



POLITECNICO DI TORINO

Master Thesis in Electrical Engineering

**syreDrive: A New Add-on to the
Motor Design Framework for
Automated Sensorless Control
Simulation**

Author:

Dario Brunelli

Supervisor:

Prof. Gianmario Pellegrino

Advisors:

Anantaram Varatharajan

Simone Ferrari

December 2020

Abstract

The aim of this work is to provide a general approach for the automatic generation of control parameters and control simulation model to evaluate the control dynamic response valid for any synchronous reluctance (SyR) motor, starting from the given motor data and the Finite Element Analysis (FEA) model of the motor. Different control methods are analyzed, both encoder-relying and sensorless techniques. A field oriented current vector control is employed in all these cases. Dealing with sensorless control, a full-speed control scheme is implemented, based on high frequency voltage signal injection techniques for zero to low speeds region and fundamental excitation-based techniques for high speeds region. Various injection and demodulation schemes are explored for the low-speed model scheme while Active Flux (AF) and Adaptive Projection vector for Position error estimation (APP) are employed for the high-speed model. The automatic control code generation is validated simulating three SyR machines of different sizes.

The work is implemented in the electric motor design platform *SyR-e*, in the form of the new feature *syreDrive*, capable of generating the dynamic model of the electric drive and simulating a default speed and torque sequence for evaluation of the the dynamic response of any selected control scheme and any given SyR motor.

Acknowledgements

I would like to thank my thesis supervisor, professor Gianmario Pellegrino, and my thesis advisors Anantaram Varatharajan and Simone Ferrari for helping me throughout the entire course of the thesis work. Their experience and expertise were the best guidance I could have received.

I am deeply grateful to my parents, as their wise advice always wiped away any discouragement, to my sister and my friends, unconditionally by my side, and to Giorgia, who enlightened my path.

Their support was the most valuable resource for reaching the end of my course of studies. The accomplishment of all this would not have been possible without them.

Thank you.

Dario Brunelli

Contents

| | |
|---|-----|
| List of Tables | III |
| List of Figures | IV |
| 1 Introduction | 1 |
| 1.1 Description of the Work | 1 |
| 1.2 <i>SyR-e</i> and <i>syreDrive</i> | 2 |
| 1.3 Thesis Outline and Contribution | 3 |
| 2 Vector Control of the Synchronous Reluctance Machine | 5 |
| 2.1 Synchronounous Reluctance motor | 5 |
| 2.1.1 SyR Machine Model | 6 |
| 2.1.2 <i>dqt</i> Model | 7 |
| 2.2 Control of SyR Motor | 11 |
| 2.2.1 Field Oriented Control | 12 |
| 2.2.2 Hybrid Flux Observer | 13 |
| 2.2.3 Inductances Online Computation | 14 |
| 3 Position Error Estimation Methods for Sensorless Control | 17 |
| 3.1 High Frequency Pulsating Voltage Injection | 18 |
| 3.1.1 Sinusoidal Pulsating Voltage Injection | 20 |
| 3.1.2 Square Wave Pulsating Voltage Injection | 20 |
| 3.2 Current Demdoulation | 21 |
| 3.2.1 Sinusoidal Injection Case | 21 |
| 3.2.2 Square Wave Injection Case | 22 |

| | | |
|----------|--|-----------|
| 3.3 | Current-Model Flux Demodulation | 23 |
| 3.3.1 | Sinusoidal Injection Case | 23 |
| 3.3.2 | Square Wave Injection Case | 24 |
| 3.4 | Saliency Analysis | 24 |
| 3.5 | Back-EMF-based position estimation methods | 26 |
| 3.5.1 | Active Flux | 26 |
| 3.5.2 | Adaptive Projection Vector for Position Error Estimation | 27 |
| 3.6 | Sensorless Control Techniques Fusion Scheme | 28 |
| 3.7 | Phase Locked Loop | 29 |
| 4 | Automated Generation of the Simulation Model and Results | 31 |
| 4.1 | <i>syreDrive</i> Implementation | 31 |
| 4.1.1 | Simulink Model Template | 36 |
| 4.1.2 | Torque and Speed Control Scheme | 40 |
| 4.2 | Simulation Results for the Encoder-based Control | 41 |
| 4.3 | Automatic Calibration Rules | 44 |
| 4.3.1 | Current loop | 45 |
| 4.3.2 | Speed loop | 46 |
| 4.3.3 | PLL | 46 |
| 4.3.4 | Hybrid flux observer and fusion coefficient | 46 |
| 4.3.5 | High Frequency Voltage Injection Parameters | 47 |
| 4.3.6 | No-Load Operation | 47 |
| 4.3.7 | High-Load operation | 48 |
| 4.3.8 | Calibration Optimization | 55 |
| 4.4 | Simulation Results for the Sensorless Control | 62 |
| 4.4.1 | Low-Speed Range | 62 |
| 4.4.2 | High-Speed Range | 63 |
| 5 | Conclusions | 71 |
| | Bibliography | 75 |

List of Tables

2.1 Elapsed times of a default simulation, using different flux maps models. 10

4.1 Ratings of the motors under test. 45

List of Figures

| | | |
|------|--|----|
| 1.1 | <i>Syr-e</i> working principle [1]. | 2 |
| 2.1 | Common reference frame of a synchronous reluctance motor [3]. . . | 6 |
| 2.2 | Experimentally evaluated flux maps of one of the 4.4 kW SyR motor under test. | 8 |
| 2.3 | Dynamic model of a SyR machine with stator flux as the state variables and inverse dq flux-map LUTs to calculate the stator current. | 8 |
| 2.4 | Mechanical model of the machine using the dq flux linkage model. . . | 9 |
| 2.5 | dqt d -flux maps of the 4.4 kW SyR motor tested for two different rotor positions. $\theta = 0^\circ$ in blue and $\theta = 30^\circ$ in red. | 9 |
| 2.6 | dqt torque maps of the 4.4 kW SyR motor tested for two different rotor positions. $\theta = 0^\circ$ in blue and $\theta = 30^\circ$ in red. | 10 |
| 2.7 | Dynamic model of a SyR machine with stator fluxes as the state variables. The orange block contains two 3D inverse dqt flux-map model LUTs, for the d - and the q -current components, to calculate the stator currents. | 11 |
| 2.8 | Mechanical model of the machine using the dqt flux linkage model. . . | 11 |
| 2.9 | MTPA trajectory of one of the 1.1 kW SyR tested. | 12 |
| 2.10 | Block diagram of the field oriented control scheme. | 13 |
| 2.11 | Block diagram of the $VI\theta$ flux observer scheme. | 14 |
| 2.12 | Graphical representation of apparent and incremental inductances. . . | 15 |
| 3.1 | Current demodulation scheme in the case of sinusoidal injected signal. | 21 |
| 3.2 | Flux demodulation scheme in the case of sinusoidal injected signal. | 23 |

| | | |
|------|---|----|
| 3.3 | Incremental saliency plots of the tested motors <i>Electro Adda</i> , <i>RawP</i> , <i>Bari</i> in the dq -currents plane in relative values. MTPA trajectory is depicted in red. | 25 |
| 3.4 | PLL scheme. | 29 |
| 4.1 | The <i>syreDrive</i> tab in the <i>SyR-e MMM</i> GUI. | 32 |
| 4.2 | The motor data header file automatically printed by the function <i>MMM_print_MotorDataH()</i> | 33 |
| 4.3 | The operating principle <i>syreDrive</i> | 35 |
| 4.4 | The quantities plotted by <i>syreDrive</i> when the control simulation ends. | 37 |
| 4.5 | The Simulink model of the whole system, comprising the difital control (blu block), the inverter model (in yellow) and the motor model (in green). | 38 |
| 4.6 | The motor model as implemented in Simulink, comprising the mechanical (a) and the electromagnetic (b) model. | 39 |
| 4.7 | Scheme of the Field Oriented sensorless control implemented. | 40 |
| 4.8 | Two torque and speed simulated sequences with the position information coming from the encoder. The simulation reported in (a) is torque-controlled, while the one in (b) is controlled in speed. The toruqe values are in p.u. relatively to the rated torque of the motor T_0 | 42 |
| 4.9 | Two graphs representing the same torque control sequence. In the simulation in (a) the dq model was employed, while in (b) the dqt model. | 43 |
| 4.10 | A torque and speed plot of a simulation to test the computation of the observed speed n_{obs} | 44 |
| 4.11 | Saliency maps of the three motors under test. A red marker is placed at 20% of the rated current value on the i_q axis. | 49 |
| 4.12 | Three simulations of the tested motors in torque control mode. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 <i>rpm</i> | 50 |
| 4.13 | Saliency maps, MTPA trajectory (in red) and torque map in <i>p.u.</i> (in black) of the three motors under test. | 52 |

| | | |
|------|---|----|
| 4.14 | A simulation of the <i>Electro Adda</i> motor in torque control mode employing flux demodulation and a PLL bandwidth of $2\pi \cdot 25 \text{ rad/s}$. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 rpm . The reference torque is lost around $T = T_0$ | 53 |
| 4.15 | A simulation of the <i>Electro Adda</i> motor in torque control mode employing current demodulation and a PLL bandwidth of $2\pi \cdot 25 \text{ rad/s}$. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 rpm . The reference torque is lost around $T = 0.75 T_0$ | 54 |
| 4.16 | Two simulations of the <i>Electro Adda</i> motor in torque control mode employing flux demodulation. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 rpm . In the plots in (a) the bandwidth of the PLL is set to $2\pi \cdot 15 \text{ rad/s}$, while in (b) it is set to $2\pi \cdot 10 \text{ rad/s}$. . . | 55 |
| 4.17 | The control parameters as they appear in the <i>User_Constants.h</i> header file. | 56 |
| 4.18 | Two simulations in speed control mode of the <i>Bari</i> motor. The one in (a) was performed using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, while the one in (b) using $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$ | 58 |
| 4.19 | Two simulations in speed control mode of the <i>Electro Adda</i> motor. The one in (a) was performed using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, while the one in (b) using $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$ | 59 |
| 4.20 | Two simulations in speed control mode of the <i>RawP</i> motor. The one in (a) was performed using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, while the one in (b) using $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$ | 60 |
| 4.21 | A simulation in speed control mode of the <i>Electro Adda</i> motor, using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1.75 \text{ rad/s}$ | 61 |
| 4.22 | A simulation in speed control mode of the <i>RawP</i> motor, using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1.8 \text{ rad/s}$ | 61 |
| 4.23 | Four torque and speed plots of torque control simulations using different low-speed position error estimation methods. | 64 |

