Control of Electric Drives and Power Converters

Assignment 1.

Antti Jormanainen 101565839

Mikko Sarén 69309U

Tasks 1 & 2.

Determining the number of pole pairs:

```
n = 1436; % Nominal motor speed [rpm]
f = 50; % Frequency [Hz]
n_p = floor(60*f/n);
```

From the calculation we can see that

$$n_p = \frac{60 * 50\,\mathrm{Hz}}{1436\,\frac{r}{\mathrm{min}}} = 2$$

For the first task of the assignment, we were asked to derive the equations for the stator current i_s^s and the rotor current i_R^s as a function of stator flux ψ_s^s and rotor flux ψ_R^s in stator coordinates. This was achieved by inserting gain and summation blocks in the "Flux equations" function bloc, as seen in the Fig. 1

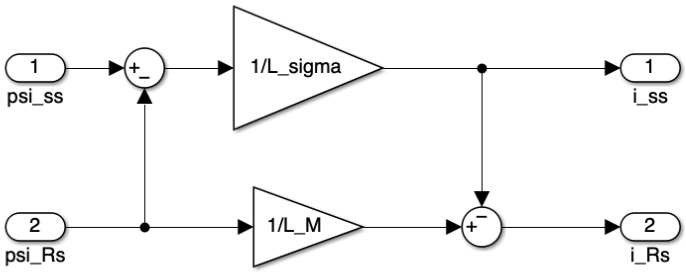


Figure 1. Flux equations fuction block

After this, the equation for electromagnetic torque was formed inside the "Electromagnetic torque" function block, as seen in the Fig. 2.

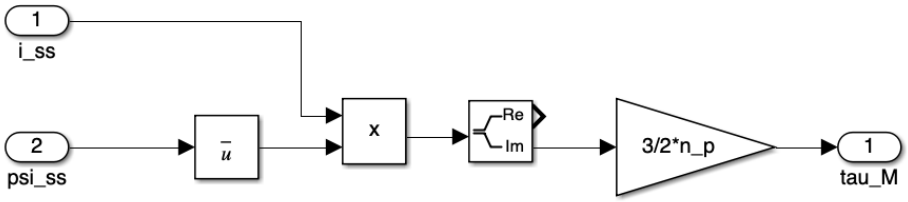


Figure 2. Electromagnetic torque control block

The nominal torque $\tau_N = \frac{P_N}{\omega_N} = \frac{P_N}{n_N \frac{2\pi}{60s}} = \frac{2200W}{1436 \frac{1}{\min} \frac{2\pi}{60s}} \approx 14.63\text{Nm}$

The nominal peak to peak line to neutral voltage $v_{p-p,\text{pn}} = \sqrt{2}\ \frac{400V}{\sqrt{3}} \approx 326.6V$

Task 3.

The missing part of the algorithm was completed by inserting the equation for u_s^s, which is

$u_s^s(k) = \psi_{s,\text{ref}} * \omega_{s,\text{ref}}(k) * e^{j\theta_s(k)}$

Running the intialization script:

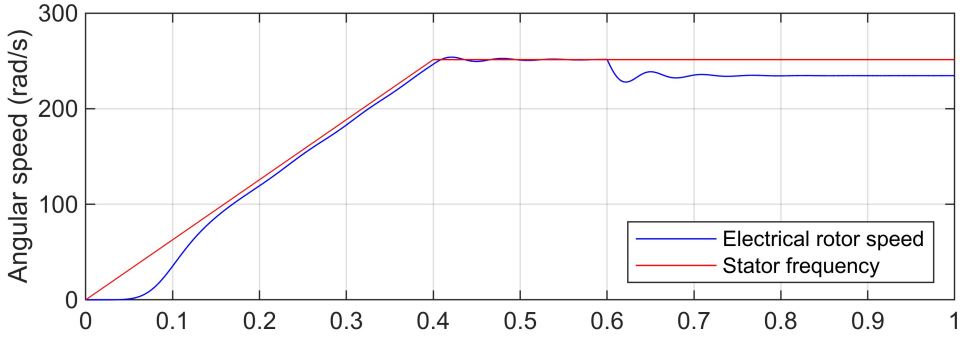
```
init_vhz;
```

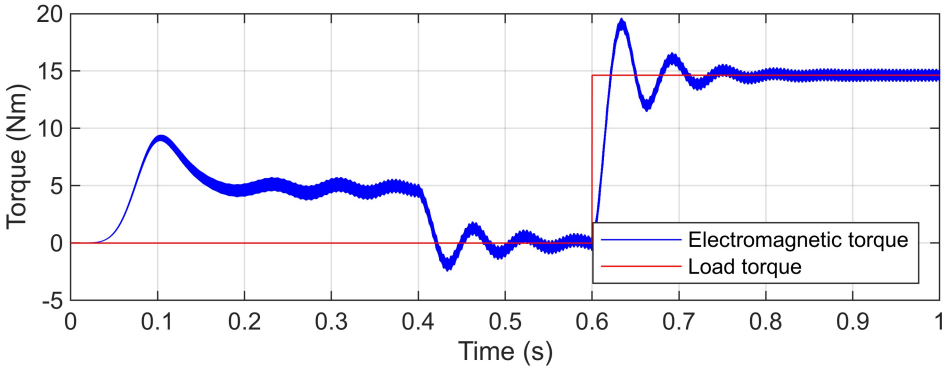
Simulating the system:

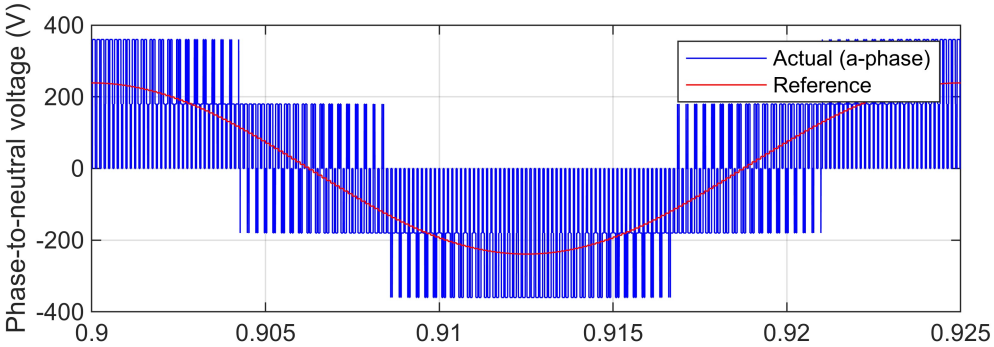
```
sim("vhz.slx");
```

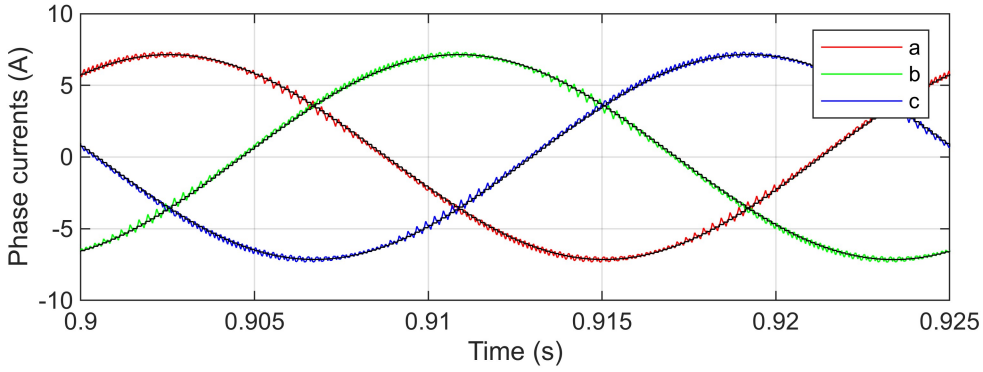
Plotting the result:

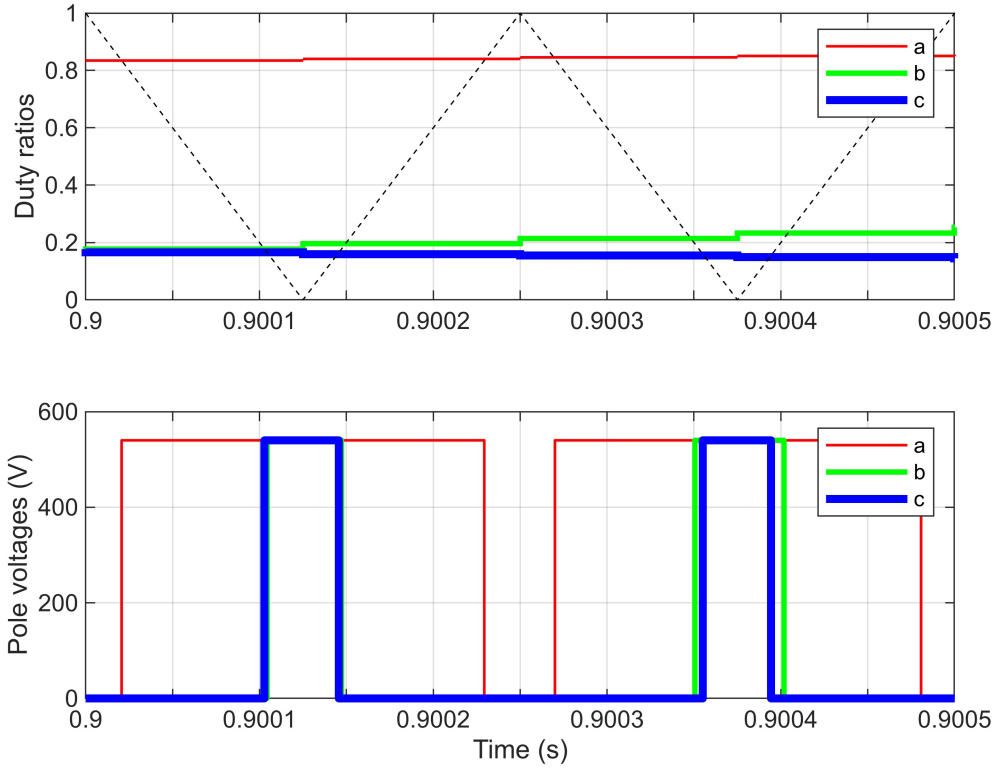
```
fig_vhz;
```


Angular speed (rad/s)
300
200
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Electrical rotor speed
Stator frequency


Torque (Nm)
20
15
10
5
0
-5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (s)
Electromagnetic torque
Load torque


Phase-to-neutral voltage (V)
400
200
0
-200
-400
0.9
0.905
0.91
0.915
0.92
0.925
Actual (a-phase)
Reference


Phase currents (A)
10
5
0
-5
-10
0.9
0.905
0.91
0.915
0.92
0.925
Time (s)
a
b
c



The figures seem to match pretty well with the one's given in the assignment for comparison.

Task 4.

The stator voltage for these calculations was approximated by using the equation

$$u_s = \omega_{s,\mathrm{ref}}\psi_{s,\mathrm{ref}} = 2\pi 40\ \frac{\mathrm{rad}}{s} * 0.95\ \mathrm{Vs} = 238.76\ V$$

The stator impedance in steady state is

$$Z_s = R_s + j\omega(L_\sigma + L_M) = 3.7\Omega + \mathrm{j}80\pi\ \frac{\mathrm{rad}}{s}\ (21\ \mathrm{mH} + 224\ \mathrm{mH}) = (3.7 + \mathrm{j}61.57)\Omega$$

And hence, the peak stator current is

$$i_s = \frac{u_s}{|Z_s|} = 3.87\ A$$

Plotting the calculated stator current value against the simulated one can be seen in Fig. 3. For the simulation, the load torque step final value was set to 0, but the rate limiter of the speed reference was kept untouched, which explains the peaks in stator current before reaching steady state.

```
T_L = 0; % load torque
sim("vhz.slx");
task_4_script;
```

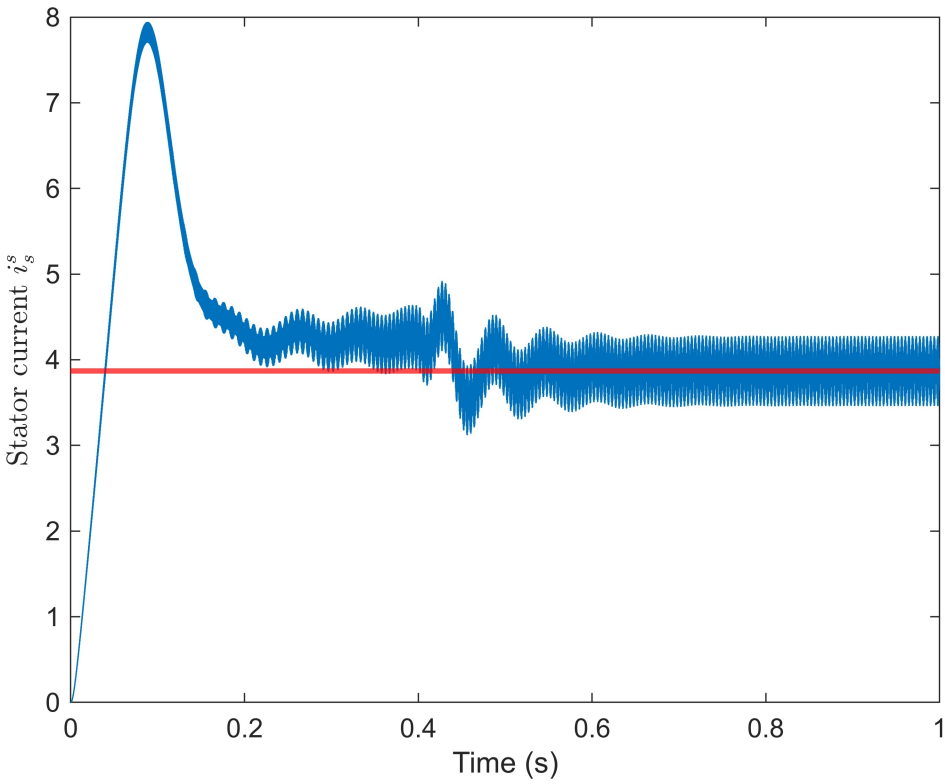


Figure 3. Simulated and calculated stator current

As can be seen from the figure the calculated stator current value matches the simulated one quite well in steady state operating area. Losses due to the stator resistance were omitted in the controller u_ref calculation, and therefore the voltage drop due to the resistance is omitted here as well.

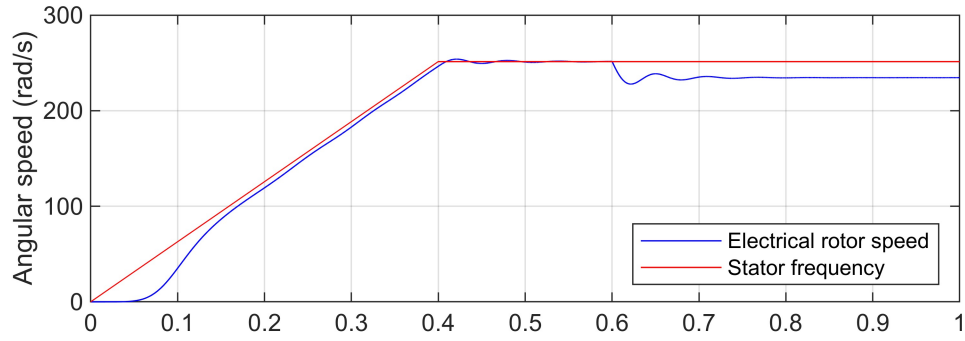
Task 5.

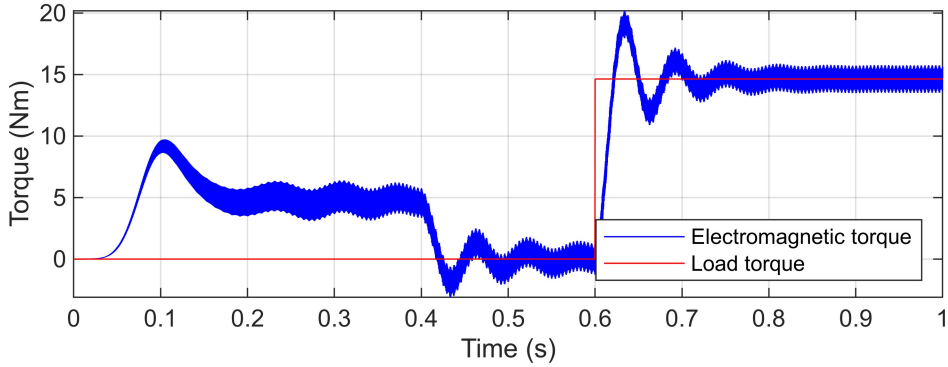
To decrease f_{sw} to 2kHz we calculated sampling period T_S to correspond to the lower switching frequency.

