POLITECNICO DI TORINO

Master Thesis in Electrical Engineering

Advanced Dynamic Model of E-motor for Control Rapid Prototyping

Supervisor
Prof. Gianmario Pellegrino

Author
Andrei Bojoi

Advisors
Anantaram Varatharajan
Paolo Pescetto

28 March 2022

Acknowledgments

I would like to thank all the people who helped me during my thesis work and guided me with patience: Prof. Gianmario Pellegrino, Paolo Pescetto, Anantharam Varatharajan and Simone Ferrari.

A special thanks to my family, who always supported me during my university career.

Thanks also to all my friends, inside and outside the university, for all the happy moments shared together.

Contents

List of Figures

List of Acronyms

PMSM   Permanent Magnet Synchornous Machine

SPM   Surface Permanent Magnet

IPM   Internal Permant Magnet

PMASR   Permanet Magnet Assisted Synchonous Reluctance

SyR   Synchronous Reluctance

FEA   Finite Element Analysis

MTPA   Maximum Torque Per Ampere

MTPV   Maximum Torque Per Volts

CCG   Controlled Current Generators

VBR   Voltage Behind Reactance

EMF   Electro-Magnetic Force

ASC   Active Short Circuit

OPF   Open Phase Fault

Thesis outline

The thesis deals with the improvement of electric motor drives simulation for control calibration under healthy and faulty conditions, including the effects of PWM voltage and electric motor harmonics fields.

The design study of an electric motor (eMotor) starts with the electromagnetic design, using Finite Element Analysis for the evaluation of the motor output figures and parameters. This is the original purpose of the SyR-e (Synchronous Reluctance-evolution) open source design environment. More recently, the SyreDrive add-on was introduced, whose purpose is to generate a Simulink model for control calibration and accurate waveform simulation, starting from the eMotor design results.

Therefore, the main purpose of the thesis is to develop an unified circuital Simulink and PLECS model for eDrive (e-motor + inverter + control) suitable for both Permanent Magnet Synchronous Machines (PMSM) and Synchronous Reluctance (SyR) machines.

The starting point of the work is the non-circuital, discrete-time average Simulink model available in SyreDrive that is considered as the benchmark for comparison of the new findings. The benchmark main limitations are of being time averaged, thus neglecting the instantaneous PWM evolution; of being non-circuital, and therefore not compatible with the analysis of fault and uncontrolled scenarios; and of requiring the inversion of the flux map tables (dq currents function of dq flux linkages).

The goal of the thesis is to set up a new circuital model of the PMSM and the inverter, valid for instantaneous and time-averaged simulations, and covering faulty operating conditions of the inverter and motor sides. Two motor modeling approaches are considered, the Controlled Current Generators (CCG) model and the Voltage Behind Reactance (VBR) model, that have been comparatively implemented using the Simscape library of Simulink and in PLECS, using a torque control scheme, with evaluation of the respective computational times

An experimental validation is carried comparing the simulated and measured waveforms in steady state and controlled fault transient. Additionally the Simulink models were validated against FEA simulations in Magnet environment. At the end of the thesis, one of the two models will be embedded into public repository of SyR-e on GitHub.

The thesis is organized in the following chapters:

1. Chapter 1 will analyse the state of art of motor modelling, concluding with the motivations of the work;

2. Chapter 2 will present the implementation of the existing motor models in Simulink and PLECS, concluding with the new proposed models: the CCG and VBR models;

3. Chapter 3 will illustrate the simulation results in Simulink and PLECS, provided by the CCG and VBR models;

4. Chapter 4 will illustrate the comparison between the results obtained by experimental tests and by CCG and VBR models in Simulink;

5. Chapter 5 will present the conclusions of the thesis.

Chapter 1

Introduction and motivations

In this chapter, the theory behind motor modeling is presented, starting from models found in literature. After that, *SyreDrive* program tool is described, concluding with the motivations and goal of the thesis.

1.1 Literature review

In [1], the analysis of motor modeling is made considering a three phase synchronous machine with the following structure: the stator has three armature windings and the rotor has one field winding and one damper winding in *d-axis* and two damper windings in *q-axis*.

For the purpose of this thesis, the analysis is adapted to permanent magnet synchronous machines (PMSM) and synchronous reluctance (SyR) machines, which are presented in Fig.1.1.

Surface mounted   V-Type Interior   PM-assisted   Synchronous
PM (SPM)   mounted PM (IPM)   SyR (PMASR)   reluctance (SyR)
motor   motor   motor   motor
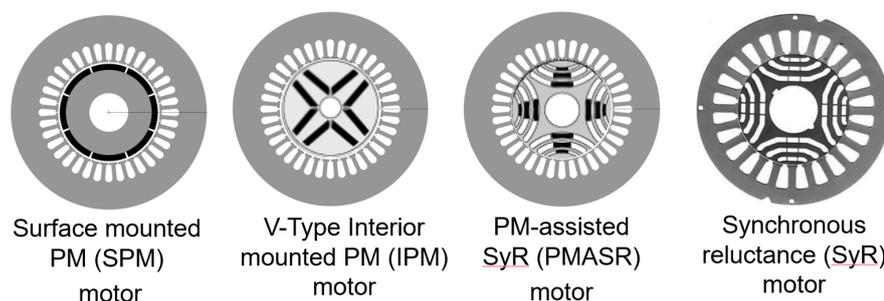
Figure 1.1: Permanent magnet synchronous machines (PMSM) and synchronous reluctance (SyR) machine.

In the rotor reference frame, the position of the dq-axes is defined by two conventions, which are presented in Fig.1.2:

(a) **PM convention**: the d-axis is oriented along the direction of minimum inductance of the rotor ($L_d < L_q$); it is generally used for SPM ($L_d = L_q$) and IPM motors and the magnet flux vector λ_m is oriented along the d-axis;

(b) **SyR convention**: the d-axis is oriented along the direction of maximum inductance of the rotor ($L_d > L_q$); it is generally used for PMASR and SyR motors; for PMASR motors the magnet flux vector λ_m is oriented along the direction of negative q-axis

For all motors, θ is the electrical angle between the d-axis and the α-axis of the stationary reference frame.
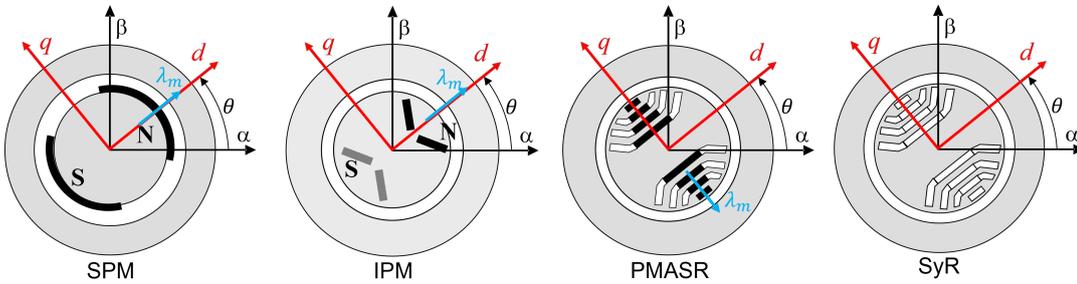
Figure 1.2: Convensions for the position od the dq axes: PM convention (for SPM and IPM) and SyR convention (for PMASR and SyR).

From the literature, the employed motor models are:

(a) **The *abc* dynamic model** ([1],[2]);

(b) **The *dq* dynamic model** ([1]);

(c) **The *dqθ* dynamic model** ([2]);

(d) **The Voltage Behind Reactance model** ([1]).

1.1.1   The *abc* dynamic model

The *abc* dynamic model (or Phase Domain model) represents the machine physical quantities in *abc* phase coordinates. The voltage equation is expressed as:

$$\overline{v}_{abc} = R_s \cdot \overline{i}_{abc} + \frac{d\overline{\lambda}_{abc}}{dt} \tag{1.1}$$

and the flux linkage equation is expressed as:

$$\overline{\lambda}_{abc} = L(\theta_r) \cdot \overline{i}_{abc} + \lambda_m(\theta_r) \quad (SPM,IPM,PMASR)$$

$$\overline{\lambda}_{abc} = L(\theta_r) \cdot \overline{i}_{abc} \qquad (SyR) \tag{1.2}$$

where $L(\theta_r)$ is the self ad mutual stator inductance 3x3 matrix, depending on the mechanical rotor position θ_r. The torque equation is expressed as:

$$T_{em} = p \left(\frac{1}{2} \cdot \overline{i}_{abc}^T \cdot \frac{\partial L(\theta_r)}{\partial \theta_r} \cdot \overline{i}_{abc} + \overline{i}_{abc}^T \cdot \lambda_m(\theta_r) \right) \quad (SPM,IPM,PMASR)$$

$$T_{em} = p \left(\frac{1}{2} \cdot \overline{i}_{abc}^T \cdot \frac{\partial L(\theta_r)}{\partial \theta_r} \cdot \overline{i}_{abc} \right) \qquad (SyR) \tag{1.3}$$

However, the time variant inductance matrix complicates the model and requires high computational burden, so more simplified motor models are utilized.

1.1.2 The *dq* dynamic model of the machine

The *dq* model represents the voltage equations in the rotating *dq* rotor frame as follows:

$$\overline{v}_{dq} = R_s \cdot \overline{i}_{dq} + \frac{d\overline{\lambda}_{dq}}{dt} + [J] \cdot \omega \cdot \overline{\lambda}_{dq} \tag{1.4}$$

where $[J] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, ω is the electrical frequency [rad/s] and R_s is the stator resistance. The flux linkage equations of the machines are given as:

$$\overline{\lambda}_{dq} = \begin{bmatrix} L_{dd} & L_{dq} \\ L_{qd} & L_{qq} \end{bmatrix} \cdot \overline{i}_{dq} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \qquad (SPM \text{ and } IPM)$$

$$\overline{\lambda}_{dq} = \begin{bmatrix} L_{dd} & L_{dq} \\ L_{qd} & L_{qq} \end{bmatrix} \cdot \overline{i}_{dq} + \begin{bmatrix} 0 \\ -\lambda_m \end{bmatrix} \qquad (PMASR) \tag{1.5}$$

$$\overline{\lambda}_{dq} = \begin{bmatrix} L_{dd} & L_{dq} \\ L_{qd} & L_{qq} \end{bmatrix} \cdot \overline{i}_{dq} \qquad (SyR)$$

where L_{dd}, L_{dq}, L_{qd} and L_{qq} are the apparent inductances, defined as:

$$L_{dd} = \frac{\lambda_d(i_d, i_q = 0)}{i_d} \qquad L_{dq} = \frac{\lambda_d(i_d, i_q) - \lambda_d(i_d, i_q = 0)}{i_q}$$

$$L_{qd} = \frac{\lambda_q(i_d, i_q) - \lambda_q(i_d = 0, i_q)}{i_d} \qquad L_{qq} = \frac{\lambda_d(i_d = 0, i_q)}{i_q} \tag{1.6}$$

The unified electromagnetic torque expression is described as:

$$T_{em} = \frac{3}{2} \cdot p \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \tag{1.7}$$

where p is the pole pairs of the machine. In particular, substituting (1.5) in (1.7) and neglecting the cross saturation inductances ($L_{dq} = L_{qd} = 0$), the torque expressions adapted to the motor type are obtained:

$$T_{em} = \frac{3}{2} \cdot p \cdot \lambda_m \cdot i_q \qquad\qquad (SPM)$$

$$T_{em} = \frac{3}{2} \cdot p \cdot [\lambda_m \cdot i_q + (L_d - L_q) \cdot i_d \cdot i_q] \quad (IPM)$$

$$T_{em} = \frac{3}{2} \cdot p \cdot [\lambda_m \cdot i_d + (L_d - L_q) \cdot i_d \cdot i_q] \quad (PMASR) \tag{1.8}$$

$$T_{em} = \frac{3}{2} \cdot p \cdot [(L_d - L_q) \cdot i_d \cdot i_q] \qquad\qquad (SyR)$$

The block diagram of the dq model, referred to (1.4) and (1.7), is presented in Fig.1.3.
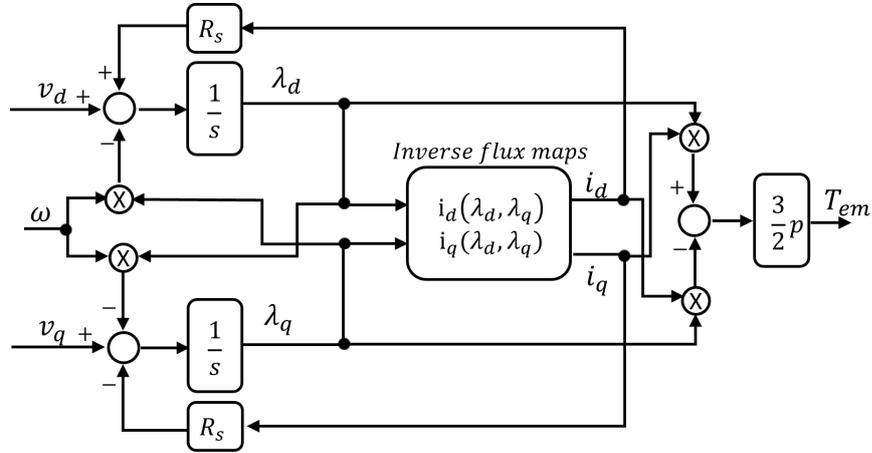
Figure 1.3: Block scheme of the dq model.

The inverse flux maps $\boldsymbol{i_{dq}(\lambda_{dq})}$ represent the maps of current function of flux linkages. The data is stored in 2D lookup tables, which are obtained by analytical inversion of the direct flux maps $\boldsymbol{\lambda_{dq}(i_{dq})}$ (flux linkage expressed in terms of dq

currents). These maps are referred as fundamental flux maps, which contain the average values of the flux linkages with respect to the rotor position angle. An example of inverse flux maps, referred to a SyR machine, is shown in Fig.1.4([2]).
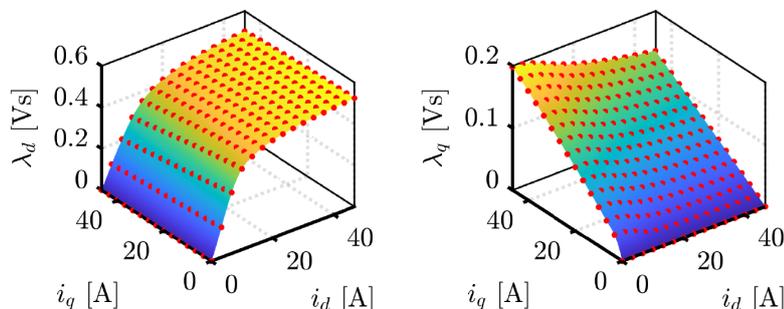
Figure 1.4: Example of inverse flux maps of a SyR machine ([2]).

1.1.3   The $dq\theta$ dynamic model of the machine

The $dq\theta$ model represents a more general approach of the dq model. The block scheme of the model is presented in Fig.1.5, which is very similar to Fig.1.3, except for the inverse flux maps. In these maps, the currents are function of flux linkage and rotor position and the data is stored in 3D lookup tables.
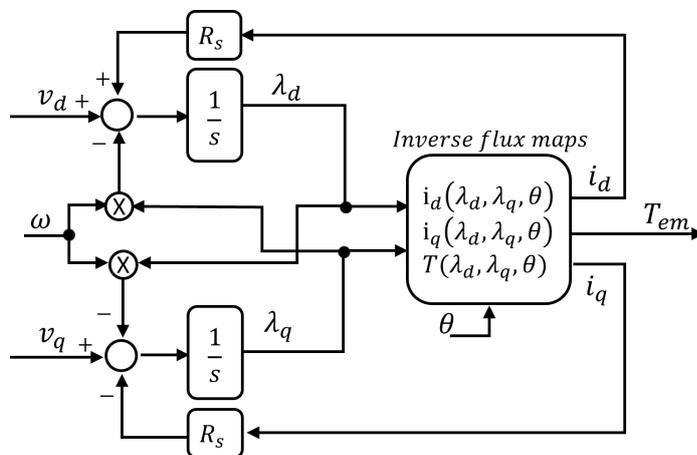
Figure 1.5: Block diagram of the $dq\theta$ model.

The additional information of the rotor position allows to represent the effects of space harmonics due to the influence of the slots. Because of information loss of the rotor position, the fundamental dq model represents average waveform of the

flux linkages and torque, while the $dq\theta$ represents instantaneous waveform of flux linkage and torque. An example of the d-axis flux linkage function of i_d, i_q and θ is shown in Fig.1.6([2]), where the 3D lookup table is composed by 2D lookup tables stacked together.
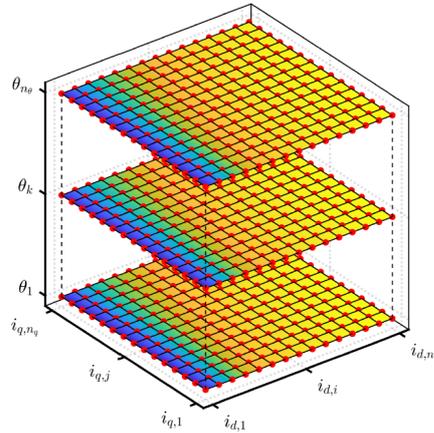
Figure 1.6: Example of a 3D lookup table, representing the d-axis flux linkage function of i_d, i_q and θ ([2]).

1.1.4   Voltage Behind Reactance model

The Voltage Behind Reactance model is a combination of the dq model and abc model, where the machine variables are expressed in abc phase coordinates starting from the dq dynamic model. In Fig.1.7, the machine is modeled like a RLE load, with 3 coupled inductors and 3 controlled voltage generators representing the back EMFs.
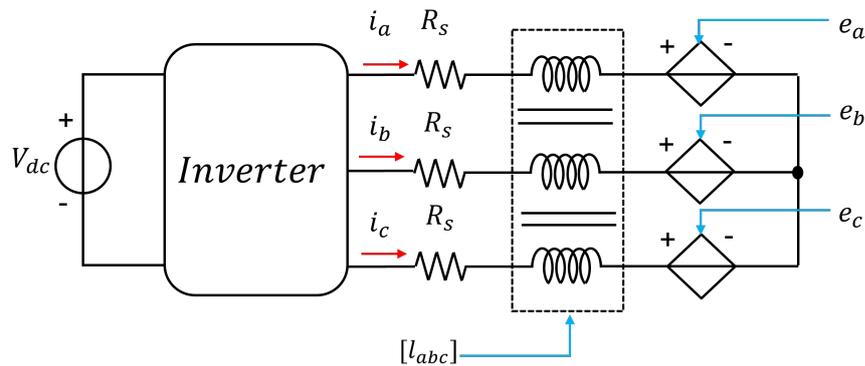
Figure 1.7: Voltage Behind Reactance circuit model.

The matrix $[l_{abc}]$ is the 3x3 tensor of incremental inductances and e_a, e_b, e_c are the back EMFs. From the voltage equation (1.4), the tensor of incremental inductances is defined as:

$$\frac{d\overline{\lambda}_{dq}}{dt} = \frac{\partial\overline{\lambda}_{dq}}{\partial\overline{i}_{dq}} \cdot \frac{d\overline{i}_{dq}}{dt} \implies \frac{\partial\overline{\lambda}_{dq}}{\partial\overline{i}_{dq}} = [l_{dq}] = \begin{bmatrix} l_{dd} & l_{dq} \\ l_{qd} & l_{qq} \end{bmatrix} \tag{1.9}$$

where the incremental inductances are defined as:

$$\begin{aligned} l_{dd} &= \frac{\partial\lambda_d(i_d, i_q)}{\partial i_d} & l_{dq} &= \frac{\partial\lambda_d(i_d, i_q)}{\partial i_q} \\ l_{qd} &= \frac{\partial\lambda_q(i_d, i_q)}{\partial i_d} & l_{qq} &= \frac{\partial\lambda_q(i_d, i_q)}{\partial i_q} \end{aligned} \tag{1.10}$$

Substituting (1.9) in (1.4), the voltage equation is expressed as follows:

$$\overline{v}_{dq} = R_s \cdot \overline{i}_{dq} + [l_{dq}] \cdot \frac{d\overline{i}_{dq}}{dt} + [J] \cdot \omega \cdot \overline{\lambda}_{dq} \tag{1.11}$$

This voltage equation is then transformed from dq-axes to abc frame. The full mathematical demonstration is found in Appendix A, but in this section the main results are presented. First, (1.11) is transformed in $\alpha\beta$-axes by applying the inverse rotational matrix:

$$A(-\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \tag{1.12}$$

The result is the voltage equation in $\alpha\beta$-axes:

$$\overline{v}_{\alpha\beta} = R_s \cdot \overline{i}_{\alpha\beta} + [l_{\alpha\beta}] \cdot \frac{d\overline{i}_{\alpha\beta}}{dt} + \cdot\overline{e}_{\alpha\beta} \tag{1.13}$$

where $[l_{\alpha\beta}]$ and $e_{\alpha\beta}$ are define as:

$$[l_{\alpha\beta}] = A(-\theta) \cdot [l_{dq}] \cdot A(\theta) = \begin{bmatrix} l_{\alpha\alpha} & l_{\alpha\beta} \\ l_{\alpha\beta} & l_{\beta\beta} \end{bmatrix} \tag{1.14}$$

$$\overline{e}_{\alpha\beta} = A(-\theta) \cdot \left([l_{dq}] \cdot (-\omega) \cdot [J] \cdot \overline{i}_{dq} + [J] \cdot \omega \cdot \overline{\lambda}_{dq} \right) = A(-\theta) \cdot \overline{e}_{dq} \tag{1.15}$$

The elements of the tensor $[l_{\alpha\beta}]$ are defined as follows:

$$[l_{\alpha\beta}] = \begin{bmatrix} l_{avg} + l_\Delta cos(2\theta) - l_{dq}sin(2\theta) & l_{dq}cos(2\theta) + L_\Delta sin(2\theta) \\ l_{dq}cos(2\theta) + L_\Delta sin(2\theta) & l_{avg} - l_\Delta cos(2\theta) + l_{dq}sin(2\theta) \end{bmatrix} \tag{1.16}$$

where l_{avg} and l_Δ are defined as:

$$l_{avg} = \frac{l_{dd} + l_{qq}}{2} \quad l_\Delta = \frac{l_{dd} - l_{qq}}{2} \tag{1.17}$$

After that, (1.13) is transformed in *abc*-axes using the inverse transformation:

$$[T]^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \tag{1.18}$$

obtaining the voltage equation in *abc* frame:

$$\overline{v}_{abc} = R_s \cdot \overline{i}_{abc} + [l_{abc}] \cdot \frac{d\overline{i}_{abc}}{dt} + \cdot\overline{e}_{abc} \tag{1.19}$$

where $[l_{abc}]$ and $\overline{e}_{abc}$ are define as:

$$[l_{abc}] = [T]^{-1} \cdot \begin{bmatrix} l_{\alpha\alpha} & l_{\alpha\beta} & 0 \\ l_{\alpha\beta} & l_{\beta\beta} & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot [T] = \begin{bmatrix} l_{aa} & l_{ab} & l_{ac} \\ l_{ab} & l_{bb} & l_{bc} \\ l_{ac} & l_{cb} & l_{cc} \end{bmatrix} \tag{1.20}$$

$$\overline{e}_{abc} = [T]^{-1} \cdot \overline{e}_{\alpha\beta} \tag{1.21}$$

The elements of the tensor $[l_{abc}]$ are defined as follows:

$$l_{aa} = \frac{2}{3} \qquad\qquad l_{ab} = l_{ba} = \frac{1}{3}\left[-l_{\alpha\alpha} + \sqrt{3}l_{\alpha\beta} \right]$$

$$l_{bb} = \frac{1}{3}\left[\frac{l_{\alpha\alpha}}{2} + \frac{3}{2}l_{\beta\beta} - \sqrt{3}l_{\alpha\beta} \right] \quad l_{ac} = l_{ca} = \frac{1}{3}\left[-l_{\alpha\alpha} - \sqrt{3}l_{\alpha\beta} \right] \tag{1.22}$$

$$l_{cc} = \frac{1}{3}\left[\frac{l_{\alpha\alpha}}{2} + \frac{3}{2}l_{\beta\beta} + \sqrt{3}l_{\alpha\beta} \right] \quad l_{bc} = l_{cb} = \frac{1}{3}\left[\frac{l_{\alpha\alpha}}{2} - \frac{3}{2}l_{\beta\beta} \right]$$

1.2   SyreDrive

The design study of a traction eMotor for eMobility involves two main steps:

1. **Step 1**: the electromagnetic design of the motor using design procedures followed by optimization and design check with Finite Element Analysis (FEA);

2. **Step 2**: the simulation of the eDrive (e-motor + inverter + control) for proper torque control calibration, including MTPA and MTPV for flux weakening speed ranges.

