POLITECNICO
MILANO 1863

DYNAMICS OF ELECTRICAL MACHINES AND DRIVES

Project Report: Induction Motor Drive

Giovanni Ploner – 970997

A.Y. 2020/21

ELECTRICAL DRIVE OF AN AIR COMPRESSOR

Consider the air treatment system of a blast furnace based on a centrifugal compressor (CC) driven by an induction motor, with the following characteristics:

- rated voltage 380V
- rated frequency 50Hz with pole pairs np = 3
- stator resistance $Rs = 0.24\ \Omega$ and inductance $Ls = 59.4mH$
- rotor resistance $Rr = 0.175\ \Omega$ and inductance $Lr = 59.1mH$
- mutual inductance $Lm = 57mH$

The compressor is coupled to the Induction Machine with a gearbox $r = \omega_L/\omega_m\ = 4$
- the equivalent inertia seen from the motor side is $J = 0.4\ kgm^2$
- friction coefficient $\beta = 0.068$

The compressor behaviour sees a load torque proportional to the square of the shaft speed as $m_l = k \times (r\,\Omega_m)^2$ where $k = 0.009\ Nm/rad^2/s^2$

EXERCISE

1- Simulate the Induction Motor without a load at nominal voltage in order to evaluate the starting transient by figuring out:
   - Stator current i_s and rotor current i_r
   - Torque m_e vs. speed Ω_m
   - Position of ψ_s and ψ_r in (α,β) reference frame

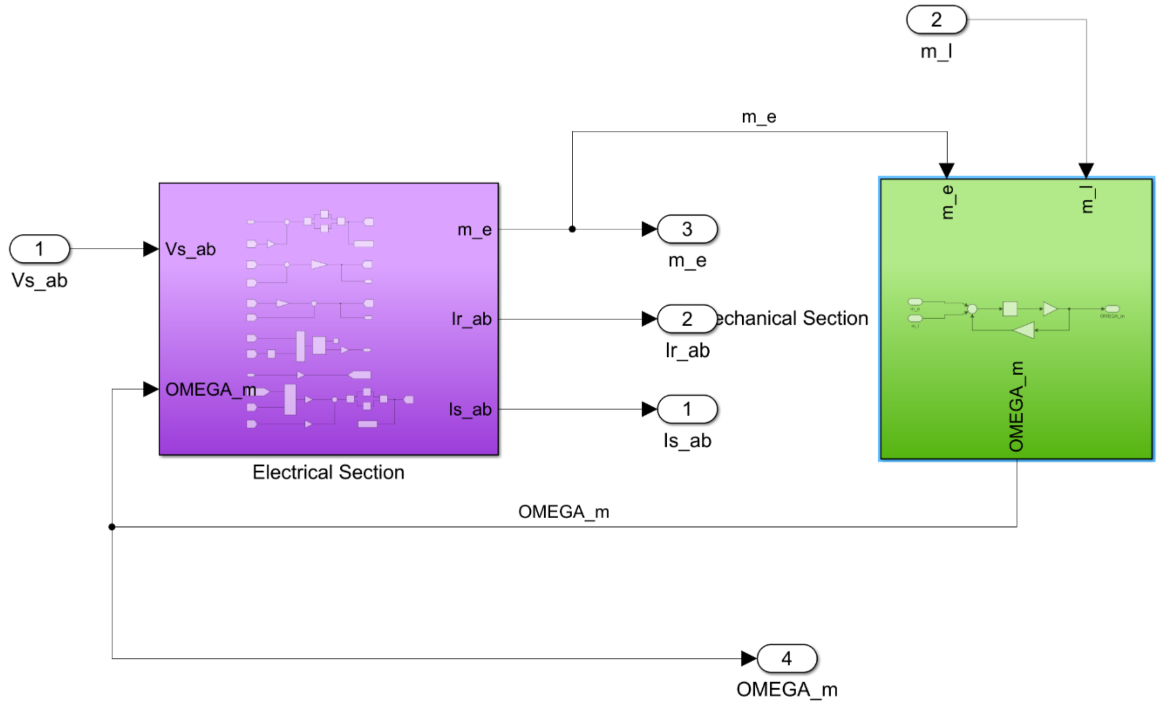
2- Design and simulate the speed, current and flux demand at least one decade faster than the intrinsic characteristic of the uncontrolled system. Figure out:
   - Currents, torque, speed profiles compared to the reference one.

INDUCTION MOTOR MODEL

Analysing the Induction Machine model, we divided the system in 2 parts:

- Electrical Section (purple);
- Mechanical Section (green).

Obtaining a total of 2 Inputs and 4 Outputs.



Electrical Section

Inputs:

- v_s in fixed (α,β) reference frame
- Ω_m mechanical speed in rad/s

Outputs:

- i_s in fixed (α,β) reference frame
- i_r in fixed (α,β) reference frame
- m_e in [N*m]

In this part of the Simulink script is represented the 4-parameters model referred to the (α,β) stationary reference frame and, more specifically, each row of the scheme is one of the following equations:

$$\psi_{s\,(\alpha,\beta)} = \int (v_{s(\alpha,\beta)} - R_s i_{s(\alpha,\beta)})\, dt$$

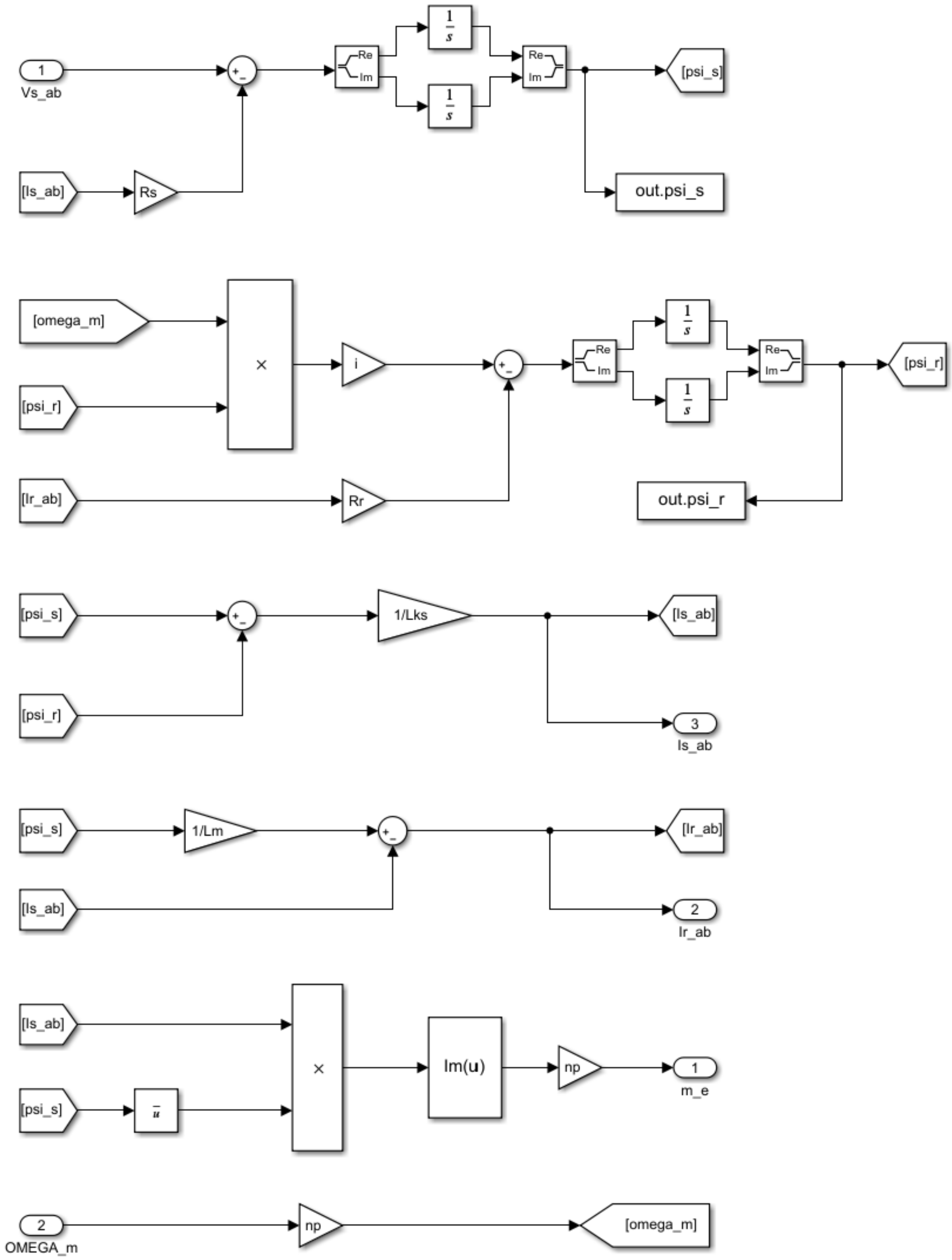
$$\psi_{r\,(\alpha,\beta)} = \int (-R_r i_{r(\alpha,\beta)} + j\omega_r \psi_{r\,(\alpha,\beta)})\, dt$$