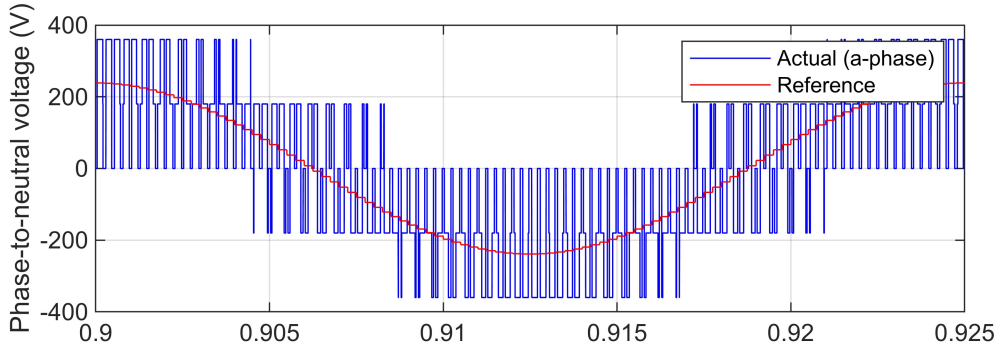
```
init_vhz;
f_sw = 2e3; % switching frequency
T_s = 1/(2*f_sw); % sampling period
T_L = 14.63; % returning the load torque back to its nominal value
```

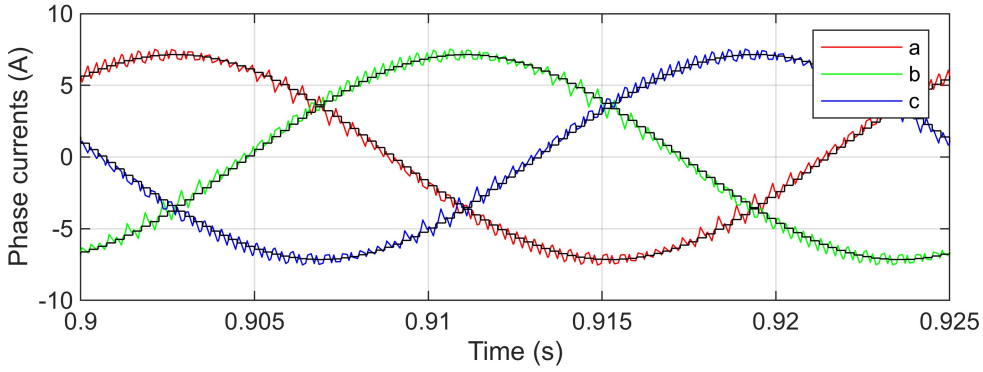
Simulating the model again with new parameters and running the plotting script.

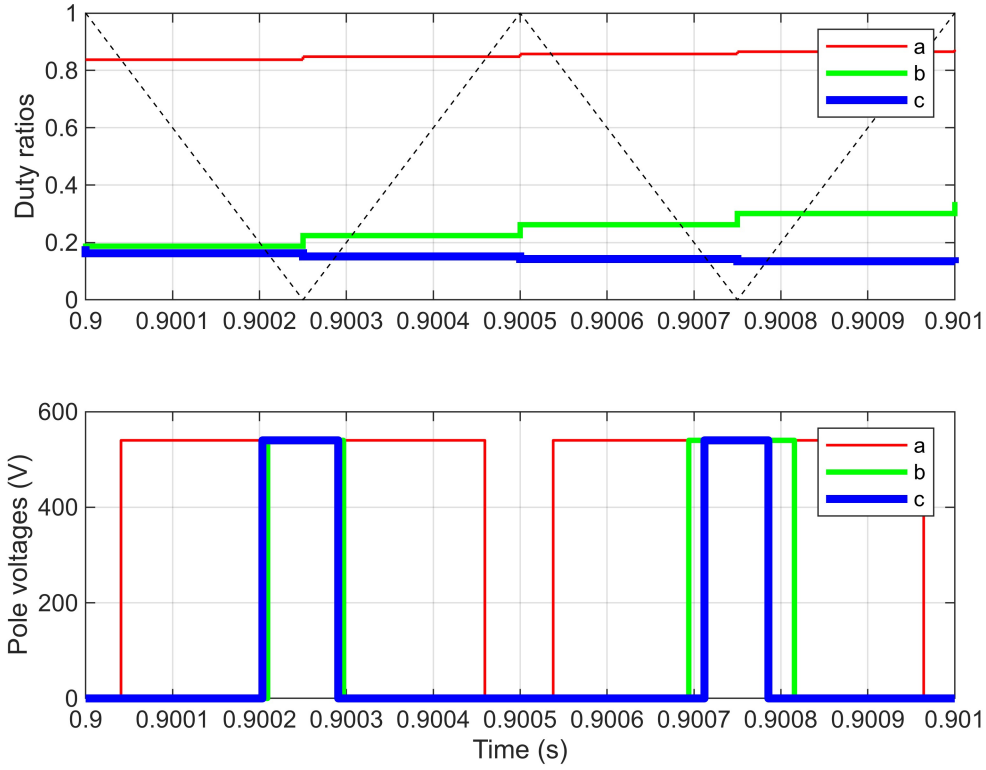
```
sim("vhz.slx");
fig_vhz;
```


Angular speed (rad/s)
300
200
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Electrical rotor speed
Stator frequency


Torque (Nm)
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (s)
Electromagnetic torque
Load torque


Phase-to-neutral voltage (V)
400
200
0
-200
-400
0.9
0.905
0.91
0.915
0.92
0.925
Actual (a-phase)
Reference


Phase currents (A)
10
5
0
-5
-10
0.9
0.905
0.91
0.915
0.92
0.925
Time (s)
a
b
c



Decreasing the switching frequency seems to increase the harmonic content in the phase currents, which leads to the harmonic content increasing in the electromagnetic torque as well, ie. torque ripple. The response of the system (overshoots, response time) remains the same, because the current remains the same effectively. While decreasing the switching frequency, the efficiency of the converter increases due to reduced switching losses, the torque ripple is not desired behaviour in some applications.

Task 6.

The transformation from abc to dq was done by implementing the transformation equations

$i_s^s = \frac{2}{3}\left(i_a + i_b e^{j2\frac{\pi}{3}} + i_c e^{j4\frac{\pi}{3}}\right)$, which is the transformation from abc to $\alpha\beta$-coordinate space vector, and

$i_s = i_s^s e^{-j\hat{\vartheta}_s}$, which is the transformation from $\alpha\beta$-coordinates to dq-coordinates

inside the abc2dq function block in vector control subsystem. The implementation was done in following manner: (code is commented, because the report is done in MATLAB Live Editor, and the code blocks are run when forming the document).

```
% Space-vector transformation and coordinate transformation
%i_ss = 2/3*(i_a+i_b*exp(1i*2*pi/3)+i_c*exp(1i*4*pi/3)); % abc to alfa-beeta
%i_s = i_ss*exp(-1i*theta_s);     % alfa-beeta to dq
```

Task 7.

The calculations for $i_{d,\text{ref}}$ and $i_{q,\text{ref}}$ were added by using the equations

$i_{d,\text{ref}} = \frac{\psi_{R,\text{ref}}}{\hat{L}_M}$ and

$i_{q,\text{ref}} = \frac{2\tau_{M,\text{ref}}}{3n_p\psi_{R,\text{ref}}}$

Algorithm was implemented as follows (again commented here to avoid problems):

```
% Current reference in estimated rotor-flux coordinates
% i_d_ref = psi_R_ref/L_M;
% i_q_ref = 2*tau_M_ref/(3*n_p*psi_R_ref);
```

Task 8.

The angular frequency of the rotor flux is estimated using the equation

$\hat{\omega}_s = \omega_m + \frac{\hat{R}_R i_q}{\hat{\psi}_R}$

The magnitude of the rotor flux is estimated using the equation

$\frac{d\psi_R}{\text{dt}} = \hat{R}_R\left(i_d - \frac{\hat{\psi}_R}{\hat{L}_M}\right)$, for which the corresponding forward Euler approximation is

$\psi_R(k+1) = \psi_R(k) + T_s\left(-\frac{R_R}{L_M}\psi_R(k) + R_R i_d\right)$

Algorithm was implemented as follows

```
%if(psi_R > 0)                          % Avoid division by zero
%    w_s = w_m+R_R*i_q/psi_R;
%else
%    w_s = w_m;
%end

% Update the states
%psi_R_new = psi_R+T_s*((-R_R/L_M)*psi_R+R_R*i_d);
```

Task 9.

Running the initialization script:

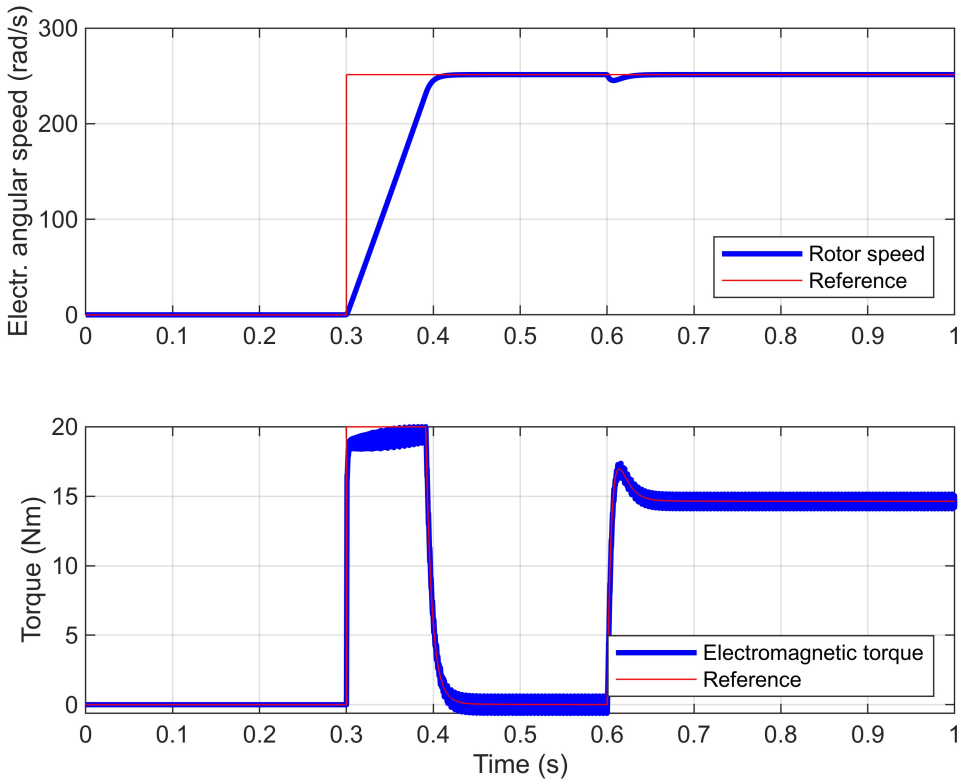
```
init_vector;
```

Simulating the model:

```
sim("vector_model.slx");
```

Plotting the result:

```
speed_torque;
```



The results seem to match those presented in the Fig. 11 of the assignment instructions figure, so the model seems to work fine.