Step 1 is the original purpose of the *Syr-e* (Synchronous Reluctance-evolution), an open source design environment developed in Matlab/Octave, initially intended only for the motor design of synchronous reluctance machines. The mode of operation of *Syr-e* is shown in Fig.1.8. A Matlab script realizes a drawing of the synchronous machine, which is sent to software FEMM, for static magnetic FEA. Then, the results are given back to Matlab for performance evaluation.
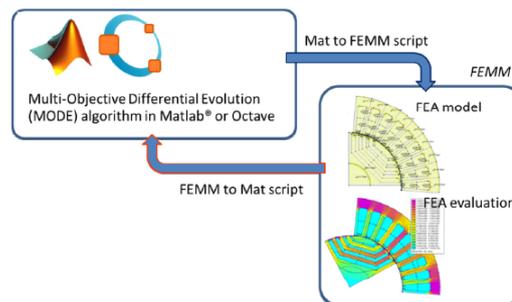
Figure 1.8: Flow Chart of *Syr-e*.

To address also Step 2, *Syr-e* was recently extended with SyreDrive, a software tool intended to be a bridge between the design environment and the control environment. The purpose of SyreDrive is to generate a Simulink model for control calibration and accurate waveform simulation, starting from the results of Step 1. The mode of operation of SyreDrive is showed in Fig1.9. The motor data obtained from the FEA evaluation is exported to Matlab/Simulink, which automatically adapts the motor model and the control code with the data of the designed machine.
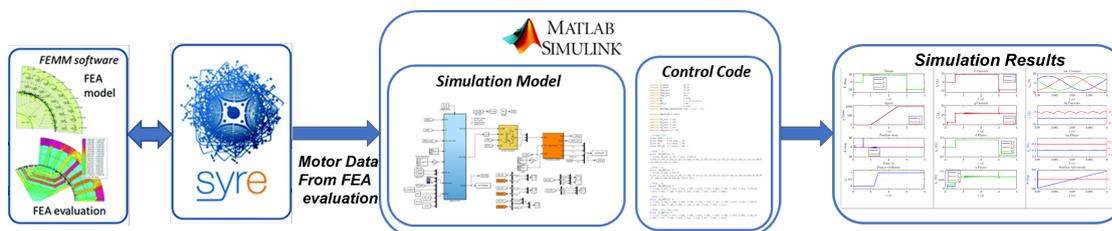
Figure 1.9: Flow chart of SyreDrive.

The user interface of SyreDrive is represented in Fig.1.10. From the tab the user

can select:

(a) Control type: current, torque or speed control;

(b) Motor model type: fundamental (based on dq model) or with harmonics (based on $dq\theta$ model);

(c) Converter data: threshold voltage and internal resistance of the power modules and the dead time;

(d) Sensorless control type if needed.
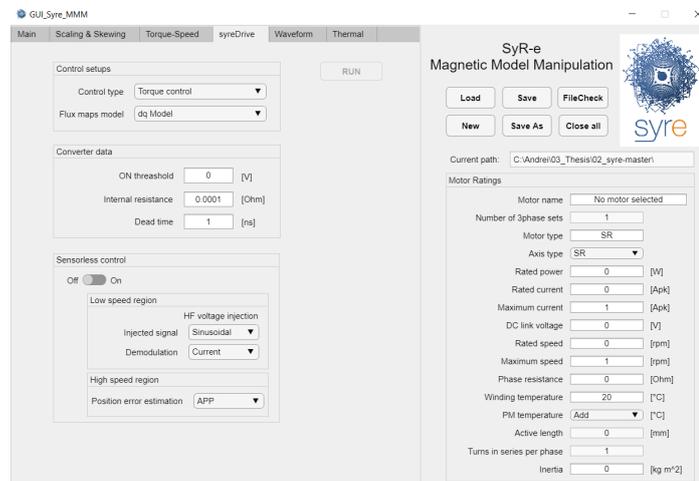
Figure 1.10: SyreDrive user interface.

When the "RUN" button is pressed, SyreDrive generates the Simulink model represented in Fig.1.11. This will be considered the benchmark and the starting point of the thesis. The benchmark model is made of three main blocks:

1. **Digital Control**: includes a discrete-time executed C-script, with configurable control;

2. **Inverter**: average non-circuital model accounting for voltage drops;

3. **Motor Model**: continuous time, voltage fed non-circuital model, using dq or $dq\theta$ model.
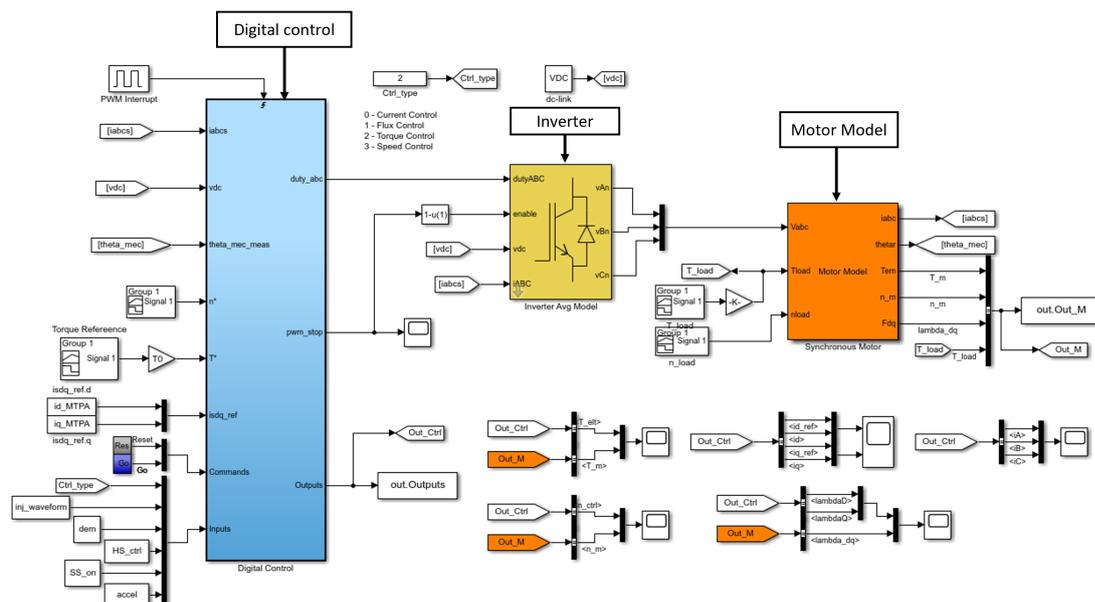
Figure 1.11: Simulink benchmark model generated by SyreDrive.

1.3   Motivation and goal of the thesis

The benchmark has four main limitations:

1. It is only time averaged so it neglects the instantaneous PWM evolution;

2. Being a non-circuital model, it cannot be used for the analysis of fault and uncontrolled scenarios;

3. The inverter voltage drops are simulated through its analytical model, it is not represented by physical components;

4. It requires the flux-map tables inversion, which limits the operating domain respect to the direct flux-map. Starting from the rectangular mesh (i_d, i_q) of the direct flux maps, the rectangular mesh (λ_d, λ_q) of the de inverse flux maps is reduced ([2]), resulting in loss of information.

Therefore, the goal of the thesis is to develop an unified circuital model for eDrive, implemented in Simulink (using the Simscape library) and PLECS. The model must be suitable for both Permanent Magnet Synchronous Machine (PMSM) and Synchronous Reluctance (SyR) machines. The thesis output will be integrated in SyreDrive, substituting the benchmark.

The main motivation of the thesis is to overcome the limitations of the bench-

mark by:

1. Building a general circuital eDrive model, which can be used for both instantaneous-time and average-time simulations;

2. Developing an appropriate tool for the simulation of faults and uncontrolled scenarios.

The validation of the Simulink model as an accurate tool to implemented in SyreDrive is made by comparison of simulation results and experimental results, obtained using an automotive IPM. The experimental validation is made of two steps:

1. Comparison of the phase current ripple in normal operating condition;

2. Comparison of the phase currents during fault scenarios.

In addition, will be added a comparison between Simulink and Magnet, a software for Finite Element Analysis.

Chapter 2

Simulink and PLECS motor models

This chapter presents the implementation of different motor models in Simulink and PLECS. First, the Simulink benchmark model is described, then the available motor models in Simulink/Simscape library and PLECS library are presented. At the end, two new developed motor models are going to be presented and implemented both in PLECS and Simulink.

2.1   SyreDrive time averaged benchmark model

The motor model implemented in the Simulink benchmark model is presented in Fig.2.1, which is placed inside the Motor Model block in Fig.1.11. The model is made by two main blocks:

1. **Electrical equations**: the inputs are the output voltages V_{abc} of the inverter and the mechanical rotor position θ_r and the outputs are the λ_{dq} flux linkages, the phase currents I_{abc} and the electromagnetic torque T_{em};

2. **Mechanical equations**: the inputs are the electromagnetic torque T_{em}, the load torque T_{load} and the load speed $n_{load}(rpm)$ and the outputs are mechanical position θ_r and the motor speed $n_m(rad/s)$.
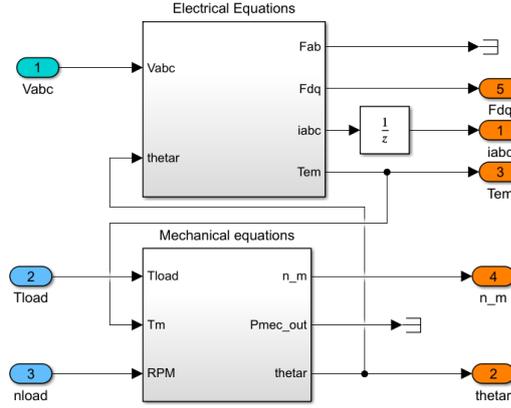
Figure 2.1: Simulink benchmark motor model.

The behavior of the mechanical equations block depends on the type of control:

1. **Current and torque control**: the outputs of the block are:

$$\theta_r = \frac{1}{s} \cdot n_{load} \cdot \frac{\pi}{30} \quad n_m = \frac{\pi}{30} \cdot n_{load} \tag{2.1}$$

2. **Speed control**: the outputs of the block are:

$$n_m = \frac{1}{s} \cdot \left(\frac{T_m - T_{load} - T_{mech_loss}}{J} \right) \quad \theta_r = \frac{1}{s} \cdot n_m \tag{2.2}$$

where $T_{mech_loss} = n_m \cdot B_m + (|n_m|^2 \cdot T_v + T_f) \cdot sgn(n_m)$, depending on the damping constant B_m, the friction torque T_f and the ventilation loss coefficient T_v. The electrical equations block is showed in Fig.2.2, which is made by two main blocks:

- **Stator equations**: the inputs are the phase voltages $v_{\alpha\beta}$ and phase currents $i_{\alpha\beta}$ and the outputs are the fluxes $\lambda_{\alpha\beta}$, calculated using:

$$\overline{\lambda}_{\alpha\beta} = \frac{1}{s} \cdot (\overline{v}_{\alpha\beta} - R_s \cdot \overline{i}_{\alpha\beta}) \tag{2.3}$$

- **Magnetic model**: the inputs are the fluxes λ_{dq} and the rotor position θ_r and the outputs are the currents i_{dq} and the electromagnetic torque T_{em}. The magnetic model can use both the inverse dq or inverse $dq\theta$ model, using the inverse magnetic model as shown in Fig.2.3a and Fig.2.3b.
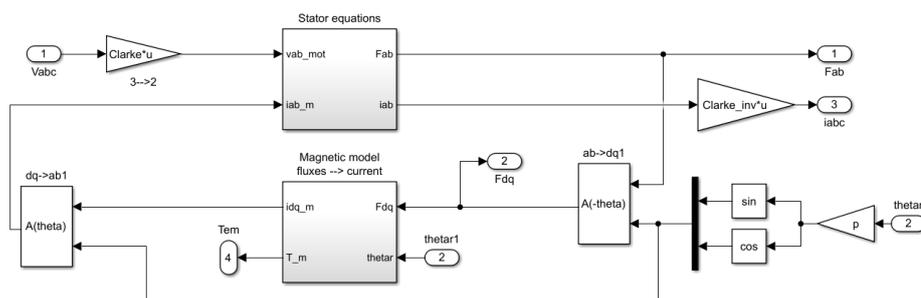
Figure 2.2: Simulink benchmark motor model.

(a) Inverse dq magnetic model.

(b) Inverse $dq\theta$ magnetic model.
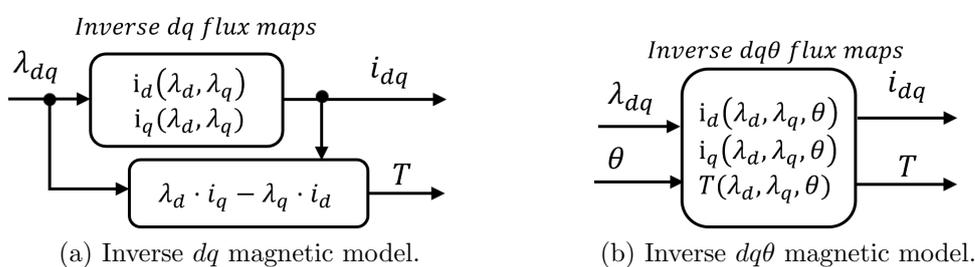
Figure 2.3: Inverse magnetic model.

The output voltage of the inverter are defined as:

$$v_{An} = \frac{2}{3} \cdot V_A - \frac{1}{3} \cdot (V_B + V_C)$$

$$v_{Bn} = \frac{2}{3} \cdot V_B - \frac{1}{3} \cdot (V_A + V_C) \tag{2.4}$$

$$v_{Cn} = \frac{2}{3} \cdot V_C - \frac{1}{3} \cdot (V_A + V_B)$$

where V_A, V_B, V_C are defined as:

$$\begin{aligned} V_A &= V_{DC} \cdot (duty_A - 0.5) - v_{errorA} \\ V_B &= V_{DC} \cdot (duty_B - 0.5) - v_{errorB} \\ V_C &= V_{DC} \cdot (duty_C - 0.5) - v_{errorC} \end{aligned} \tag{2.5}$$

and $v_{errorA,B,C}$ are the voltage error caused by the dead time and the voltage drop of the inverter components.

2.2 Existing Simulink models

This section presents two blocks of interest from the Simscape library:

1. Synchronous reluctance machine ([3]);

2. Permanent magnet synchronous machine ([4]).

2.2.1 Synchronous Reluctance Machine

The Synchronous Reluctance Machine block represents a SyR motor with sinusoidal flux distribution. The block and the user tab are presented in Fig.2.4.
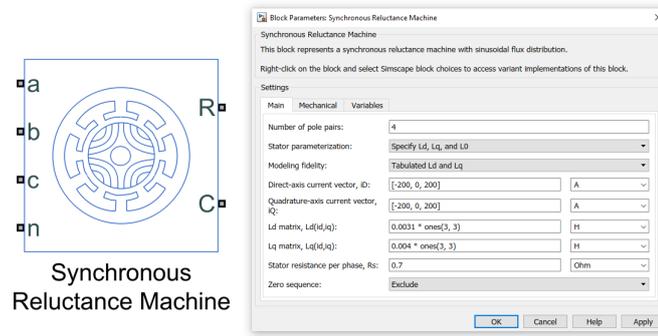
Figure 2.4: Synchronous Reluctance machine in Simscape.

The voltage equations in dq frame are expressed as follows:

$$v_d = R_s \cdot i_d + L_d \cdot \frac{di_d}{dt} - \omega \cdot i_q \cdot L_q$$

$$v_q = R_s \cdot i_q + L_q \cdot \frac{di_d}{dt} + \omega \cdot i_d \cdot L_d$$

(2.6)

where L_d and L_q are the apparent inductances of the machine. These equations are simplified because they don't take into account the incremental inductances.

The main data required are:

- direct axis current vector i_d;

- quadrature axis current vector i_q;

- L_d matrix $L_d(i_d, i_q)$;

- L_q matrix $L_d(i_d, i_q)$;

2.2.2   FEM Parameterized PMSM

This component represents a permanent magnet synchronous motor (PMSM) in terms of magnetic flux linkage. The block is parameterized by providing the magnetic flux as tabulated data, function of currents and rotor angle (dq or $dq\theta$ model). The machine uses the PM convention, with the magnet flux oriented along the d-axis. In Fig.2.5 is shown the Simscape symbol and the user tab for the component settings.
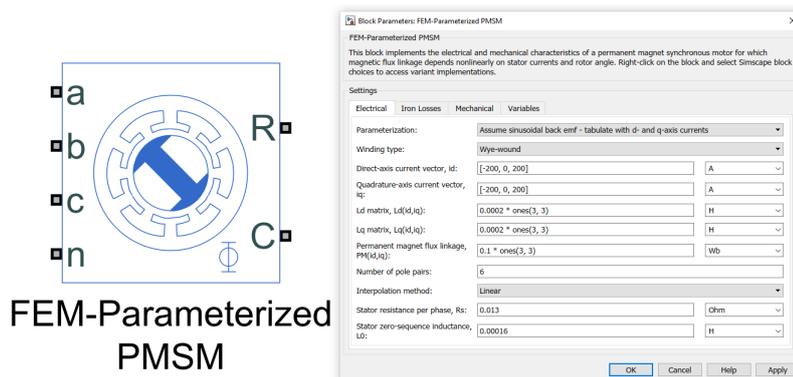
Figure 2.5: FEM-Parametrized PMSM in Simscape.

The user can choose between four types of parametrization:

1. 2D partial derivative data;

2. 3D partial derivative data;

3. 4D partial derivative data;

4. 3D flux linkage data.

2D partial derivative data   The model is providing two settings:

1. **Constant Mutual Inductance**: the flux linkage of a phase linearly depends on the currents of the other two phases, using constant mutual parameters and non linearly depend o the current of the same phase, as shown in the expression below:

$$
\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} 0 & -M_s & -M_s \\ -M_s & 0 & -M_s \\ -M_S & -M_s & 0 \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \lambda_a(i_a, \theta_r) \\ \lambda_b(i_b, \theta_r - \frac{2\pi}{3p}) \\ \lambda_c(i_c, \theta_r + \frac{2\pi}{3p}) \end{bmatrix} \tag{2.7}
$$

2. **Sinusoidal back EMFs**: the permanent magnet flux linkage is considered sinusoidal and the flux linkage is considered non linearly depend on all three currents, as shown in the expression below:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d(i_d, i_q) & \\ & L_q(i_d, i_q) \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m(i_d, i_q) \end{bmatrix} \tag{2.8}$$

3D partial derivative data   The flux linkage depends on the direct and quadrature currents and the rotor angle, as shown below:

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} \lambda_a(i_d, i_q, \theta_r) \\ \lambda_b(i_d, i_q, \theta_r - \dfrac{2\pi}{3p}) \\ \lambda_c(i_d, i_q, \theta_r + \dfrac{2\pi}{3p}) \end{bmatrix} \tag{2.9}$$

4D partial derivative data   The flux linkage depend on the three phase currents and rotor position angle, as shown below:

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} \lambda_a(i_a, i_b, i_c, \theta_r) \\ \lambda_b(i_a, i_b, i_c, \theta_r - \dfrac{2\pi}{3p}) \\ \lambda_c(i_a, i_b.i_c, \theta_r + \dfrac{2\pi}{3p}) \end{bmatrix} \tag{2.10}$$

3D Flux linkage data   This setting do not need to calculate partial derivatives, instead it can receive the flux linkage data in four formats:

- dq axes flux linkages as a function of i_{dq} currents and rotor angle;

- dq axes flux linkages as a function of peak current magnitude, current advance angle and rotor angle;

- a-phase flux linkage as a function of i_{dq} currents and rotor angle;

- a-phase flux linkage as a function of peak current magnitude, current advance angle, and rotor angle.

2.3 Existing PLECS models

This section presents two block of interest in PLECS library:

1. Non-Excited Synchronous Machine ([5]);

2. Permanent Magnet SM ([6]).

2.3.1   Non-Excited Synchronous Machine

This component represents a three-phase synchronous machine with optional permanent magnets in the rotor. The component is able to simulate magnetization, saliency, saturation and cross coupling by using the Flux Maps and the Inductance Maps. The PLECS symbol and the user tab are presented in Fig.2.6
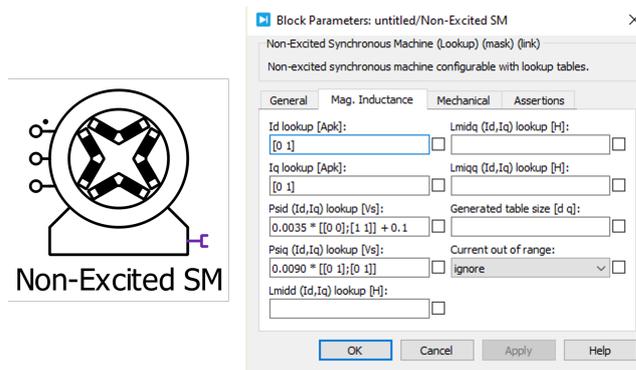

Figure 2.6: Non Exited Synchronous machine in PLECS

The electrical sistem is modeled using the Voltage Behind Reactance model, showed in Fig.2.7.
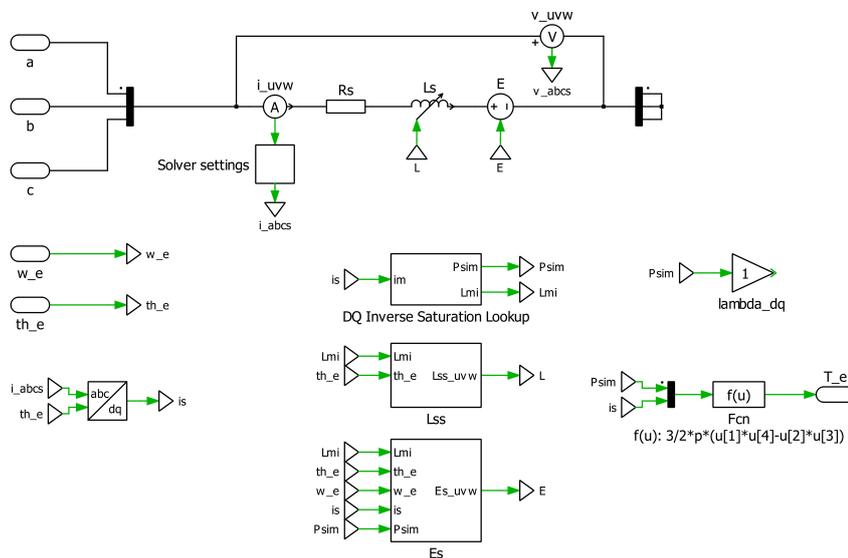

Figure 2.7: Voltage Behind Reactance model of the Non-Excited Synchronous Machine in PLECS.

The electrical equation in dq frame using space-vector notation:

$$\overline{v}_s = R_s \cdot \overline{i}_s + (\boldsymbol{L_{\sigma s}} + \boldsymbol{L_{mi}}) \cdot \left(\frac{d\overline{i}_s}{dt} + j \cdot \omega \cdot \overline{i}_s \right) + \overline{e}_s \qquad (2.11)$$

$$\overline{e}_s = j \cdot \omega \cdot \lambda_m - \boldsymbol{L_{mi}} \cdot j\omega \cdot \overline{i}_s \qquad (2.12)$$

where $\boldsymbol{L_{\sigma s}}$ is the 2x2 matrix of leakage inductances and $\boldsymbol{L_{mi}}$ the 2x2 matrix of incremental inductances.

The component requires the following input data:

(a) I_d lookup vector: d-axis current vector;

(b) I_q lookup vector: q-axis current vector;

(c) $\lambda_d(I_d, I_q)$ lookup table: d-axis flux linkage lookup table;

(d) $\lambda_q(I_d, I_q)$ lookup table: q-axis flux linkage lookup table;

(e) $L_{dd}(I_d, I_q)$ look up table: d-axis self incremental inductance lookup table;

(f) $L_{qq}(I_d, I_q)$ look up table: q-axis self incremental inductance lookup table;

(g) $L_{dq}(I_d, I_q)$ look up table: mutual incremental inductance lookup table.