$$i_{s(\alpha,\beta)} = \frac{1}{L_{ks}} (\psi_{s\,(\alpha,\beta)} - \psi_{r\,(\alpha,\beta)})$$

$$i_{r(\alpha,\beta)} = \frac{\psi_{r\,(\alpha,\beta)}}{L_m} - i_{s(\alpha,\beta)}$$

$$m_e = n_p Im(\bar{i}_{s(\alpha,\beta)} \underline{\psi}_{s\,(\alpha,\beta)})$$

$$\omega_m = n_p \Omega_m$$



Complex decomposition block are used for equations (a) and (b) to avoid possible SIMULINK errors.

Mechanical Section

Inputs:

- m_e in [N*m]
- m_l in [N*m]

Outputs:
- Ω_m mechanical speed in [rad/s]

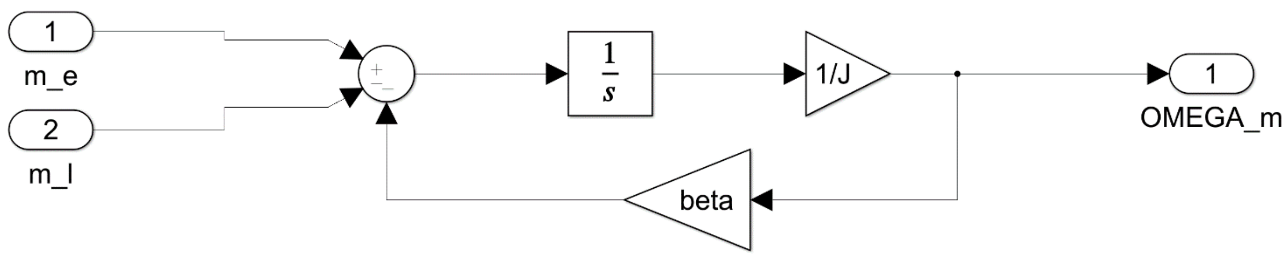
The equation implemented in the SIMULINK model is:

$$\Omega_m = \frac{1}{J}\int (m_e - m_l - \beta\Omega_m)dt$$

Where β is a friction coefficient and m_l a load torque given by the equation:

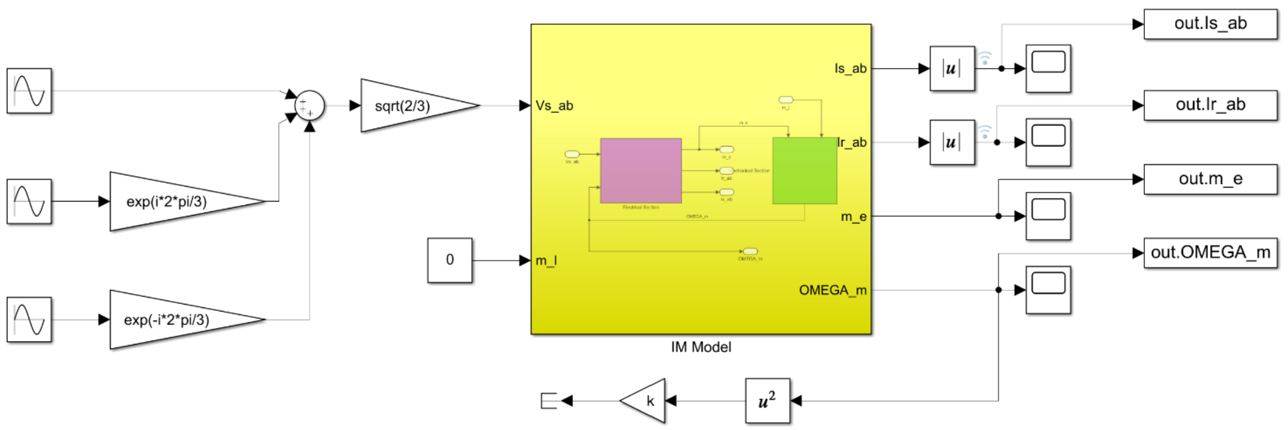
$$m_l = K(r\Omega_m)^2$$

Which is represented outside the I.M. Model subsystem, as can be seen in the full induction machine scheme.



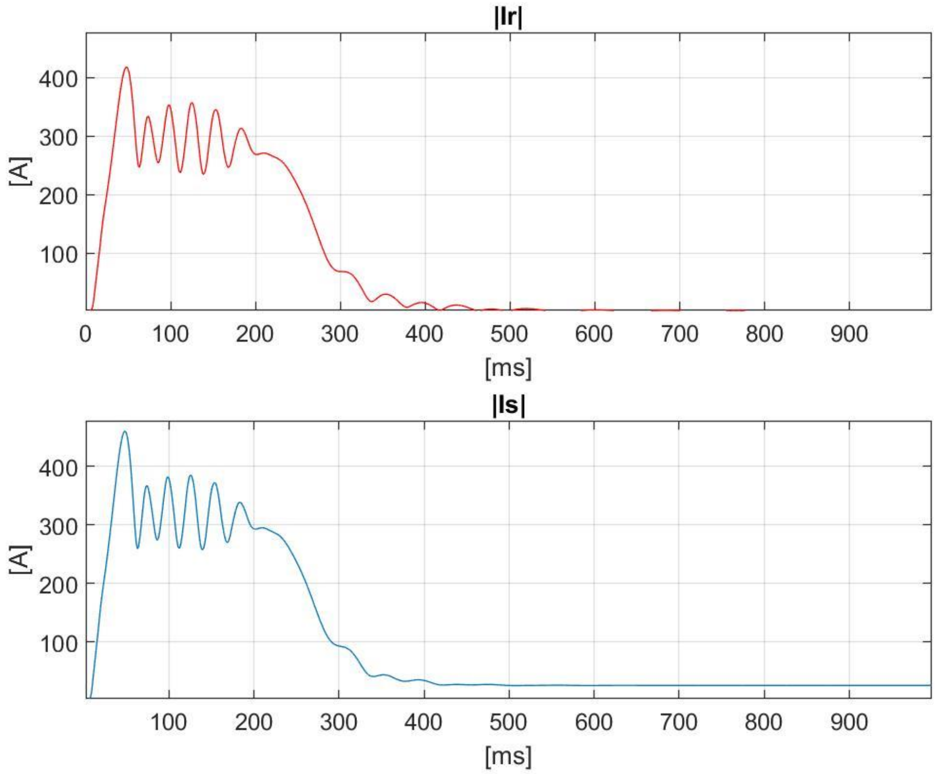
STARTING TRANSIENT SIMULATION

The induction motor (whose model I have depicted in yellow) is simulated with no load (m_l = 0) at the given 3-phase nominal voltage to evaluate the starting transient.
Can also be seen the space phasor transformation of the voltage V_s in the picture below.