Task 10.

To calculate i_d, i_q and $|\mathbf{i}_s|$ we use following equations.

$$i_d = \frac{\psi_R}{L_M}$$

$$i_q = \frac{2\tau_M}{3n_p\psi_R}$$

$$\boldsymbol{i}_s = i_d + \mathrm{j}i_q$$

where τ_M=14.63Nm, $n_p = 2$, $\psi_R = 0.9Vs$ and $L_M = 224mH$

For u_d, u_q and $|\mathbf{u}_s|$ we use equations:

$$\omega_m = n_p \omega_M$$

$$T_r = \frac{L_M}{R_R}$$

$$\tau_M = \frac{3n_p}{2} \mathrm{Im}\left\{ \frac{j\omega_r \psi_{\boldsymbol{R}}}{R_R} \psi_{\boldsymbol{R}}^* \right\} = \frac{3n_p}{2} \frac{\psi_R^2}{R_R} \omega_r$$

$$\omega_r = \frac{2\tau_M}{3n_p} \frac{R_R}{\psi^2{}_R}$$

$$\boldsymbol{i_s} = \frac{\psi_s}{L_\sigma + \dfrac{L_M}{1 + j\omega_r T_r}}$$

$$\psi_s = \boldsymbol{i_s} \left(L_\sigma + \frac{L_M}{1 + j\omega_r T_r} \right)$$

$$\omega_s = \omega_r + \omega_m$$

$$\boldsymbol{u_s} = R_s \boldsymbol{i_s} + j\omega_s \psi_s$$

$$u_d = \mathrm{Re}\{\boldsymbol{u_s}\}$$

$$u_q = \mathrm{Im}\{\boldsymbol{u_s}\}$$

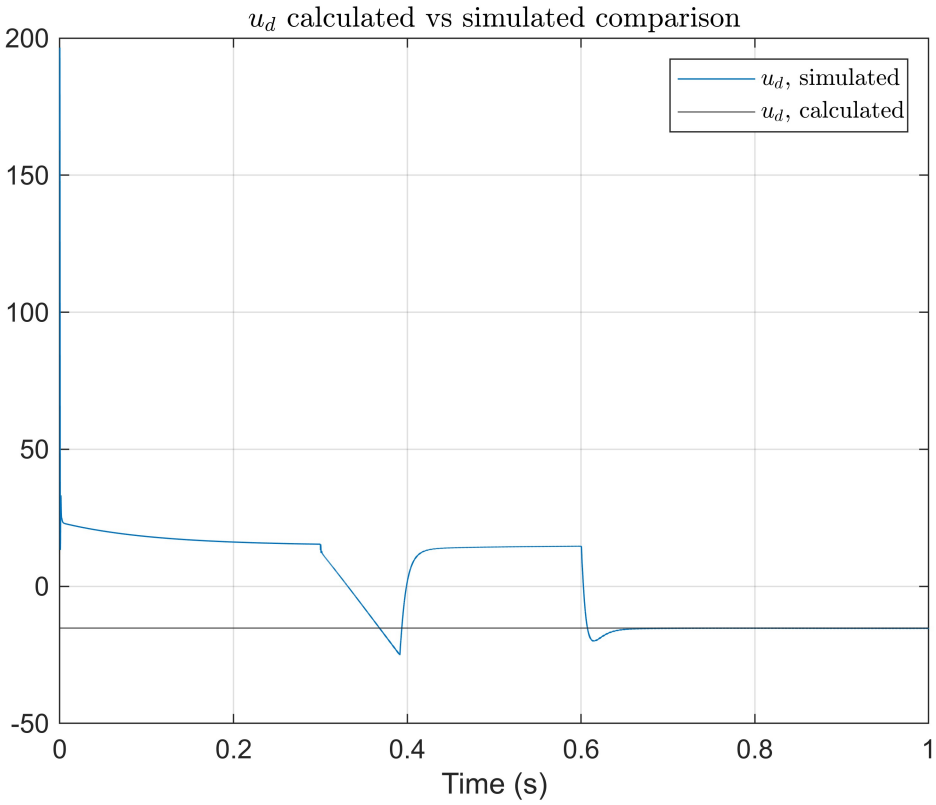
Calculating the values using the above equations with a script:

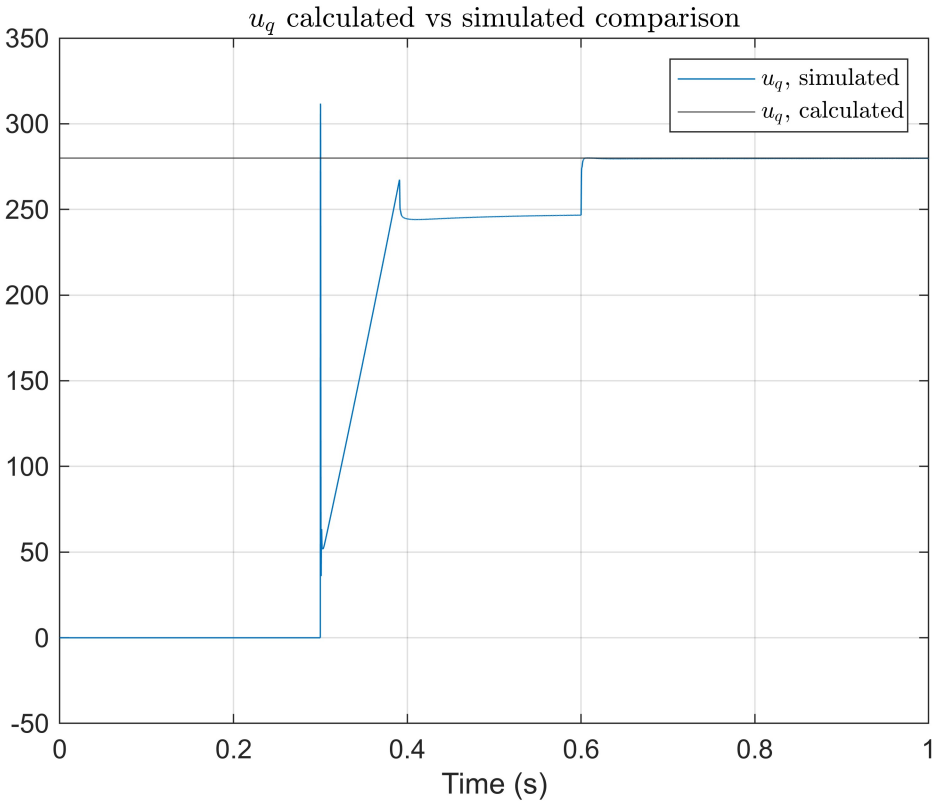
```
task10;
```

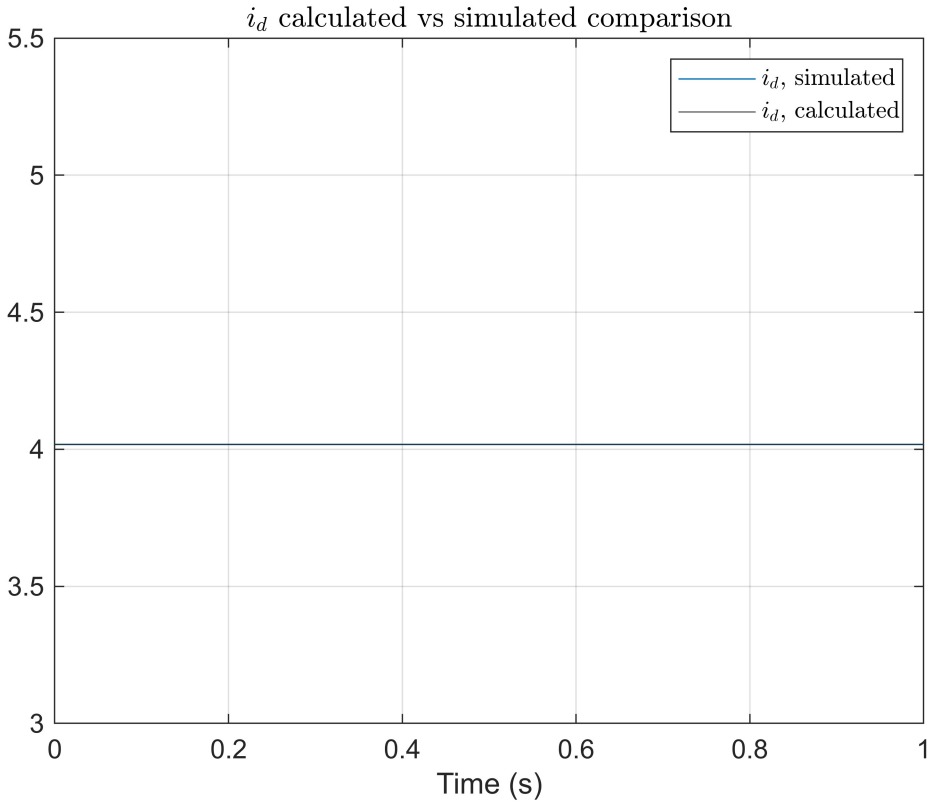
```
i_d = 4.0179
i_q = 5.4185
i_s_abs = 6.7456
u_d_calc = -15.1709
u_q_calc = 279.8946
u_s_abs = 280.3054
```

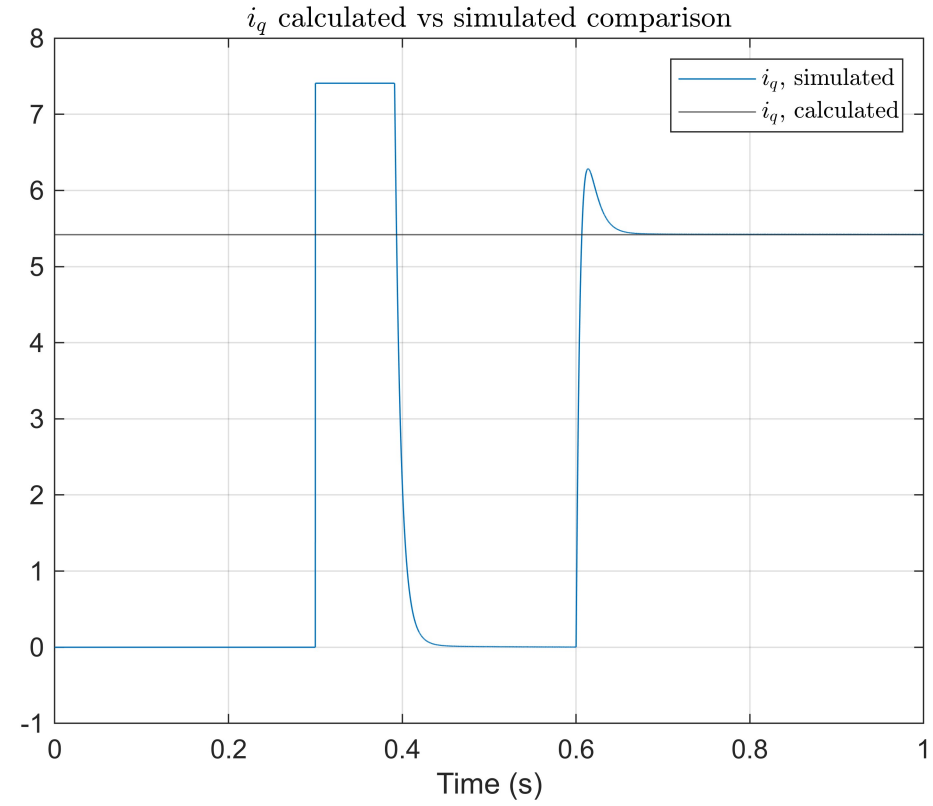
The results can be seen above. In comparison to the simulated values, the calculated ones match well once the system has reached steady state. Comparison can be seen in the following figures.

```
task_10_script;
```


u_d calculated vs simulated comparison
u_d, simulated
u_d, calculated
200
150
100
50
0
-50
0
0.2
0.4
0.6
0.8
1
Time (s)


u_q calculated vs simulated comparison
u_q, simulated
u_q, calculated
350
300
250
200
150
100
50
0
-50
0
0.2
0.4
0.6
0.8
1
Time (s)


i_d calculated vs simulated comparison
i_d, simulated
i_d, calculated
5.5
5
4.5
4
3.5
3
0
0.2
0.4
0.6
0.8
1
Time (s)

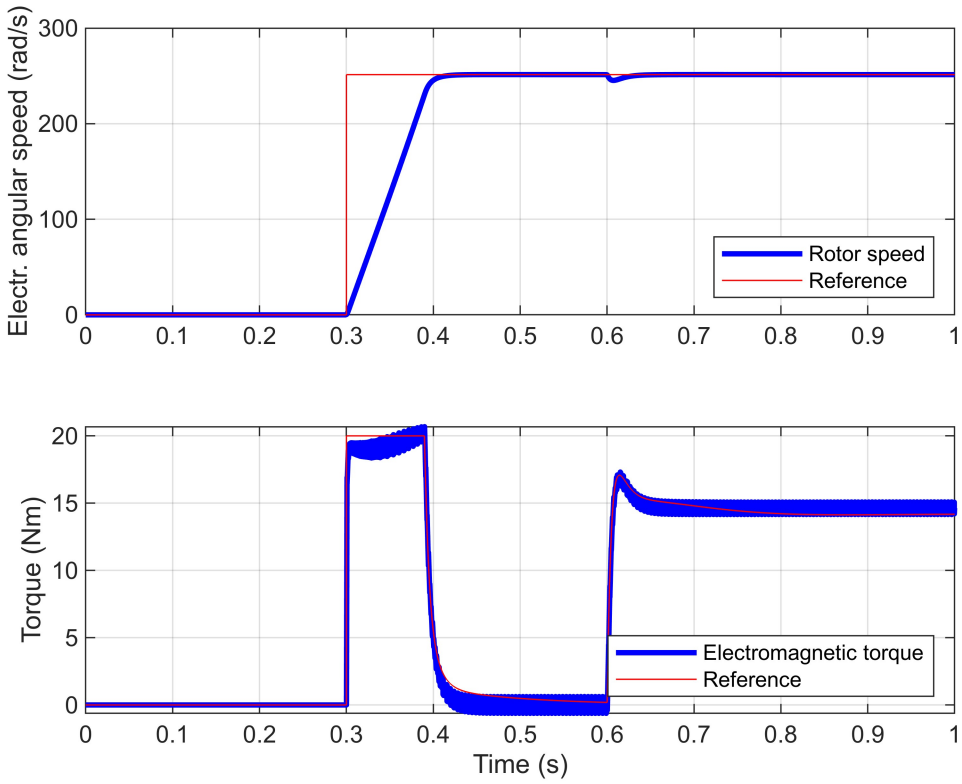

i_q calculated vs simulated comparison
i_q, simulated
i_q, calculated
8
7
6
5
4
3
2
1
0
-1
0
0.2
0.4
0.6
0.8
1
Time (s)

Task 11.

For this task, we initialize a new variable $R_{R,\text{actual}}$ and set it for the rotor resistance in the induction motor simulation model mask. This way we can alter this value, without the value inside the control system changing.

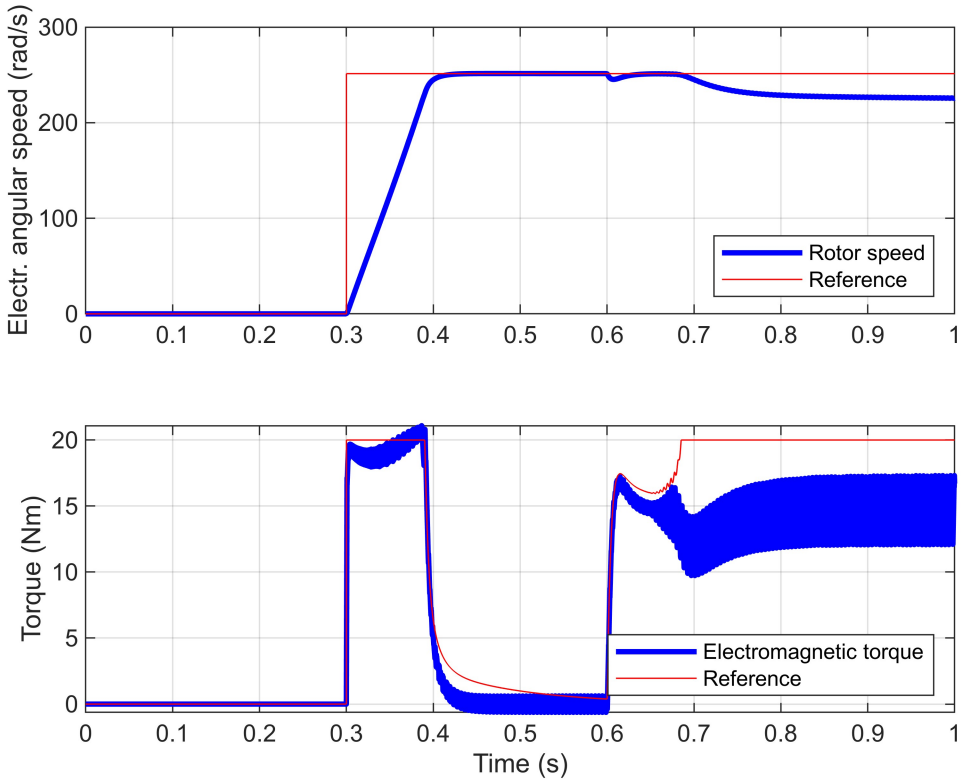
Simulation with 120 % rotor resistance value:

```
init_vector;
R_R_actual = 1.2*R_R;
sim("vector_model.slx");
speed_torque;
```



Simulation with 150 % rotor resistance:

```
R_R_actual=R_R*1.5;
sim("vector_model.slx");
speed_torque;
```



The system seems to withstand 120 % rotor resistance quite well, but with 150 % value the torque ripple and the offset to reference in steady state increaces noticeably. This is due to the flux estimation being incorrect, which directly affects the torque reference.