The I_d and I_q currents represents the inputs to the flux linkage and incremental inductance lookup tables, whose number of rows and columns must match the size of I_d and I_q

2.3.2   Permanent Magnet SM

The block represents a three-phase permanent magnet synchronous machine with sinusoidal back EMF. The flux of the magnet is oriented along the direction of the d-axis. The machine model offers two different implementations the electrical system:

(a) Rotor Reference Frame;

(b) Voltage Behind Reactance model;
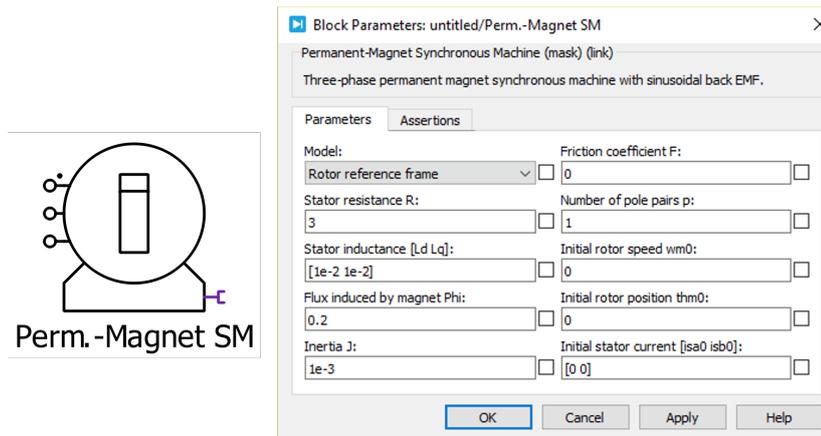
Figure 2.8: Permanent Magnet Synchronous machine in PLECS.

Rotor Reference Frame model

In this model, shown in Fig.2.9, the motor is modeled with two controlled current generators. The three phase physical variables are transformed in dq rotor reference frame.
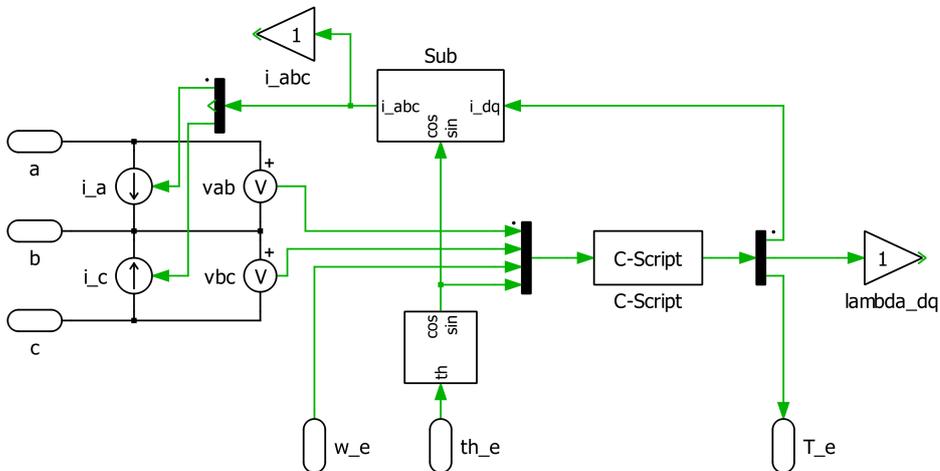
Figure 2.9: Rotor Reference Frame model of the Permanent Magnet SM in PLECS.

The block scheme for the computation of the currents to pilot the current generators is shown in Fig.2.10.
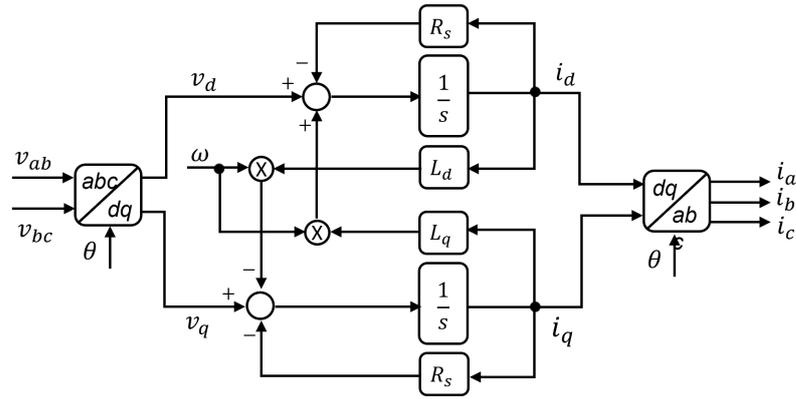
Figure 2.10: Sceme block of the Rotor Reference Frame model of the Permanent Magnet SM in PLECS.

Voltage Behind Reactance model

The VBR model was already introduced in section 1.1.4, however this model is simplified because it does not take in to account the coupled inductances. The circuit is shown in Fig.2.11 and (2.13) show the computation of the phase inductances and back EMFs.
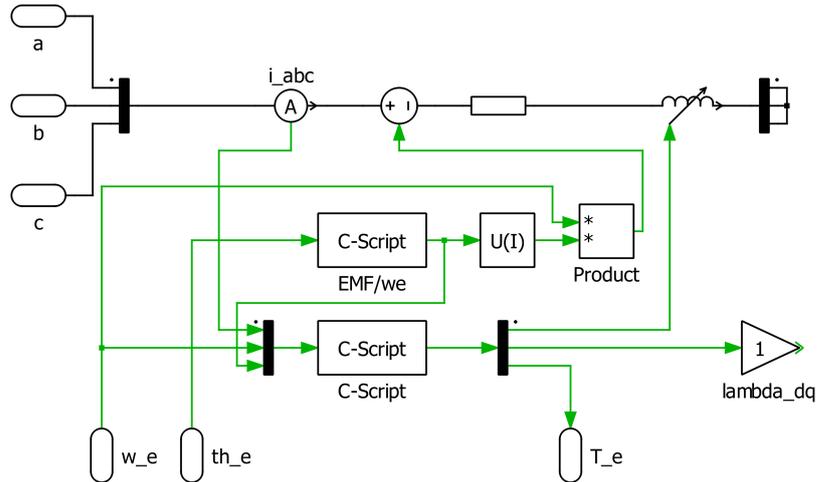
Figure 2.11: Voltage Behind Reactance model of the Permanent Magnet SM in PLECS.

$$
\begin{aligned}
L_a &= \frac{L_d + L_q}{2} + (L_d - L_q) \cdot cos(2\theta_r) & E_a &= -\omega_e \cdot \lambda_m \cdot sin(\theta_r) \\
L_b &= \frac{L_d + L_q}{2} + (L_d - L_q) \cdot cos(2\theta_r - \frac{2\pi}{3}) & E_b &= -\omega_e \cdot \lambda_m \cdot sin(\theta_r - \frac{2\pi}{3}) \\
L_c &= \frac{L_d + L_q}{2} + (L_d - L_q) \cdot cos(2\theta_r + \frac{2\pi}{3}) & E_c &= -\omega_e \cdot \lambda_m \cdot sin(\theta_r + \frac{2\pi}{3})
\end{aligned}
$$

$$(2.13)$$

2.4   New proposed models

After reviewing the available models in PLECS and Simulink, for the purpose of the thesis two motor models are proposed.

- Controlled Current Generators model.

- Voltage Behind Reactance model.

These models are suitable for all types of synchronous machines and both are implemented in PLECS and Simulink/Simscape. The presented models are going to substitute the Electrical equation block in of Fig.2.1, while the Mechanical equation block remains the same.

2.4.1   Controlled current generators model

The starting point of the controlled current generators (CCG) model is the Rotor Reference Frame model presented in section 2.3.2. As shown in Fig.2.12, the motor is modeled as three current generators, in parallel with very large resistors and all together in series with the phase resistances R_s. The current generators are piloted by the three phase currents of the machine and the voltage drop across the resistors represents the back EMFs of the machine.
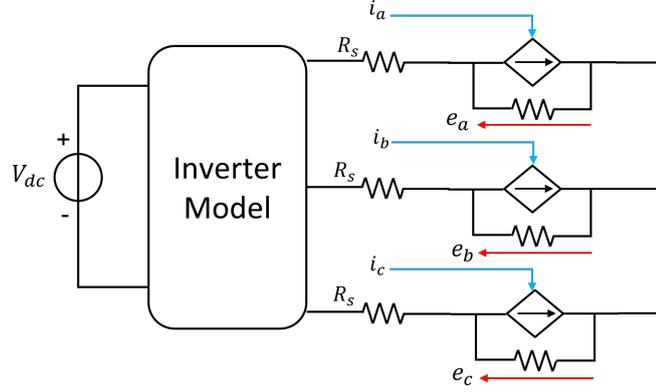
Figure 2.12: Circuital representation of the CCG model.

The block diagram for the calculation of the currents i_{abc} is represented in Fig.2.13.
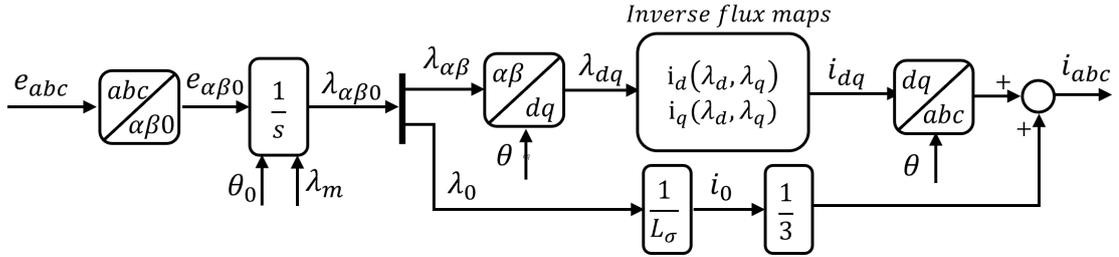

Figure 2.13: Block diagram for the computation of the three phase currents of the CCG model.

The back EMFs e_{abc} are transformed in $\alpha\beta0$ axes using the Clarke transformation:

$$\begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} X_\alpha \\ X_\beta \\ X_0 \end{bmatrix} \tag{2.14}$$

The integration of $e_{\alpha\beta0}$ provides the flux linkage in $\alpha\beta0$ axis $\lambda_{\alpha\beta0}$. The integral initialization depends on the position of the magnet flux λ_m and the initial rotor position θ_0:

1. *SyR*: the integrals are initialized to zero ($\lambda_m = 0$);

2. *PMASR*: the integral in α-axis is initialized to $\lambda_m sin(\theta_0)$ and in β-axis is initialized to $\lambda_m cos(\theta_0)$ (Fig.2.14a) ;

3. *SPM and IPM*: the integral in α-axis is initialized to $\lambda_m cos(\theta_0)$ and in β-axis is initialized to $\lambda_m sin(\theta_0)$ (Fig.2.14b).


(a) PM convention   (b) SyR convention
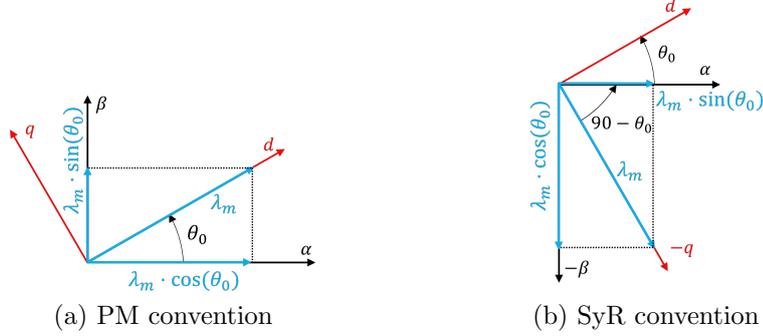
Figure 2.14: Magnet flux λ_m position on $\alpha\beta$ axis based on initial rotor position θ_0.

After that, the fluxes $\lambda_{\alpha\beta}$ are transformed in dq-axis λ_{dq} using the rotational transformation matrix:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & sin(\theta) \\ -\sin(\theta) & cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} \tag{2.15}$$

The fluxes λ_{dq} represent the input of the inverse magnetic model, used to retrieve the currents i_{dq}. These currents are then transformed in abc axis using the inverse Clarke-Parke transformation:

$$\begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos(\theta) & -sin(\theta) \\ cos(\theta - \frac{2\pi}{3}) & -sin(\theta - \frac{2\pi}{3}) \\ cos(\theta + \frac{2\pi}{3}) & -sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \cdot \begin{bmatrix} X_d \\ X_q \end{bmatrix} \tag{2.16}$$

At the end, the contribution of the homopolar flux λ_0 is given by summing $\frac{1}{3} \cdot i_0$ to each phase current, where i_0 is the homopolar current defined as $\frac{\lambda_0}{L_\sigma}$, where L_σ is the leakage inductance of the machine. In normal operating condition λ_0 is negligible but could be significant in fault operating conditions.

The implementations in PLECS and Simulink of the CCG model are shown respectively in Fig.2.15 and Fig.2.16. The model in Simulink was implemented using the Simscape Electrical library. In both models a large resistor is placed between the neutral point and ground, for stability reason, in case the sum of the three phase currents is not perfectly zero.
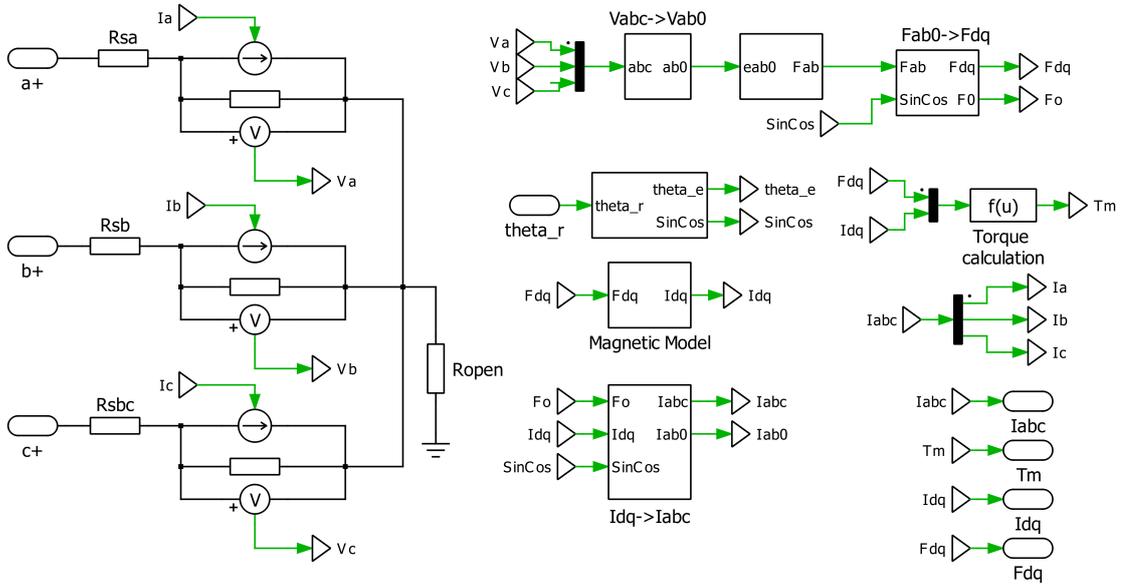
Figure 2.15: Current generators model in PLECS.
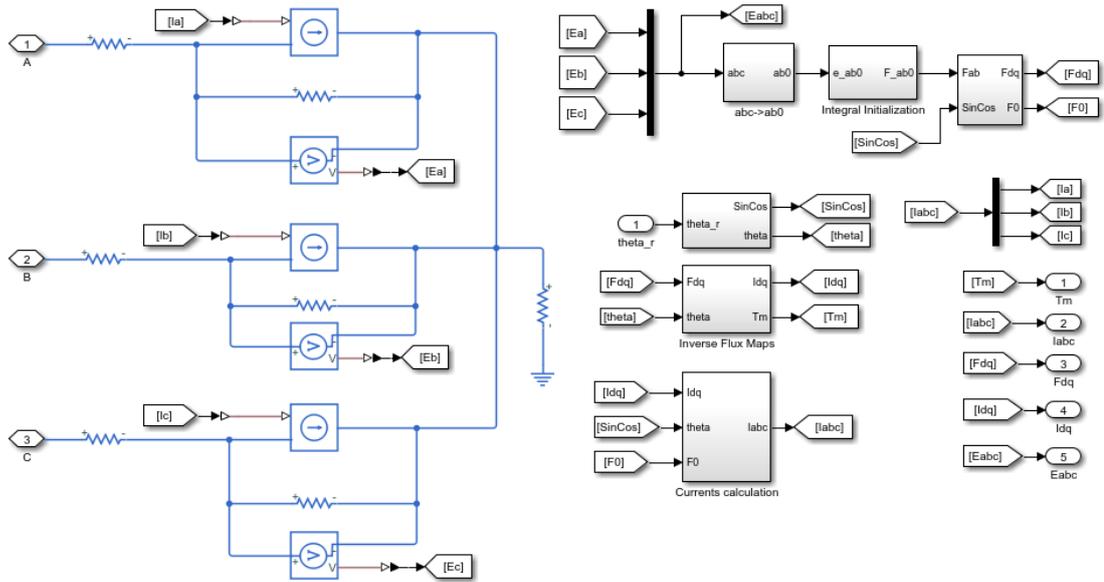
Figure 2.16: Current generators model in Simulink/Simscape Electrical.

2.4.2   Voltage behind reactance model

The theory behind the Voltage Behind Reactance model was already introduced in Chapter 1. The circuital implementation is based on the model of the Non-

Exited Syncronous Machine in PLECS library (Fig.2.7). The diagram block of model is shown in Fig.2.17.
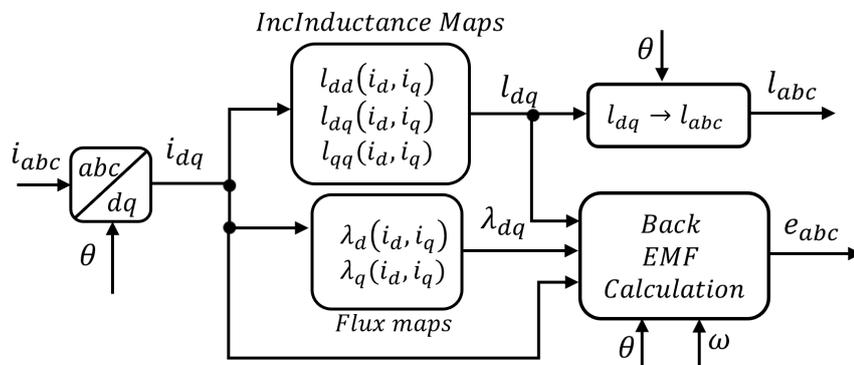
Figure 2.17: Block scheme for the computation of the inductance matrix and back EMFs in the VBR model.

The phase currents i_{abc} are transformed in i_{dq} using the Clarke-Park transformation:

$$\begin{bmatrix} X_d \\ X_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos(\theta) & cos(\theta - \frac{2\pi}{3}) & cos(\theta + \frac{2\pi}{3}) \\ -sin(\theta) & -sin(\theta - \frac{2\pi}{3}) & -sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \cdot \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \qquad (2.17)$$

The i_{dq} currents are used to obtain the incremental inductances l_d, l_q, l_{dq} from the incremental inductance maps and the fluxes λ_{dq} from the magnetic model. Then the computation of $[l_{abc}]$ and e_{abc} was introduced in section 1.1.4 and it can be found also in Appendix A. The implementations in PLECS and Simulink of the CCG model are shown respectively in Fig.2.18 and Fig.2.19. In PLECS model there are two elements added for stability reason: a very small inductor L_q and a very large resistor R_{open}. The model in Simulink was implemented using the Simscape Electrical library.

In PLECS, the three coupled inductors are simulated using the component *variable inductor*, which receives as input the elements of the incremental inductance matrix $[l_{abc}]$. In Simscape there is no such component in the standard library, but it is possible to generate a custom component, by using the Simscape language.
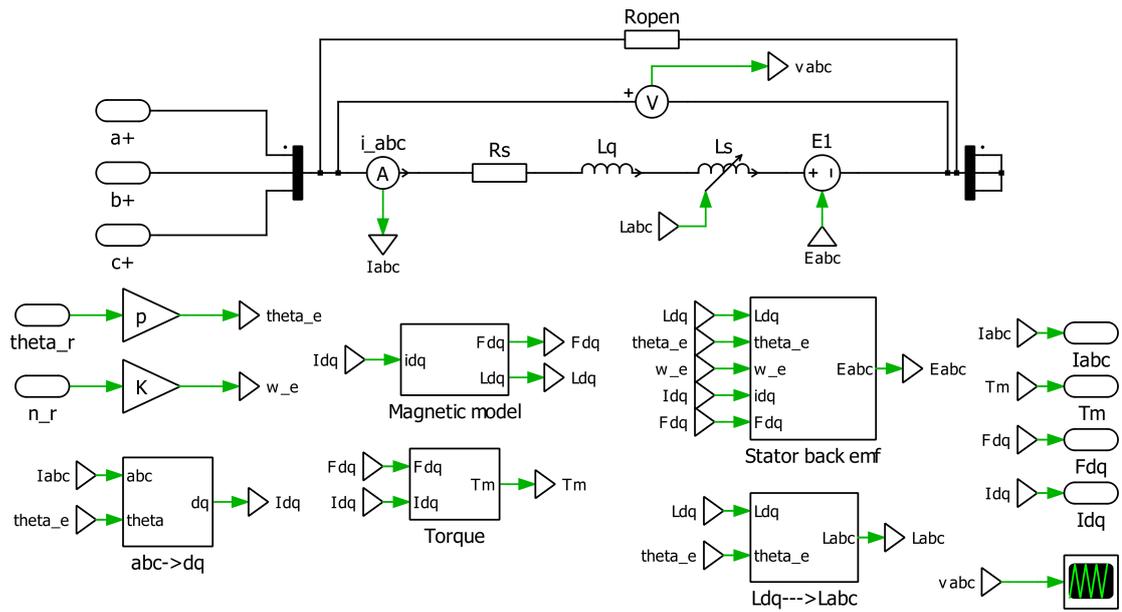
Figure 2.18: Voltage Behind Reactance model in PLECS.
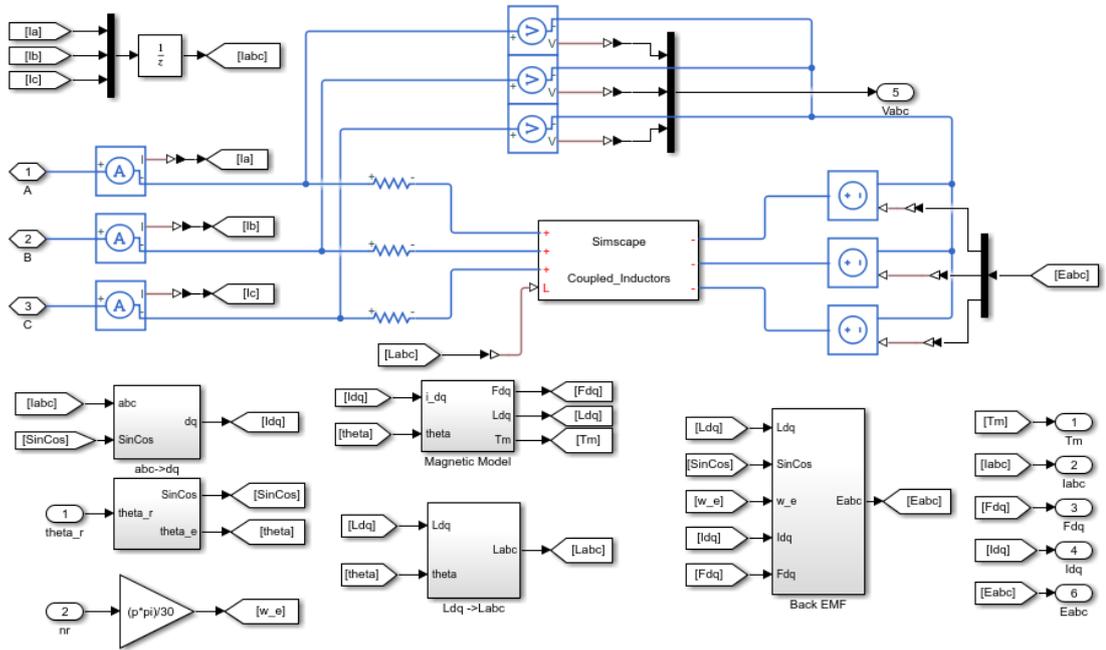
Figure 2.19: Voltage Behind Reactance model in Simulink/Simscape Electrical.

Chapter 3

Simulations of the proposed models

This chapter presents the results obtained by simulating the CCG model and the VBR both in PLECS and Simulink. First, the output waveforms of both models are presented and then a comparison is made, based primary on the computational time. The motor utilized in simulation is a synchronous reluctance machine, whose specifications are shown in Tab.3.1.