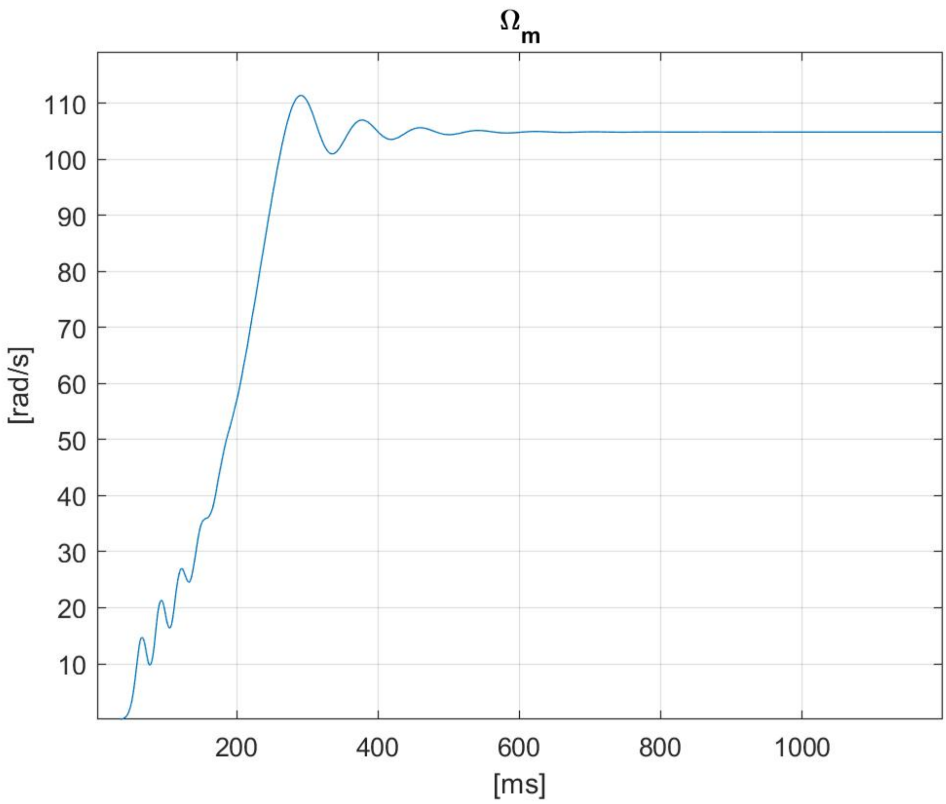


The results are the following:

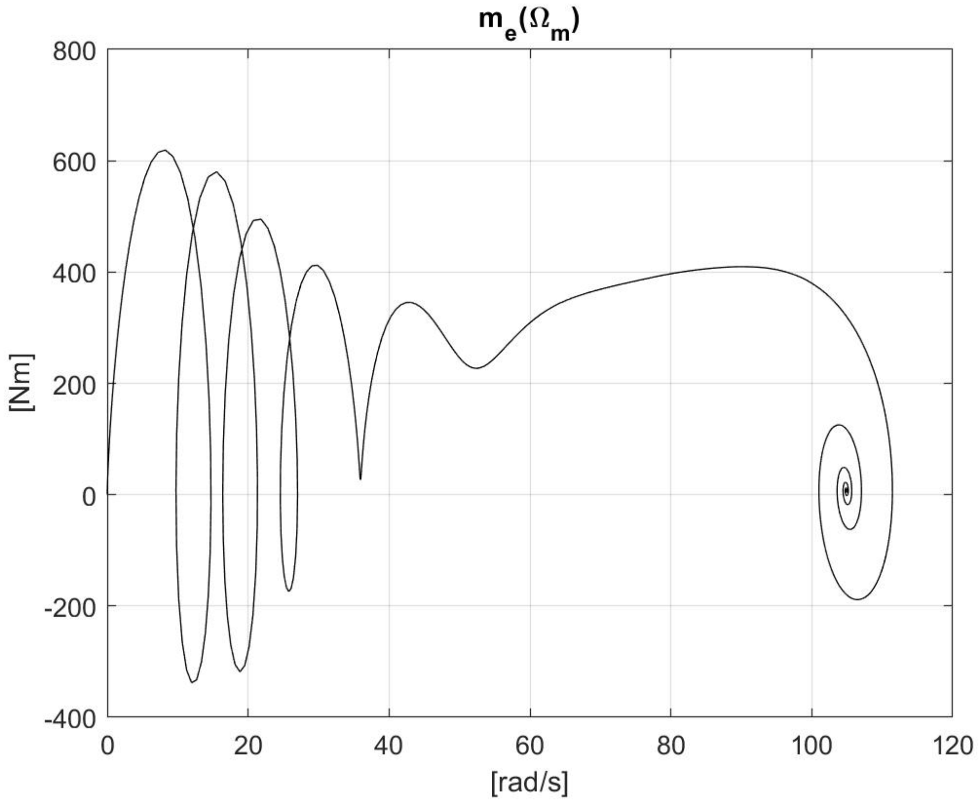
1. Absolute value of rotor and stator current in (α,β) reference frame