| | | |
|------|---|----|
| 4.24 | From top to bottom: the actual torque, the observed torque and the load torque in <i>p.u.</i> ; the reference speed, the actual speed and the observed speed in <i>rpm</i> ; the fusion coefficient f_ω ; the real position θ and the observed position $\hat{\theta}$ in <i>rad</i> ; the position error $\tilde{\theta}$ in degrees. The plots are taken from a simulation in speed control employing high frequency voltage injection and flux demodulation in the low-speed region, and APP during the high-speed operations. | 66 |
| 4.25 | From top to bottom: the actual torque, the observed torque and the load torque in <i>p.u.</i> ; the reference speed, the actual speed and the observed speed in <i>rpm</i> ; the fusion coefficient f_ω ; the real position and the observed position; the position error. The plots are taken from a simulation in speed control employing high frequency voltage injection and flux demodulation in the low-speed region, and active flux during the high-speeds operation. Notice that a minimum torque load is applied through out the whole simulation. | 68 |
| 4.26 | From top to bottom: the actual torque, the observed torque and the load torque in <i>p.u.</i> ; the reference speed, the actual speed and the observed speed in <i>rpm</i> ; the fusion coefficient f_ω ; the real position and the observed position; the position error. The plots are taken from a simulation in speed control employing high frequency voltage injection and flux demodulation in the low-speed region, and active flux during the high-speeds operation. No torque load is applied in this case. | 69 |

Chapter 1

Introduction

This work aims at developing an automated procedure to evaluate the performance of synchronous reluctance motors through auto-generated control simulations within the motor design framework, provided the parameters and the flux maps of the motor.

The main motivations are:

1. To build a general control algorithm that is independent of the machine specifications.
2. To provide a quick control performance evaluation for machine design prototypes.

1.1 Description of the Work

A template of control files that is general in nature is generated and automatically customized with the data of the motor under consideration. The same approach is used for the simulation: a general Simulink model, suitable for any synchronous reluctance motor, is loaded with the motor data and the suitable C-files of the control, then the simulation is run.

The control provides for torque or speed control modes and can be encoder-relying or position sensorless. Multiple sensorless control methods are integrated and available to the user in a simple GUI. The user will be able to adjudicate the results of

the simulation with the selected control type.

This tool will provide a first judgement parameter for the control strategy of a freshly designed motor or any machine for which the control is to be evaluated; in particular, it will be an instrument of assessment for considering the implementation of position sensorless control, which is a focus of this work.

The control files generated are portable to any floating-point 32-bit microcontroller with minor modifications and are suitable to be implemented in a digital control and utilized for practical tests or for further dedicated tuning of the control.

1.2 *SyR-e* and *syreDrive*

SyR-e (Synchronous Reluctance - evolution) is a Matlab and FEMM based software for design and simulation of synchronous motors [1].

SyR-e interacts with its main client FEMM [2] for static magnetic FEA and embeds design equations, mainly but not only for magnetic design, besides a number of other possibilities.

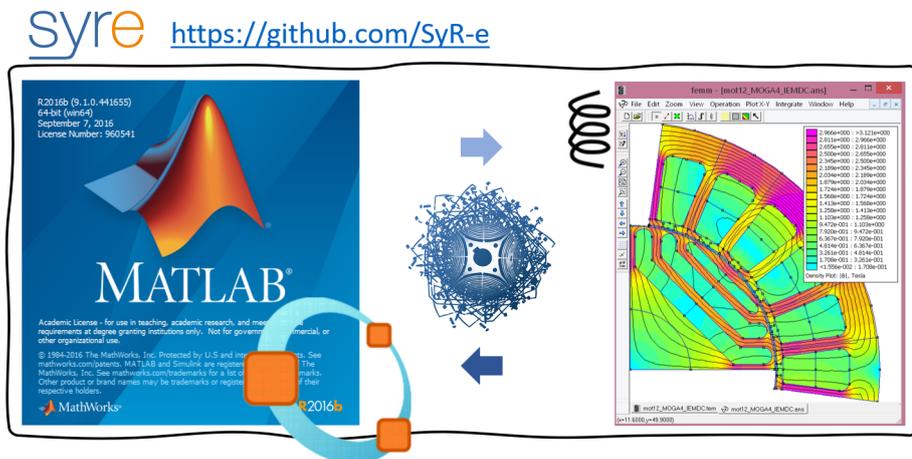


Figure 1.1: *SyR-e* working principle [1].

The work developed in the course of this thesis is supplemented to *SyR-e* in the form of a new add-on to this motor design tool. Given a motor designed in *SyR-e*, the new *syreDrive* feature will automatically build a Simulink model comprising the

electric motor, inverter and current vector control blocks for control development purposes. This will enable the fast and seamless transition from the motor design environment to the digital control environment.

1.3 Thesis Outline and Contribution

In the first section of Chapter 2, a general overview on synchronous reluctance motor is provided and its fundamental mathematical model is presented. *dqt* flux maps model is described and a simplified *dqt* model which reduces the computational time during simulation is introduced. In Section 2.2 the control strategy for SyR machines is outlined. The flux observer employed in the control is described and the online computation of apparent and incremental inductances is discussed. Chapter 3 is dedicated to the position sensorless techniques employed. The position error estimation methods are classified as low-speed region and high-speed region techniques. The former rely upon the saliency of the machine and the injection of a high frequency voltage signal, while the latter are based on the fundamental model of the motor. A fusion scheme of the two speed regions techniques frames a full-speed sensorless position estimation algorithm. The phase-locked loop (PLL) employed to track the position is described.

Chapter 4 deals with describing the fundamental blocks which constitute the core of this work: the Simulink model for the control simulation and the structure of the control code. The calibration of the control parameters is dealt with. The automatization of the process and the implementation of the work in *SyR-e* are described. The results of the simulations are presented and discussed.

Finally the conclusions are drawn in Chapter 5.

My personal contributions to this work regarded:

- review of technical literature on traditional and sensorless control algorithms for SyR motors;
- comparative analysis of multiple sensorless position error estimation techniques, including low- and high-speed range schemes;

- automatic calibration of the full speed sensorless control scheme, with different options for the low- and high- speed schemes;
- validation of the above findings for different motor sizes;
- implementation of the work in *SyR-e* in the form of a new add-on.

Chapter 2

Vector Control of the Synchronous Reluctance Machine

2.1 Synchronous Reluctance motor

Nowadays, synchronous reluctance (SyR) motors are an established technology in the electric motors market because of the advantages they demonstrate over other types of motors. Their performance mainly depends on the cutting geometry of the rotor, as the shape of the flux barriers heavily affects the saliency ratio of the machine. Their rotor structure makes them simple to manufacture, with no windings required, and cheap to produce because of the absence of permanent magnets (PM). Rotor windings in induction motors and magnets in PM synchronous machines make very high speed applications challenging for this kind of motors because of mechanical complications due to the significant centrifugal force their rotors undergo. The rotor of synchronous reluctance motors solely consists of punched iron, which makes them excellent candidates when it comes to spinning at very high speeds. Furthermore, their rotor produces considerably lower losses than the one of induction machines, reducing the complication of cooling the rotor and increasing the capability of the motor to produce torque. Moreover, stator configuration of SyR motors is the same as the one of induction and PM synchronous motors; this is an advantage for manufacturers who decide to produce more than one type of

motor.

2.1.1 SyR Machine Model

Unlike all other synchronous machines, the d -axis of the dq synchronous rotor reference frame of the SyR motors is aligned along the direction of minimum permeance, as shown in Fig. 2.1. Space vectors in the stationary reference frame are denoted by subscript $\alpha\beta$.

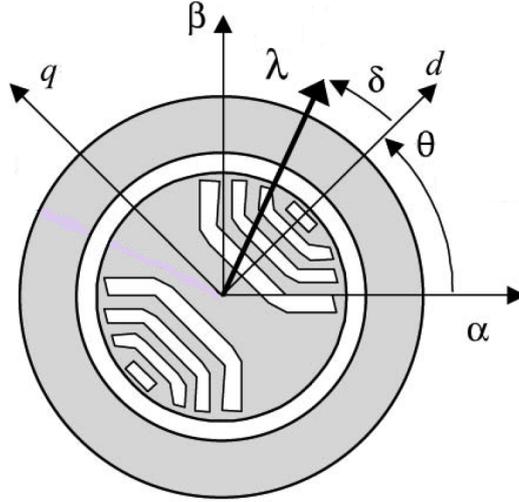


Figure 2.1: Common reference frame of a synchronous reluctance motor [3].

The voltage equations of a SyR machine expressed in the rotor coordinates are:

$$s\boldsymbol{\lambda}_{dq} = \mathbf{v}_{dq} - R_s \mathbf{i}_{dq} - \omega \mathbf{J} \boldsymbol{\lambda}_{dq} \quad (2.1)$$

where s is the differential operator d/dt , R_s is the stator resistance, $\boldsymbol{\lambda}_{dq} = [\lambda_d \lambda_q]^T$ is the stator flux linkage space vector, ω is the electrical angular speed and $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is the orthogonal rotational matrix.

The electromagnetic torque is given by:

$$T = \frac{3}{2} p (\boldsymbol{\lambda}_{dq} \times \mathbf{i}_{dq}) \quad (2.2)$$

Where p is the number pole pairs.