Table 3.1: Specification of SyR motor.

V_{DC}	565 V
nominal current	15 A
max current	30 A
nominal torque	17 Nm
max torque	43 Nm
nominal speed	2500 rpm
max speed	6000 rpm
pole pairs	3

The waveforms shown in this chapter are obtained using a torque control, whose block diagram is presented in Fig.3.1. The PI regulator is provided with feed-forward and anti wind-up mechanism. The reference torque and the imposed speed of the control are shown in Fig.3.2.
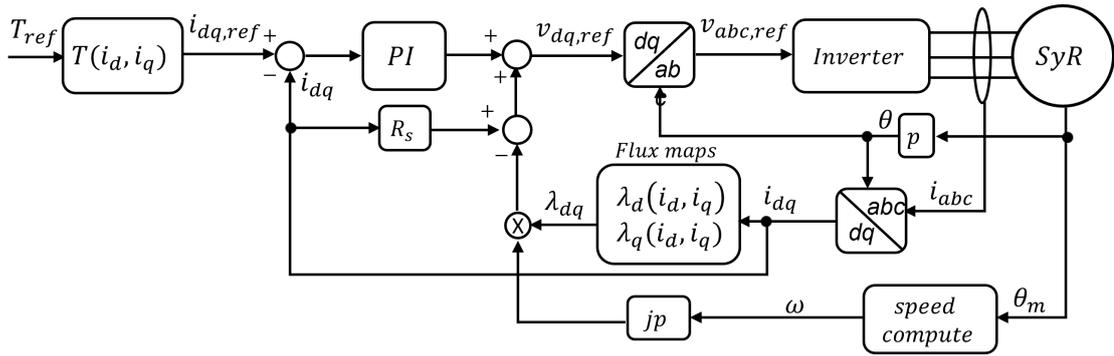
Figure 3.1: Torque control diagram.

The reference torque T_{ref} has two steps variations:

1. from 0 to nominal torque T_n, by imposing positive $I_{d,ref}$ and $I_{q,ref}$;

2. from T_n to $-T_n$, by reversing $I_{d,ref}$ (for a SyR motor the torque reversal can also be done by reversing $I_{q,ref}$).
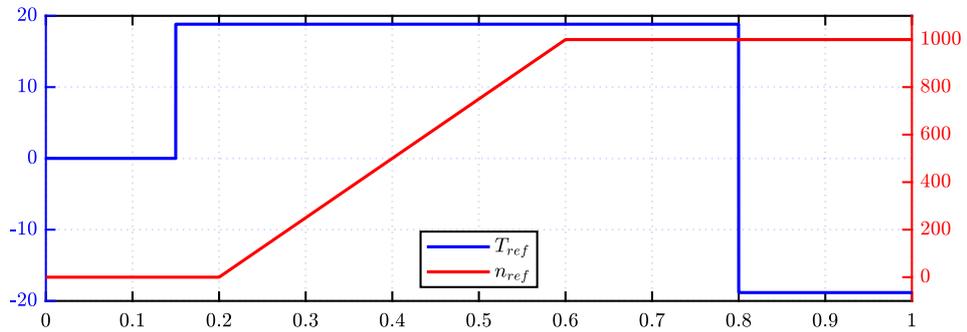

Figure 3.2: Reference torque and imposed speed of the torque control.

In the following sections, the presented results are:

1. **D-axis current waveform**;

2. **Q-axis current waveform**;

3. **Torque waveform**;

3.1   PLECS comparative results

The PLECS simulation is presented in Fig.3.3. The model has the same structure of the benchmark shown in Fig.1.11, with the same three main blocks: the Digital Control, the Inverter and the Motor. The Digital Control block includes the same C-script for the execution of the algorithm, while the Motor Model is using the CCG or the VBR model.
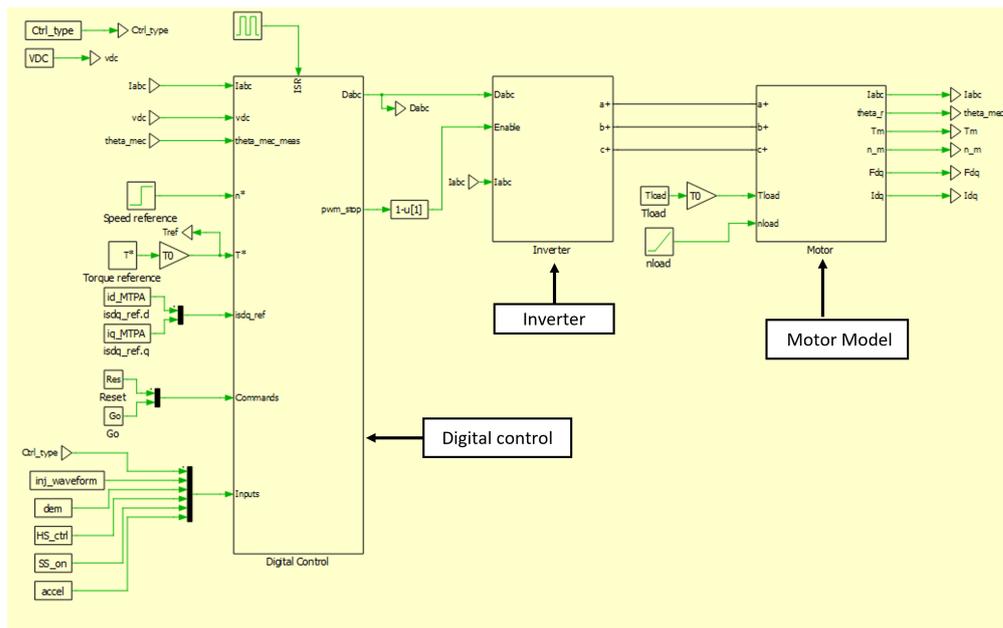

Figure 3.3: PLECS simulation model.

The Inverter model is presented in Fig.3.4. The inverter component of PLECS library is using ideal IGBTs and can work with two settings:

1. Switching model: the signals that commands the transistors are the six switching functions, obtained from the PWM regulation;

2. Average model: the signals that command the transistors are the duty cycle of each transistor.

The waveforms presented in this chapter are obtained using the switching configuration of the inverter.
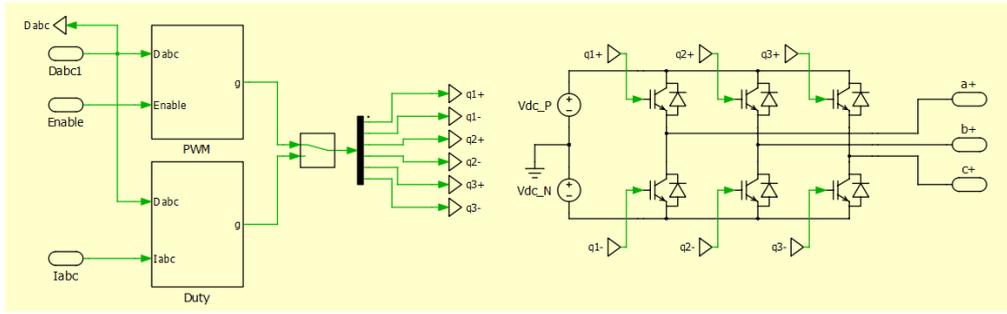
Figure 3.4: PLECS inverter model.

3.1.1   D-axis current waveform
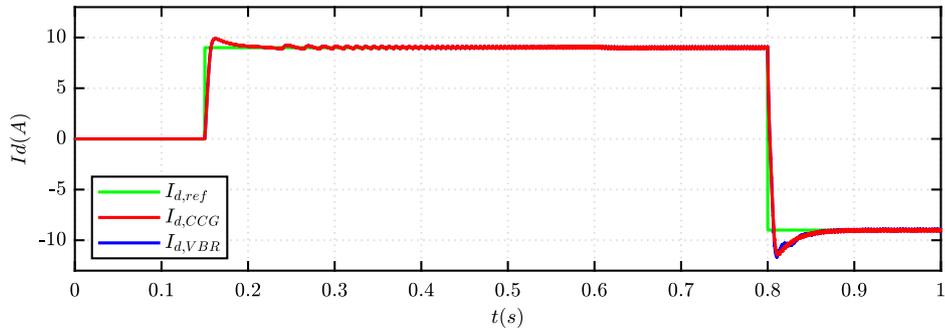
Figure 3.5: Reference current $I_{d,ref}$ and CCG I_d and VBR I_d.
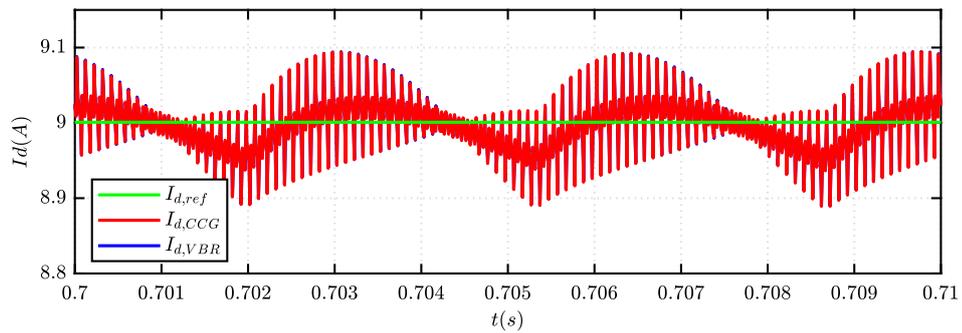
Figure 3.6: Zoom on CCG I_d and VBR I_d when steady state is reached, with positive T_{ref} and positive $I_{d,ref}$.
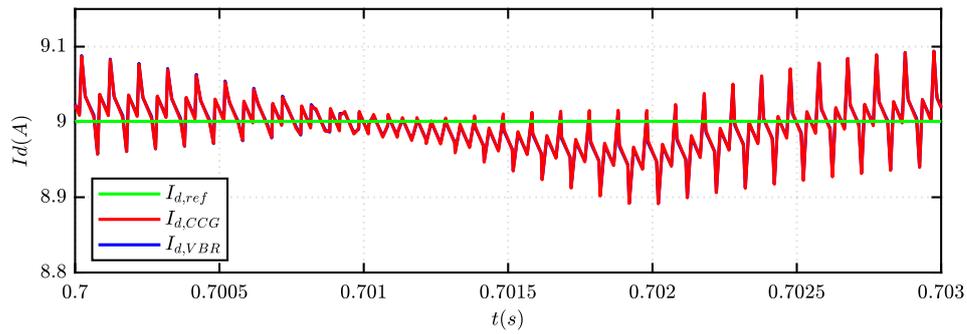
Figure 3.7: Zoom on the ripple of CCG I_d and VBR I_d, when steady state is reached, with positive T_{ref} and positive $I_{d,ref}$.
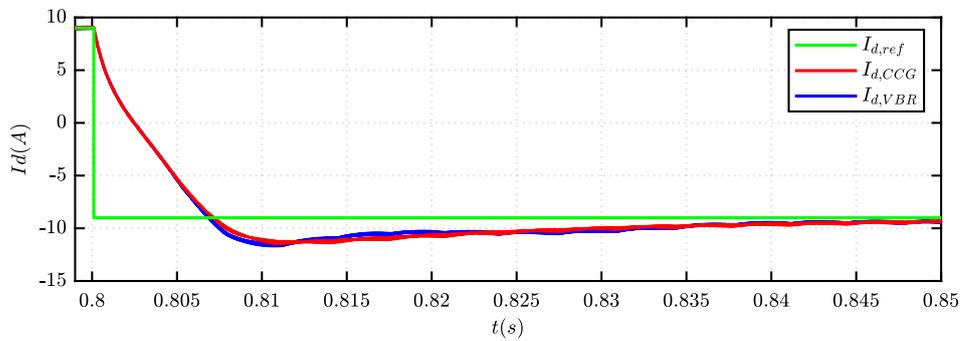

Figure 3.8: Zoom on CCG I_d and VBR I_d when the T_{ref} is reversed, resulting a negative step variation of $I_{d,ref}$.
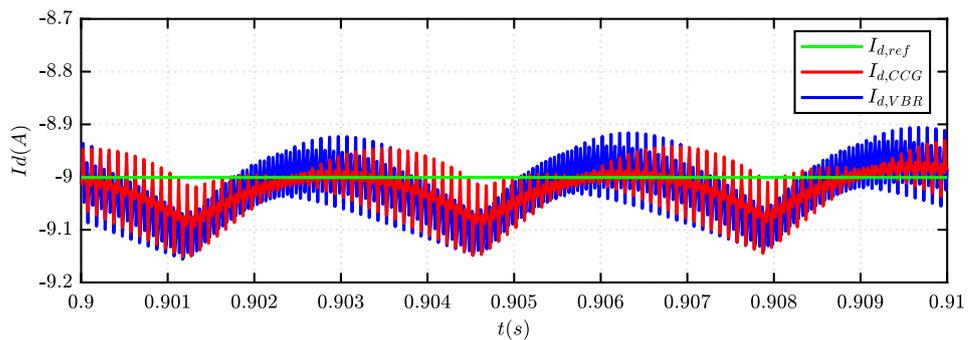

Figure 3.9: Zoom on CCG I_d and VBR I_d when steady state is reached, with negative T_{ref} and negative $I_{d,ref}$.
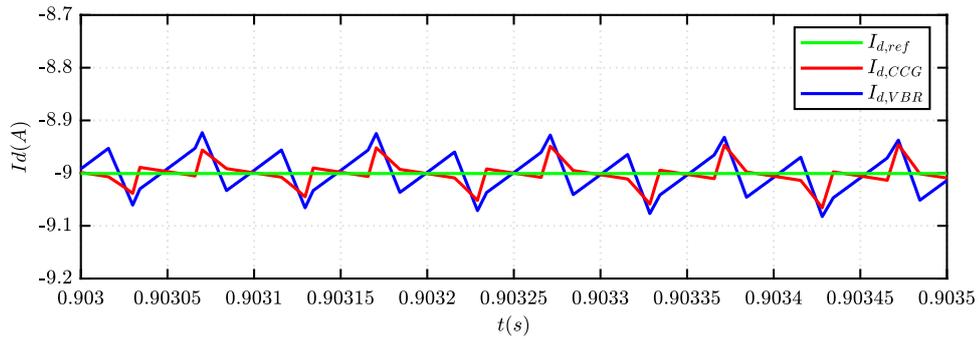
Figure 3.10: Zoom on the ripple of CCG I_d and VBR I_d, when steady state is reached, with negative T_{ref} and negative $I_{d,ref}$.

3.1.2   Q-axis current waveform
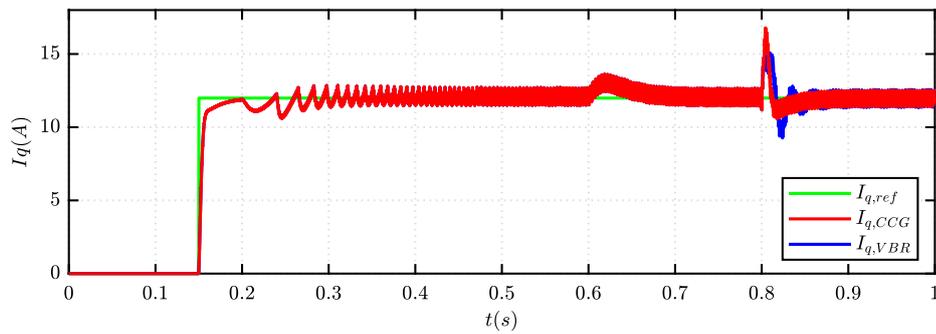
Figure 3.11: Reference $I_{q,ref}$ and CCG I_q and VBR I_q.
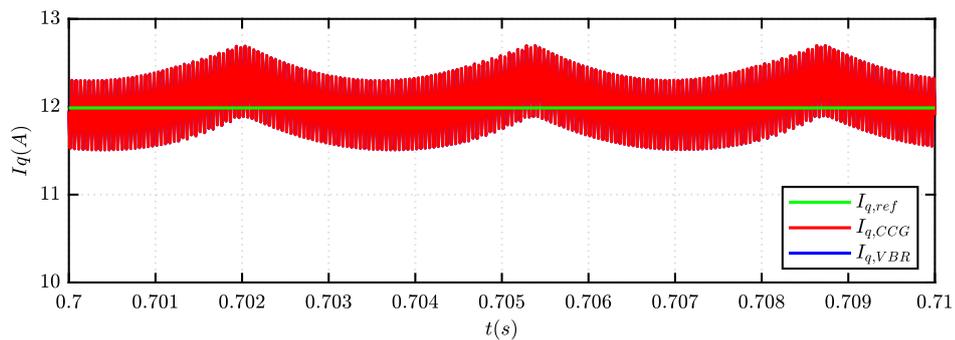
Figure 3.12: Zoom on CCG I_d and VBR I_d when steady state is reached, with positive T_{ref} and positive $I_{q,ref}$.
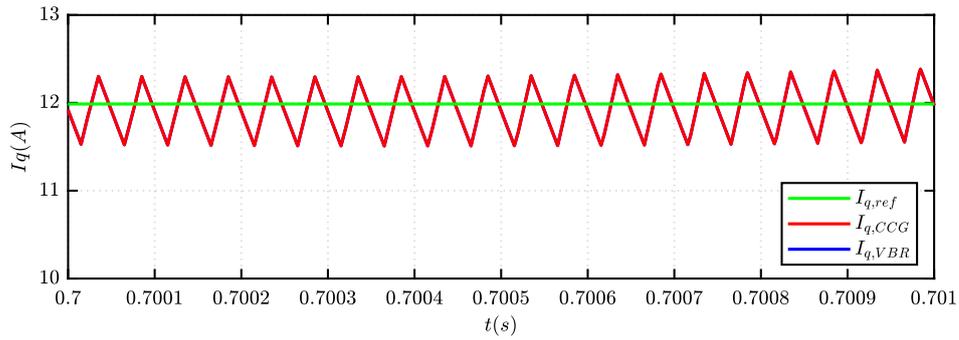
Figure 3.13: Zoom on the ripple of CCG I_q and VBR I_q, when steady state is reached, with positive T_{ref} with positive $I_{q,ref}$.
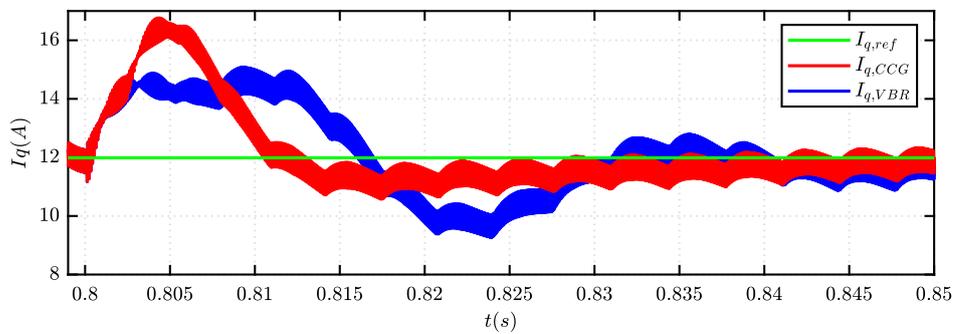
Figure 3.14: Zoom on CCG I_d and VBR I_d when the T_{ref} is reversed, maintaining a positive $I_{q,ref}$

.
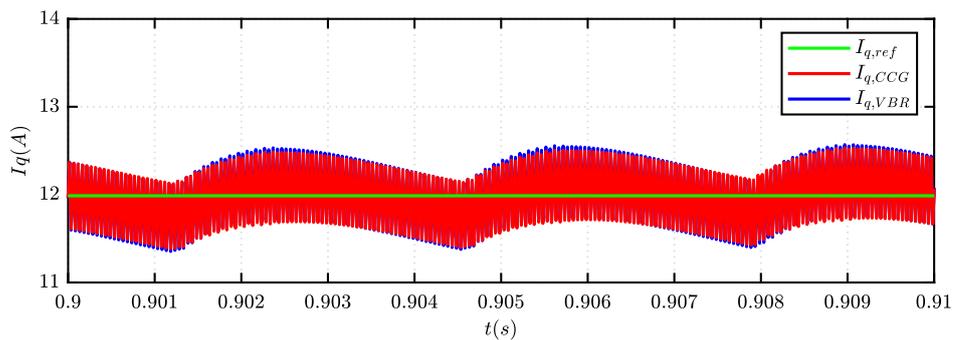
Figure 3.15: Zoom on CCG I_q and VBR I_q when steady state is reached, with negative T_{ref} and positive $I_{q,ref}$.
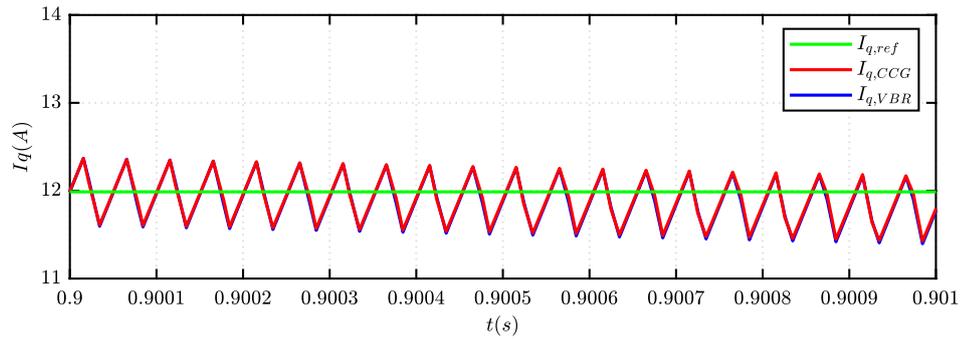
Figure 3.16: Zoom on the ripple of CCG I_q and VBR I_q, when steady state is reached, with negative T_{ref} and positive $I_{q,ref}$.

