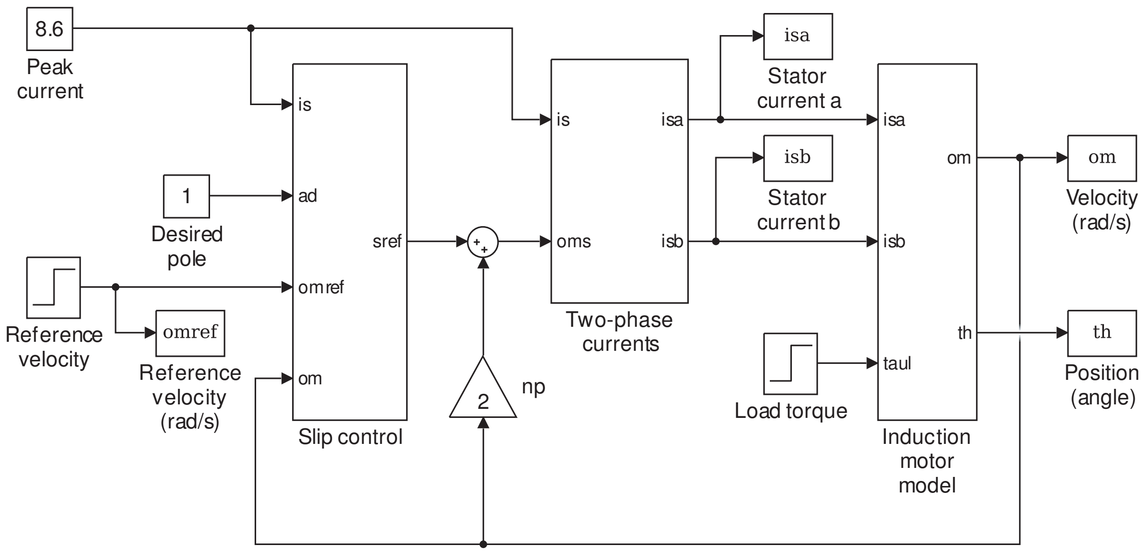

Figure 24: Diagram for the simulation of the slip control scheme

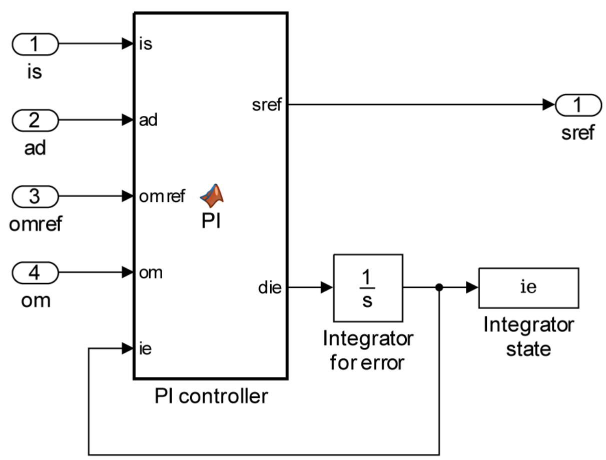

Figure 25: Diagram for the slip control block

5 Modified slip control

In the last simulation, the slip control scheme is modified using a phase compensation signal. Adjust the two-phase current block so that (73) is replaced by (79). Fig. 26 and Fig. 27 show the diagrams of a possible implementation. Run the simulation over 10 s and observe that the parameter a_D can be increased without producing oscillations. Plot the speed and the speed reference (in rpm) as functions of time for $a_D = 10$ rad/s. Observe that the transient response is significantly accelerated and that the effect of the load is much reduced using a remarkably simple modification.