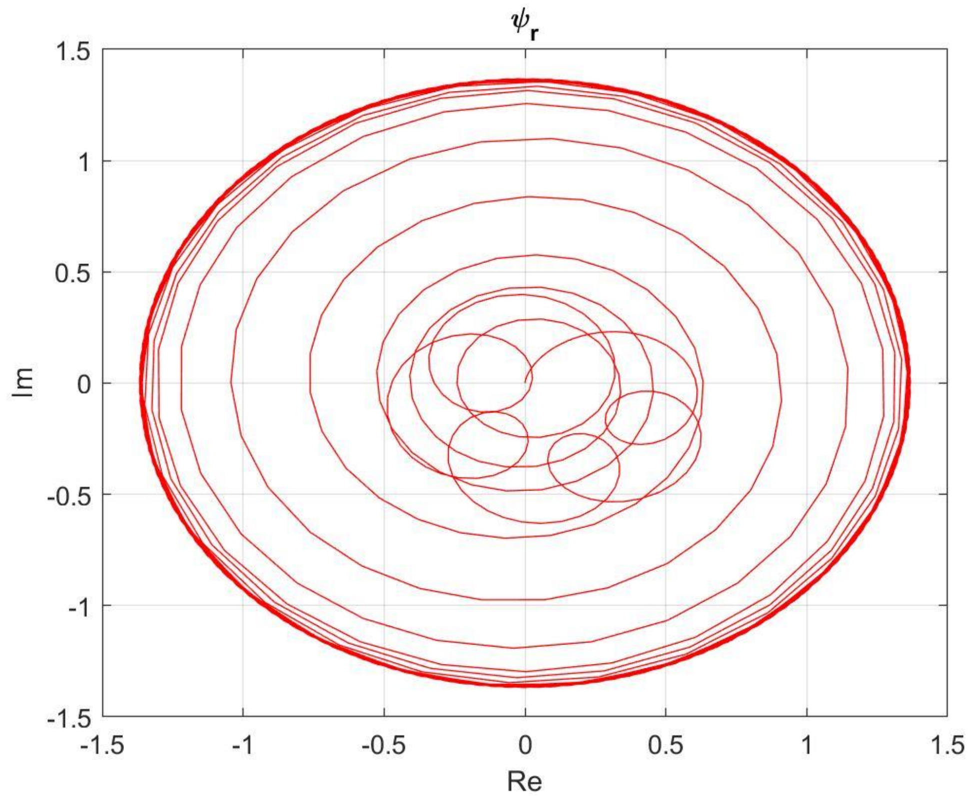
2. Motor mechanical speed Ω_m

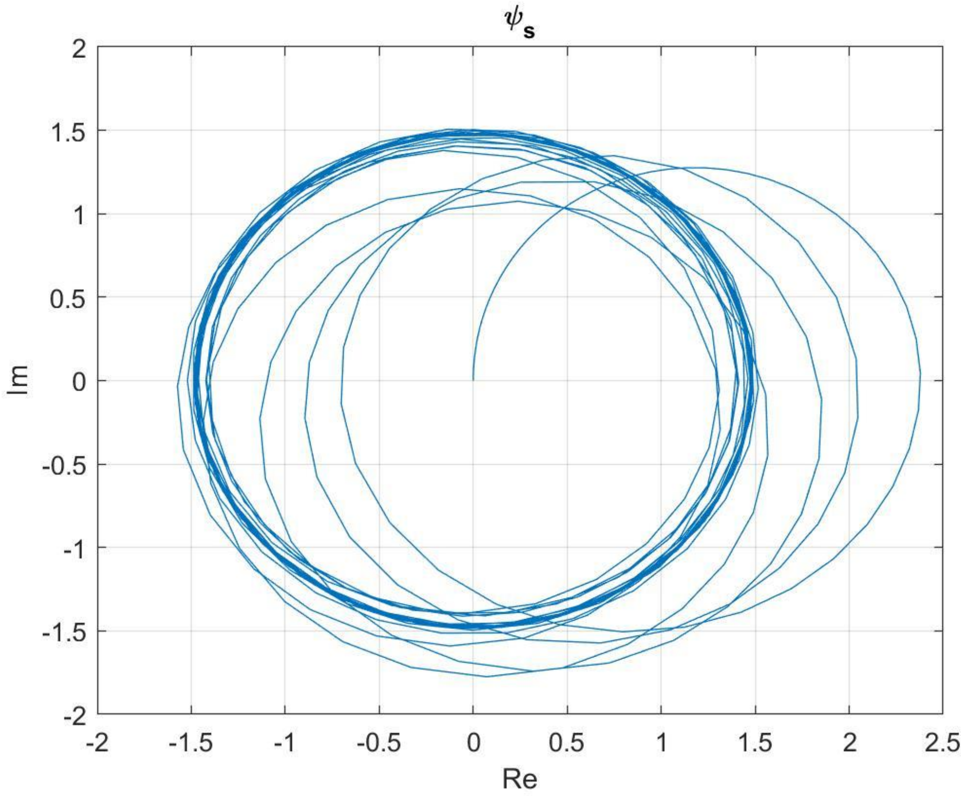


3. Torque vs. Speed relation



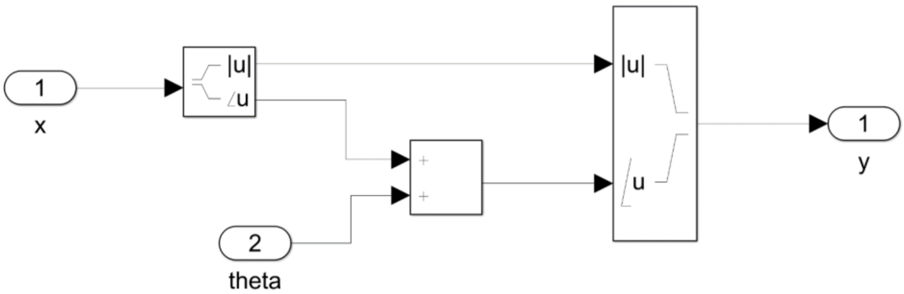
4. $\psi_{r\,(\alpha,\beta)}$ *and* $\psi_{s\,(\alpha,\beta)}$ in (α,β) reference system





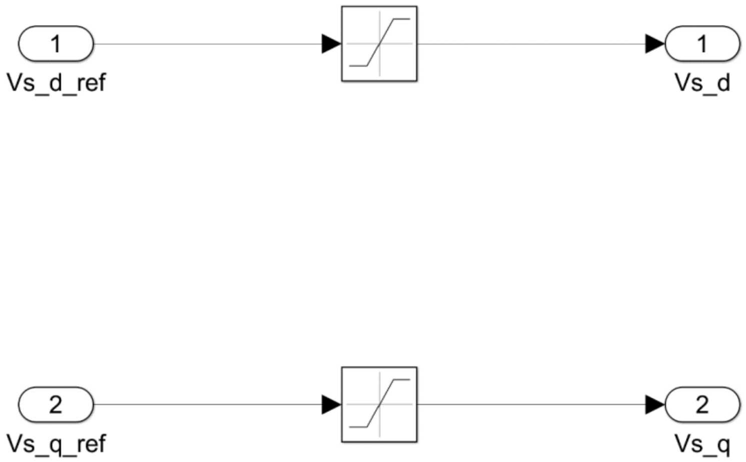
REFERENCE FRAME CONVERSION

To rotate a vector by a generic angle θ, the following subsystem (called "Rotation" block), is used:



In order to move from (d,q) reference frame to (α,β) a rotation of θ_s is needed, while to move from (α,β) to (d,q) it is necessary to rotate by an angle of - θ_s.

POWER SUPPLY MODEL



For the sake of simplicity, the power supply is assumed to have a very fast dynamics compared to the rest of the system.
For this reason it is modelled as a unit gain saturated by the maximum possible voltage reachable by the inverter.