3.1.3 Torque waveform
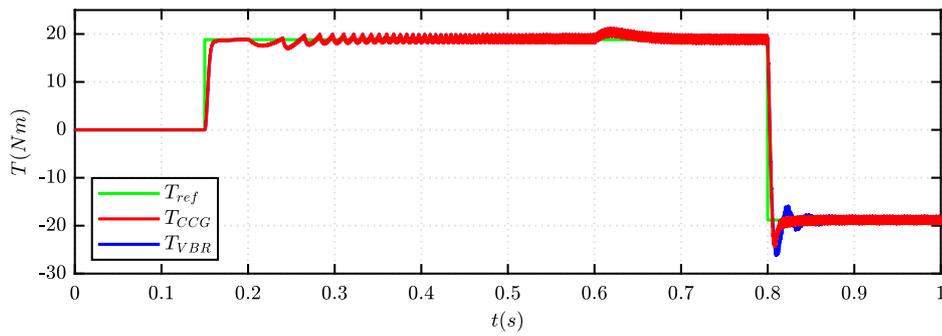
Figure 3.17: Reference T_{ref} and CCG T and VBR T.
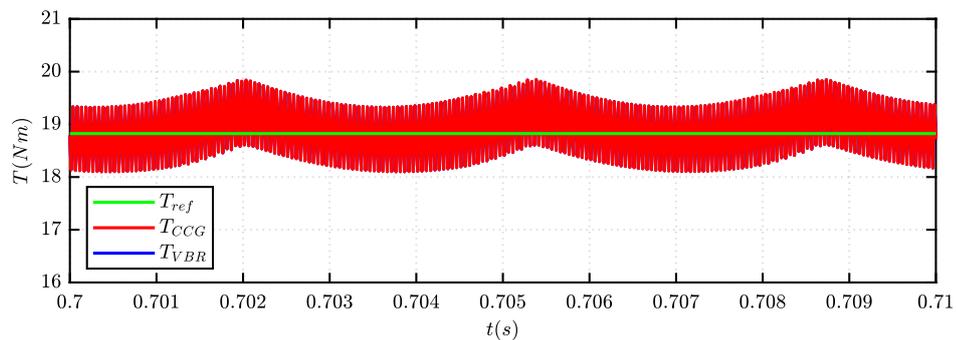
Figure 3.18: Zoom on CCG T and VBR T when steady state is reached, with positive T_{ref}.
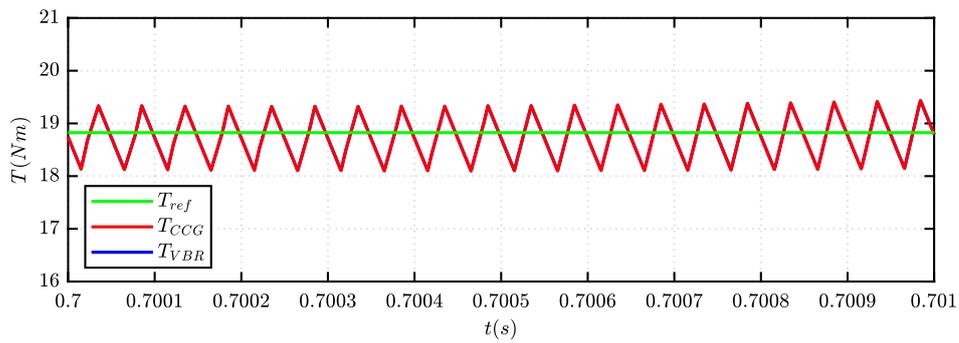
Figure 3.19: Zoom on the ripple of CCG T and VBR T, when steady state is reached, with positive T_{ref}.
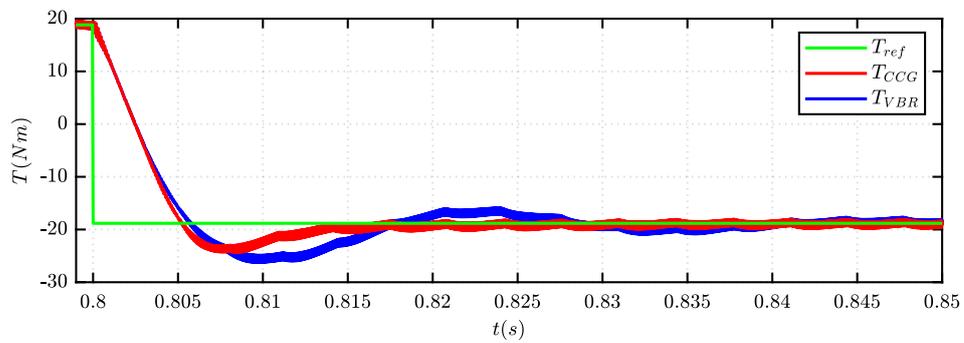
Figure 3.20: Zoom on CCG T and VBR T when the T_{ref} is reversed.
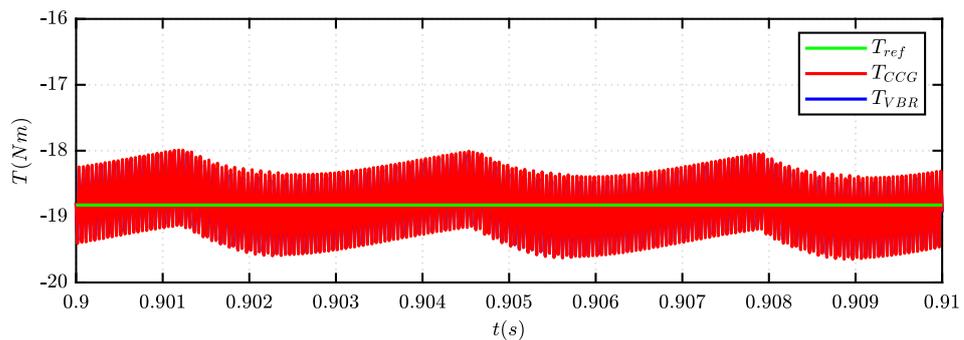
Figure 3.21: Zoom on CCG T and VBR T when steady state is reached, with negative T_{ref}.
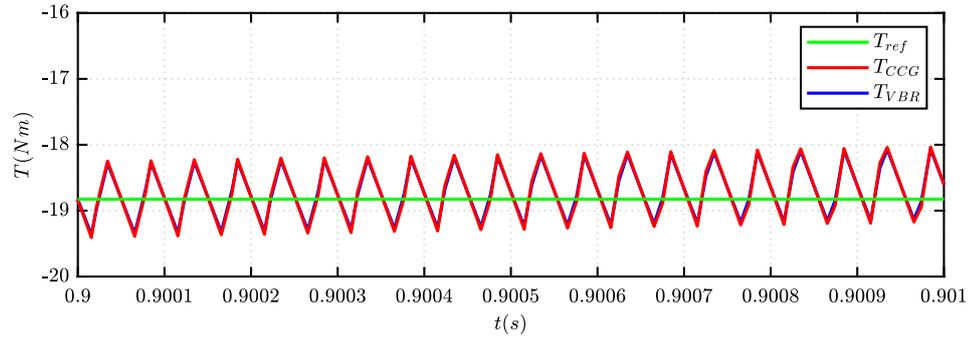
Figure 3.22: Zoom on the ripple of CCG T and VBR T, when steady state is reached, with negative T.

3.2 Simulink comparative results

The Simulink model is presented in Fig.3.23 and, similarly to the benchmark and PLECS model, it has structure, with the same three main blocks: Digital Control, Inverter Model and Motor Model, which is usign the CCG or the VBR model.
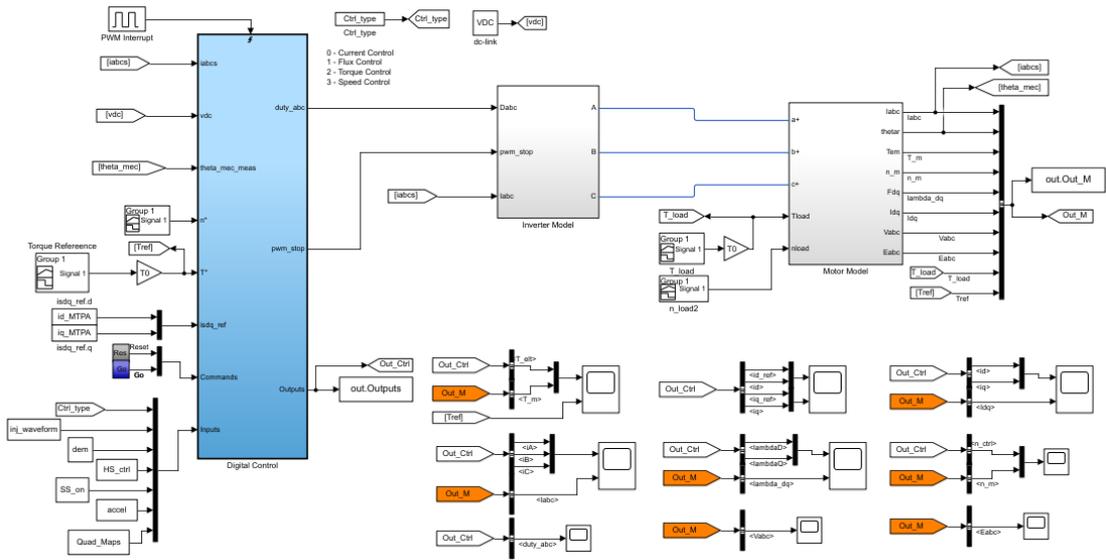
Figure 3.23: Simulink simulation model.

The Inverter model, presented in Fig.3.24, can use different components (MOS-FET, IGBT or Ideal Semiconductor) and as in PLECS in Fig.3.4, the same two settings are proposed again:

1. Switching model: the signals that commands the transistors are the six switching functions, obtained from the PWM regulation;

2. Average model: the signals that command the transistors are the duty cycle of each power module.

Again, the waveform are obtained using the inverter with the switching configuration.
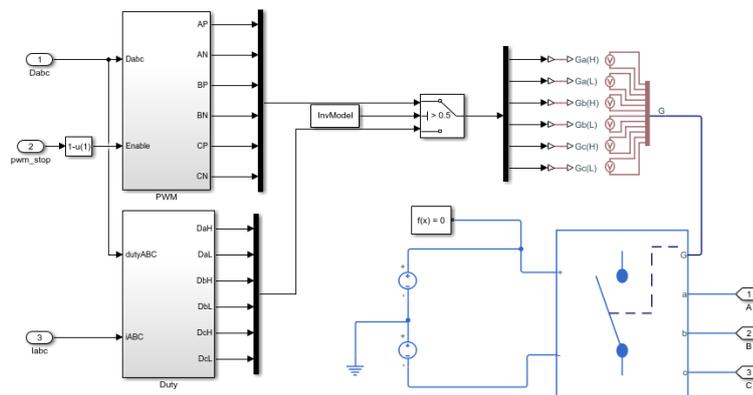
Figure 3.24: Simulink inverter model.

3.2.1   D-axis current waveform
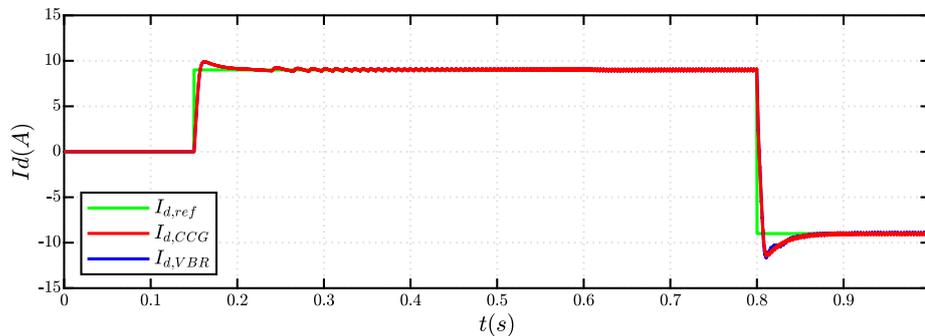
Figure 3.25: Reference current $I_{d,ref}$ and CCG I_d and VBR I_d.
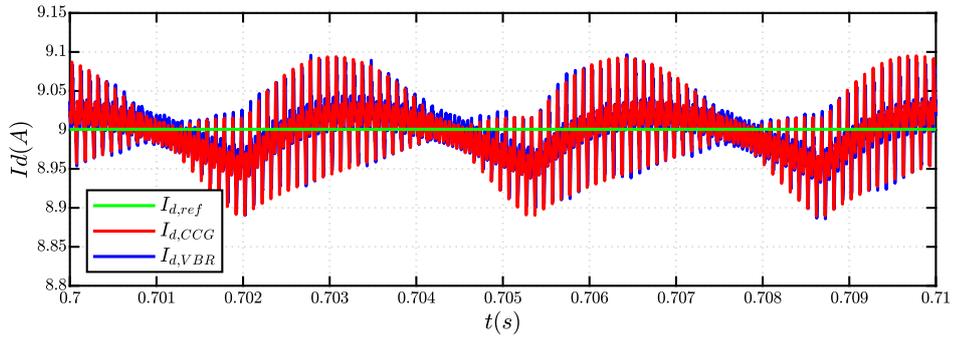
Figure 3.26: Zoom on CCG I_d and VBR I_d when steady state is reached, with positive T_{ref} and positive $I_{d,ref}$.
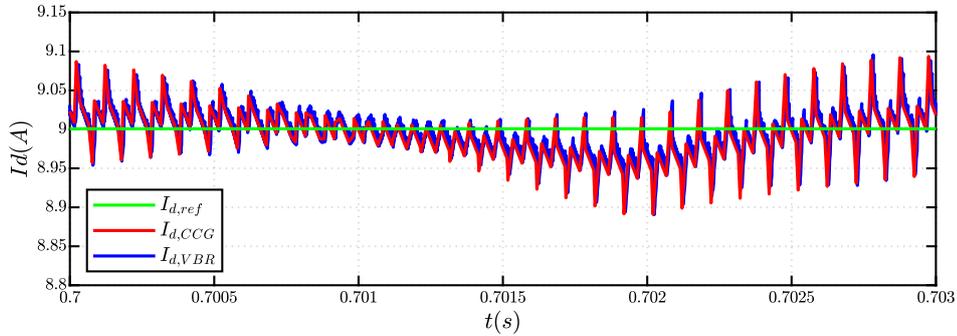
Figure 3.27: Zoom on the ripple of CCG I_d and VBR I_d, when steady state is reached, with positive T_{ref} and positive $I_{d,ref}$.
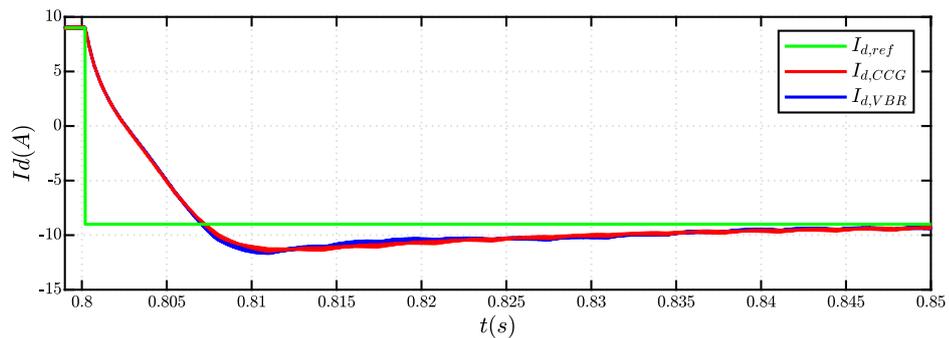
Figure 3.28: Zoom on CCG I_d and VBR I_d when the T_{ref} is reversed, resulting a negative step variation of $I_{d,ref}$.
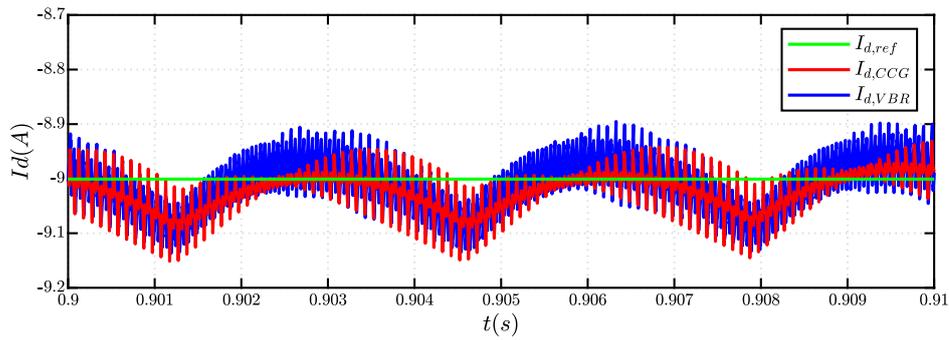
Figure 3.29: Zoom on CCG I_d and VBR I_d when steady state is reached, with negative T_{ref} and negative $I_{d,ref}$.
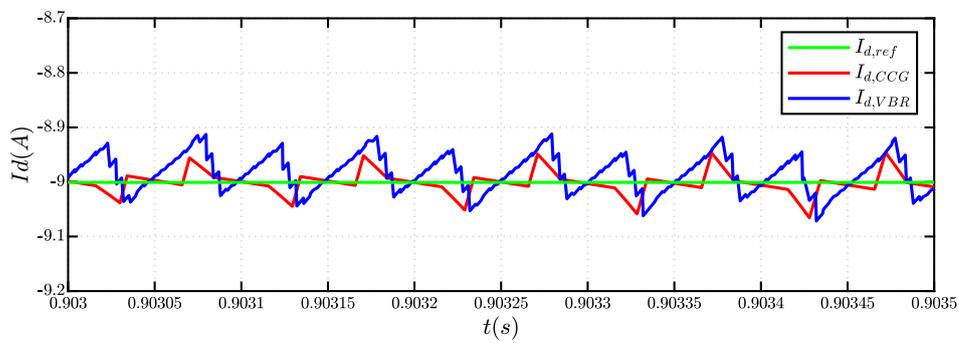
Figure 3.30: Zoom on the ripple of CCG I_d and VBR I_d, when steady state is reached, with negative T_{ref} and negative $I_{d,ref}$.

3.2.2   Q-axis current waveform
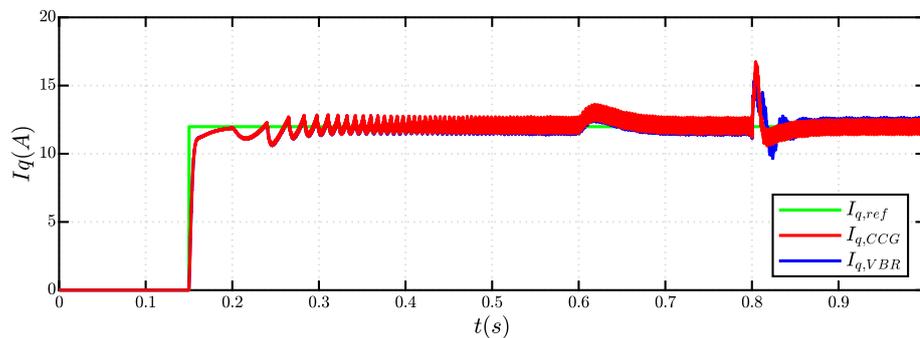
Figure 3.31: Reference $I_{q,ref}$ and CCG I_q and VBR I_q.
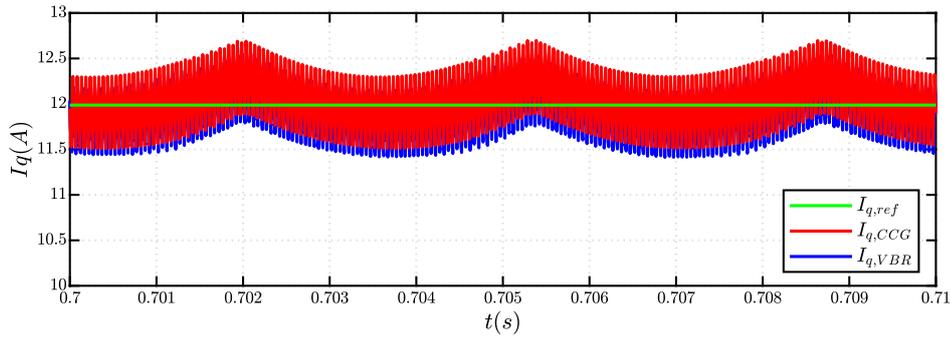
Figure 3.32: Zoom on CCG I_d and VBR I_d when steady state is reached, with positive T_{ref} and positive $I_{q,ref}$.
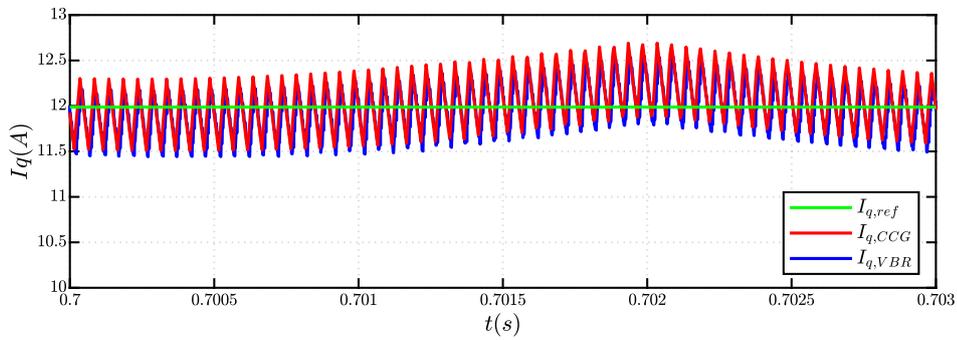
Figure 3.33: Zoom on the ripple of CCG I_q and VBR I_q, when steady state is reached, with positive T_{ref} with positive $I_{q,ref}$.
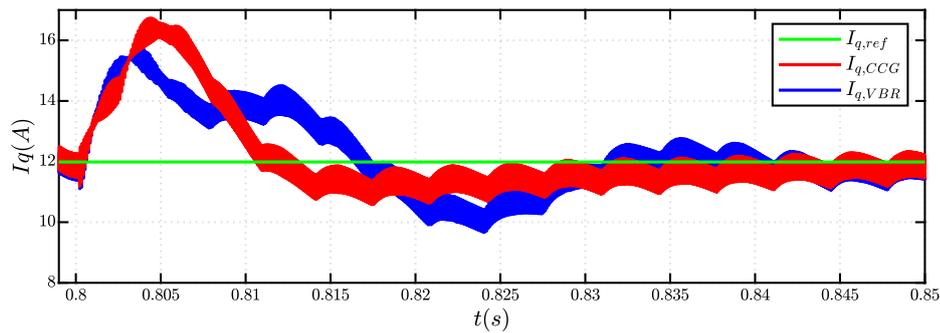
Figure 3.34: Zoom on CCG I_d and VBR I_d when the T_{ref} is reversed, maintaining a positive $I_{q,ref}$

.
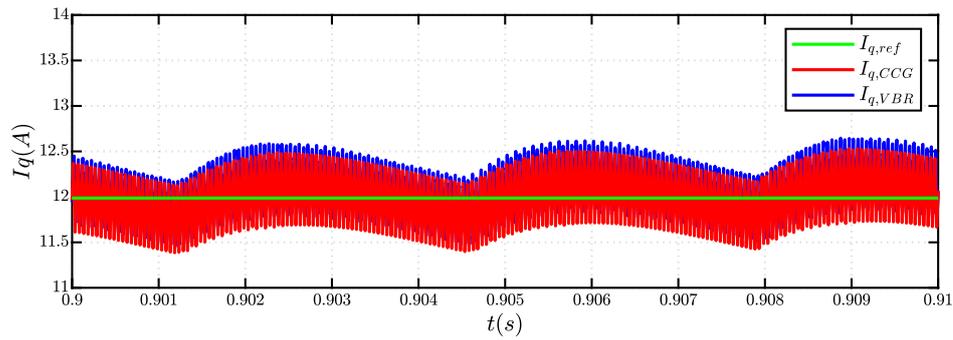
Figure 3.35: Zoom on CCG I_q and VBR I_q when steady state is reached, with negative T_{ref} and positive $I_{q,ref}$.
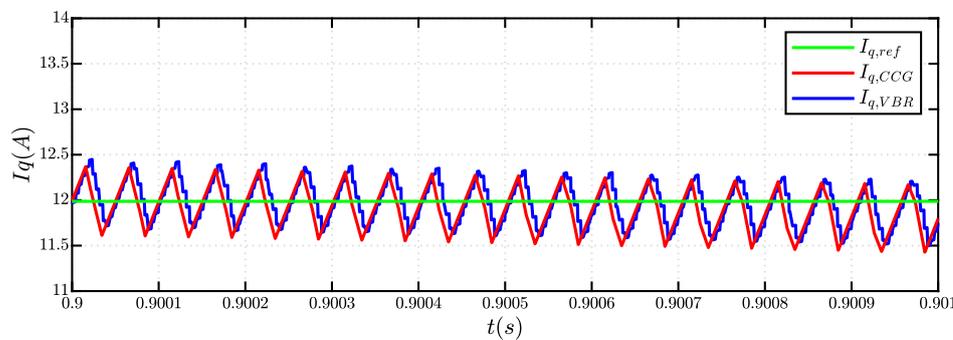
Figure 3.36: Zoom on the ripple of CCG I_q and VBR I_q, when steady state is reached, with negative T_{ref} and positive $I_{q,ref}$.