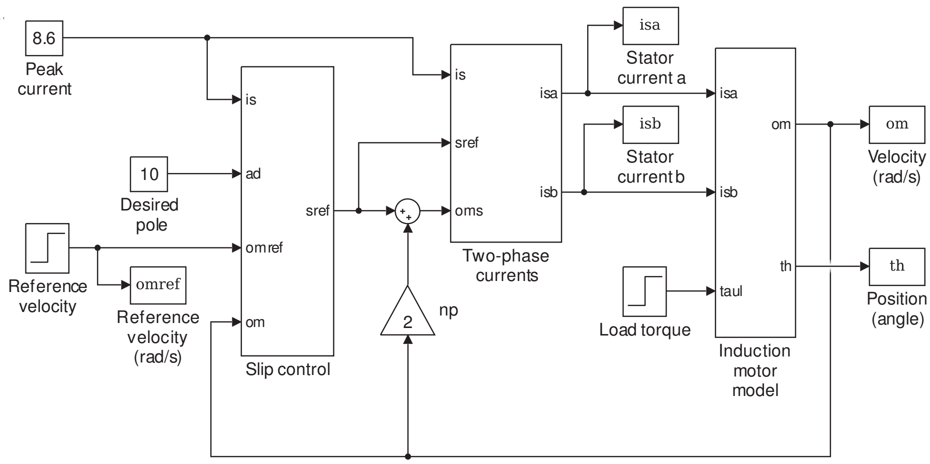

Figure 26: Modified slip control

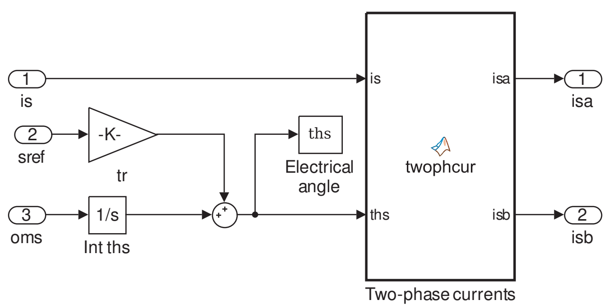

Figure 27: Modified slip control (two-phase current block)

6 Report at a glance

Be sure to include:

- Plot of the speed and speed reference for currents of fixed magnitude and frequency. Final values of speed and normalized slip.
- Plots of the speed and speed reference for the slip controller with $a_D = 1$ rad/s and with $a_D = 5$ rad/s.
- Plot of the speed and speed reference for the modified slip controller with $a_D = 10$ rad/s.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units on all of your values.

Lab 9 – Field-oriented control of induction motors

Objectives

The objective of the lab is to implement a field-oriented control method for the tracking of speed references by a squirrel-cage induction motor. For simplicity, current command is assumed. The current in the direct axis maintains a desired rotor flux magnitude, while the current in the quadrature axis controls the torque. Commands are limited to satisfy current limits.

1 Introduction

The model of a two-phase squirrel-cage induction motor with current command expressed in the AB coordinates and using rotor fluxes as state variables is given by

$$\begin{aligned}
\frac{d\psi_{RA}}{dt} &= -\frac{1}{T_R}\psi_{RA} + \frac{M}{T_R} i_{SA} - n_P \omega \psi_{RB} \\
\frac{d\psi_{RB}}{dt} &= -\frac{1}{T_R}\psi_{RB} + \frac{M}{T_R} i_{SB} + n_P \omega\ \psi_{RA} \\
J\frac{d\omega}{dt} &= \frac{n_P M}{L_R}\left(i_{SB}\ \psi_{RA} - i_{SA}\ \psi_{RB}\right) - \tau_{LF}.
\end{aligned} \tag{80}$$

In the model, i_{SA}, i_{SB} (A) are the stator currents, ψ_{RA}, ψ_{RB} (Wb) are the components of the rotor flux, ω (rad/s) is the angular velocity of the motor, and τ_{LF} (N m) is the load and friction torque. The parameters are:

- T_R (s), the rotor time constant.
- L_R (H), the self-inductance of a rotor winding.
- R_R (Ω), the resistance of a rotor winding.
- M (H), the mutual inductance between a stator winding and a rotor winding when aligned.
- J (kg m^2), the rotor's moment of inertia.
- n_P, the number of pole pairs.

A field-oriented model is obtained through a DQ transformation

$$\begin{pmatrix} i_{Sd} \\ i_{Sq} \end{pmatrix} = \begin{pmatrix} \cos(\rho) & \sin(\rho) \\ -\sin(\rho) & \cos(\rho) \end{pmatrix} \begin{pmatrix} i_{SA} \\ i_{SB} \end{pmatrix}, \tag{81}$$

with inverse

$$\begin{pmatrix} i_{SA} \\ i_{SB} \end{pmatrix} = \begin{pmatrix} \cos(\rho) & -\sin(\rho) \\ \sin(\rho) & \cos(\rho) \end{pmatrix} \begin{pmatrix} i_{Sd} \\ i_{Sq} \end{pmatrix}. \tag{82}$$

The angle ρ is obtained from the AB rotor fluxes through

$$\rho = \measuredangle\left(\psi_{RA} + j\psi_{RB}\right). \tag{83}$$

Then, the DQ rotor fluxes are

$$\psi_d = \sqrt{\psi_{RA}^2 + \psi_{RB}^2} \triangleq \psi, \ \psi_q = 0. \tag{84}$$

The equations of a two-phase motor with current command become

$$\begin{aligned} \frac{d\psi}{dt} &= -\frac{1}{T_R}\psi + \frac{M}{T_R} i_{Sd} \\ \frac{d\rho}{dt} &= n_P\omega + \frac{M}{T_R}\frac{i_{Sq}}{\psi} \\ \frac{d\omega}{dt} &= \frac{n_P M}{J L_R}\psi i_{Sq} - \frac{\tau_{LF}}{J}. \end{aligned} \tag{85}$$

With i_{Sd} constant, the flux in steady-state becomes

$$\psi = M i_{Sd}. \tag{86}$$

Then, the motoring torque becomes proportional to i_{Sq}, which can be used to control the motor. Because ψ appears on the denominator for $d\rho/dt$, the use of the model requires that ψ be nonzero. A current $i_{Sd} > 0$ must be applied starting at $t = 0$ so that $\psi > 0$ for $t > 0$.