The flux linkage and its derivative in terms of incremental inductance \mathbf{L}_∂ and apparent inductance \mathbf{L} are the following:

$$\boldsymbol{\lambda}_{dq} = \mathbf{L} \mathbf{i}_{dq} \quad (2.3a)$$

$$s\boldsymbol{\lambda}_{dq} = \mathbf{L}_\partial s \mathbf{i}_{dq} \quad (2.3b)$$

The matrices \mathbf{L}_∂ and \mathbf{L} are reported in Equations (2.4). Matrix \mathbf{L}_∂ consists of the incremental inductances l_d and l_q , respectively the incremental inductance along direct d and quadrature q axis, and the cross-saturation term l_{dq} . Apparent inductances are defined likewise.

$$\mathbf{L}_\partial(\mathbf{i}_{dq}) = \begin{bmatrix} l_d & l_{dq} \\ l_{dq} & l_q \end{bmatrix} \quad \mathbf{L}(\mathbf{i}_{dq}) = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \quad (2.4)$$

To accurately describe saturation and cross-saturation effects and to achieve precise torque estimation at zero speed, the current-to-flux linkage maps, denoted with $\boldsymbol{\Lambda}$, should be stored in the form of lookup tables. They can be automatically evaluated using methods such as self-commissioning, described in [4], [5] and [6], or determined by means of finite element analysis. Experimental flux maps $\boldsymbol{\lambda}_{dq} = \boldsymbol{\Lambda}(\mathbf{i}_{dq})$ of one of the motor examined later in this work are represented in figure 2.2.

The electromagnetic model is represented in Fig. 2.3. Here, an inverse magnetic model is employed. It consists of flux-to-current maps, obtained offline from the direct model and stored in lookup tables.

The mechanical model equation is:

$$T = B \omega_r + J \frac{d\omega_r}{dt} + T_L \quad (2.5)$$

Where B is the viscous friction coefficient, J is the equivalent inertia of the rotor and load, ω_r is the mechanical angular speed, $\omega_r = s\theta_r$, where θ_r is the rotor position and T_L is the load torque.

The mechanical block model is represented in Fig. 2.4.

2.1.2 *dqt* Model

To take into account secondary space and slot harmonics and better represent the response of the motor to excitation, *dq-theta* model, or *dqt*, is implemented in the

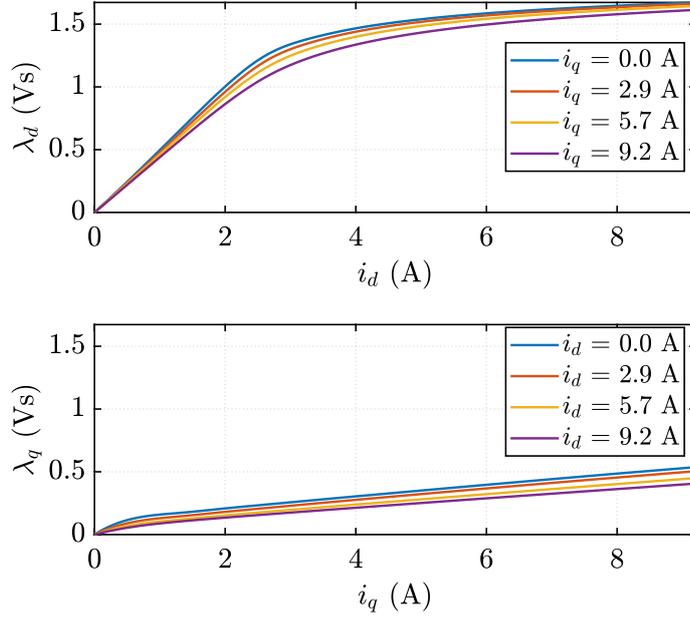


Figure 2.2: Experimentally evaluated flux maps of one of the 4.4 kW SyR motor under test.

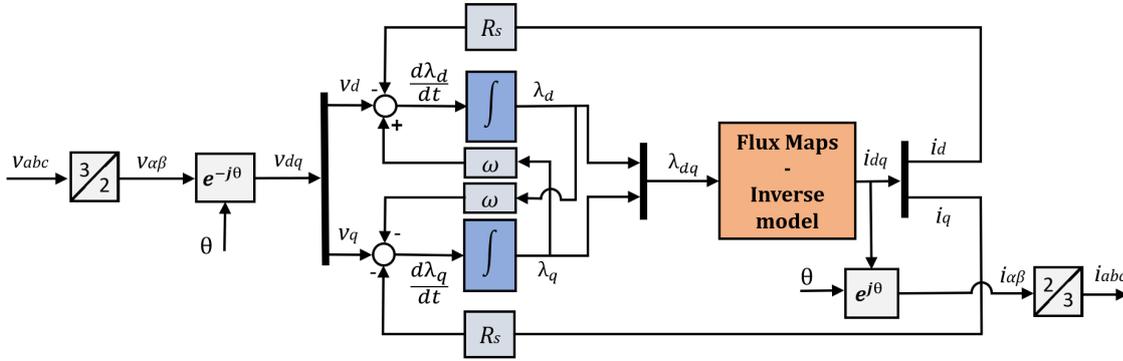


Figure 2.3: Dynamic model of a SyR machine with stator flux as the state variables and inverse dq flux-map LUTs to calculate the stator current.

control [24] [25].

This model considers the flux to be function of dq -currents and of the rotor position $\lambda_{dq}(\mathbf{i}_{dq}, \theta_r)$, accounting in this way of the influence of the slots. Fig. 2.5 shows the d -flux map at $\theta_r = 0^\circ$ in blue, and $\theta_r = 30^\circ$ in red.

When using dqt model, the *Flux Maps - Inverse model* block of the electromagnetic

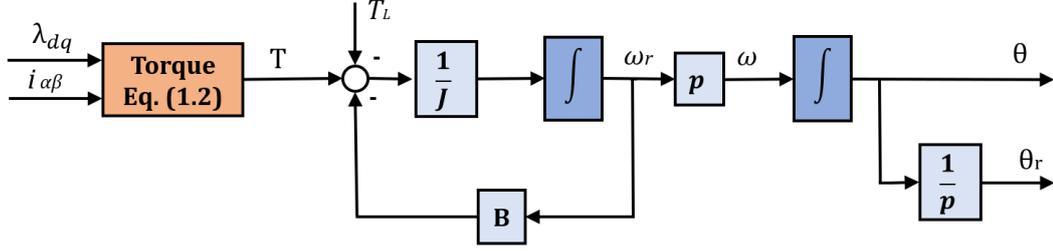


Figure 2.4: Mechanical model of the machine using the dq flux linkage model.

model represented in Fig. 2.3 is substituted by two 3D-lookup tables, one for each current, containing the inverse model data and function of dq flux and rotor position $\mathbf{I}_d(\lambda_{dq}, \theta)$ and $\mathbf{I}_q(\lambda_{dq}, \theta)$. The block diagram of the electromagnetic equations employing the dqt model is represented in Fig. 2.7

While regarding the mechanical model, the *Torque Equation* block in Fig. 2.4 is substituted by the 3D-lookup table $\mathbf{T}(\lambda_{dq}, \theta)$, shown in 2.8. Torque is then determined by interpolating a dqt Torque map, as a function of the dq flux and the rotor position. Fig. 2.6 shows two torque maps at two different rotor positions, $\theta = 0^\circ$ in blue, and $\theta = 30^\circ$ in red.

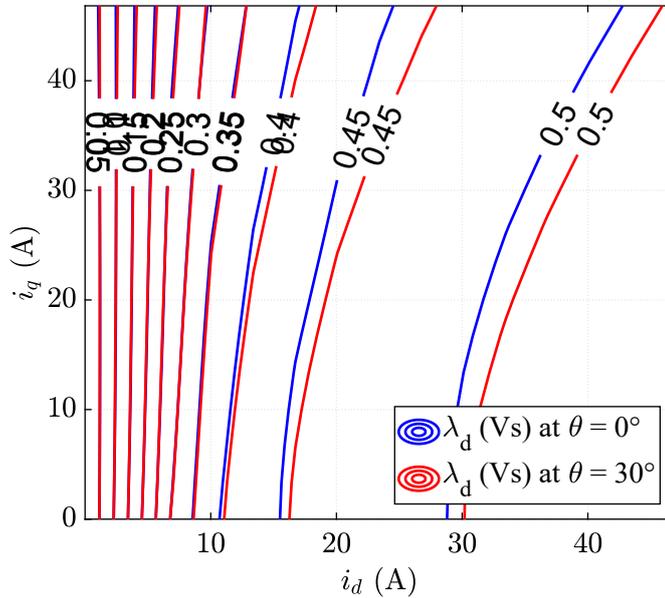


Figure 2.5: dqt d -flux maps of the 4.4 kW SyR motor tested for two different rotor positions. $\theta = 0^\circ$ in blue and $\theta = 30^\circ$ in red.

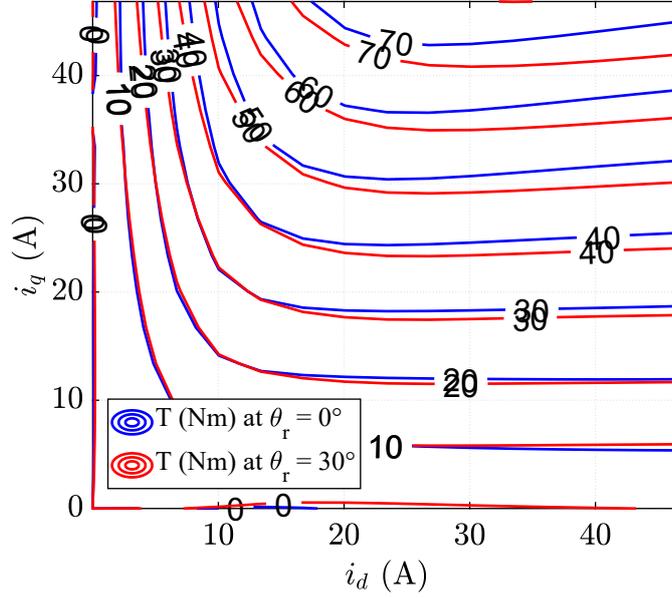


Figure 2.6: dqt torque maps of the 4.4 kW SyR motor tested for two different rotor positions. $\theta = 0^\circ$ in blue and $\theta = 30^\circ$ in red.

The flux maps computed are then three-dimensional data structures. Reading these 3D-lookup tables may be computationally demanding. To reduce the simulation time, the lookup tables of the dqt model are cut to the symmetrical angular span of periodicity of the machine. The simplified dqt maps represent only the data in the angle span of 0° to 60° as opposed to the complete dqt map with the span of 0° to 360° . Using this simplified dqt model, the dimensions of the flux maps are considerably reduced, as is the computational time.