3.2.3   Torque waveform
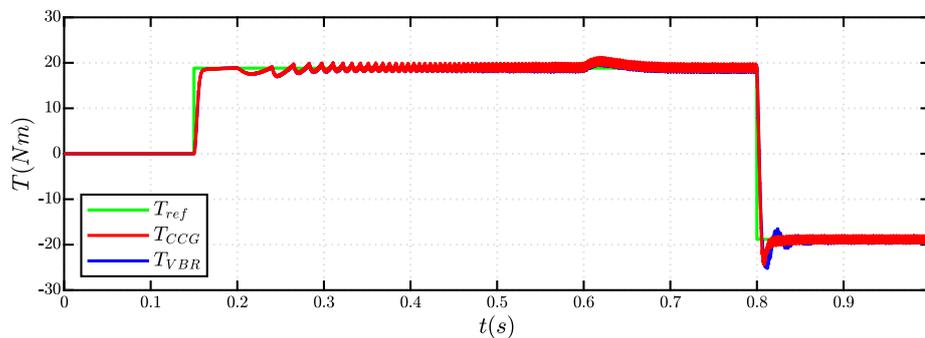
Figure 3.37: Reference T_{ref} and CCG T and VBR T.
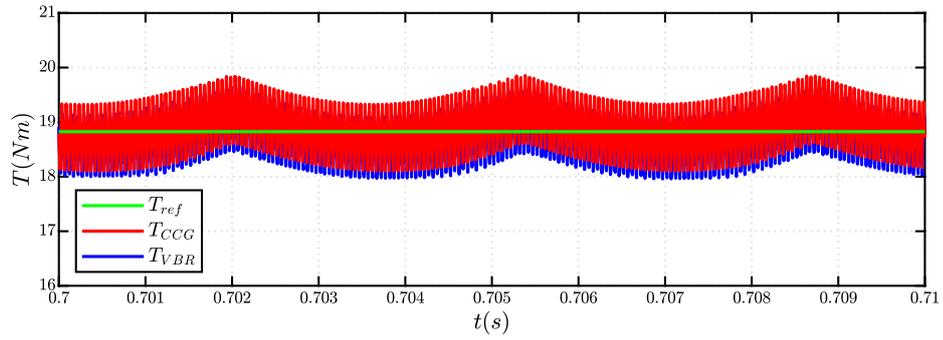
Figure 3.38: Zoom on CCG T and VBR T when steady state is reached, with positive T_{ref}.
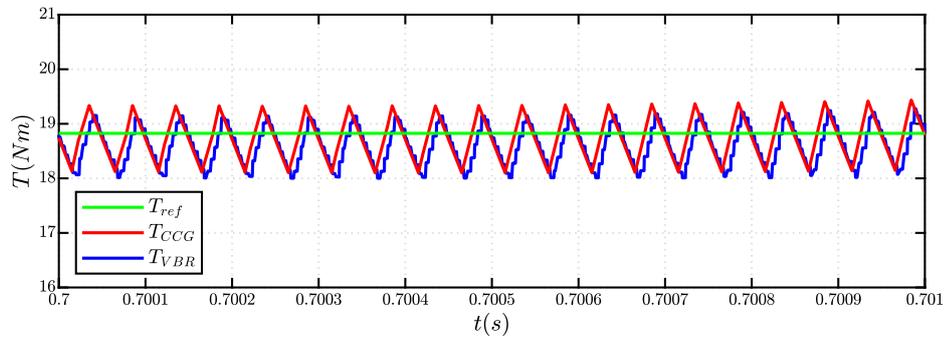
Figure 3.39: Zoom on the ripple of CCG T and VBR T, when steady state is reached, with positive T_{ref}.
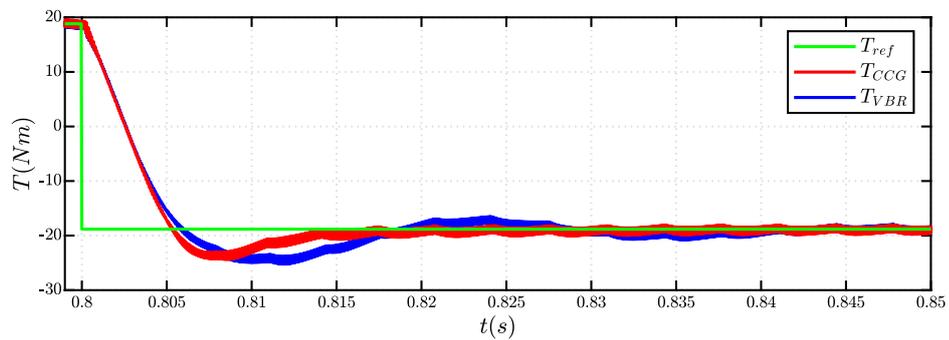
Figure 3.40: Zoom on CCG T and VBR T when the T_{ref} is reversed.
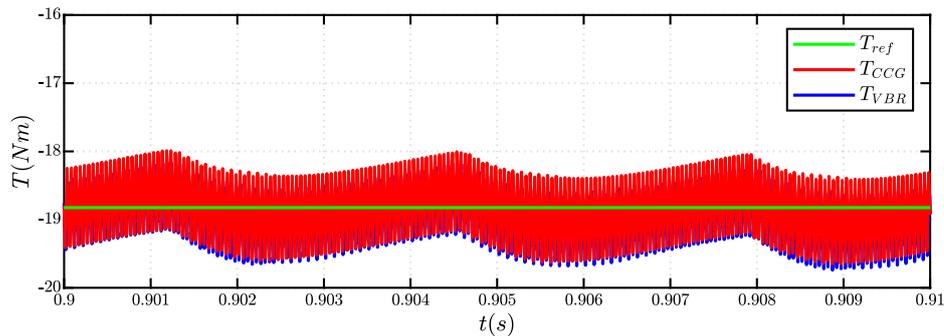
Figure 3.41: Zoom on CCG T and VBR T when steady state is reached, with negative T_{ref}.
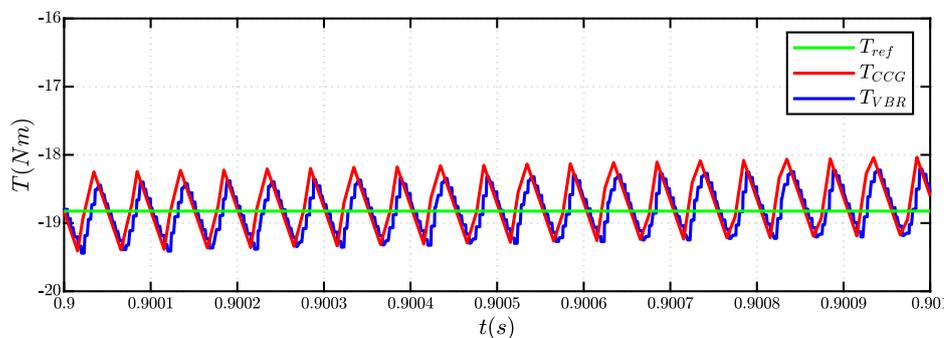
Figure 3.42: Zoom on the ripple of CCG T and VBR T, when steady state is reached, with negative T.

3.3   Summary of the comparison

The results provided by CCG and VBR models are very similar at steady state but not during the transients. When the first torque step is applied (from 0 to T_n) and when the steady state is reached the I_d and I_q currents and torque are basically equal (the waveforms are perfectly overlapped). Instead, when the second torque step is applied (from T_n to $-T_n$), the behavior of the two models is quite different: the I_d currents differ very little, instead there is a big difference between the I_q currents ad thus the torques. The reason for this difference is still unclear. In the end, at steady state with negative T_{ref}, the I_q currents and torques are again basically equal. However, there is a difference in the ripple waveform of the I_d currents (also this reason is unclear). It can be noted that in Simulink, the waveforms provided by the VBR model are more 'discretized'. This is caused

by the delay block applied to the currents measurements (Fig.2.19), necessary for braking an algebraically loop.

For the comparison of the CCG model and VBR model, it is very important to take into account the execution time of the simulation, which depends on different factors: processor of the computer, tasks of the simulation, data loaded into the model, hours of operation of the computer and, if it is a laptop, it depends if it is powered from the battery or from the grid.

The results presented in this section are obtained using a laptop computer, just turned on and plugged to the grid, with the following specifications:

1. i7 Intel Core (2.60GHz);

2. 16 GB of RAM memory.

3.3.1 Computational time assessment: PLECS model

The settings of the PLECS solver are shown in Tab.3.2.

Type	Variable step
Time step	$2e^{-6}$
Solver	RADAU(stiff)
Relative tolerance	1e-3
Absolute tolerance	1e-3
Simulation time	1s

Table 3.2: PLECS solver configuration.

The comparison of the execution time of the models is presented in Fig.3.43.

(a) Switching model execution time.   (b) Average model execution time.
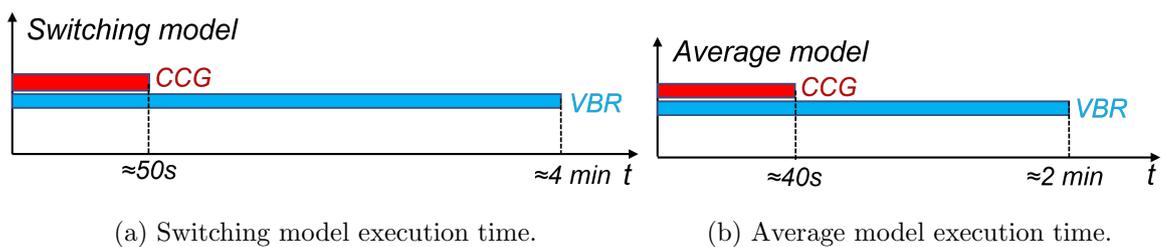
Figure 3.43: Execution time of PLECS models for 1s of simulation time.

3.3.2 Computational time assessment: Simulink model

The settings of the Simulink solver are shown in Tab.3.3.

Type	Variable step
Time step	$2e^{-6}$
Solver	ode15s(stiff/NDF)
Relative tolerance	1e-3
Absolute tolerance	1e-3
Simulation time	1s

Table 3.3: Simulink solver configuration.

The comparison of the execution time of the models is presented in Fig.3.44.

(a) Switching model execution time.   (b) Average model execution time.
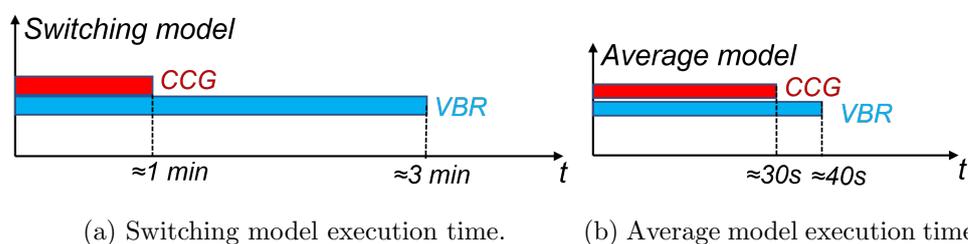
Figure 3.44: Execution time of Simulink models for 1s of simulation time.

In Fig.3.45, the benchmark model execution time for the same simulation is compared with the two models used in the time averaged configuration.
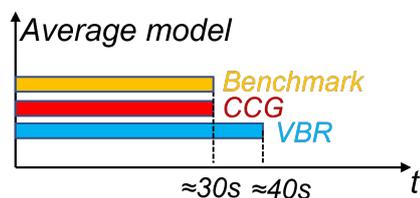
Figure 3.45: Comparison of the execution time between the benchmark model and the time averaged CCG and VBR models, for 1s of simulation time.

3.4   Simulation conclusions

In conclusion, the main characteristics of the two model are described below:

1. **CCG model**: the model requires a minor computational time because the circuit has less elements and it requires only the 2D lookup table of the *inverse flux maps*. But these are also the main disadvantage of the model

because the computation of the *inverse flux maps* reduces the operating domain with respect to the *flux maps*;

2. **VBR model**: the model requires a heavier computation burden, probably because the circuits has more elements and it needs to process more data, the 2D look tables of the *flux maps* and *incremental inductance maps*.

For now, the most convenient model is the CCG model used in the switching configuration. The execution time is only 30s more than the benchmark model, despite the fact that it is a switching model and that the Simscape components are heavier computationally speaking respect to Simulink blocks.

Chapter 4

Experimental validation

This chapter presents the results obtained during the experimental validation in the laboratory. The goal is to compare the waveforms obtained in the laboratory tests and in simulation, in order to verify the accuracy of the Simulink models in representing the behavior of a real machine. First, the workbench used for the laboratory tests is presented and then the following analysis are described:

1. PWM current ripple analysis in normal operating condition;

2. Currents analysis during a controlled Active Short Circuit (ASC) transient.

Additionally, a Open Phase Fault (OPF) was simulated in Simulink and Magnet, an environment for Finite Element Analysis, for comparison of the phase currents.

4.1   Experimental setup

The machine under test used in the laboratory is a IPM synchronous machine from the company BRUSA Elektronik, whose main specifications are shown in Tab.4.1. The driving machine coupled with the machine under test is an asynchronous machine from the company LENZE, whose main specifications are shown in Tab.4.2.

V_{DC}	400 V
nominal speed	4200 rpm
continuous torque	130 Nm
max torque	220 Nm
continuous power	70 kW
max power	96 kW
max speed	12000 rpm
pole pairs	3

Table 4.1: Main specifications of the machine under test.

V_{DC}	380 V
nominal speed	2100 rpm
continuous torque	100 Nm
max torque	140 Nm
continuous power	22 kW
max power	30 kW
max speed	6400 rpm
pole pairs	4

Table 4.2: Main specifications of the driving machine.

The two motors coupled together are shown in Fig.4.1, while Fig.4.2 presents the whole laboratory workbench.
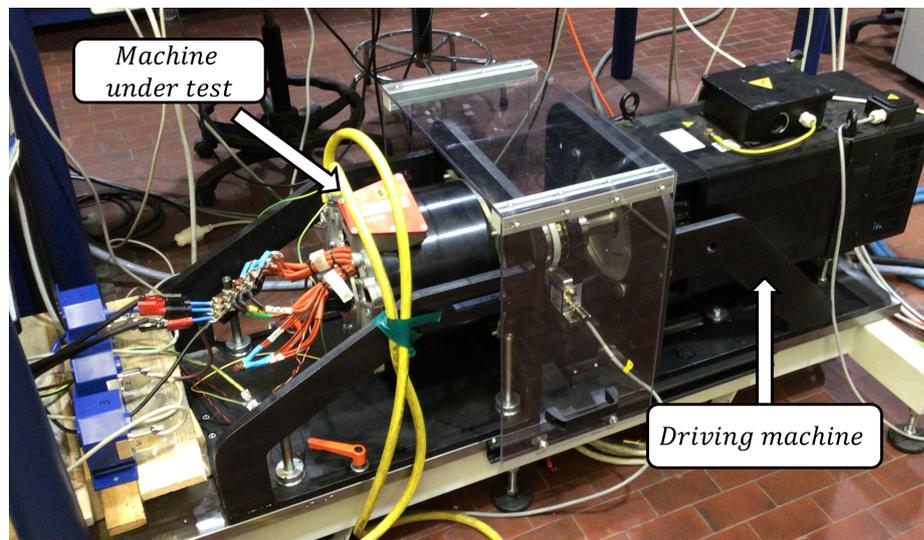
Figure 4.1: Workbench with machine under test and driving machine.
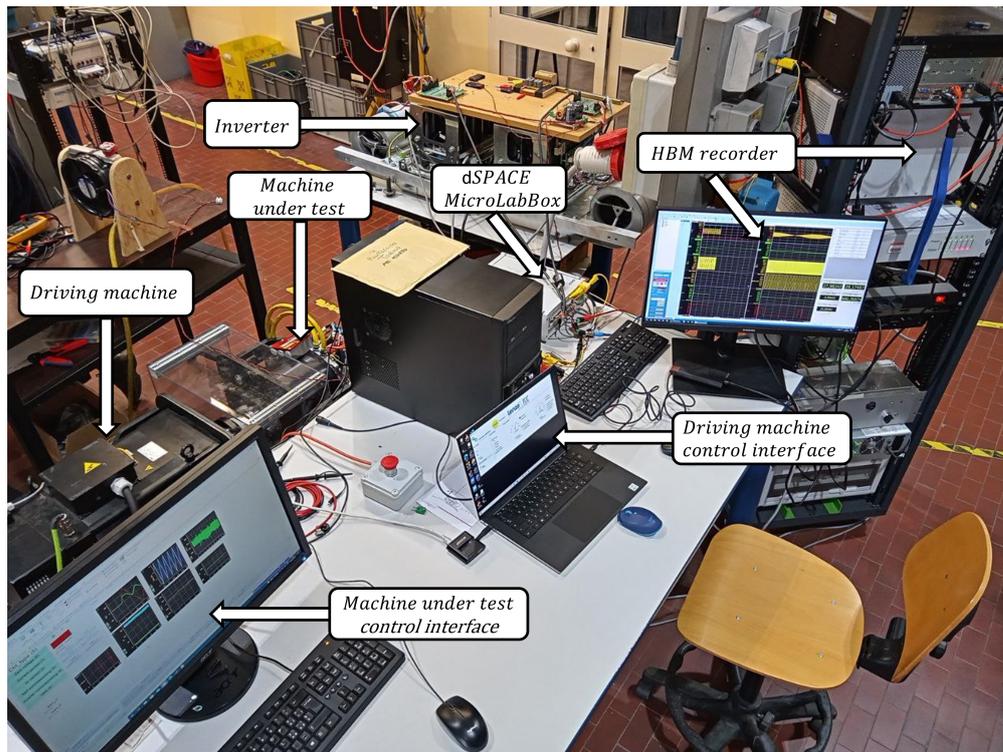
Figure 4.2: Laboratory workbench.

The main elements of the workbench are:

- Machine under test and driving machine, whose main specifications are in Tab.4.1 and Tab.4.2;

- Control interface for the driving machine which is speed controlled, using an algorithm written in LabVIEW software;

- Control interface for the machine under test which is current controlled, using an algorithm written in Simulink and loaded into the dSPACE MicroLabBox;

- dSPACE MicroLabBox, which is development a system, with real time processor and user programmable FPGA, used for electrical drives control;

- HBM data recorder, which is a powerful data acquisition system with very high sampling rate (400MB/s); it receives the data provides by a torque sensor, resolver and current sensors;

- Three-phase two-levels inverter, whose main specifications are showed in Tab.4.3.

I_{out} rated	180 A_{rms}
I_{out} max	200 A_{rms}
V_{DC} max	750 V
F_{out}	500 Hz
F_{sw} max	20 kHz

Table 4.3: Main specifications of the inverter.

4.2   PWM current ripple analysis

The aim of the PWM analysis is to verify the accuracy of the Simulink models in simulating the phase currents PWM ripple at steady-state operating conditions. By imposing different working points of speed and dq currents, the machine under test phase currents are measured through the HBM data recorder and then they are compared with the phase currents obtained in simulation.

Both CCG and VBR models are analyzed, by simulating two operating points:

- $I_d = -98A$ and $I_q = 68A$ (around the nominal torque T_n) at 1000 rpm;

- $I_d = -98A$ and $I_q = 68A$ at 2000 rpm.

The figures in the next pages show the comparison of the phase A current I_a between the results obtained in simulation and in laboratory.

Controlled Current Generators

The comparison of current I_a at 1000rpm is presented in Fig.4.3, while Fig.4.4 shows the same comparison at 2000rpm. It can be noted that in both operating points the CCG model simulates the real phase current with sufficient accuracy.
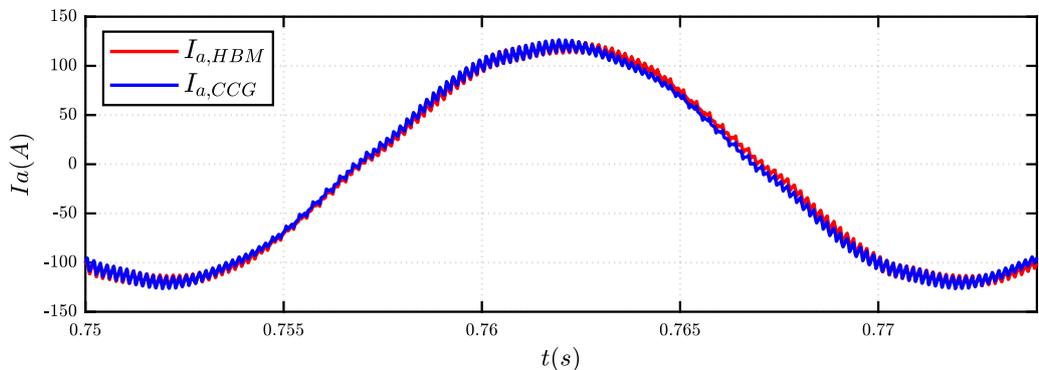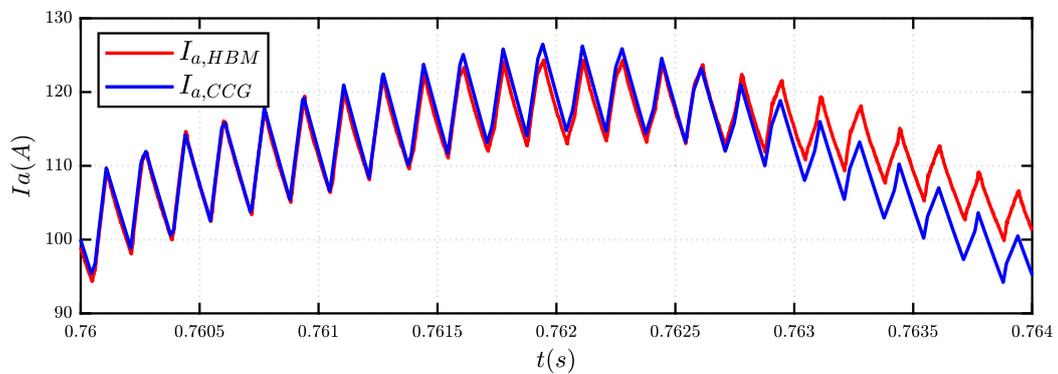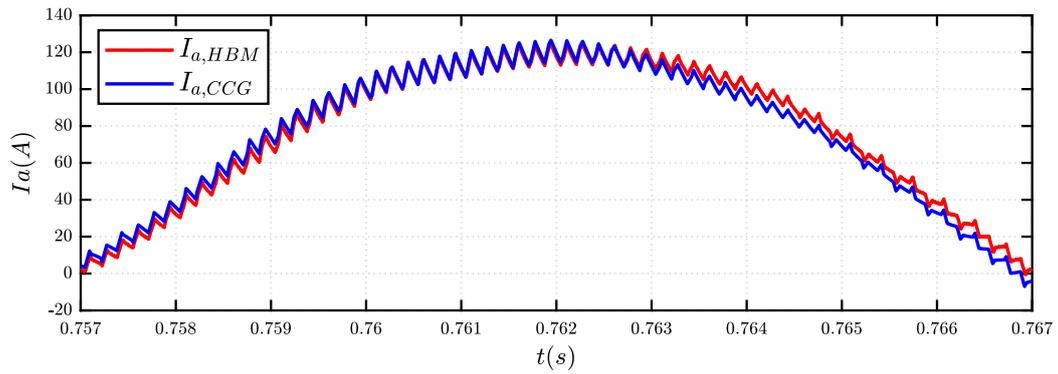
Figure 4.3: Comparison of I_a current between CCG model and HBM data recorder acquisition, at 1000rpm.
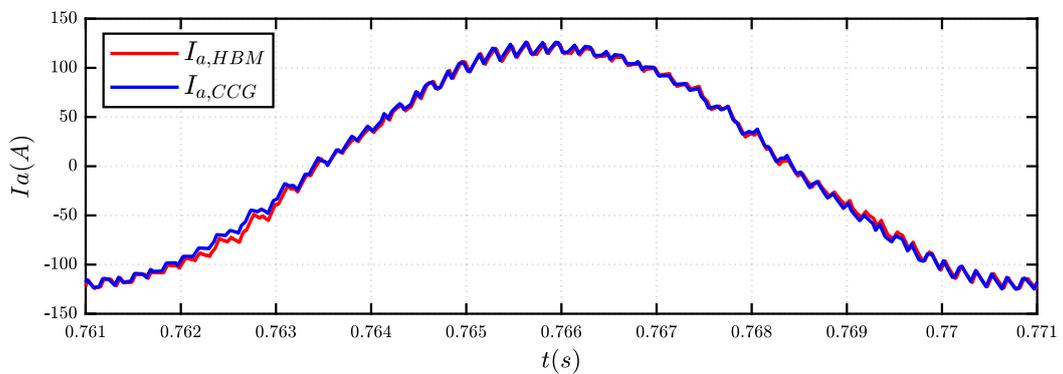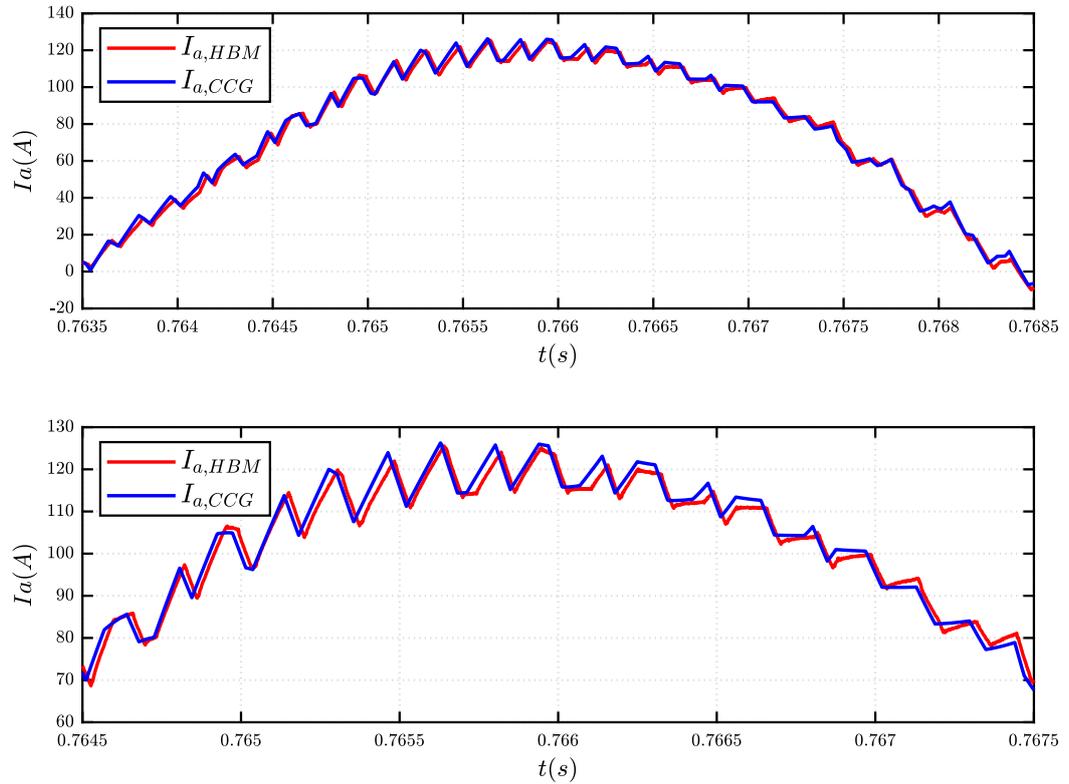
Figure 4.4: Comparison of I_a current between CCG model and HBM acquisition, at 2000rpm.