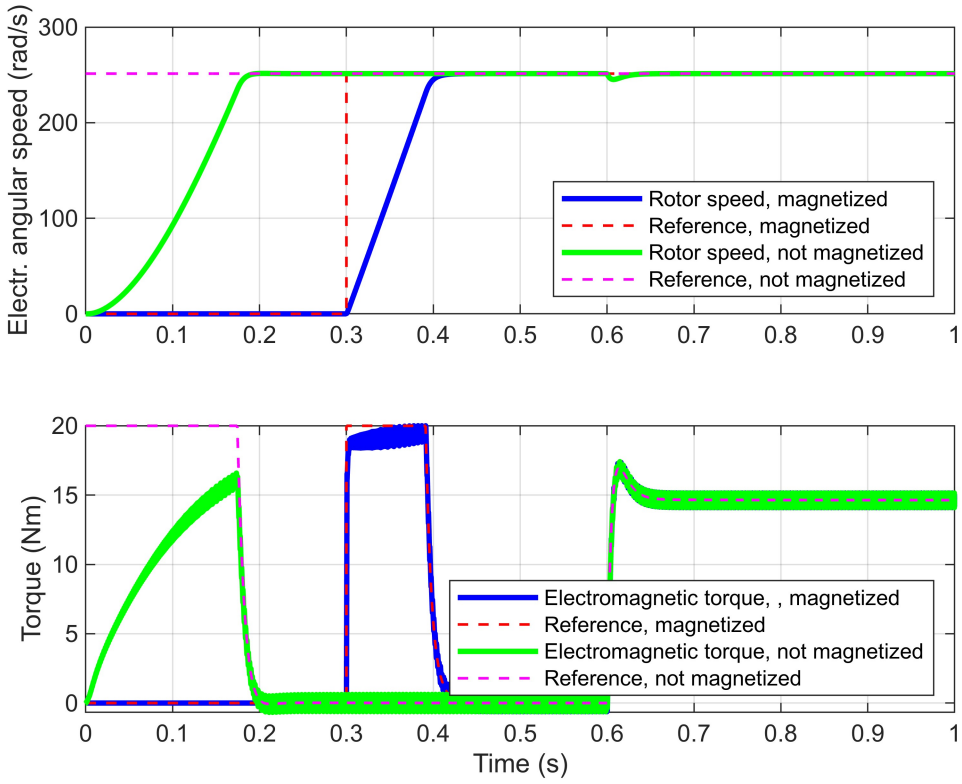
Setting the value back to nominal:

```
R_R_actual = R_R;
```

Task 12.

In this task we changed the speed reference step time to zero. Therefore the stator was not magnetized before the motor starts to accelerate.

```
task12_script;
```

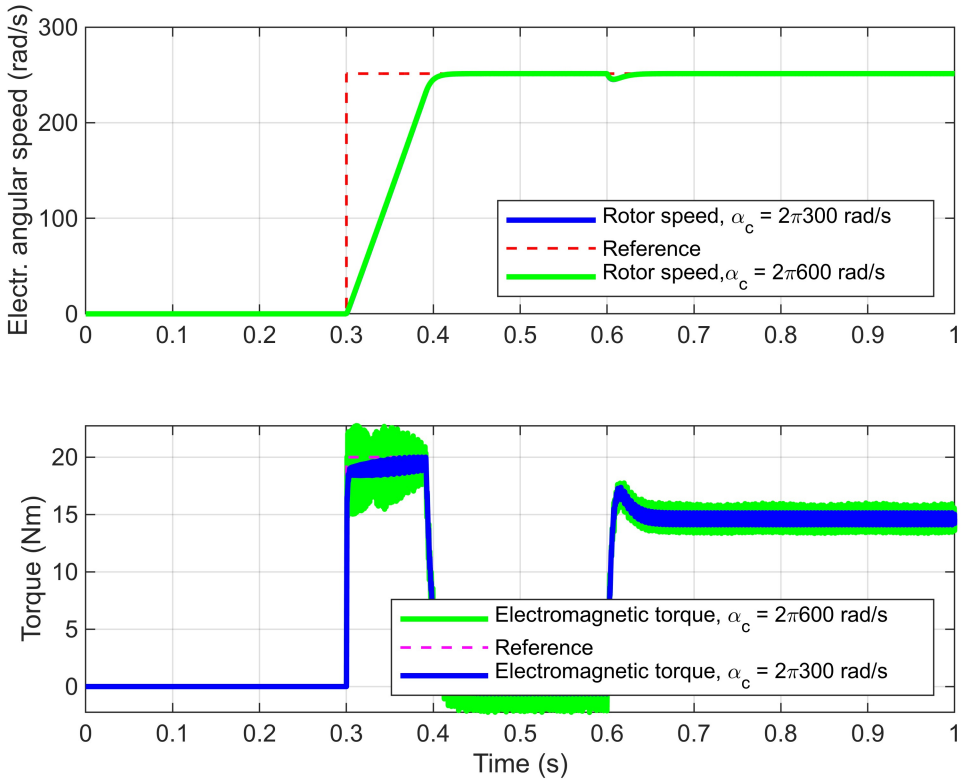


As can be seen on figure above, the acceleration time is almost twice as long when the stator is not magnetized before starting. This phenomena is also visible in the torque curve, the torque buildup takes time when stator is not already magnetized.

Task 13.

In this task we study the model behaviour when the current controller bandwith was doubled to $\alpha_c = 2\pi \cdot 600 \text{rad}/s$.

```
task13;
```



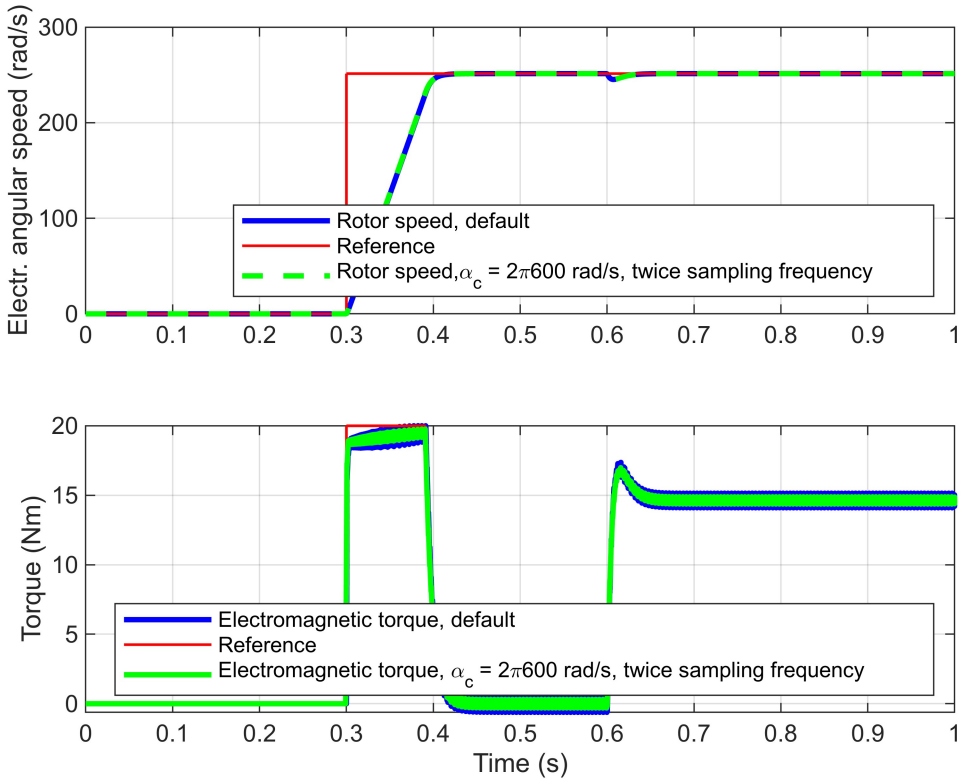
As can be seen the figure above, the increased control-bandwith α_c causes gain stability robustness issues. In 2DOF controller, the bandwidth should not exceed 4% of the angular sampling frequency (Harneforss et al, Control of Voltage-Source Converters and Variable-Speed Drives). In this case the α_c is 7.5% of angular sampling frequency.