2 Field-oriented control

The diagram of the field-oriented control scheme is shown in Fig. 28. The principle of operation is that the transfer function from i_{Sq} to ω is

$$\frac{\omega(s)}{i_{Sq}(s)} = \frac{k_0}{s}, \text{ with } k_0 = \frac{n_P M}{J L_R}\psi. \tag{87}$$

A PI controller can be used as in Lab 8, but with a different value of k_0.

In the PI controller, limiting should be included so that $|i_{SA}| \leqslant I_{MAX}$ and $|i_{SB}| \leqslant I_{MAX}$. The result is obtained if

$$i_{Sd}^2 + i_{Sq}^2 \leqslant I_{MAX}^2. \tag{88}$$

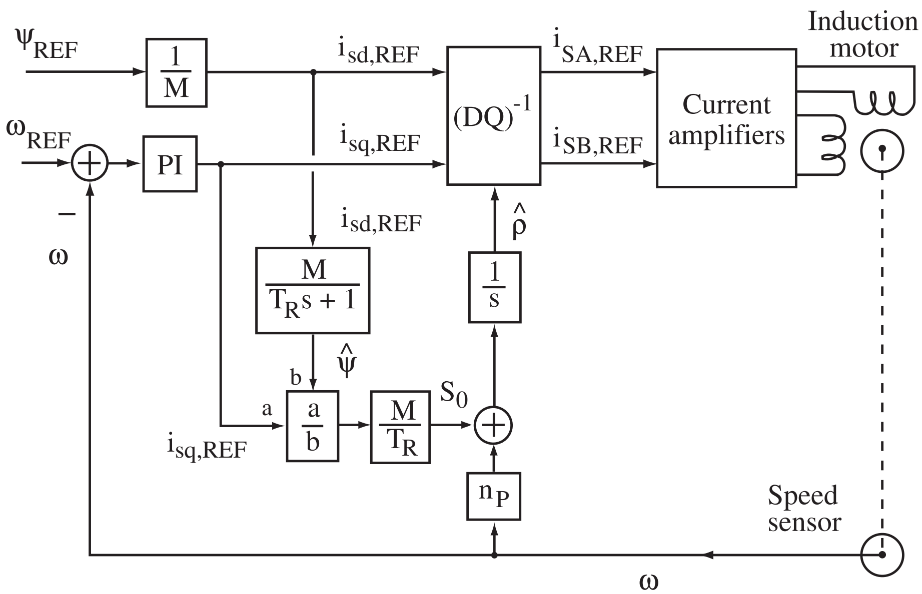

Figure 28: Field-oriented control scheme

Given that i_{Sd} is independently specified, the following limit is applied

$$|i_{Sq}| \leqslant \sqrt{I_{MAX}^2 - i_{Sd}^2}. \tag{89}$$

Integration is stopped in the PI controller when the limit is reached.

The rotor flux variables are estimated using an open-loop observer that consists in repeating the differential equations, i.e.,

$$\begin{aligned} \frac{d\hat{\psi}}{dt} &= -\frac{1}{T_R}\hat{\psi} + \frac{M}{T_R} i_{Sd} \\ \frac{d\hat{\rho}}{dt} &= n_P\omega + \frac{M}{T_R}\frac{i_{Sq}}{\hat{\psi}}. \end{aligned} \tag{90}$$

The second equation is only used when $\hat{\psi} \geqslant \psi_{MIN}$, where $\psi_{MIN} > 0$ is arbitrary. When $\hat{\psi} < \psi_{MIN}$, integration is stopped in the PI controller, i_{Sq} is set to zero, and the differential equation for $\hat{\rho}$ is replaced by

$$\frac{d\hat{\rho}}{dt} = n_P\omega. \tag{91}$$

3 Experiments

Implement the field-oriented control algorithm starting from the file *Lab8.mdl*. Fig. 29 shows the *Simulink* diagram of a possible implementation. Fig. 30 shows the contents of the field-oriented control block. Note that the flux reference is specified through a reference on the current i_{Sd}.