Three simulation times are reported in the table below. The first one from a simulation which employed dq model, the second from a simulation using a traditional dqt model and the last one from a simulation which used the simplified dqt model with the lookup tables of the flux maps cut.

| Flux Maps Model | Elapsed time (s) |
|------------------|------------------|
| dq | 54.2 |
| dqt | 178.4 |
| Simplified dqt | 124.7 |

Table 2.1: Elapsed times of a default simulation, using different flux maps models.

It is evident that the simplified dqt flux maps model reduces the time required for the simulation by 30 %.

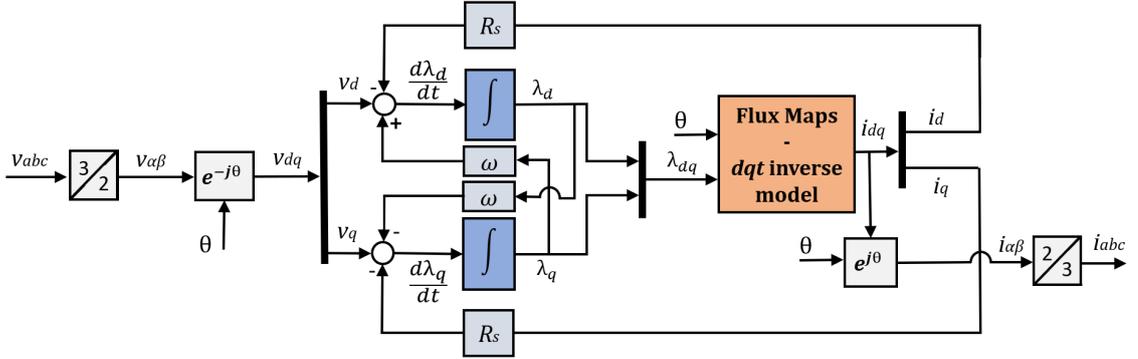


Figure 2.7: Dynamic model of a SyR machine with stator fluxes as the state variables. The orange block contains two 3D inverse dqt flux-map model LUTs, for the d - and the q -current components, to calculate the stator currents.

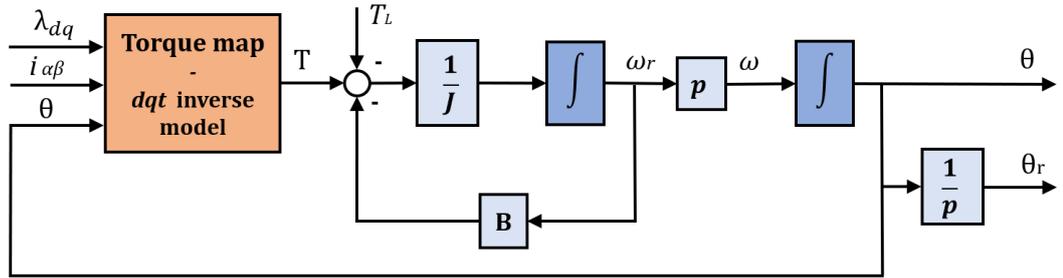


Figure 2.8: Mechanical model of the machine using the dqt flux linkage model.

2.2 Control of SyR Motor

Many control strategies for synchronous reluctance motors are available in literature, such as direct torque control (DTC), direct flux vector control (DFVC) [7], [8], [9] or field oriented control.

As this work aims at laying a general groundwork for developing of control for any SyR machine, field oriented control is selected to be implemented for its simplicity.

2.2.1 Field Oriented Control

Field Oriented Control (FOC) is implemented. d - and q -current values are directly regulated to match the reference values using two PI current regulators to determine the reference voltage [10]. Maximum Torque Per Ampere (MTPA) strategy is adopted for determining the current reference values. The MTPA trajectory optimizes the output torque at any given inverter current, approximating maximum efficiency when neglecting iron losses. This approximation may not be valid at high speed operations, as iron losses are proportional to ω^2 . Implementing the current control along MTPA requires the knowledge of the MTPA locus, i.e. the correct correlation between reference torque T^* and reference d and q currents i_d^* and i_q^* , which is stored in the form of two lookup tables, shown in Fig. 2.9.

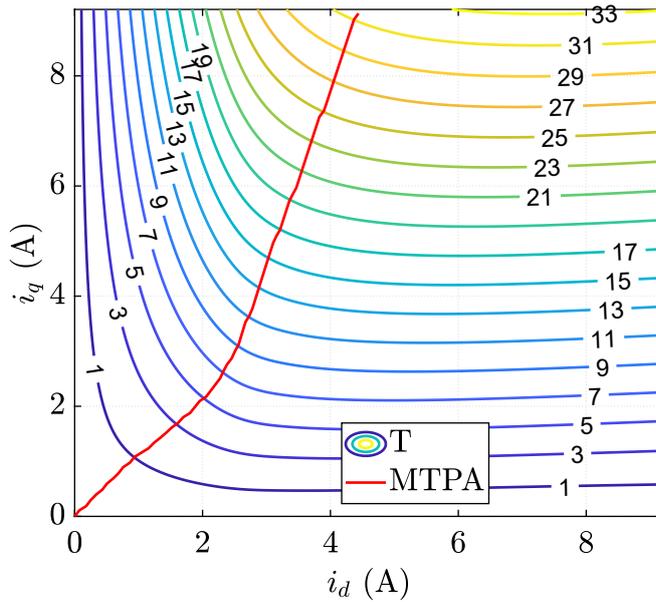


Figure 2.9: MTPA trajectory of one of the 1.1 kW SyR tested.

The torque reference T^* is obtained from the speed PI controller that is cascaded to the current controllers. The reference d and q voltages are compensated with the back-EMF feedforward calculated with the observed flux in the flux observer. The FOC scheme is represented in Fig. 2.10. The flux observer in the yellow block

in the figure is discussed in the next section.

The maximum voltage is limited to $v_{dc}/\sqrt{3}$ where v_{dc} is the DC-link voltage, while current amplitude is automatically bounded by limiting torque to the maximum value that the motor can withstand.

The position information needed for the determination of the dq reference frame and the speed calculation is provided by an encoder. Sensorless control is discussed in the next chapter.

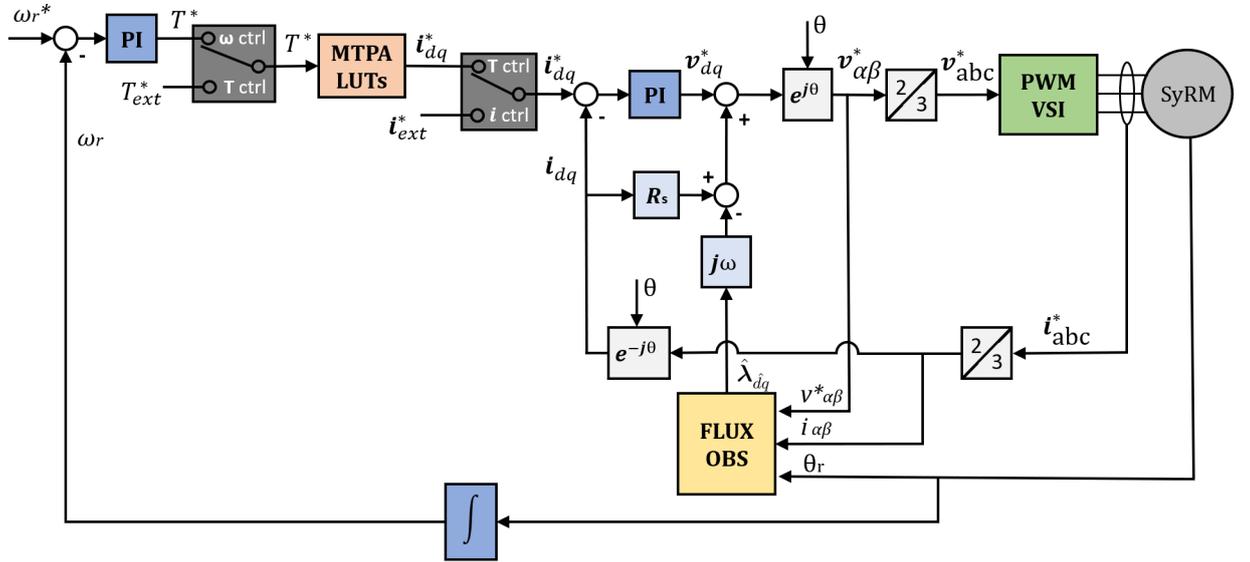


Figure 2.10: Block diagram of the field oriented control scheme.

2.2.2 Hybrid Flux Observer

A $VI\theta$ hybrid flux observer (HFO) is employed to estimate the stator flux linkage $\hat{\lambda}$. The term hybrid indicates that the flux observer is based on the magnetic model flux-map LUTs (current-model) at low speed and on back-electromotive force integration (voltage model) at high speed [3], and the crossover frequency between the two models coincides with the observer gain g . The flux observer is implemented in stator reference frame as in its scheme reported in Fig. 2.11.

The current model apparent inductance L^i in estimated rotor reference frame computed with the flux map lookup tables is obtained as $L^i(\mathbf{i}_{\hat{d}q}) \cdot \mathbf{i}_{\hat{d}q} = \Lambda(\mathbf{i}_{\hat{d}q})$, where Λ is the flux linkage map shown in Fig. 2.2.

The state equation of the hybrid flux observer is then defined as

$$s\hat{\lambda}_{\alpha\beta} = v_{\alpha\beta} - R_s i_{\alpha\beta} + \mathbf{G} \left(e^{J\theta} L^i e^{-J\theta} i_{\alpha\beta} - \hat{\lambda}_{\alpha\beta} \right) \quad (2.6)$$

where $\mathbf{G} = g\mathbf{I}$ and \mathbf{I} is the 2x2 diagonal matrix.

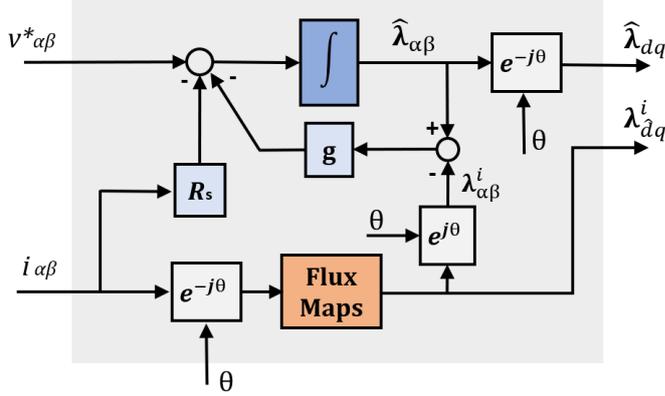


Figure 2.11: Block diagram of the $VI\theta$ flux observer scheme.