CONTROLLER DESIGN

Speed Controller

Starting from the equation:

$$\dot{\Omega}_m = (m_e - m_l - \beta\Omega_m)\frac{1}{J}$$

Considering m_l as a disturbance, the transfer function between torque and speed is:

$$G_{\Omega m} = \frac{1}{sJ + \beta}$$

The outer loop of the controller regulates the speed by acting on the torque m_e (it is assumed that the inner current loop is much faster so that its dynamics can be neglected).
We must find PI controller that guarantees a ω_c at least one decade faster than the intrinsic characteristic of the uncontrolled system and a phase margin of at least 60°.
Looking at the speed profile we can assume $\omega_{mecc} = 20[rad/s]$, which is higher than the maximum requested slope of 1 [rad/s].
To fullfill the requirements, the parameters chosen for the PI are:

- Kp = 8;
- Ki = 1.36;
- Saturation at $m_{e,n}$

Flux Controller

We can distinguish between two operating regions for the controller:

a- If $\Omega_m \leq \Omega_b$ the motor operates in the constant torque region, all the ferromagnetic material is properly used such that we have the maximum admissible torque. The flux reference is:

$$\psi_{ref} = \psi_{rn}$$

b- If $\Omega_m > \Omega_b$ the motor enters in the flux weakening region, e.m.f and the power are kept constant while the flux is decreased inversely proportionally with respect to the speed.
Given:

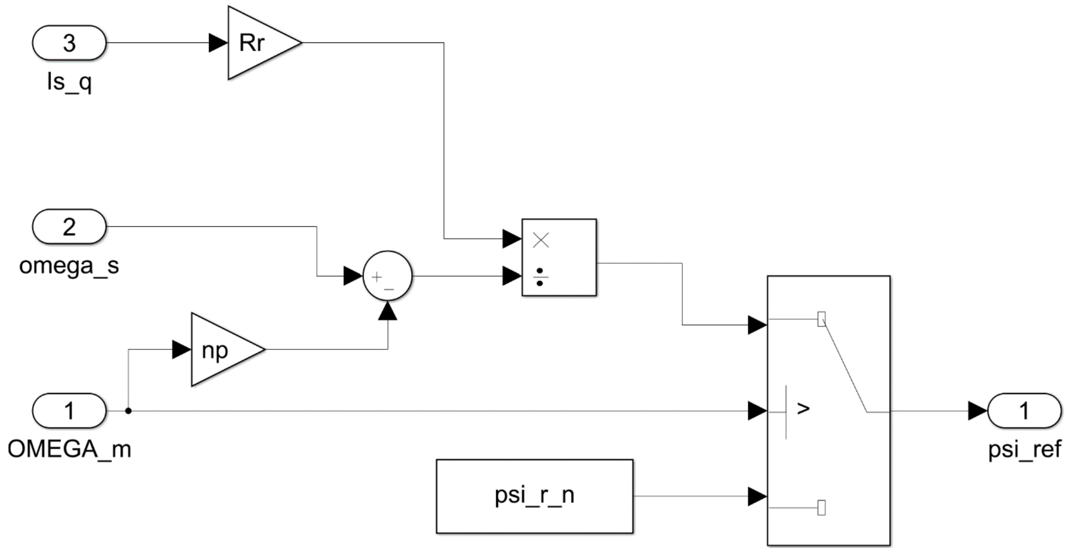
$$i_{sq} = i_{sq,n} = const$$

$$0 = R_r i_{sq} - (\omega_s - \omega_r)\psi_r$$

The reference flux is given by the equation:

$$\psi_{ref} = \frac{R_r i_{sq}}{\omega_s - n_p \Omega_m}$$

In our scheme it is used the "switch" SIMULINK block to represent the two possible cases, even though in this exercise Ω_m never exceeds the base speed.



Also in this situation we have chosen a PI controller, tuned on the transfer function:

$$G_{\psi i} = \frac{R_r}{s + \frac{R_r}{L_m}}$$

With parameters:
- Kp = 1.14×10^3;
- Ki = 3.51×10^3;
- Saturation at $i_{sd,n}$

Current Controller
The flux and speed controller set the reference for i_{sq} and i_{sd} currents.

The controller for these currents must have a ω_c much higher than the outer control loops ($\omega_{elec} = 100 \times \omega_{mecc}$).
The plant transfer function seen by the PI controller used in this case is:

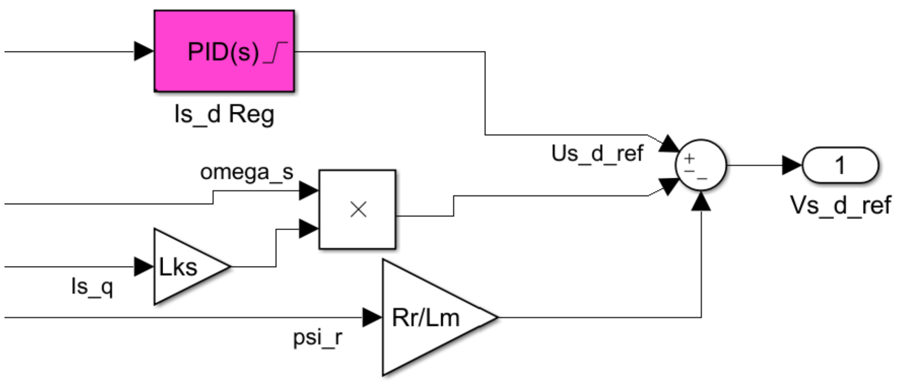
$$G_i = \frac{1}{R_{ks} + sL_{ks}}$$

Given the requirements, the following parameters were chosen:
- Kp = 8.85;
- Ki = 830;
- Saturation at $2 * V_n * \frac{\sqrt{(2)}}{\sqrt{(3)}}$

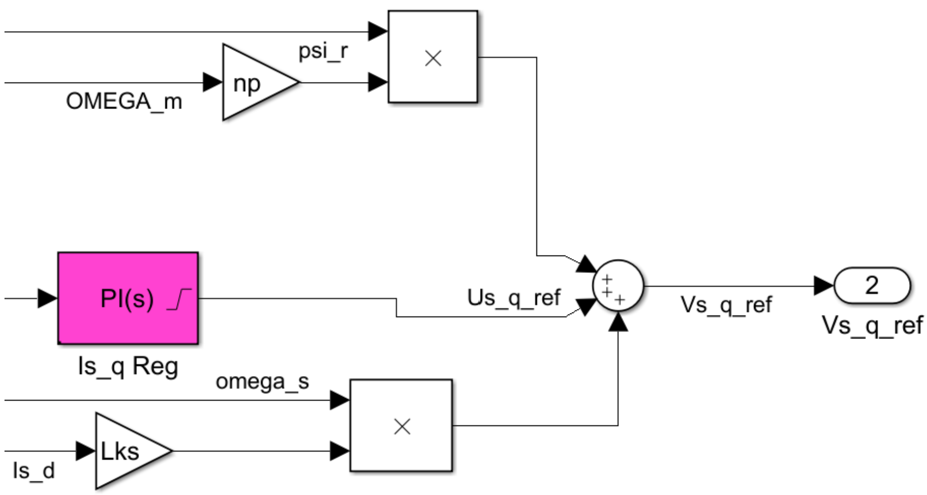
COMPENSATION TERMS

Some compensation terms must be added to compensate mutual coupling terms between the two loops and the term representing the electromotive force: they are, in fact, treated as disturbances inside the control scheme. Cross coupling terms are used to mitigate cross coupling or big and rapid changes in speed, current and flux linkage at steady state.

$$v_{sd} = u_{sd} - {}^{R_r}/_{L_m}\, \psi_r - \omega_s L_{ks} i_{sq}$$

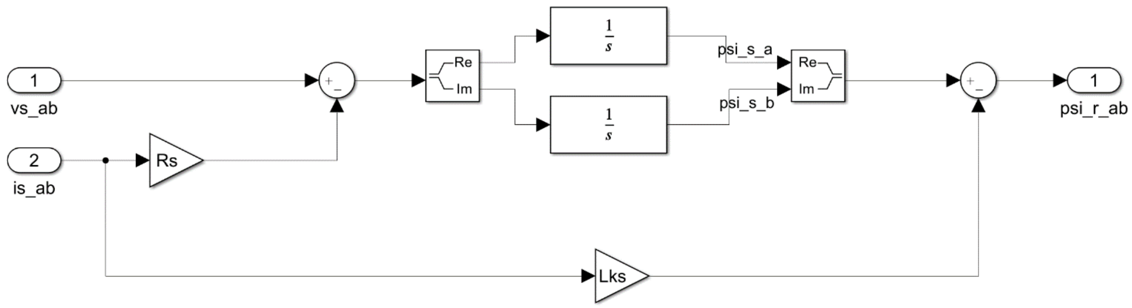


$$v_{sq} = u_{sq} + n_p \Omega_m \psi_r + \omega_s L_{ks} i_{sd}$$



FLUX ESTIMATOR

Rotor flux cannot be directly measured and must be estimated. To do that a V-I estimator (or observer) is used:



The equations implemented in Simulink are:

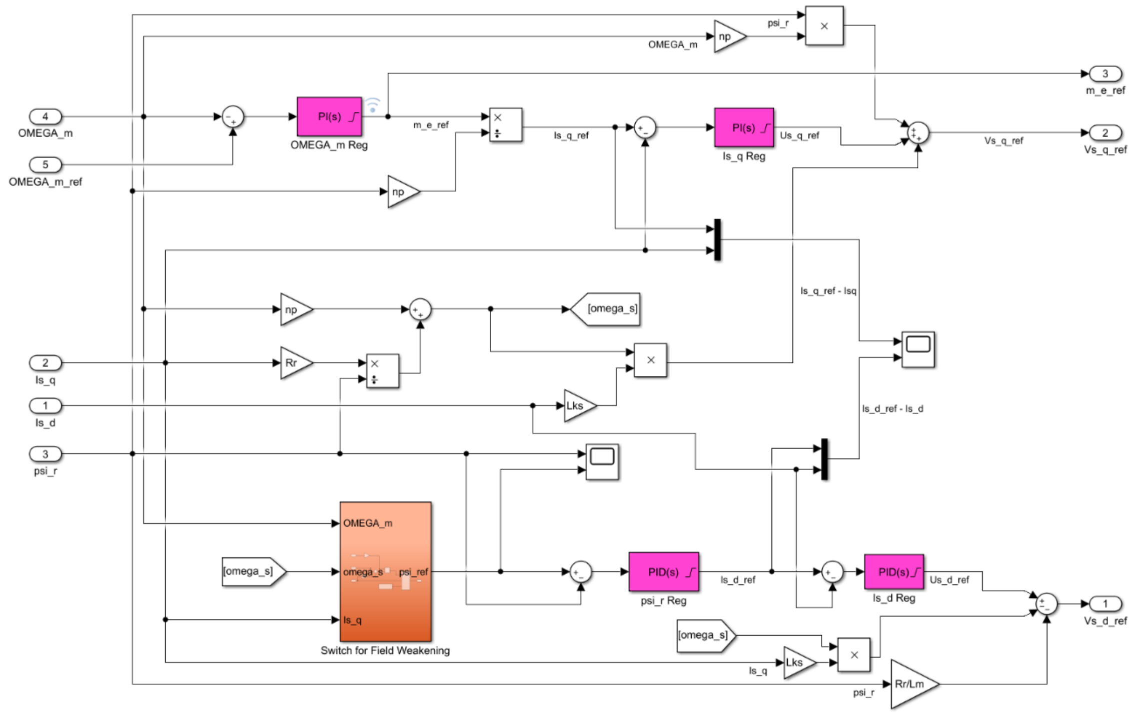
$$\psi_{s\ (\alpha,\beta)} = \int (v_{s(\alpha,\beta)} - R_s i_{s(\alpha,\beta)})\, dt$$

$$\psi_{r\ (\alpha,\beta)} = \psi_{s\ (\alpha,\beta)} - L_{ks} i_{s(\alpha,\beta)}$$

Instead of using a pure integrator, which can make the estimation diverge, we can put a low-pass filter $\frac{1}{1+s}$. These solution, however, doesn't provide us anymore a correct estimation for very low speed values.
The orientation of the flux $\psi_{r\ (\alpha,\beta)}$ is given by $\theta_s = \measuredangle(\psi_{r\ (\alpha,\beta)})$.
Using this angle it is possible to move from (α,β) reference system to (d,q) one thanks to the "Rotation" block scheme.

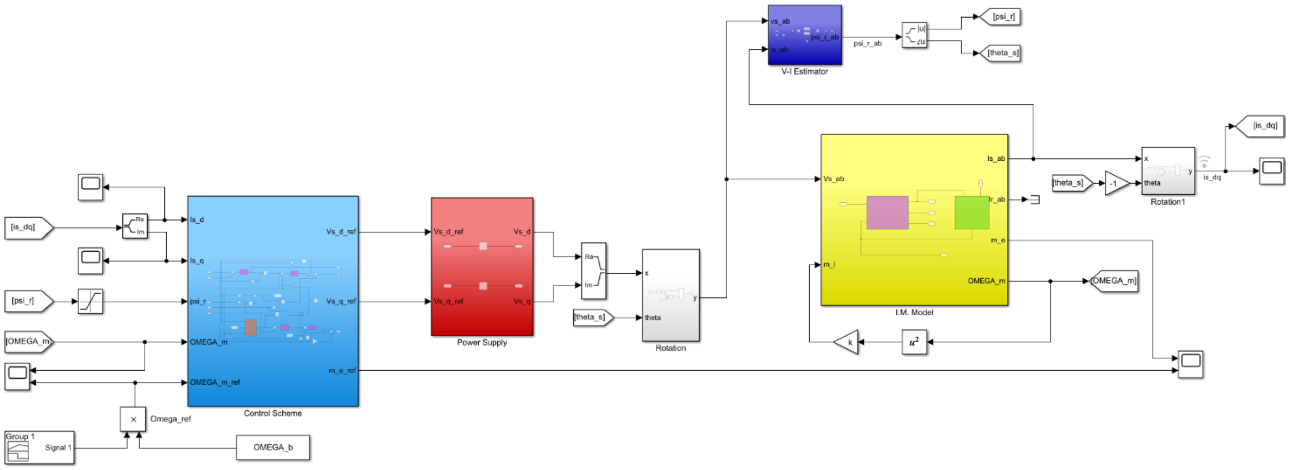
COMPLETE CONTROLLER SCHEME

The complete controller scheme is shown in the picture below; the flux switch block is coloured in orange, while the 4 PI controllers are in pink.



CONTROLLER AND INDUCTION MACHINE MODEL

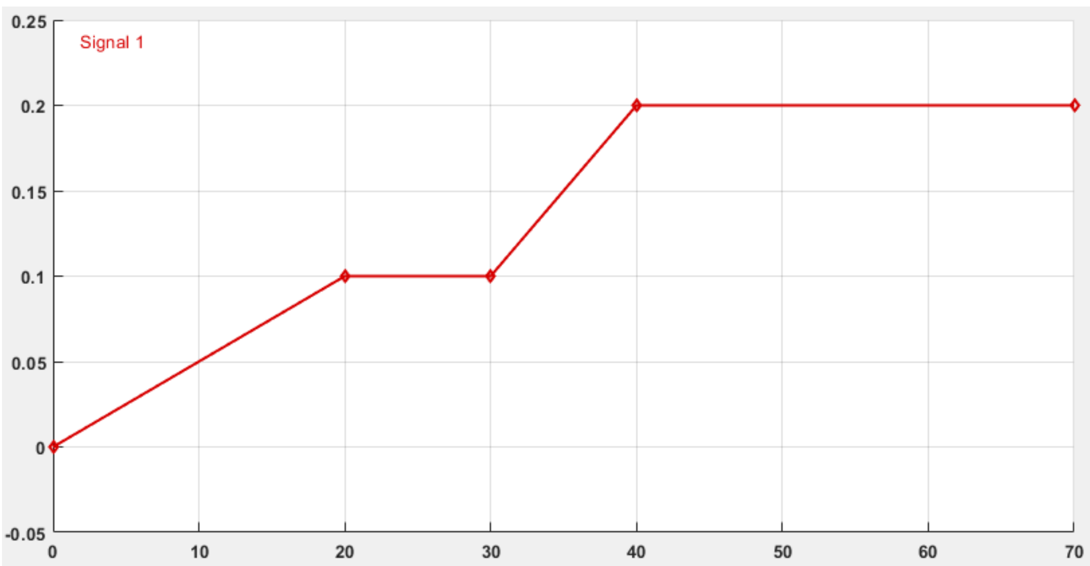
Connecting together all the blocks previously described, the following complete model is obtained.