Voltage Behind Reactance

The comparison of current I_a at 1000rpm is presented in Fig.4.5, while Fig.4.6 shows the same comparison at 2000rpm. It can be noted that in both operating points also the VBR model recreates the real current with sufficient accuracy.
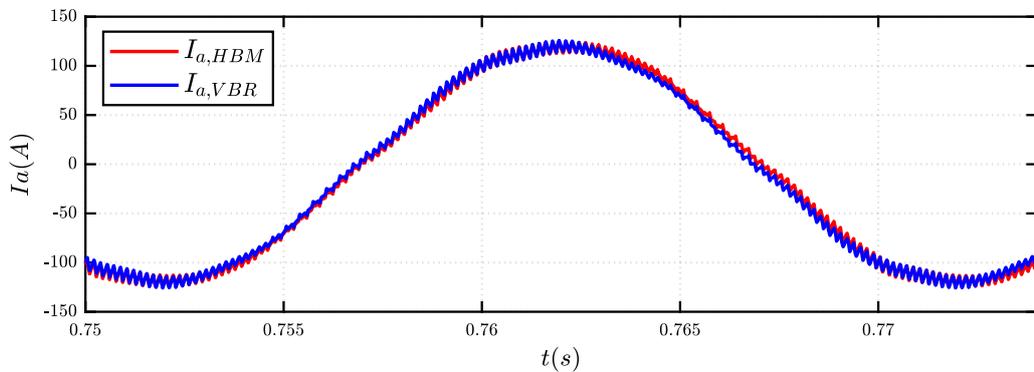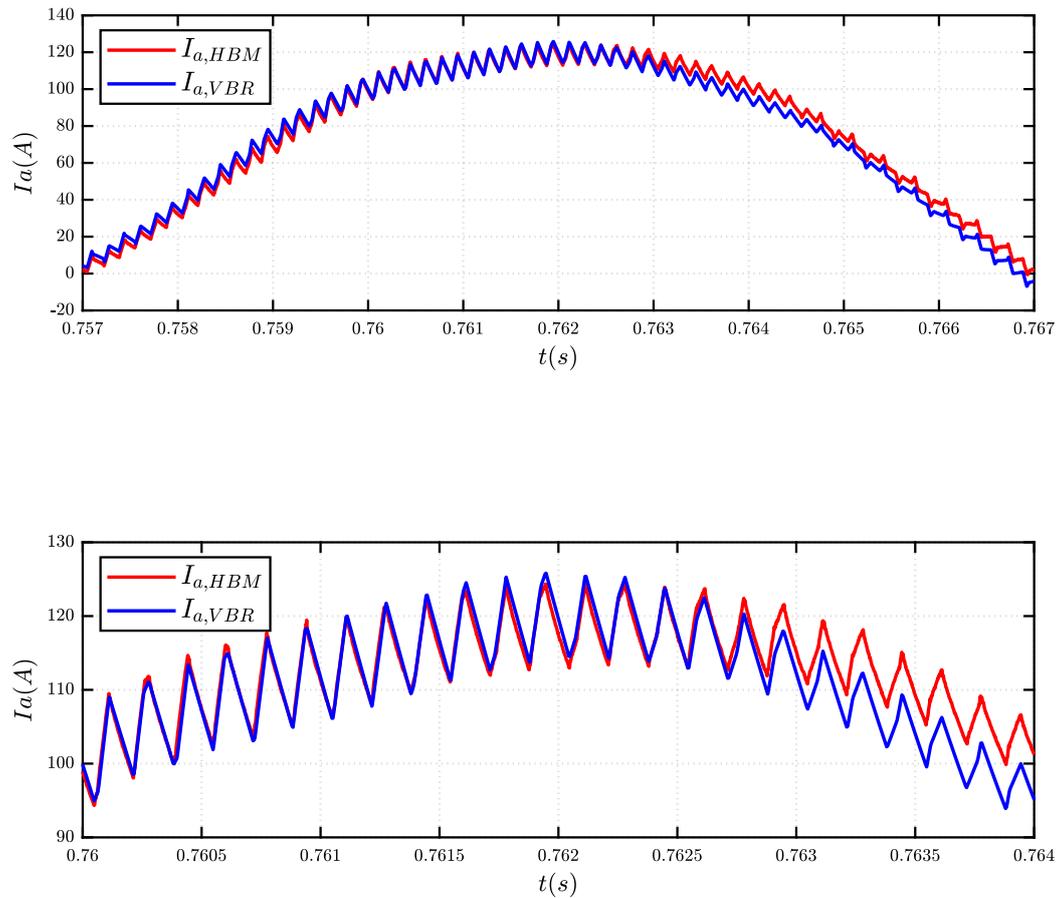
Figure 4.5: Comparison of I_a current between VBR model and HBM acquisition, at the working point of nominal torque and 1000rpm.
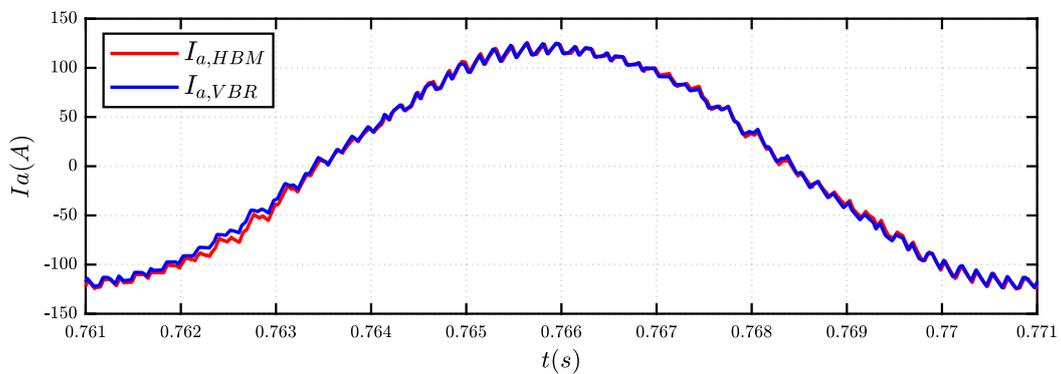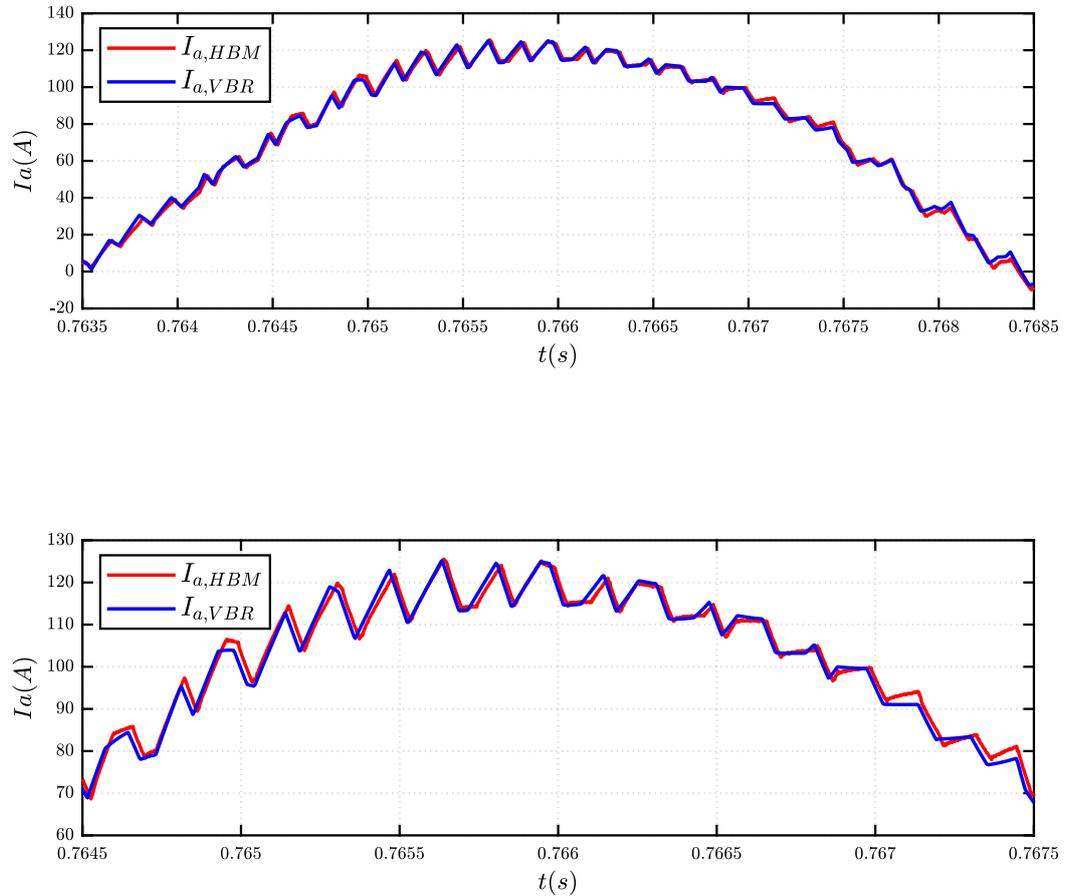
Figure 4.6: Comparison of I_a current between VBR model and HBM acquisition, at the working point of nominal torque and 2000rpm.

In conclusion, both CCG model and VBR model are very accurate in simulating the phase current of the machine.

4.3   Fault condition analysis

The aim of the fault analysis is to verify the accuracy of the Simulink model in simulating the torque and phase currents during a controlled Active Short Circuit. The operating point of the motor before the fault is:

1. speed at 500rpm;

2. currents $I_d = 0A$ and $I_q = 0A$.

The driving machine is simulated like shown in Fig.4.7. The blue block implements a speed loop, using a PI regulator with anti wind-up, which generates the torque to the machine under test, in order to maintain the reference speed (in this case 500rpm). Unfortunately, the parameters of the PI regulator (k_p and k_i) are unknown, so in simulation the gains of the speed loop were roughly tuned, in order to obtain the same transient response of the real motor, but this limits the accuracy of the simulation.
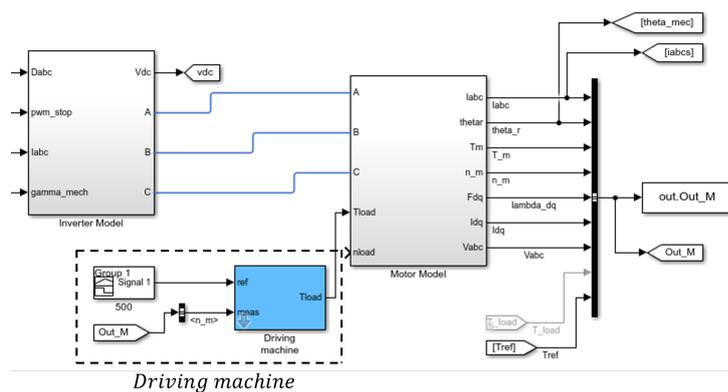
Figure 4.7: Implementation of the driving machine in Simulink model.

The flux maps (and the inverse flux maps) implemented in the Simulink models were obtained experimentally, but due to some difficulties, these maps have a limited domain. The working point of the machine at steady-state of the ASC is exceeding the domain of the maps, so analytical extrapolation is required, limiting the accuracy of the simulation. Also the flux maps and inverse flux maps obtained with Finite Element Analysis were implemented in the Simulink for the comparison of the CCG and VBR models.

Controlled Current Generators

When the ASC occurs, the speed is reduced but then the driving machine sets back the reference speed. This transient is shown in Fig.4.8, in which the measured speed and the simulated speed are compared. Because the parameters of the speed regulator of the driving machine are roughly tuned, the simulation does not replicate exactly the measured speed. The dq currents computed by dSPACE are compared with the dq currents obtained in Simulink in Fig.4.9. At steady state the dq currents are accurately represented but during the transient there is a difference probably due to the analytical extrapolation.
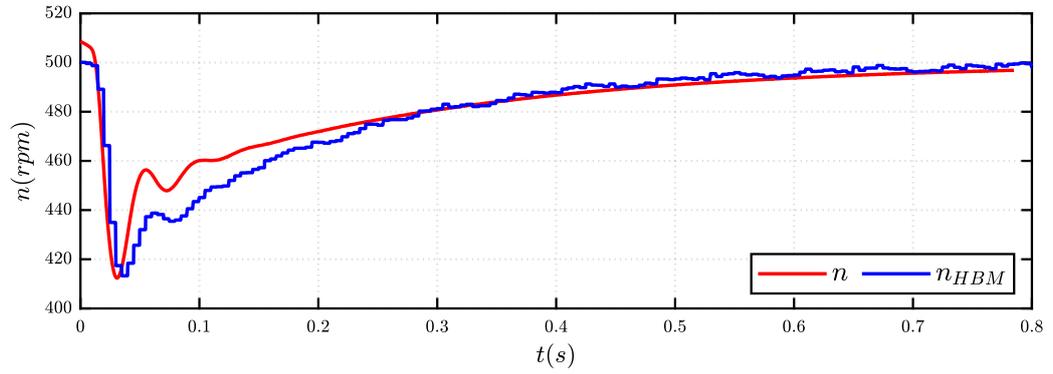
Figure 4.8: Speed measured with HBM and speed in the CCG model.
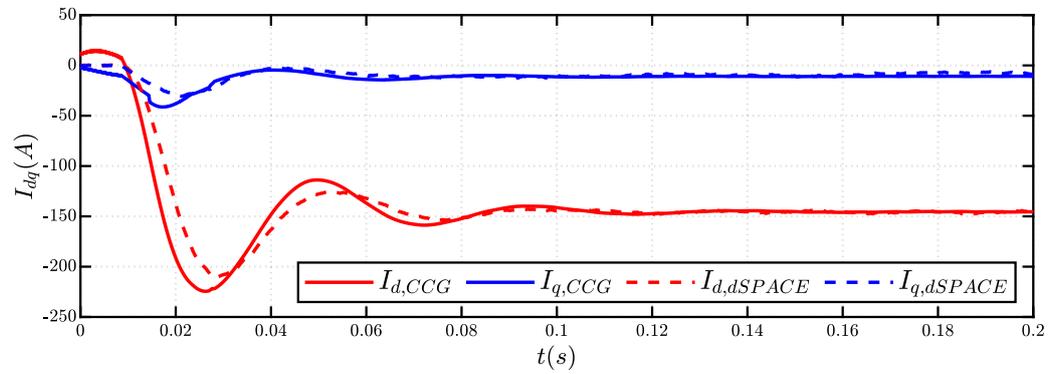
Figure 4.9: *Dq* currents computed in DSpace and *dq* currents in the CCG model.
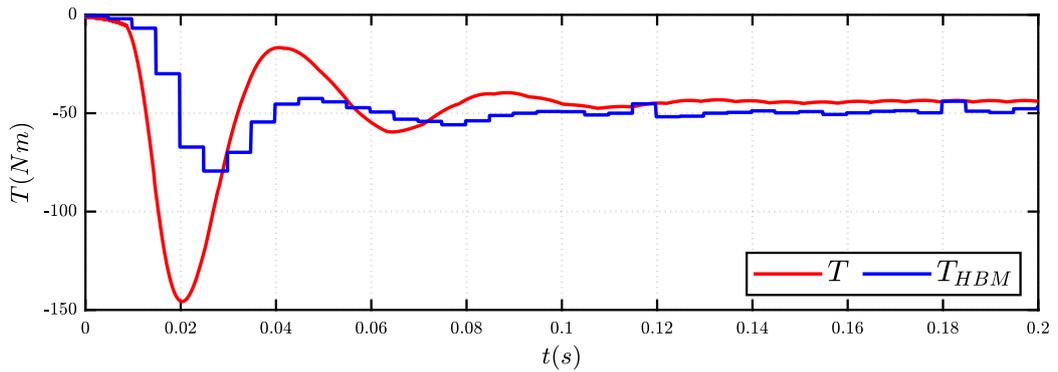
Figure 4.10: Torque measured by HBM and torque in the CCG model.

The torque comparison is shown in Fig.4.10. It must be noted that during the

transient the torque sensor measured the torque at the shaft, which is the machine torque minus the following term: $J \cdot \dot{\omega}$, where J is the total inertia of the system and the $\dot{\omega}$ is the variation of speed during the transient. Then, because the of the extrapolation of the currents, also the simulated torque during the transient is not very reliable, but at steady state there is real little difference with the measured torque. At the end, in Fig.4.11 the measured phase currents are compared with he simulated phase currents, which are pretty accurate, because of the dq currents. The phase shift between the measured and simulated currents is caused by the difference in speed showed in Fig.4.8.
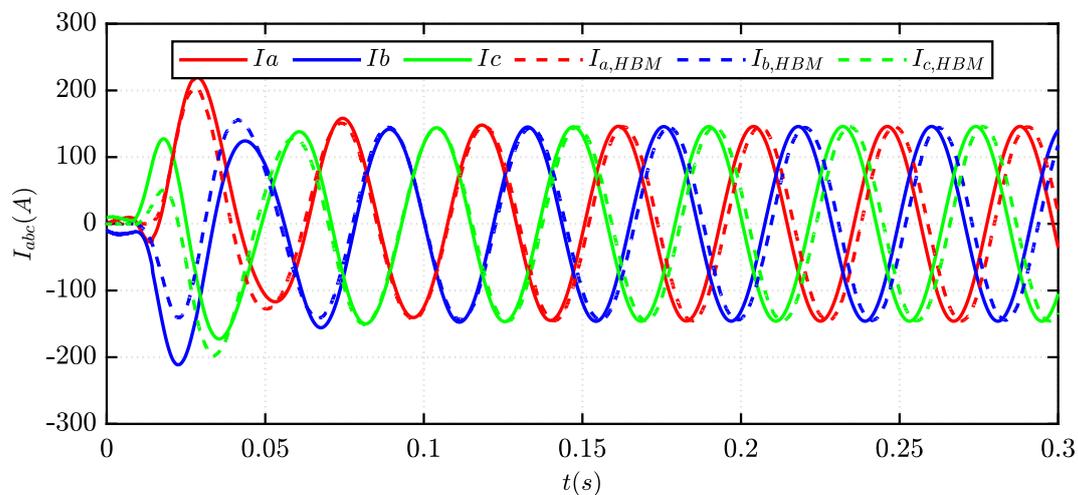
Figure 4.11: Phase currents measured with HBM and phase currents in the CCG model.

Voltage Behind Reactance

The speed transient when the ASC occurs is shown in Fig.4.12. It is still unclear why the simulated speed is so different respect to the speed of the other model. The dq currents computed by dSPACE are compared with the dq currents obtained in Simulink in Fig.4.13. It can be noted that the simulated dq currents are less accurate than the other model during the transient, but represents the correct values at steady state. The torque comparison is shown in Fig.4.14, where it can be noted that the simulated torque does not provide an acceptable representation, because it has some irregularities and it is similar to the torque measurement, which is the torque value of the shaft, not of the motor. These inaccuracies during the transient could probably be caused by the data extrapolation from the maps, but more research needs to be done. At the end, in Fig.4.11 the measured phase currents are compared with he simulated phase currents, which are have some

differences during the transient, due to the difference in the dq currents and the speed.
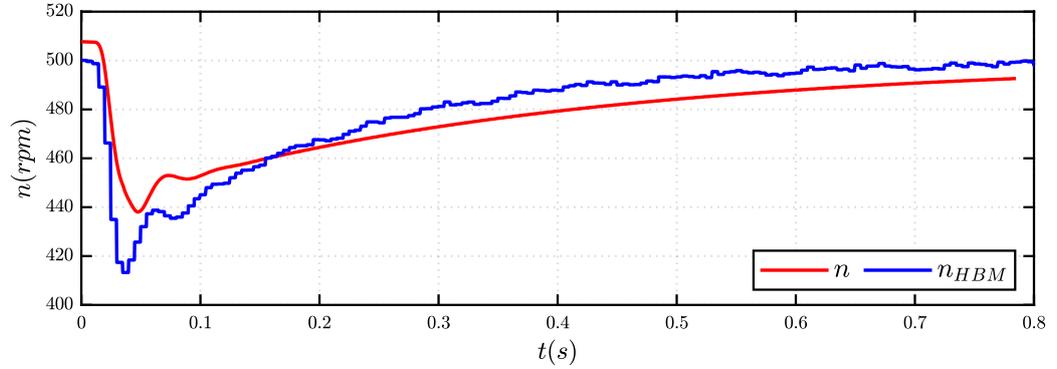
Figure 4.12: Speed measured with HBM and speed in the VBR model.
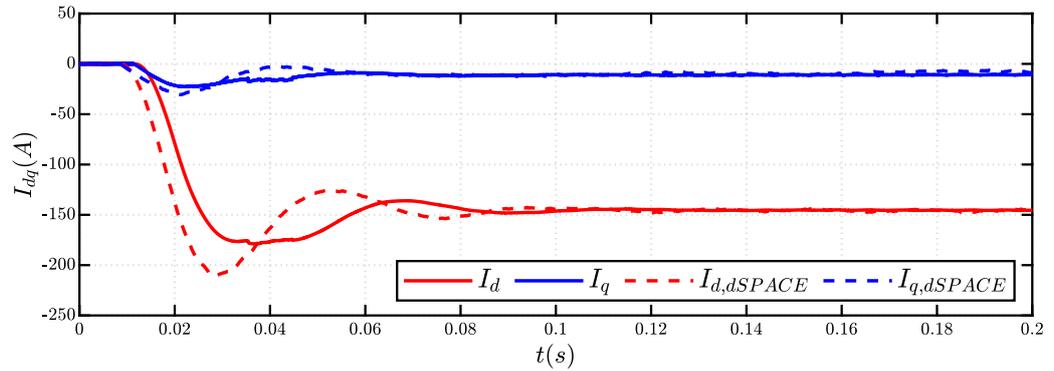
Figure 4.13: Dq currents computed in DSpace and dq currents in the VBR model.
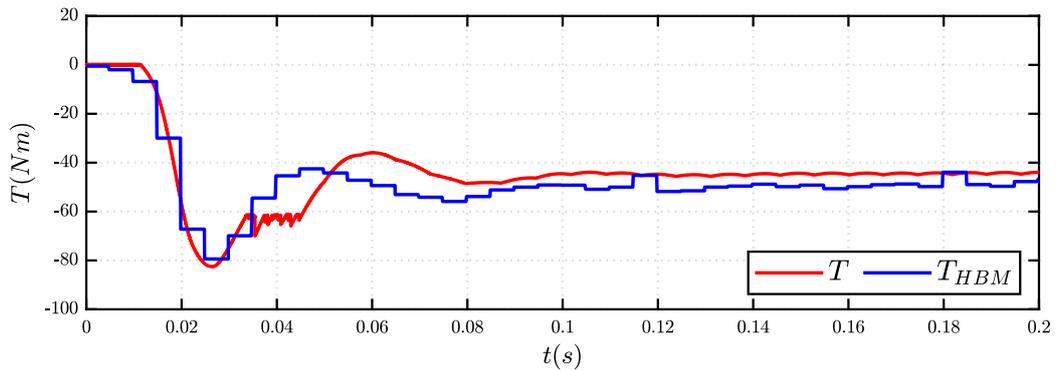
Figure 4.14: Torque measured by HBM and torque in the VBR model.
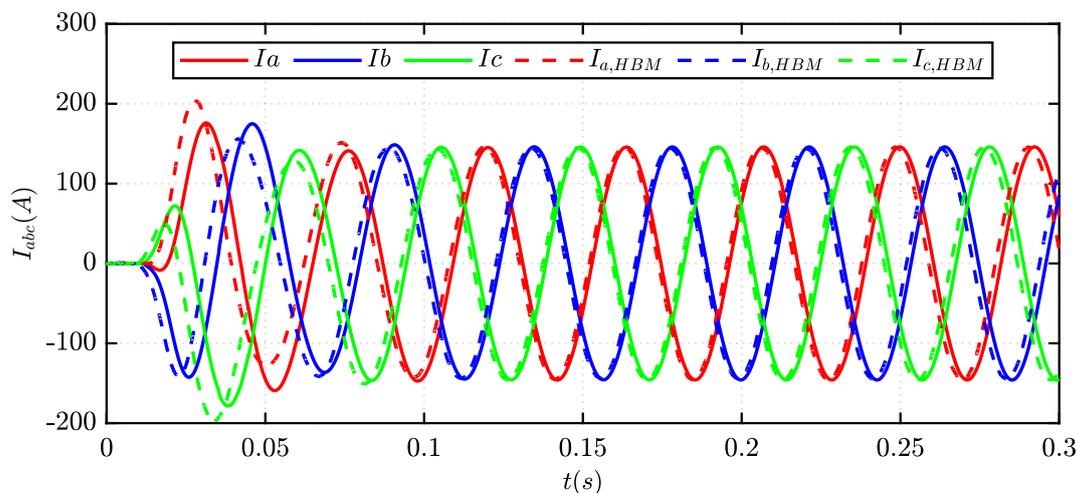
Figure 4.15: Phase currents measured with HBM and phase currents in the VBR model.