2.2.3 Inductances Online Computation

The value of the apparent and incremental inductances depend on the operating point. To gain a better accuracy in the position error estimation methods presented in the following chapter, apparent and incremental inductances are calculated online.

Apparent inductances are defined as the ratio of the flux linkage and the relative-axis current, while the incremental inductances are the partial derivative of the flux linkage respect to the current, evaluated for any operating point $i_{dq0} = [i_{d0} \ i_{q0}]^T$ as depicted in Fig. 2.12.

$$\begin{cases} L_d^i(\mathbf{i}_{dq0}) = \frac{\lambda_d^i(\mathbf{i}_{dq0})}{i_{d0}} \\ L_q^i(\mathbf{i}_{dq0}) = \frac{\lambda_q^i(\mathbf{i}_{dq0})}{i_{q0}} \end{cases} \quad (2.7)$$

$$\left\{ \begin{array}{l} l_d^i(\mathbf{i}_{dq0}) = \frac{\lambda_d^i(i_{d0} + \partial i_d, i_{q0}) - \lambda_d^i(i_{d0}, i_{q0})}{\partial i_d} \\ l_q^i(\mathbf{i}_{dq0}) = \frac{\lambda_q^i(i_{d0}, i_{q0} + \partial i_q) - \lambda_q^i(i_{d0}, i_{q0})}{\partial i_q} \\ l_{dq}^i(\mathbf{i}_{dq0}) = \frac{\lambda_d^i(i_{d0}, i_{q0} + \partial i_q) - \lambda_d^i(i_{d0}, i_{q0})}{\partial i_q} \end{array} \right. \quad (2.8)$$

Where $\boldsymbol{\lambda}_{dq0}^i = [\lambda_{d0}^i \ \lambda_{q0}^i]^T = \boldsymbol{\Lambda}(\mathbf{i}_{dq0})$ and ∂i is a small delta of current equal to 10 mA. Note that $l_{dq}^i = l_{qd}^i$.

The apparent and incremental inductances matrices are

$$\mathbf{L}^i(\mathbf{i}_{dq0}) = \begin{bmatrix} L_d^i & 0 \\ 0 & L_q^i \end{bmatrix} \quad (2.9a)$$

$$\mathbf{L}_{\partial}^i(\mathbf{i}_{dq0}) = \begin{bmatrix} l_d^i & l_{dq}^i \\ l_{dq}^i & l_q^i \end{bmatrix} \quad (2.9b)$$

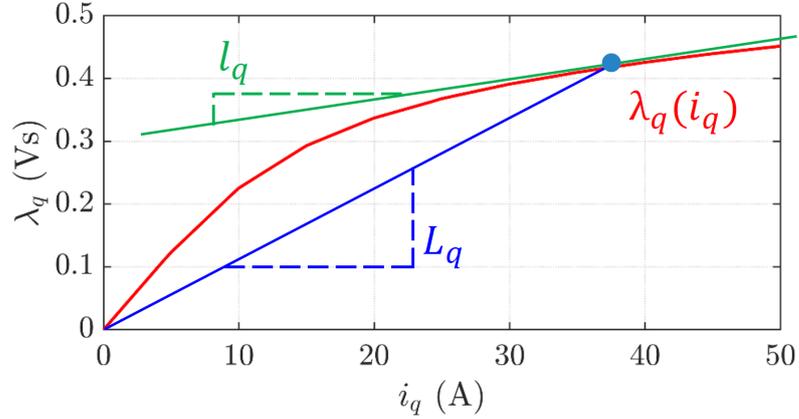


Figure 2.12: Graphical representation of apparent and incremental inductances.

Chapter 3

Position Error Estimation

Methods for Sensorless Control

Sensorless control is implemented. This type of control allows to fully control the machine without the rotor position information coming from the position transducer. The advantages of this method are the higher reliability of the hardware due to the absence of the position/speed sensor, avoiding the presence of supply and signal conditioning requirements and the possibility of cable disconnection and fault concerning the sensor. Moreover, its absence reduces the cost and the size of the machine and fosters the use of the motor in harsh environments, such as operation at high temperature, immersed in a liquid or vacuum packed, and in confined spaces. Furthermore, high speed operation is not limited by the presence of the sensor, both in bandwidth and in measurement accuracy.

When implementing sensorless control techniques the motor model equations presented in Section 2.1 are expressed in coordinates of estimated rotor reference frame, denoted by subscript $\hat{d}q$, whose d -axis is at $\hat{\theta} = \theta - \tilde{\theta}$, where $\tilde{\theta}$ is the angular position error between the estimated coordinates and the real ones.

The main techniques employed for sensorless control are:

- **Fundamental model-based position estimation methods:** they rely on stator voltage equations to be used in rotor equations, which are position-dependent. The back-electromotive force (EMF) in the voltage equation is only reliable at medium-high speed.
- **Saliency-based methods:** they take advantage of high frequency (HF) excitation. They are employed at zero/low speed where the back-EMF is not reliable or absent.

Multiple injection and demodulation techniques are available in literature, such as high frequency current, voltage or flux signal injection, as in [11] and [12]. Besides being pulsating, the injected signal can be rotating, continuous or discontinuous, periodic or PWM-based.

In this work, high frequency voltage injection is analyzed and the demodulated quantities considered are stator current and current-model flux.

The reader should keep in mind scrolling through the next sections of this chapter that all the apparent and incremental inductances which appear in the following analysis are computed online from the magnetic current model, as anticipated in Section 2.2.3, and for the sake of clarity the superscript i , denoting the current model parameters, will not be noted.

The advantages and disadvantages of each method are dealt with in more detail in the following.

3.1 High Frequency Pulsating Voltage Injection

The high frequency voltage injection is a saliency-based sensorless control technique which takes advantage of the anisotropy of the machine, making this method suitable for SyR motors, to determine the rotor position and speed by demodulating the high-frequency stator current or current-model flux component.

If a high-frequency signal is injected in the voltage d -axis component, the quadrature high-frequency response in stator current or current-model flux can be isolated

to determine rotor position, as their magnitudes are proportional to this quantity, as demonstrated in [13] and [14].

This method is particularly suited for zero or low speed operations, as it does not rely on back-EMF, which is only trustworthy at higher speeds, for determining the rotor position.

Since high frequency injection is a saliency-based method, attention should be paid to the operating points at which saliency is low. A saliency analysis is carried out in Section 3.4.

A drawback of this method is the fact that part of the DC-link voltage available is intended for being injected as part of the high-frequency signal and therefore, resulting in reduced torque dynamics. At high speeds, the injection is relegated in favor of back-emf based methods for the absence of acoustic noise.

The machine voltage, current and flux are separated in two terms:

- Fundamental component: used for motor control.
- HF component: used for position estimation.

Several saliency-based algorithms are possible, depending on the type of injected signal and the demodulation method.

In this work the following techniques are employed. They are discerned according to:

- Type of pulsating voltage injection:
 1. Sinusoidal;
 2. Square wave;
- Demodulated quantity for position error signal:
 1. Stator current;
 2. Stator current-model flux;

The different injection and demodulation techniques will be analyzed in the following.

3.1.1 Sinusoidal Pulsating Voltage Injection

A high frequency voltage signal $v_{\hat{d}qh}$ is injected in the d -axis, while the high frequency component of the q -axis quantity, stator current or current-model flux, is extracted with demodulation to retrieve the position error signal [13] [15]. Note that the high-frequency components are denoted by subscript h .

$$v_{\hat{d}qh} = v_h \begin{bmatrix} \cos \omega_h t \\ 0 \end{bmatrix} \quad (3.1)$$

Where $\omega_h = 2\pi f_h$, f_h is the frequency of the injected signal and v_h is its magnitude.

A drawback of this method is that to be able to reconstruct the high frequency injected signal, the following relation must be respected:

$$f_h \leq \frac{1}{10} f_s \quad (3.2)$$

where f_h is the frequency of the injected signal and f_s is the switching frequency.

A low-pass filter (LPF) is employed to determine the amplitude of the demodulated high frequency signal. The bandwidth of the position tracking loop is limited under the cut-off frequency of the low-pass filter, resulting in poor dynamic characteristics. As asserted in [14], the suggested bandwidth of the PI regulator of the PLL should be at least three times smaller than the cut-off frequency of the LPF:

$$\Omega_{PLL} \leq \frac{1}{3} \Omega_{0,LPF} \quad (3.3)$$

Square wave injection overcomes this problem, as the low-pass filter can be entirely removed.

3.1.2 Square Wave Pulsating Voltage Injection

The maximum possible frequency of the square wave injected signal for synchronous sampling is up to half the switching frequency. This allows to conspicuously increase the cut-off frequency of the LPF or to entirely eliminate the filter, enabling the bandwidth of the whole control system to increase, resulting in a higher stability

of the overall system. [17].

The squarewave signal is injected as

$$v_{dh}^k = \begin{cases} +v_h, & \text{if } k = 2n \\ -v_h, & \text{if } k = 2n + 1 \end{cases} \quad (3.4)$$

where v_h is the magnitude of the injected voltage and the superscript k denotes the discrete time instant [18].

3.2 Current Demodulation

Current values are easily accessible thanks to the current sensors which are already part of the drive system. Besides, having current as the demodulated signal increases resilience to parameter errors in flux-map LUTS.

3.2.1 Sinusoidal Injection Case

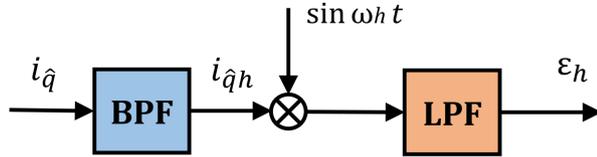


Figure 3.1: Current demodulation scheme in the case of sinusoidal injected signal.