$$\alpha_{c,\%} = \frac{2\pi 600 \frac{\text{rad}}{s}}{\frac{2\pi}{125 \cdot 10^{-6}} \frac{\text{rad}}{s}} \cdot 100\% = 7.5\%$$

When also the sampling frequency is doubled the behaviour becomes stable again. Now the:

$$\alpha_{c,\%} = \frac{2\pi 600 \frac{\text{rad}}{s}}{\frac{2\pi}{\frac{(125 \cdot 10^{-6})}{2}} \frac{\text{rad}}{s}} \cdot 100\% = 3.75\%$$

```
task13_2;
```

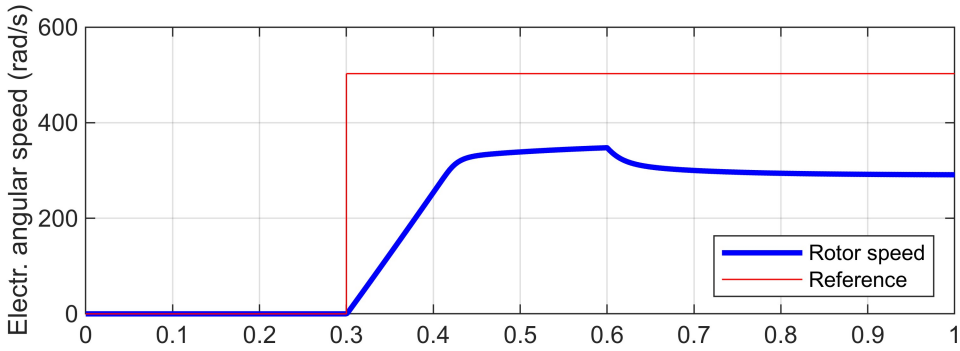


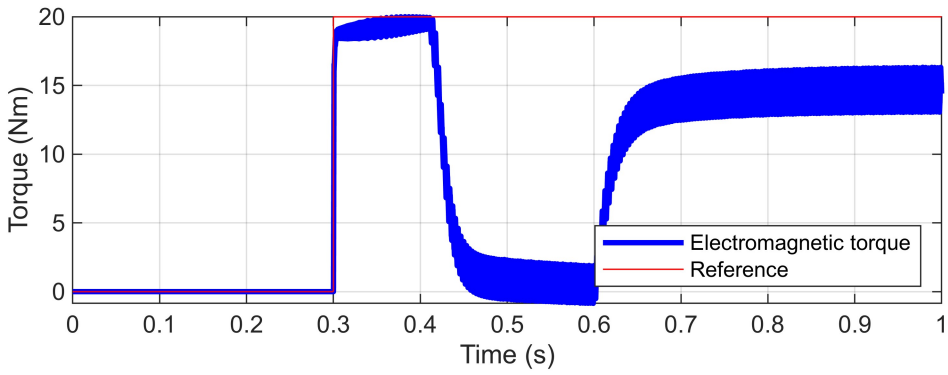
As can be seen figure above, the torque ripple decreases when control-bandwith increases and the sampling frequency is high enough to keep control system gain stability in robust operating area ($\alpha_{c,\%} \leq 4\%$). The speed behaviour is identical to default case, because the current control bandwith doesn't affect the speed controller it at all.

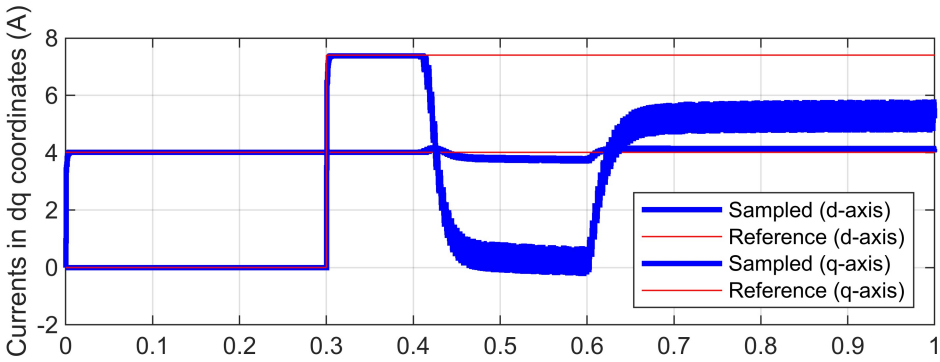
Task 14.

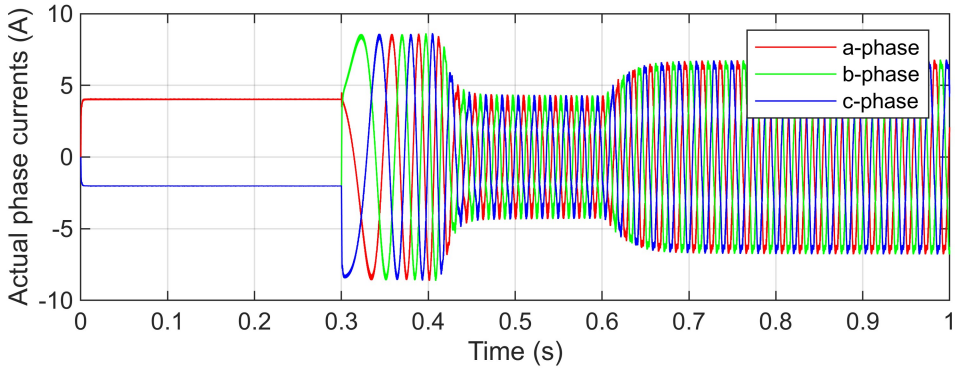
Initially, the speed reference step final value is set to $\omega_M = 2\pi * \frac{40}{n_p} \frac{\text{rad}}{s}$, with $n_p = 2$. It was requested to set the final value to $\omega_M = 2\pi * 40 \frac{\text{rad}}{s}$, so we handle this by simply multiplying the value by 2 which efficiently sets the speed reference $\omega_M = 2\pi * 40 \frac{\text{rad}}{s}$.

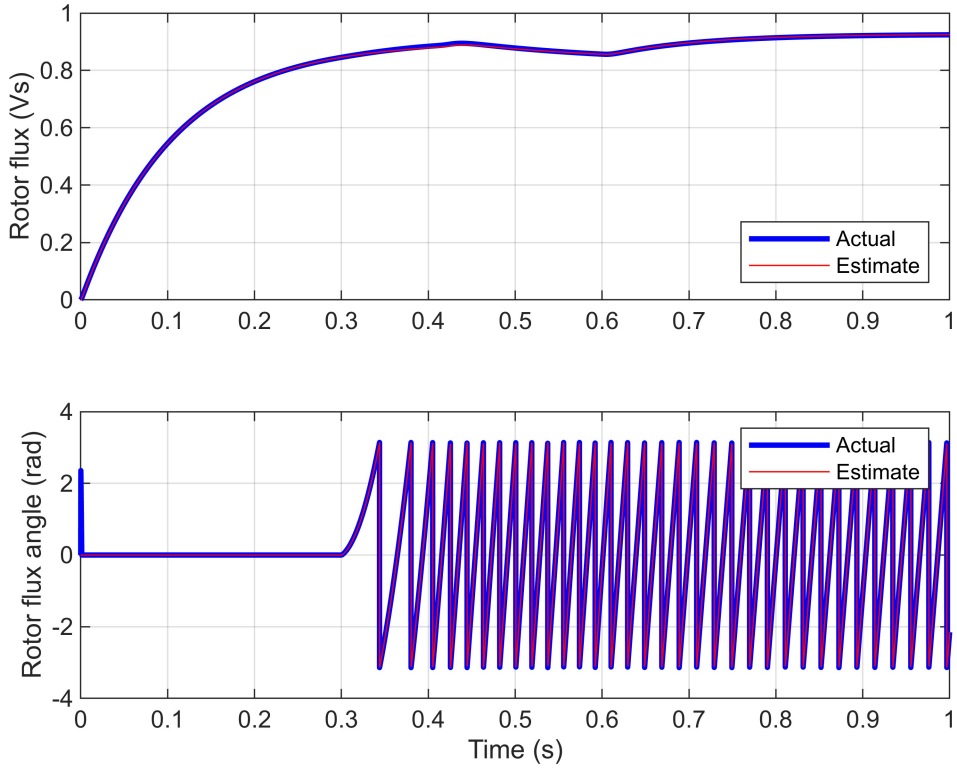
```
w_m_ref = 2*w_m_ref; %Changing the speed reference w_M to 2pi*40 rad/s
sim("vector_model.slx");
fig_vector; % plotting speed and torque
```


Electr. angular speed (rad/s)
600
400
200
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Rotor speed
Reference