The I.M. is depicted in yellow, while the controller is in light-blue. The V-I estimator, the power supply and the "Rotation" blocks are respectively in blue, red and white.

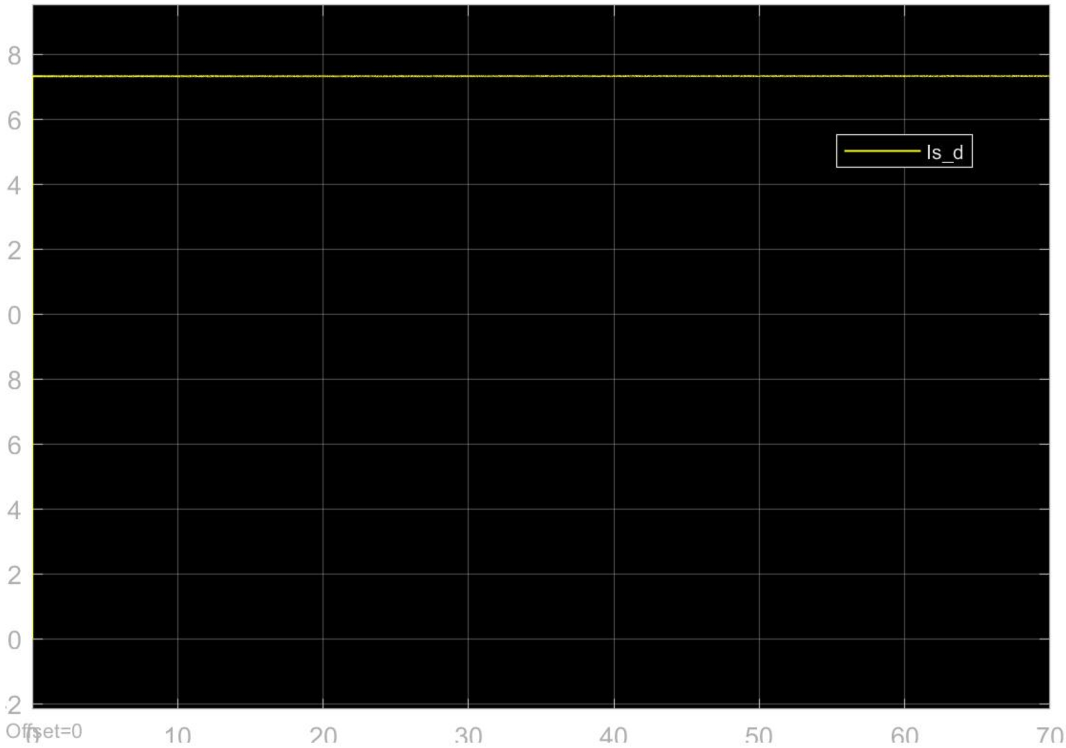
SIMULATION RESULTS

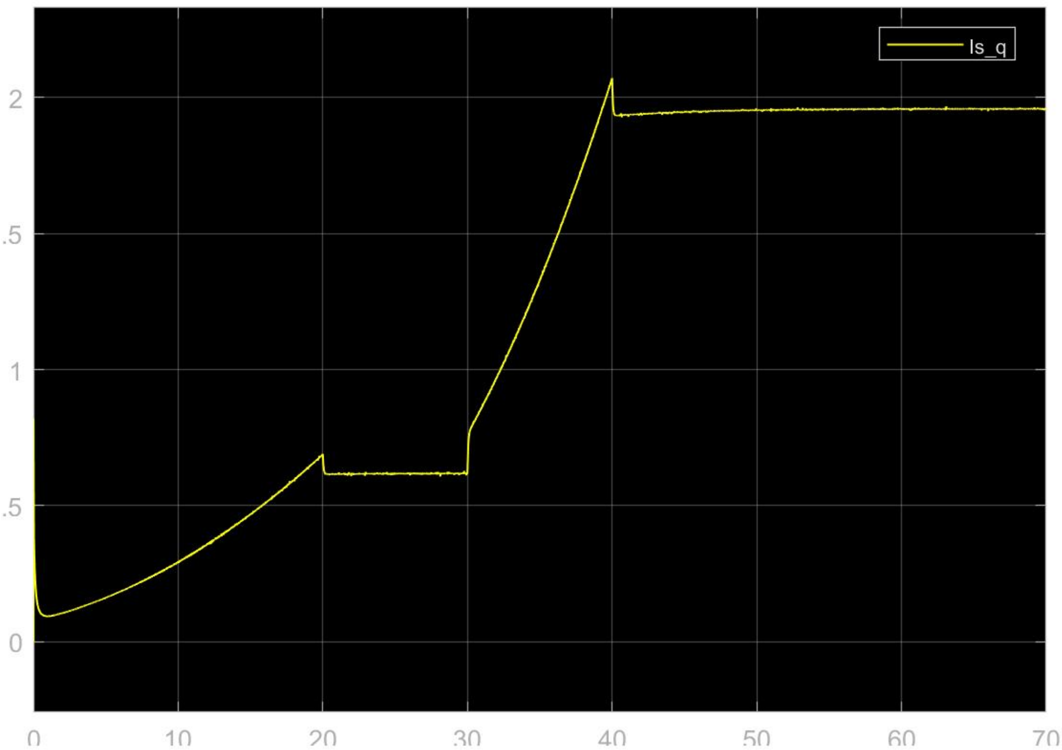
The input of the controller is the following reference speed profile, expressed in percentage with respect to the nominal motor speed Ω_b:



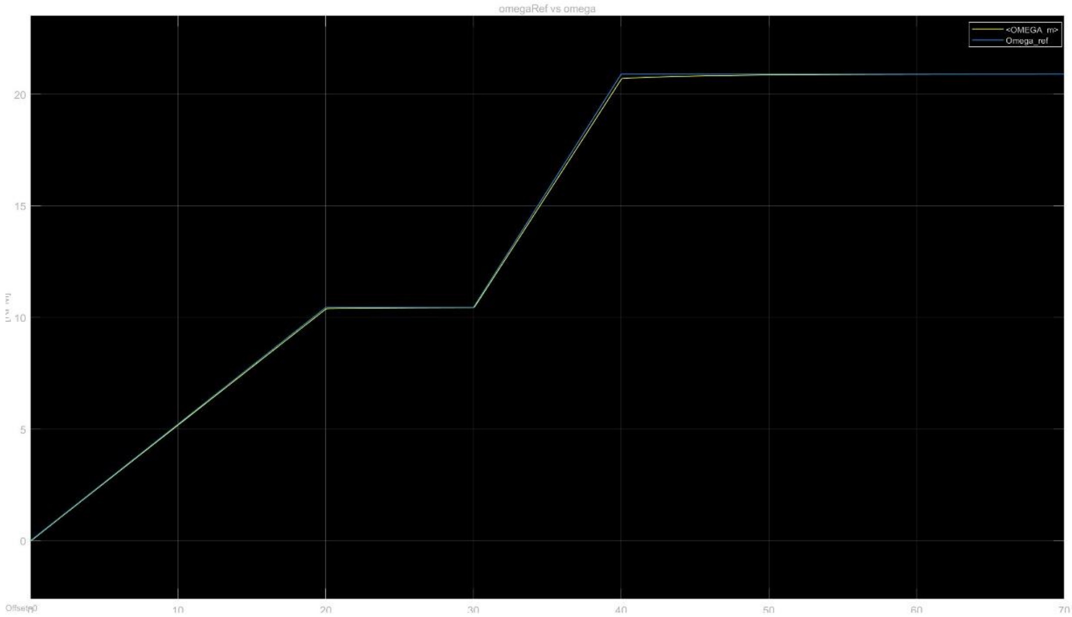
Outputs of the simulation are the following:

1. Direct and Quadrature Currents;

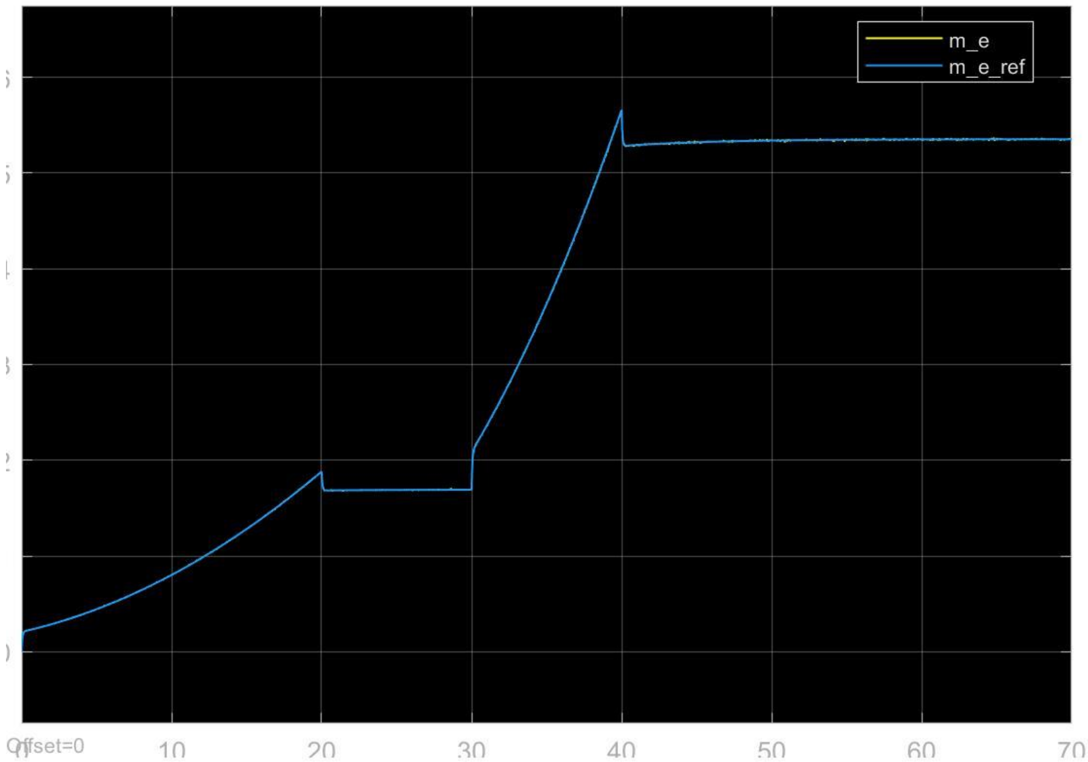




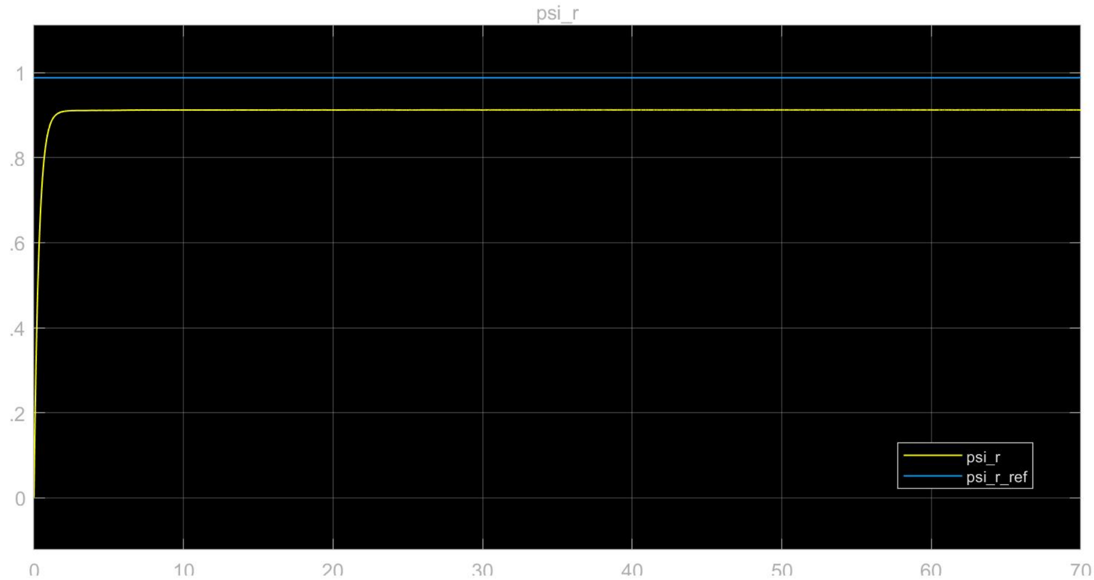
2. Mechanical Speed Ω_m vs Ω_m^{ref};



3. Electromagnetic Torque m_e vs m_e^{ref};



4. Rotor Flux ψ_r vs ψ_r^{ref}.



MATLAB SCRIPT

The following matlab script contains the declaration of the model's parameters and the computations to obtain controller parameters.

```
close all
clear all
%% Given Data
%Electrical
Vn = 380;
fn = 50;
np = 3;
Rs = 0.24;
Ls = 59.4e-3;
Rr = 0.175;
Lr = 59.1e-3;
Lm = 57e-3;
%Mechanical
r = 4;
J = 0.4;
beta = 0.068;
k = 0.009;
%% Computed Parameters
Lks = Ls - Lm^2/Lr;
Rks = Rs+Rr;
OMEGA_b = 104.5;
omega_n=2*pi*50/np;
m_e_n = k*omega_n^2+beta*omega_n;
psi_r_n = Vn*(sqrt2/sqrt3)/(2*pi*fn);
is_d_n = psi_r_n/Lm;
is_q_n = m_e_n/np/psi_r_d_n;
%% Controllers Parameters
ome_mecc=20; %taken from the speed profile and taking the most critical situation.
kp_ome=J*ome_mecc;
ki_ome=beta*ome_mecc;
ome_elec=2000; %cutoff frequency
ki_is=Lks*ome_elec;
kp_is=(Rs+Rr)*ome_elec;
ome_flux=200;
ki_psi=ome_flux/Rr;
kp_psi=ome_flux/Lm;
```