It follows from reversed Eq. (2.3b) in estimated rotor position reference frame that

$$i_{\hat{d}q} = e^{J\tilde{\theta}} \mathbf{L}_{\tilde{\theta}}^{-1} e^{-J\tilde{\theta}} \lambda_{\hat{d}q} \quad (3.5a)$$

$$i_{\hat{q}h} = \frac{-l_{\Delta} \sin 2\tilde{\theta} - l_{dq} \cos 2\tilde{\theta}}{l_d l_q - l_{dq}^2} \frac{v_h \sin \omega_h t}{\omega_h} \quad (3.5b)$$

where $l_{\Delta} = \frac{l_d - l_q}{2}$.

Being the magnitude of the high frequency q -current component $i_{\hat{q}h}$ proportional to the position error, it is demodulated from the measured current and filtered to

retrieve the position error as shown in Fig. 3.1 [13].

By demodulating (3.5b), the position error signal for the low speed model ϵ_h is obtained.

$$\epsilon_h = -\frac{\omega_h}{v_h} \frac{2 l_d l_q}{(l_d - l_q)} \text{LPF} [i_{\hat{q}h} \sin \omega_h t] \quad (3.6a)$$

$$\epsilon_h = k_e \cdot \text{LPF} [i_{\hat{q}h} \sin \omega_h t] \quad (3.6b)$$

where k_e is

$$k_e = -\frac{\omega_h}{v_h} \frac{2 l_d l_q}{(l_d - l_q)} \quad (3.7)$$

Current demodulation results in a steady-state error due to cross saturation given by

$$\tilde{\theta}_0 = -\frac{1}{2} \tan^{-1} \frac{l_{dq}}{l_{\Delta}} \quad (3.8)$$

In literature, some papers propose active compensation of the steady state position error derived from the demodulation of the high frequency current component. This method requires accurate knowledge of the model parameters and the assumption of small position error to be consistent [16].

3.2.2 Square Wave Injection Case

In case of square wave injected signal the band-pass filter and the LPF are eliminated, as it is enough to subtract the measured current at $(k-1)^{th}$ step from the k^{th} step one to retrieve the high frequency component, as follows:

$$i_{\hat{q}h}^{(k)} = i_{\hat{q}}^{(k)} - i_{\hat{q}}^{(k-1)} \quad (3.9)$$

The position error signal ϵ_h proportional to $i_{\hat{q}h}^{(k)}$ is then determined similarly to Eq. (3.6) as

$$\epsilon_h^{(k)} = \frac{f_h}{v_h} \frac{2 l_d^{(k)} l_q^{(k)}}{(l_d^{(k)} - l_q^{(k)})} \text{LPF} [i_{\hat{q}h}^{(k)} \cos \pi k] \quad (3.10a)$$

$$\epsilon_h^{(k)} = k_e^{(k)} \cdot \text{LPF} [i_{\hat{q}h}^{(k)} \cos \pi k] \quad (3.10b)$$

Where $k_e^{(k)}$ is

$$k_e^{(k)} = \frac{f_h}{v_h} \frac{2 l_d^{(k)} l_q^{(k)}}{(l_d^{(k)} - l_q^{(k)})} \quad (3.11)$$

3.3 Current-Model Flux Demodulation

To overcome the steady-state position error problem due to cross saturation, current-model high-frequency flux q -component is demodulated to retrieve the position information instead of current [14].

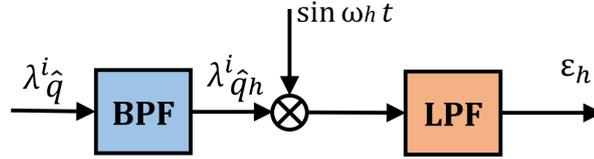


Figure 3.2: Flux demodulation scheme in the case of sinusoidal injected signal.

3.3.1 Sinusoidal Injection Case

The flux to be demodulated and filtered comes from the current model flux linkage based on the flux map lookup tables. It is identified with superscript i and defined as

$$\lambda_{\hat{d}qh}^i = \mathbf{L}_{\partial}^i i_{\hat{d}qh} \quad (3.12a)$$

$$\lambda_{\hat{d}qh}^i = \mathbf{L}_{\partial}^i e^{\mathbf{J}\tilde{\theta}} \mathbf{L}_{\partial}^{-1} e^{-\mathbf{J}\tilde{\theta}} \lambda_{\hat{d}q} \quad (3.12b)$$

where \mathbf{L}_{∂}^i is the current model incremental inductance matrix computed online. Assuming small position error $\tilde{\theta}$ and constant incremental inductance in proximity of the operating point, that is to say that \mathbf{L}_{∂}^i is equal to the real \mathbf{L}_{∂} , linearization is possible:

$$\lambda_{\hat{q}h}^i = -\frac{l_q l_{\Delta} - l_{dq}^2}{l_d l_q - l_{dq}^2} \frac{v_h \sin \omega_h t}{\omega_h} \sin 2\tilde{\theta} \quad (3.13)$$

After demodulating and filtering Eq. (3.13) like in Fig. 3.2, the position error signal for the low speed model ϵ_h is obtained as

$$\epsilon_h = -\frac{\omega_h (l_d l_q - l_{dq}^2)}{v_h (l_q l_\Delta - l_{dq}^2)} \text{LPF} [\lambda_{\hat{q}h}^i \sin \omega_h t] \quad (3.14a)$$

$$\epsilon_h = k_e \cdot \text{LPF} [\lambda_{\hat{q}h}^i \sin \omega_h t] \quad (3.14b)$$

Where k_e is

$$k_e = -\frac{\omega_h (l_d l_q - l_{dq}^2)}{v_h (-l_q l_\Delta + l_{dq}^2)} \quad (3.15)$$

3.3.2 Square Wave Injection Case

Similar to the current demodulation case, if a square wave voltage is injected, it is sufficient to calculate the difference of two current model q -fluxes belonging to two consecutive computational steps to obtain the high frequency flux component $\lambda_{\hat{q}h}^i$, eliminating the band-pass filter and the low-pass filter, as

$$\lambda_{\hat{q}h}^{i(k)} = \lambda_{\hat{q}}^{i(k)} - \lambda_{\hat{q}}^{i(k-1)} \quad (3.16)$$

The position error signal ϵ_h is calculated similarly to Eq. (3.17) as follows:

$$\epsilon_h^{(k)} = \frac{f_h (l_d^{(k)} l_q^{(k)} - l_{dq}^{(k)2})}{v_h (l_q^{(k)} l_\Delta^{(k)} - l_{dq}^{(k)2})} [\lambda_{\hat{q}h}^{(k)i} \cos \pi k] \quad (3.17a)$$

$$\epsilon_h^{(k)} = k_e^{(k)} \cdot [\lambda_{\hat{q}h}^{(k)i} \cos \pi k] \quad (3.17b)$$

where f_h is the high frequency of the injected signal and $k_e^{(k)}$ is

$$k_e^{(k)} = \frac{f_h (l_d^{(k)} l_q^{(k)} - l_{dq}^{(k)2})}{v_h (l_q^{(k)} l_\Delta^{(k)} - l_{dq}^{(k)2})} \quad (3.18)$$

3.4 Saliency Analysis

Since high-frequency voltage injection is an incremental saliency-based technique, it is appropriate to perform a saliency analysis of the tested motors to be aware of the operating points at which the position error estimation method may become

troublesome.

High frequency-based position error estimation methods rely on incremental saliency, defined as the ratio of incremental inductances l_d/l_q .

The incremental saliency maps of the three investigated motors are reported in Fig. 3.3. For the ratings of the motors, see table 4.1.

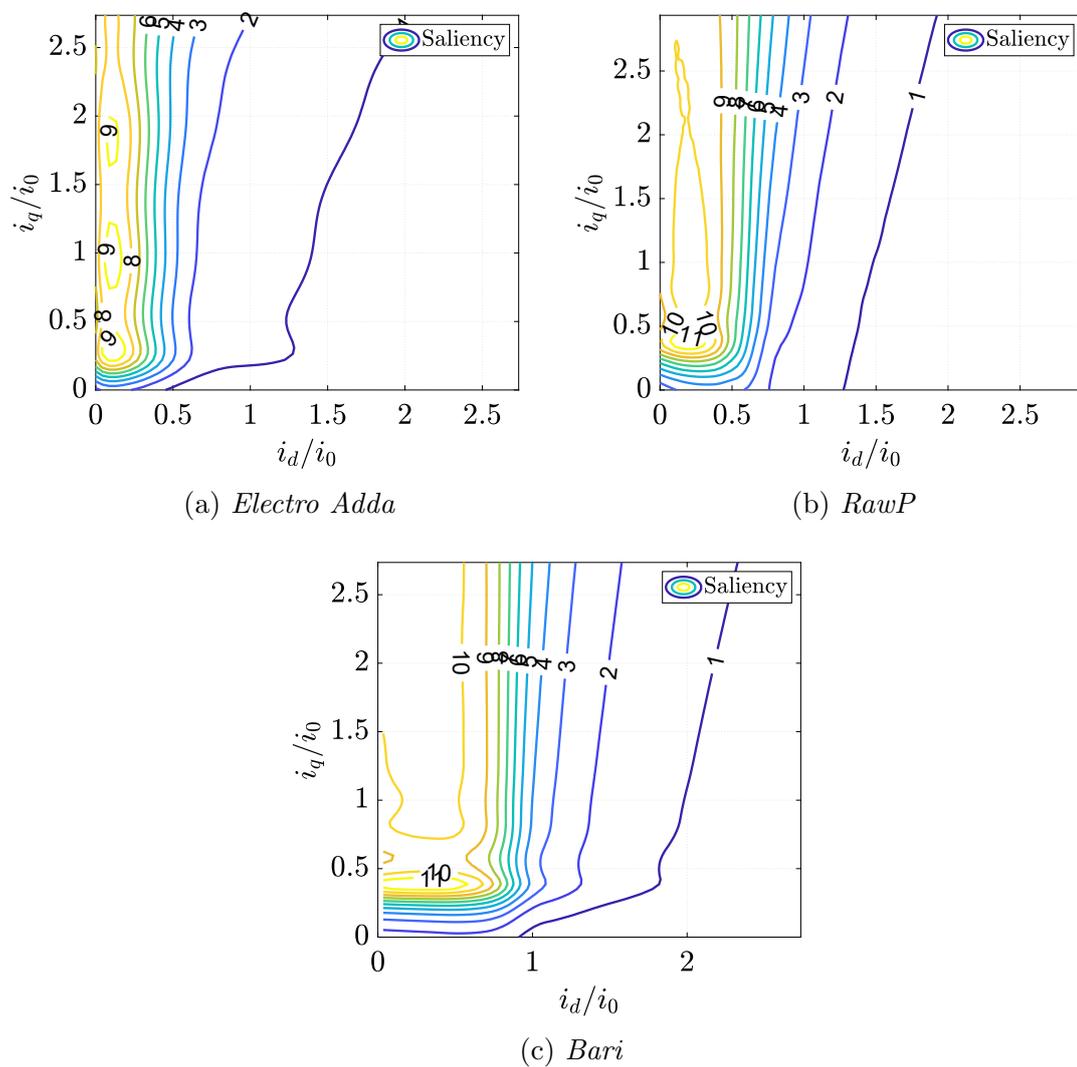


Figure 3.3: Incremental saliency plots of the tested motors *Electro Adda*, *RawP*, *Bari* in the dq -currents plane in relative values. MTPA trajectory is depicted in red.