Torque (Nm)
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (s)
Electromagnetic torque
Reference


Currents in dq coordinates (A)
8
6
4
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Sampled (d-axis)
Reference (d-axis)
Sampled (q-axis)
Reference (q-axis)


Actual phase currents (A)
10
5
0
-5
-10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Time (s)
a-phase
b-phase
c-phase

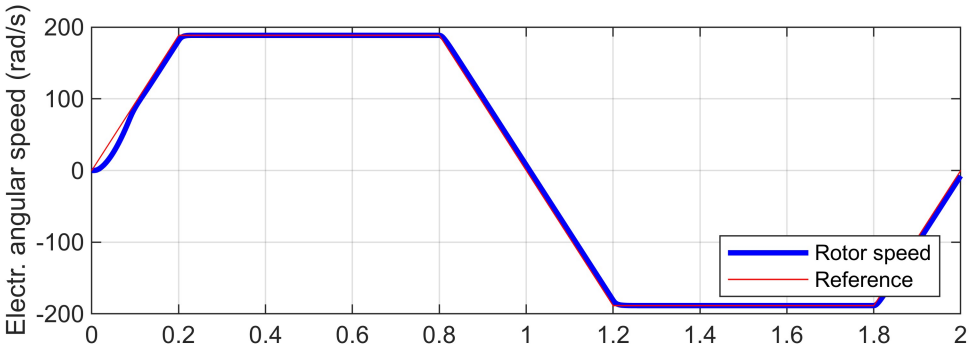


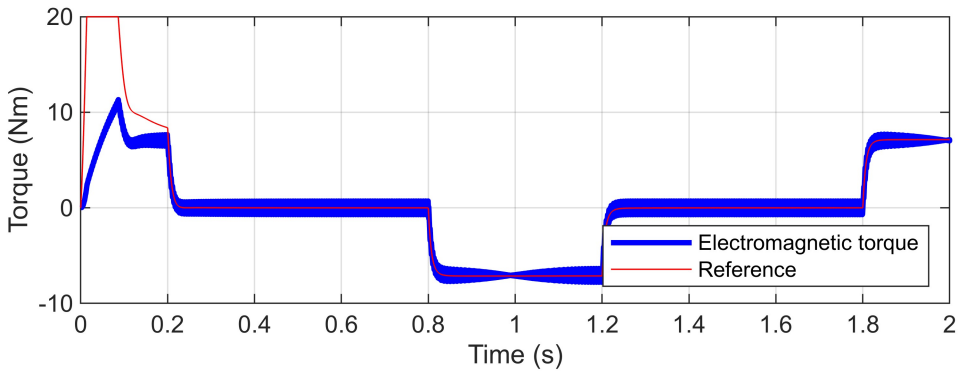

```
w_m_ref = w_m_ref / 2; % reset speed refence to the default
```

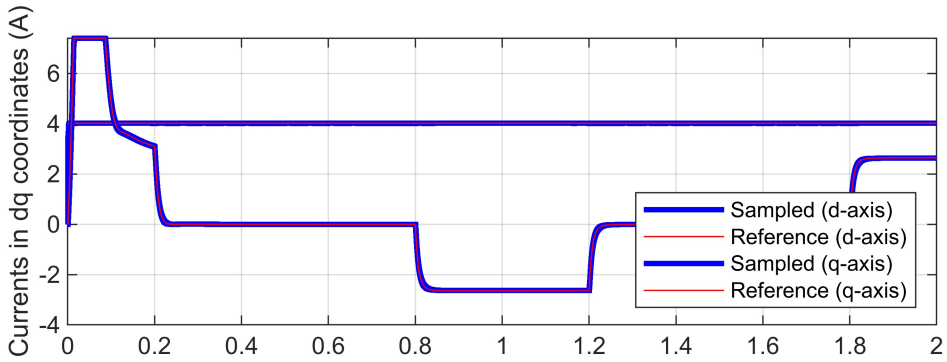
The speed reference is higher than the maximum speed of the motor without the field weakening. Therefore speed controller keeps the torque reference in the maximum value after the speed step, but then the back_EMF limit the current available and therefore also the torque. When the nominal load torque occurs the slip of the motor increases resulting desired torque. The high torque ripple is due to the poor reference tracking ie. speed controller sets the torque reference to the maximum to achieve the reference speed.

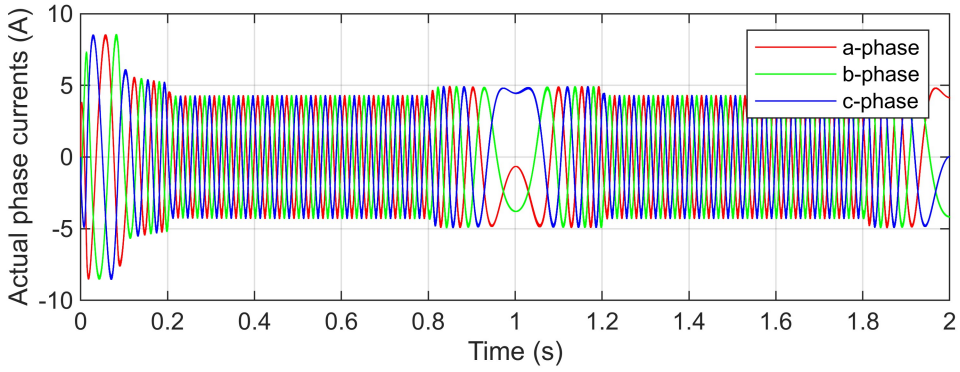
Task 15.

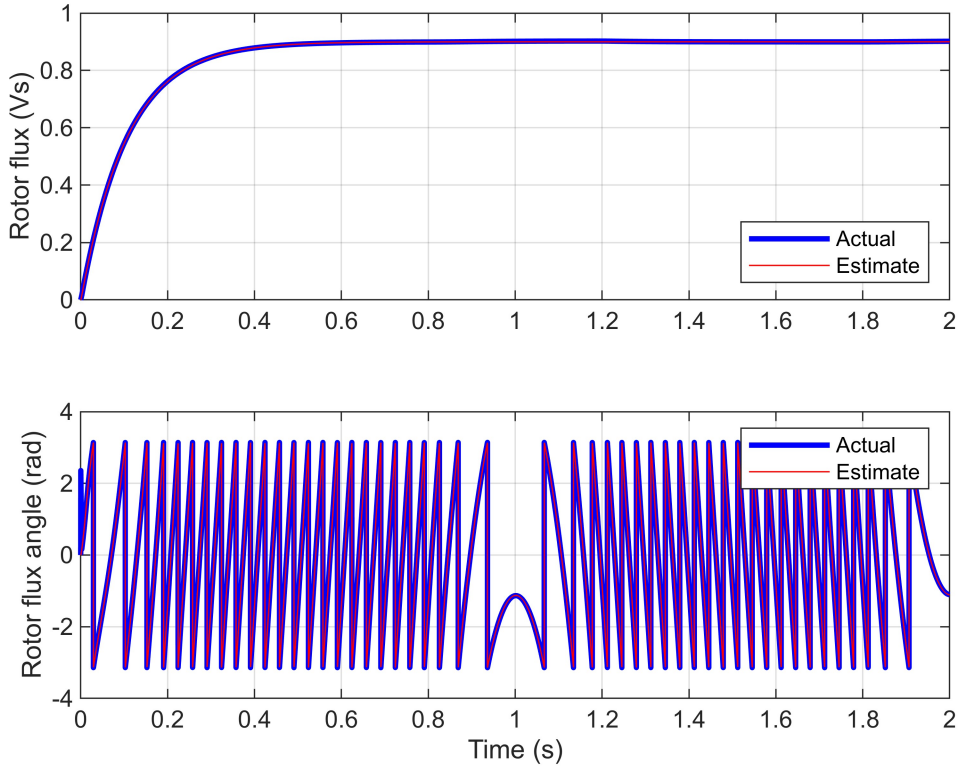
```
set_param(gcb,'sw','0'); % set the switch to the "Repeating Sequence "
T_L = 0; % load torque to zero
sim("vector_model.slx",2)
fig_vector;
```


Electr. angular speed (rad/s)
200
100
0
-100
-200
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Rotor speed
Reference


Torque (Nm)
20
10
0
-10
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (s)
Electromagnetic torque
Reference


Currents in dq coordinates (A)
6
4
2
0
-2
-4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Sampled (d-axis)
Reference (d-axis)
Sampled (q-axis)
Reference (q-axis)


Actual phase currents (A)
10
5
0
-5
-10
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (s)
a-phase
b-phase
c-phase




```
set_param(gcb,'sw','1'); % set the switch to the "step";
T_L = 14.63; %load torque back to default
```

At the starts the motor torque cannot produce desired torque, because the stator is not magnetized. After the stator have reached the constant flux, the torque reference tracking is almost perfect, even when motor changes the direction. From the phase current figure, we can see that when the change of the direction of rotation happens, the order of phases is changed. It looks like phases b and c switch places. The change of direction of rotation is also seen in the rotor flux angle waveform.