In addition, the ASC is simulated also using the FEA flux maps and inverse flux maps and the dq currents and torque waveform are shown respectively in Fig.4.16 and Fig.4.17. As for the experimental maps, the models have different behavior during the transient, but are equal at steady state.
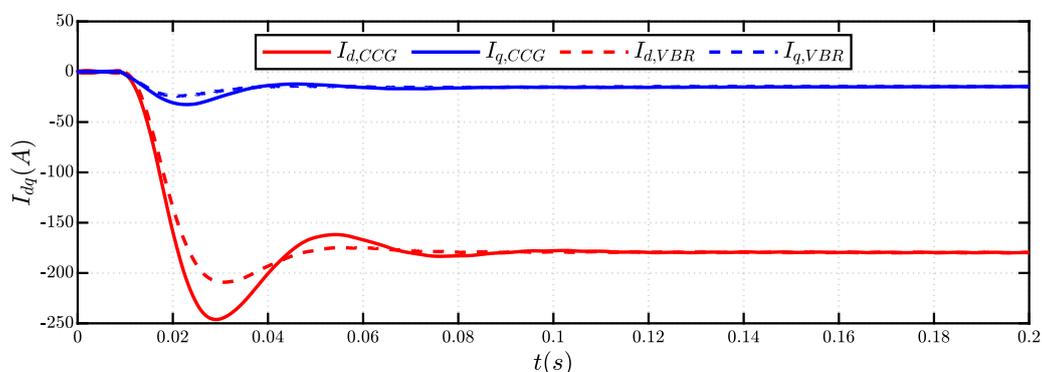
Figure 4.16: Currents I_{dq} waveform of CCG and VBR models, during the ASC.

In conclusion, the CCG model and the VBR model provide the same results regarding the dq currents and torque at steady-state but there is a great difference during the transient, for both the FEA and experimental flux maps. For the purpose of this thesis, the torque comparison is set aside and the dq currents is

taken into account, concluding that the CCG model is more accurate in simulating the ASC circuit.
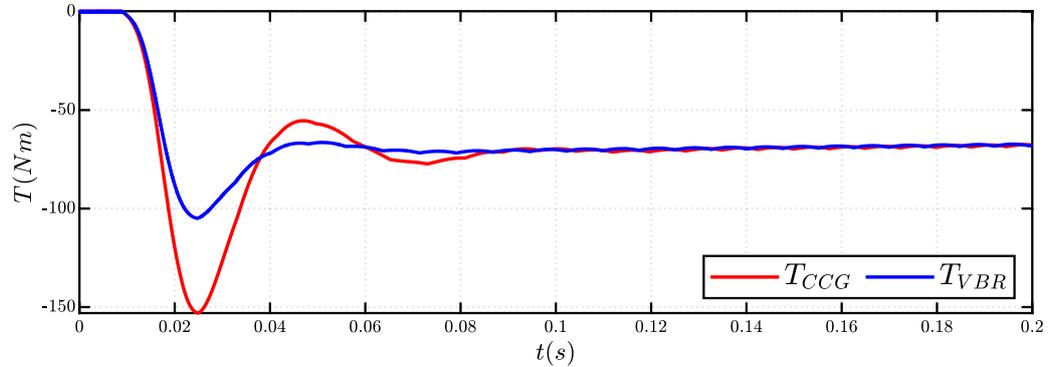
Figure 4.17: Torque waveform of CCG and VBR models, during the ASC.

4.4   Comparison with transient FEA

This section presents a comparison between Simulink and Magnet, a software for FEA analysis. The aim of this comparison is to verify the accuracy of both software in simulating an Open Phase Fault of an PMASR machine.
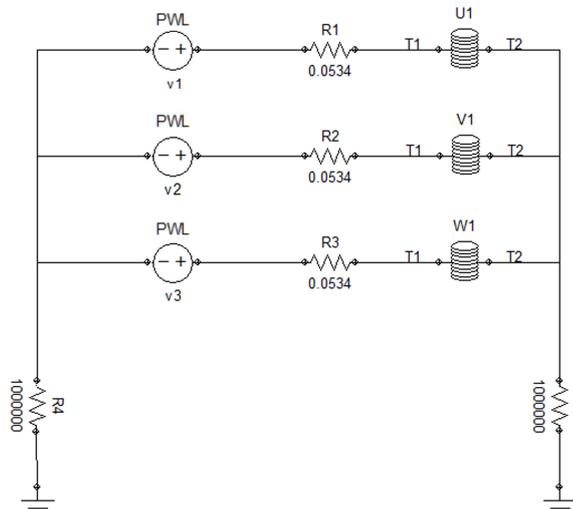
Figure 4.18: Circuit used in Magnet for FEA simulation.

The circuit used in Magnet is shown in Fig.4.18, which is made by three voltage generators supplying an RL load and the inductors are magnetically coupled. First,

the open phase fault is simulated in Simulink. The computed phase voltages are then used to pilot the voltage generators of the Magnet circuit. After the FEA analysis, the software generates as outputs the phase currents and the torque, which are compared to the ones obtained in Simulink.

Controlled Current Generators

The comparison between the phase currents is shown in Fig.4.19, while the comparison of the torque waveform is shown in Fig.4.20. It can be noticed that the currents match very well, while the torques have some discrepancy. This is caused by the fact that Magnet takes into account the presence of the slots, which introduces space harmonics in the currents and fluxes, while the Simulink model is using the dq model, that doesn't take into account the influence of the slots.
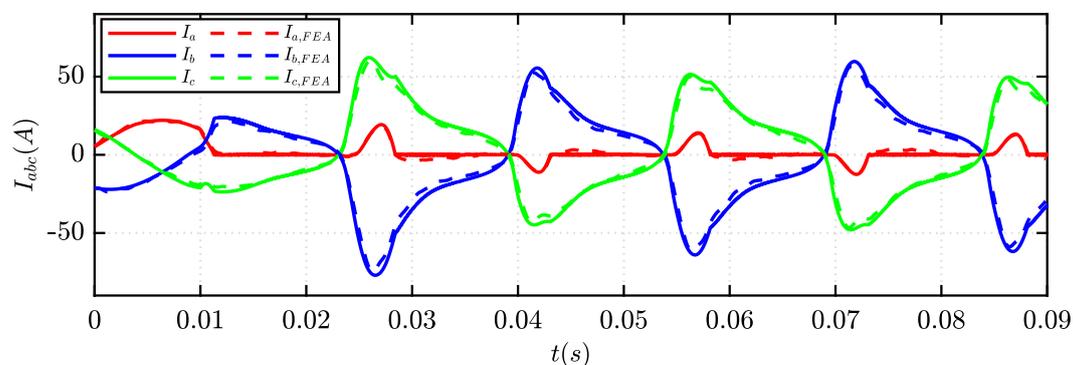
Figure 4.19: Phase currents comparison between Simulink CCG model and Magnet during the open phase fault.
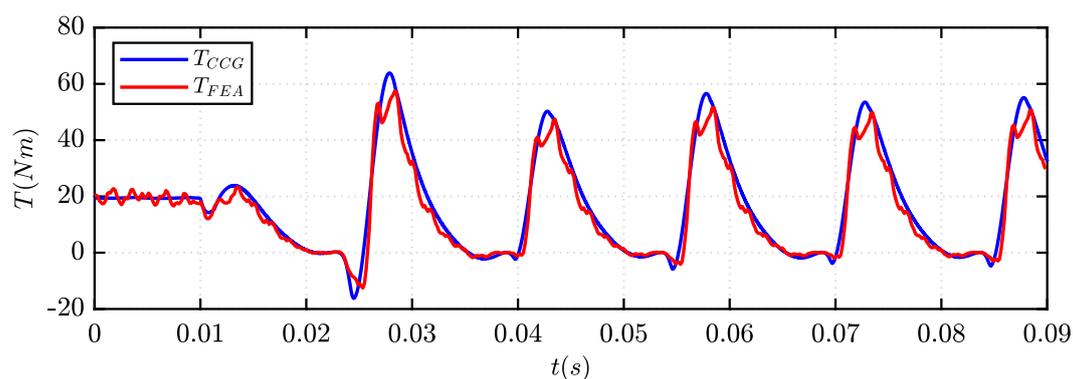
Figure 4.20: Torque comparison Phase between Simulink CCG model and Magnet during the open phase fault.

Voltage Behind Reactance

The comparison between the phase currents is shown in Fig.4.21, while the comparison of the torque waveform is shown in Fig.4.22. It can be noted that the VBR model provides basically the waveforms as the CCG model.
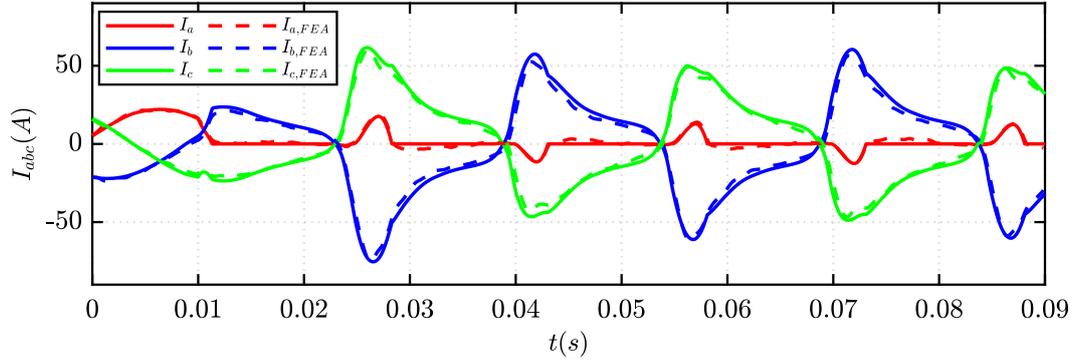
Figure 4.21: Phase currents comparison between Simulink VBR model and Magnet during the open phase fault.
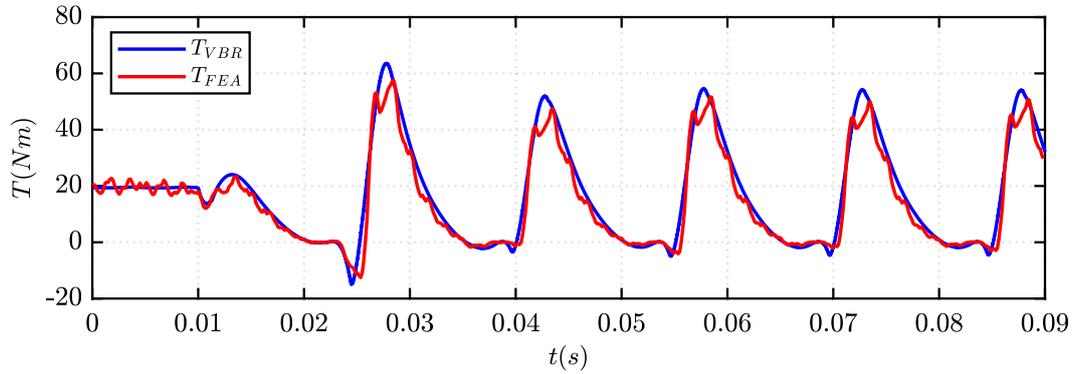
Figure 4.22: Torque comparison Phase between Simulink VBR model and Magnet during the open phase fault.

Chapter 5

Conclusions

The purpose of the thesis was to find a circuital eDrive model, suitable for permanent magnet synchronous machines and synchronous reluctance machines, capable of simulating the behavior of the motor in normal operating conditions and fault conditions. Two modeling approaches were analyzed: the Controlled Current Generators (CCG) and the Voltage Behind Reactance (VBR) models, both implemented in PLECS and Simulink. The latter ones were used for experimental validation, for the analysis of the PWM current ripple and the analysis of a controlled Active Short Circuit.

The simulation results suggest that the CCG model allows faster simulations, but it requires the computation of the inverse flux maps. The VBR model requires the direct flux maps, but it is heavier computational wise.

The experimental validation results suggest that the two model have the same behavior at steady state, but the CCG model is more accurate in simulating the ASC.

In conclusion, the CCG model is chosen to be embedded in SyreDrive. However, both CCG and VBR models still need some refinements. Some ideas for future works are listed below:

1. extension of the model to the $dq\theta$ flux maps;

2. research in the simulation of faults and uncontrolled scenarios;

3. extension of the control algorithm, including new features as flux weakening operating area and Direct Flux Vector Control;

4. extention of the motor models to other types of motors, like induction machines or multi-phase machines;

Appendix A

VBR model mathematical demostration

The voltage equation in dq frame is expressed as:

$$\overline{v}_{dq} = R_s \cdot \overline{i}_{dq} + [l_{dq}] \cdot \frac{d\overline{i}_{dq}}{dt} + [J] \cdot \omega \cdot \overline{\lambda}_{dq} \qquad (A.1)$$

First, the voltage equation is transformed in $\alpha\beta$ axis by applying the inverse rotational matrix:

$$A(-\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \qquad (A.2)$$

obtaining the voltage equation in $\alpha\beta$-axes:

$$A(-\theta) \cdot \overline{v}_{dq} = R_s \cdot A(-\theta) \cdot \overline{i}_{dq} + A(-\theta) \cdot [l_{dq}] \cdot \frac{d\overline{i}_{dq}}{dt} + A(-\theta) \cdot [J] \cdot \omega \cdot \overline{\lambda}_{dq} \quad (A.3)$$

$$\overline{v}_{\alpha\beta} = R_s \cdot \overline{i}_{\alpha\beta} + A(-\theta) \cdot [l_{dq}] \cdot \frac{d\overline{i}_{dq}}{dt} + A(-\theta) \cdot [J] \cdot \omega \cdot \overline{\lambda}_{dq} \qquad (A.4)$$

The second term of the previous equation can be transformed as follows:

$$
\begin{aligned}
A(-\theta) \cdot [l_{dq}] \cdot \frac{d\overline{i}_{dq}}{dt} &= A(-\theta) \cdot [l_{dq}] \cdot \frac{d}{dt}\left(A(-\theta) \cdot \overline{i}_{\alpha\beta}\right) = \\
A(-\theta) \cdot [l_{dq}] \cdot A(\theta) \cdot \frac{d\overline{i}_{\alpha\beta}}{dt} &+ A(-\theta) \cdot [l_{dq}] \cdot \frac{d}{dt} A(\theta) \cdot \overline{i}_{\alpha\beta}
\end{aligned}
\qquad (A.5)
$$

The derivative of the rotation matrix is expressed as:

$$\frac{d}{dt} A(\theta) = \frac{d}{dt}\begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} = -\omega \cdot \begin{bmatrix} sin(\theta) & -cos(\theta) \\ cos(\theta) & sin(\theta) \end{bmatrix} \qquad (A.6)$$

The tensor of incremental inductance in $\alpha\beta$-axes $[l_{\alpha\beta}]$ is defined as:

$$[l_{\alpha\beta}] = A(-\theta) \cdot [l_{dq}] \cdot A(\theta) = \begin{bmatrix} l_{\alpha\alpha} & l_{\alpha\beta} \\ l_{\alpha\beta} & l_{\beta\beta} \end{bmatrix} \tag{A.7}$$

Each component of $[l_{\alpha\beta}]$ is defined as:

$$l_{\alpha\alpha} = l_{dd}cos^2(\theta) + l_{qq}sin^2(\theta) - 2l_{dq}cos(\theta)sin(\theta)$$

$$l_{\alpha\beta} = l_{\beta\alpha} = l_{dq}(cos^2(\theta) - sin^2(\theta)) + l_{dd}cos(\theta)sin(\theta) - l_{dd}cos(\theta)sin(\theta) \tag{A.8}$$

$$l_{\alpha\alpha} = l_{qq}cos^2(\theta) + l_{qq}sin^2(\theta) + 2l_{dq}cos(\theta)sin(\theta)$$

Considering the following trigonometric relations:

$$cos(2\theta) = cos^2(\theta) + sin^2(\theta) = 1 - sin^2(\theta) = 2cos^2(\theta) - 1$$

$$cos^2(\theta) = \frac{1 + cos(2\theta)}{2} \qquad sin^2(\theta) = \frac{1 - cos(2\theta)}{2} \tag{A.9}$$

$$sin(2\theta) = 2cos(\theta)sin(\theta)$$

the tensor $[l_{\alpha\beta}]$ can be written as:

$$[l_{\alpha\beta}] = \begin{bmatrix} l_{avg} + l_{\Delta}cos(2\theta) - l_{dq}sin(2\theta) & l_{dq}cos(2\theta) + L_{\Delta}sin(2\theta) \\ l_{dq}cos(2\theta) + L_{\Delta}sin(2\theta) & l_{avg} - l_{\Delta}cos(2\theta) + l_{dq}sin(2\theta) \end{bmatrix} \tag{A.10}$$

where l_{avg} and l_{Δ} are defined as:

$$l_{avg} = \frac{l_{dd} + l_{qq}}{2} \quad l_{\Delta} = \frac{l_{dd} - l_{qq}}{2} \tag{A.11}$$

Returning back to (A.5):

$$A(-\theta) \cdot [l_{dq}] \cdot \frac{d}{dt} A(\theta) \cdot \bar{i}_{\alpha\beta} = A(-\theta) \cdot [l_{dq}] \cdot (-\omega) \cdot \begin{bmatrix} sin(\theta) & -cos(\theta) \\ cos(\theta) & sin(\theta) \end{bmatrix} \cdot \bar{i}_{\alpha\beta} \tag{A.12}$$

and substituting $\bar{i}_{\alpha\beta} = A(\theta) \cdot \bar{i}_{dq}$, the result is:

$$A(-\theta) \cdot [l_{dq}] \cdot (-\omega) \cdot \begin{bmatrix} sin(\theta) & -cos(\theta) \\ cos(\theta) & sin(\theta) \end{bmatrix} \cdot \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \cdot i_{dq} =$$

$$= [A(-\theta)] \cdot [l_{dq}] \cdot (-\omega) \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \bar{i}_{dq} = A(-\theta) \cdot [l_{dq}] \cdot (-\omega) \cdot [J] \cdot \bar{i}_{dq}$$

(A.13)

Returning to (A.4):

$$\bar{v}_{\alpha\beta} = R_s \cdot \bar{i}_{\alpha\beta} + [l_{\alpha\beta}] \cdot \frac{d\bar{i}_{\alpha\beta}}{dt} + A(-\theta) \cdot [l_{dq}] \cdot (-\omega) \cdot [J] \cdot \bar{i}_{dq} + A(-\theta) \cdot [J] \cdot \omega \cdot \overline{\lambda}_{dq} =$$

$$= R_s \cdot \bar{i}_{\alpha\beta} + [l_{\alpha\beta}] \cdot \frac{d\bar{i}_{\alpha\beta}}{dt} + A(-\theta) \cdot \left([l_{dq}] \cdot (-\omega) \cdot [J] \cdot \bar{i}_{dq} + [J] \cdot \omega \cdot \overline{\lambda}_{dq} \right)$$

(A.14)

The back EMFs in $\alpha\beta$ axes are defines as:

$$\bar{e}_{\alpha\beta} = A(-\theta) \cdot \left([l_{dq}] \cdot (-\omega) \cdot [J] \cdot \bar{i}_{dq} + J \cdot \omega \cdot \overline{\lambda}_{dq} \right) = A(-\theta) \cdot \bar{e}_{dq}$$ (A.15)

The voltage equation in $\alpha\beta$ axes is:

$$\bar{v}_{\alpha\beta} = R_s \cdot \bar{i}_{\alpha\beta} + [l_{\alpha\beta}] \cdot \frac{d\bar{i}_{\alpha\beta}}{dt} + \bar{e}_{\alpha\beta}$$ (A.16)

The (1.13) is transformed in abc axes using the inverse transformation:

$$[T]^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix}$$ (A.17)

obtaining the voltage equation in abc frame:

$$[T]^{-1} \cdot \bar{v}_{\alpha\beta} = R_s \cdot [T]^{-1} \cdot \bar{i}_{\alpha\beta} + [T]^{-1} \cdot [l_{\alpha\beta}] \cdot \frac{d\bar{i}_{\alpha\beta}}{dt} + [T]^{-1} \cdot \bar{e}_{\alpha\beta}$$ (A.18)

$$\bar{v}_{abc} = R_s \cdot \bar{i}_{abc} + [T]^{-1} \cdot [l_{\alpha\beta}] \cdot \frac{d\bar{i}_{\alpha\beta}}{dt} + \bar{e}_{abc}$$ (A.19)

Considering $\bar{i}_{\alpha\beta} = [T] \cdot \bar{i}_{abc}$

$$\bar{v}_{abc} = R_s \cdot \bar{i}_{abc} + [T]^{-1} \cdot [l_{\alpha\beta}] \cdot [T] \cdot \frac{d\bar{i}_{abc}}{dt} + \bar{e}_{abc}$$ (A.20)

The tensor in abc frame $[l_{abc}]$ can be expressed as:

$$[l_{abc}] = [T]^{-1} \cdot [l_{\alpha\beta}] \cdot [T] = \begin{bmatrix} l_{aa} & l_{ab} & l_{ac} \\ l_{ba} & l_{bb} & l_{bc} \\ l_{ca} & l_{cb} & l_{cc} \end{bmatrix} \tag{A.21}$$

where each component is define as:

$$l_{aa} = \frac{2}{3} \qquad\qquad l_{ab} = l_{ba} = \frac{1}{3}\left[-l_{\alpha\alpha} + \sqrt{3}l_{\alpha\beta}\right]$$

$$l_{bb} = \frac{1}{3}\left[\frac{l_{\alpha\alpha}}{2} + \frac{3}{2}l_{\beta\beta} - \sqrt{3}l_{\alpha\beta}\right] \quad l_{ac} = l_{ca} = \frac{1}{3}\left[-l_{\alpha\alpha} - \sqrt{3}l_{\alpha\beta}\right] \tag{A.22}$$

$$l_{cc} = \frac{1}{3}\left[\frac{l_{\alpha\alpha}}{2} + \frac{3}{2}l_{\beta\beta} + \sqrt{3}l_{\alpha\beta}\right] \quad l_{bc} = l_{cb} = \frac{1}{3}\left[\frac{l_{\alpha\alpha}}{2} - \frac{3}{2}l_{\beta\beta}\right]$$

Bibliography

[1] L. Wang, J. Jatskevich and H. W. Dommel, "Re-examination of Synchronous Machine Modeling Techniques for Electromagnetic Transient Simulations," in IEEE Transactions on Power Systems, vol. 22, no. 3, pp. 1221-1230, Aug. 2007, doi: 10.1109/TPWRS.2007.901308.

[2] S. Ferrari, G. Dilevrano, P. Ragazzo and G. Pellegrino, "The dq-theta Flux Map Model of Synchronous Machines," 2021 IEEE Energy Conversion Congress and Exposition (ECCE), 2021, pp. 3716-3723, doi: 10.1109/ECCE47101.2021.9595187.

[3] https://it.mathworks.com/help/physmod/sps/ref/synchronousreluctancemachine.html

[4] Mathworks documentation about the FEM-Parametrized PMSM https://it.mathworks.com/help/physmod/sps/ref/femparameterizedpmsm.html

[5] PLECS User Manual version 4.6 pag.566

[6] PLECS User Manual version 4.6 pag.577