As shown in the contour plots in Fig. 3.3, attention should be paid when a saliency-based position error estimation technique is employed at no load, namely close to the origin, where currents value is low or zero. When operating in this circumstance, the ribs directed along the q -axis are not saturated, making the machine isotropic. It is noticeable from Fig. 3.3 the convenience of imposing a low value of current along the q -axis to increase the saliency and circumvent the problem.

Whereas when high load operation is required, the saturation along d -axis reduces the saliency of the machine along the MTPA trajectory. A maximum torque limit should be set when operating with saliency-based position error estimation methods to avoid the loss of control in this conditions.

3.5 Back-EMF-based position estimation methods

Since these methods rely on back-EMF to retrieve rotor position, they are only employed for medium to high speed operations, as at low speed or standstill, the back-EMF is too low to be exploited or even null. When the speed is sufficiently high, fundamental model based techniques are always preferred to the others, presenting good robustness and accuracy.

In this work, Active Flux and Adaptive Projection Vectro for Position Error Estimation methods are investigated.

3.5.1 Active Flux

Active Flux is a well known position error estimation method, appreciated for its simplicity and ease of implementation. It has been first introduced in [19] and then widely investigated in the literautre [14], [20].

The idea of active flux is to turn a salient machine into an isotropic one. The q -axis component of active flux is proportional to the position error and is defined

under position error as

$$\boldsymbol{\lambda}_{\hat{d}q}^{af} = \boldsymbol{\lambda}_{\hat{d}q} - L_q^i \cdot \mathbf{i}_{\hat{d}q} = \begin{bmatrix} (L_d^i - L_q^i) i_{\hat{d}} \\ 0 \end{bmatrix} + \boldsymbol{\lambda}_{\hat{d}q}^a \tilde{\theta} \quad (3.19)$$

where $\boldsymbol{\lambda}_{\hat{d}q}^a$ is the auxiliary flux vector, introduced in [21] and developed in [20]. It is defined as

$$\boldsymbol{\lambda}_{\hat{d}q}^a = (\mathbf{J}\mathbf{L}^i - \mathbf{L}_{\partial}\mathbf{J}) = \begin{bmatrix} (l_d - L_q^i) i_{\hat{q}} - l_{dq} i_{\hat{d}} \\ (L_d^i - l_q) i_{\hat{d}} - l_{dq} i_{\hat{q}} \end{bmatrix}. \quad (3.20)$$

The observed active flux allows to determine the position error signal for the high speed model ϵ_θ .

$$\epsilon_\theta = \frac{1}{2L_{\Delta}^i i_{\hat{d}}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \left(\boldsymbol{\lambda}_{\hat{d}q} - \boldsymbol{\lambda}_{\hat{d}q}^i \right) \quad (3.21)$$

$L_{\Delta}^i = \frac{L_d^i - L_q^i}{2}$ and L_d^i and L_q^i are the current model apparent inductances computed online.

Unlike saliency-based position error estimation, imposing a minimum current in q -axis, as asserted in Section 3.4, is not favorable for active flux as, at no load, it becomes unobservable. This makes active flux incompatible with low-speed saliency-based position error estimation, resulting in some intricacies when a full-speed sensorless control algorithm is constituted with the two techniques together. This will be proven later in Section 4.4.2.

3.5.2 Adaptive Projection Vector for Position Error Estimation

An Adaptive Projection vector for Position error estimation was introduced and investigated in [22] and further analysed in [23]. Using this method, the position error signal is determined as the projection on the vector Φ_θ of the difference between the observed and the current model fluxes, as described by.

$$\epsilon_\theta = \Phi_\theta^T \left(\hat{\boldsymbol{\lambda}}_{\hat{d}q} - \boldsymbol{\lambda}_{\hat{d}q}^i \right) \quad (3.22)$$

The transfer function K_θ from the position error signal to the position error $\epsilon_\theta \rightarrow \tilde{\theta}$ is derived in [22] through the linear error dynamics of the state observer (2.6) and reported in the following:

$$K_\theta = \frac{\epsilon_\theta}{\tilde{\theta}} = \Phi_\theta^T (s\mathbf{I} + \mathbf{G} + \omega\mathbf{I})^{-1} (s\mathbf{I} + \omega\mathbf{J}) \boldsymbol{\lambda}_{dq}^a \quad (3.23)$$

where $\boldsymbol{\lambda}_{dq}^a$ is the auxiliary flux vector reported in Eq. (3.20).

in steady state, when $s = 0$, it follows from (3.23) that the projection vector to obtain $K_\theta = 1$ is

$$\Phi_\theta^T = \frac{-1}{\omega \|\boldsymbol{\lambda}_{dq}^a\|^2} \boldsymbol{\lambda}_{dq}^{aT} \mathbf{J} (\mathbf{G} + \omega\mathbf{J}) \quad (3.24)$$

where \mathbf{G} is the 2x2 flux observer gain matrix.

3.6 Sensorless Control Techniques Fusion Scheme

As mentioned in the previous sections, high-frequency voltage injection methods are more suited for zero-to-low speed applications, as part of the available DC-link voltage is employed to inject the high-frequency signal and cannot be exploited by the motor to reach the maximum speed. Whereas fundamental model-based position error estimation methods such as active flux or APP encounter problems at standstill or low speed, where back-EMF on which they rely on is absent or scarce, while perform well at medium-to-high speed range.

The two techniques are fused together to form a full-speed sensorless control algorithm with the linear fusion scheme employed in [23] and reported below, exploiting the fusion coefficient f_ω .

$$\epsilon = f_\omega \cdot \epsilon_h^k + (1 - f_\omega) \cdot \epsilon_\theta^k \quad (3.25)$$

$$f_\omega = \begin{cases} 1, & \text{if } |\hat{\omega}^k| < g - \omega_g \\ 0, & \text{if } |\hat{\omega}^k| > g + \omega_g \\ \frac{g + \omega_g - |\hat{\omega}^k|}{2\omega_g}, & \text{otherwise} \end{cases} \quad (3.26)$$

Where $\hat{\omega}^k$ is the operating speed, ω_g is the speed span of transition on either sides of the cross-over frequency g . The fusion coefficient f_ω ensures a smooth transition between the low-speed and the high-speed position error estimation methods.

3.7 Phase Locked Loop

Most of the sensorless control algorithms adopt Phase Locked Loop (PLL) structures.

The PI regulator of the PLL drives the position error signal ϵ to zero producing as output the observed rotor position used in the control. The PLL also acts like a low-pass filter, meaning that the noise in the observed quantity is considerably reduced. It also permits to retrieve the observed speed $\hat{\omega}$. The PI regulator equations of the PLL are reported in the following and its scheme is represented in Fig. 3.4.

$$\hat{\omega} = k_p \epsilon + \int k_i \epsilon dt \qquad \hat{\theta} = \int \hat{\omega} dt \qquad (3.27)$$

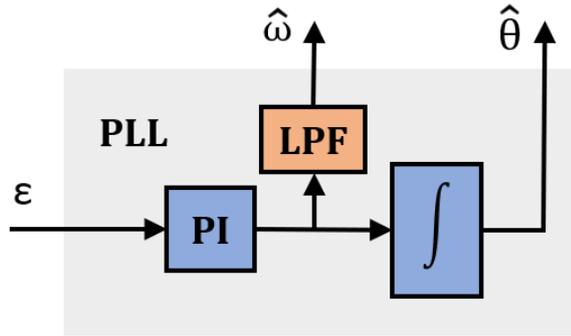


Figure 3.4: PLL scheme.

where k_p and k_i are the proportional and the integral gains, respectively. As discussed later in Section 4.3, they are tuned for a critically damped response. The bandwidth of the PI regulator of the PLL is tuned separately for the sinusoidal and for the square wave high frequency injection cases.

Chapter 4

Automated Generation of the Simulation Model and Results

4.1 *syreDrive* Implementation

The whole work is implemented in the software *SyR-e* in the form of a new add-on, *syreDrive*. This feature allows to start a simulation in Simulink of a standard sequence for control evaluation purposes of a synchronous reluctance machine, given solely the motor parameters. The control files, as well as the Simulink model for the simulation, are automatically generated, so that the user only deals with the graphical user interface (GUI) of *SyR-e* dedicated to magnetic model manipulation (MMM) and does not need to tackle any matters related to the control, aside from deciding which control to use.

In the main folder of the software is a folder named *Syr_ctrl_SIM* which contains a default Simulink model, shown in Fig. 4.5, and predefined C-files for the control. Both the Simulink model and the C-files are suitable for any SyR machine, they are customized for the selected motor from *SyR-e*.

The *syreDrive* tab appears as in Fig. 4.1. As it can be seen from the picture, the user is allowed to change the control settings. It is possible to choose the control type between current, torque or speed control, and the flux maps model, *dq* or

dqt. Sensorless control can be enabled or not with a switch in the GUI, and if on, sensorless control methods can be set up. Regarding the position error estimation methods for the low-speed region, the user can decide whether to use sinusoidal or square wave high frequency pulsating voltage injection and whether to demodulate the current or the flux q -component. For the high speed-region, active flux or APP can be set.