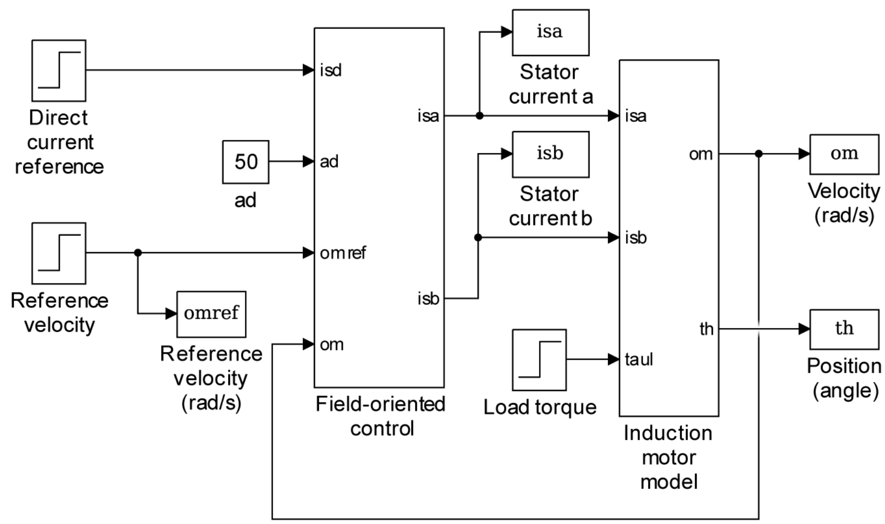

Figure 29: Diagram for the simulation of the field-oriented control scheme

Using the parameters from the simulation model, compute k_0 as a function of ψ using (87), but insert a factor 3/2 because the simulation implements the two-phase equivalent of a three-phase motor. For the controller, let $a_D = 50$ rad/s, $k_F = 2/3$, $I_{MAX} = 8.6$ A, and $\psi_{MIN} = 0.1$ Wb. For the input signals, let $i_{Sd} = 2.5$ A at first, then increase the current to 3 A at 3 s. Let the speed reference be zero at first, then increase to 1500 rpm at $t = 1$ s. Let the load torque be zero at first, then increase to 15.1 N m at $t = 2$ s.

Simulate the response over 4 s. Plot the speed with the speed reference in rpm as functions of time. Also plot the rotor flux magnitude and the peak stator current using

$$\psi = \sqrt{\psi_{RA}^2 + \psi_{RB}^2}, \; I_S = \sqrt{i_{SA}^2 + i_{SB}^2}. \tag{92}$$

The speed response should be consistent with the choice of poles and zero, and will show a small dip when the load torque is applied. The stator current should reach the limit during the acceleration, return to the value of i_{Sd} when the speed reference is reached, and increase again when the load torque is applied. The step in i_{Sd} should not impact the speed response and, even though i_{Sd} increases, the overall stator current decreases because a smaller current i_{Sq} is needed to achieve the desired torque with a larger flux ψ.

Given that many non-ideal effects are neglected, the results may not be completely realistic. Still, the simulation illustrates the elegant and powerful solution provided by field-oriented control.

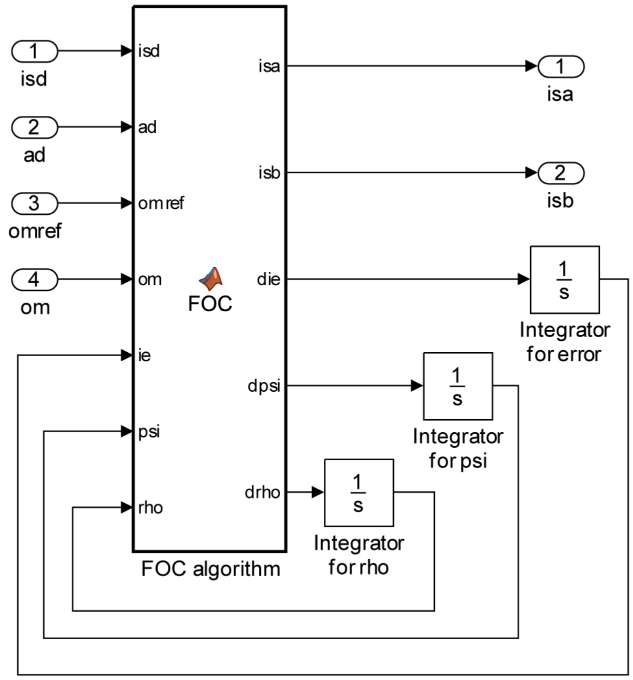

Figure 30: Diagram for the velocity control block using field-oriented control

4 Report at a glance

Be sure to include:

- Plot of speed and speed reference.
- Plot of rotor flux magnitude.
- Plot of peak stator current.

In addition, the report should include comments and explanations for the work performed and the results observed/produced. Be sure to label the axes of all your plots, and to use line widths and font sizes that provide good readability. Also include units on all of your values.