Some converter data can be set as well, like the on-threshold voltage of the semiconductor devices, their internal resistance and the dead time.

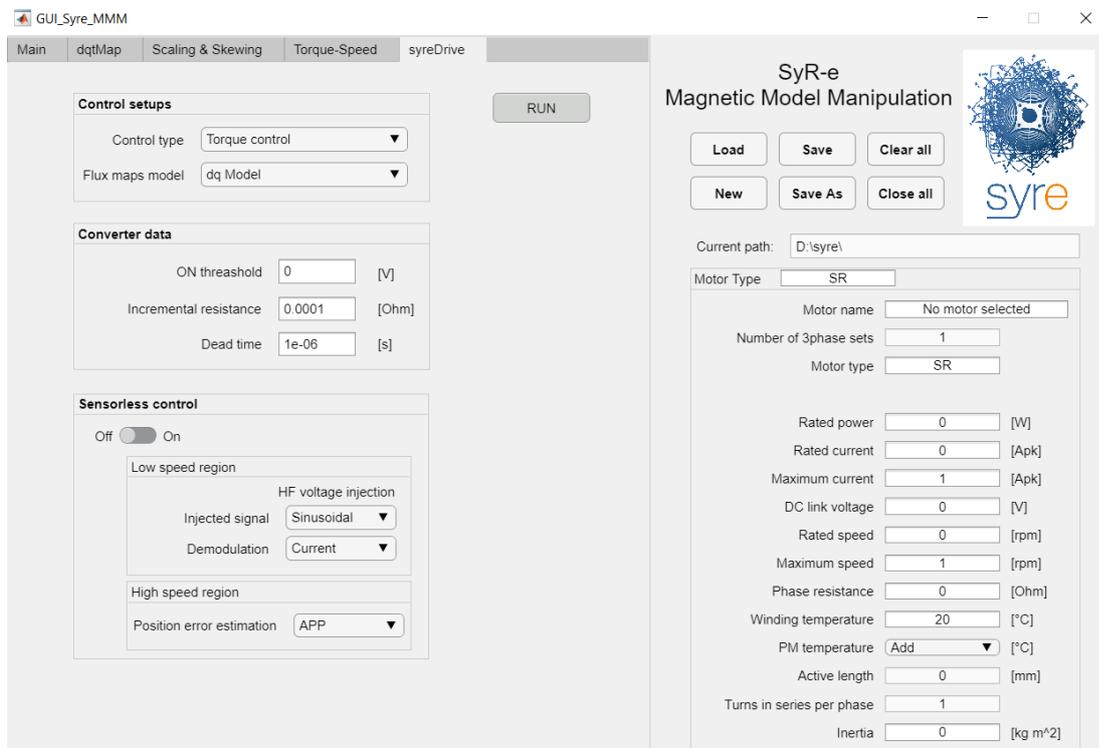


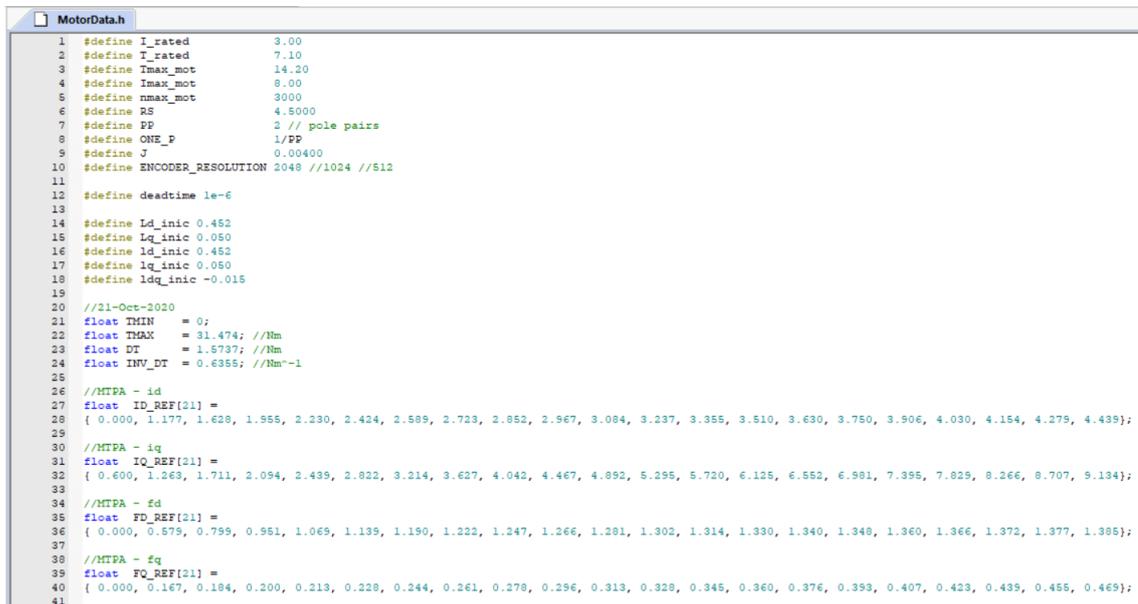
Figure 4.1: The *syreDrive* tab in the *SyR-e MMM* GUI.

The control settings decided by the user are saved in a data structure called *motorModel.m* which is generated by *SyR-e* or can be loaded in the software. This structure also holds all the motor data of the selected machine, comprising the flux maps and even the motor name and path directory where *motorModel.m* is saved.

When button *RUN* in Fig. 4.1 is pressed, the MATLAB function *MMM_CtrlSIM()* is triggered. This function admits as input argument the data

structure *motorModel.m*.

The script of the function then copies the default folder *Syr_ctrl_SIM* in the folder where *motorModel.m* is saved. The copy is customized for the loaded machine by automatically writing the motor data header file used in the digital control and by saving it in the copied folder. The function that writes the motor data header file which is set off in the script of *MMM_CtrlSIM()* is *MMM_print_MotorDataH()*. This function prints the header file with the motor data displaced as required by the control, drawing the data directly from the structure *motorModel.m*. The informations contained in the motor data header file are used to customize the control for the selected SyR motor, allowing the control to draw the data it needs from such a file, like motor ratings, flux maps, MTPA trajectory and the values of the quantities for calibrating the control parameters. Fig. 4.2 shows an example of automatically printed motor data header file for the *Bari* motor. The file contains the ratings of the motor and the 1D lookup tables of the MTPA for retrieving the reference currents and the 2D LUTs of the flux maps, not reported in the figure for the sake of brevity.



```

1 #define I_rated      3.00
2 #define I_rated      7.10
3 #define Tmax_mot     14.20
4 #define Imax_mot     8.00
5 #define nmax_mot     3000
6 #define RS           4.5000
7 #define PP           2 // pole pairs
8 #define ONE_P        1/PP
9 #define J            0.00400
10 #define ENCODER_RESOLUTION 2048 //1024 //512
11
12 #define deadtime 1e-6
13
14 #define Ld_inic 0.452
15 #define Lq_inic 0.050
16 #define ld_inic 0.452
17 #define lq_inic 0.050
18 #define ldq_inic -0.015
19
20 //21-Oct-2020
21 float THIN = 0;
22 float THAX = 31.474; //Nm
23 float DT = 1.5737; //Nm
24 float INV_DT = 0.6355; //Nm^-1
25
26 //MTPA - id
27 float ID_REF[21] =
28 { 0.000, 1.177, 1.628, 1.955, 2.230, 2.424, 2.585, 2.723, 2.852, 2.967, 3.084, 3.237, 3.355, 3.510, 3.630, 3.750, 3.906, 4.030, 4.154, 4.279, 4.439};
29
30 //MTPA - iq
31 float IQ_REF[21] =
32 { 0.600, 1.263, 1.711, 2.094, 2.439, 2.822, 3.214, 3.627, 4.042, 4.467, 4.892, 5.295, 5.720, 6.125, 6.552, 6.981, 7.395, 7.829, 8.266, 8.707, 9.134};
33
34 //MTPA - fd
35 float FD_REF[21] =
36 { 0.000, 0.579, 0.799, 0.951, 1.069, 1.139, 1.190, 1.222, 1.247, 1.266, 1.281, 1.302, 1.314, 1.330, 1.340, 1.348, 1.360, 1.366, 1.372, 1.377, 1.385};
37
38 //MTPA - fq
39 float FQ_REF[21] =
40 { 0.000, 0.167, 0.184, 0.200, 0.213, 0.228, 0.244, 0.261, 0.278, 0.296, 0.313, 0.328, 0.345, 0.360, 0.376, 0.393, 0.407, 0.423, 0.439, 0.455, 0.469};
41

```

Figure 4.2: The motor data header file automatically printed by the function *MMM_print_MotorDataH()*.

The default folder *Syr_ctrl_SIM* also holds the script *init_sim.m*. This script is

run in the callback functions of the Simulink model before starting the simulation. It loads in the MATLAB's workspace the data held in *motorModel.m* that will be used in the simulation. By copying the whole *Syr_ctrl_SIM* folder in the motor folder to be customized, *init_sim.m* file is also copied there.

The Simulink simulation in the copied folder is finally started. It employs the C-files contained in the S-function inside the blue block in Fig. 4.5, including the printed motor data header file and the *motorModel.m* informations, hence customizing the simulation. The *syreDrive* flowchart is reported in Fig. 4.3.

When the simulation is over, *SyR-e* automatically plots some graphs like the ones in Fig. 4.4 for the case of speed control, which are

1. torque control case:
 - reference torque T^* , real torque T and observed torque T_{obs} in *p.u.* relatively to the rated torque of the motor;
 - external speed imposed n and observed speed n_{obs} in *rpm*;
2. speed control case:
 - real torque T , observed torque T_{obs} and load torque T_L in *p.u.* relatively to the rated torque of the motor;
 - reference speed n^* , real speed n and observed speed n_{obs} in *rpm*;
3. for both cases:
 - rotor position θ and estimated position θ_{PLL} in rad;
 - real rotor position estimation error $\tilde{\theta}$ and expected steady state position estimation error due to cross saturation $\tilde{\theta}_0$ in degrees;
 - fusion coefficient f_ω ;
 - d and q currents in *A*;
 - d and q fluxes in *Vs*.

The observed quantities are plotted as well, so that the behaviour of the flux observer and PLL can be assessed.

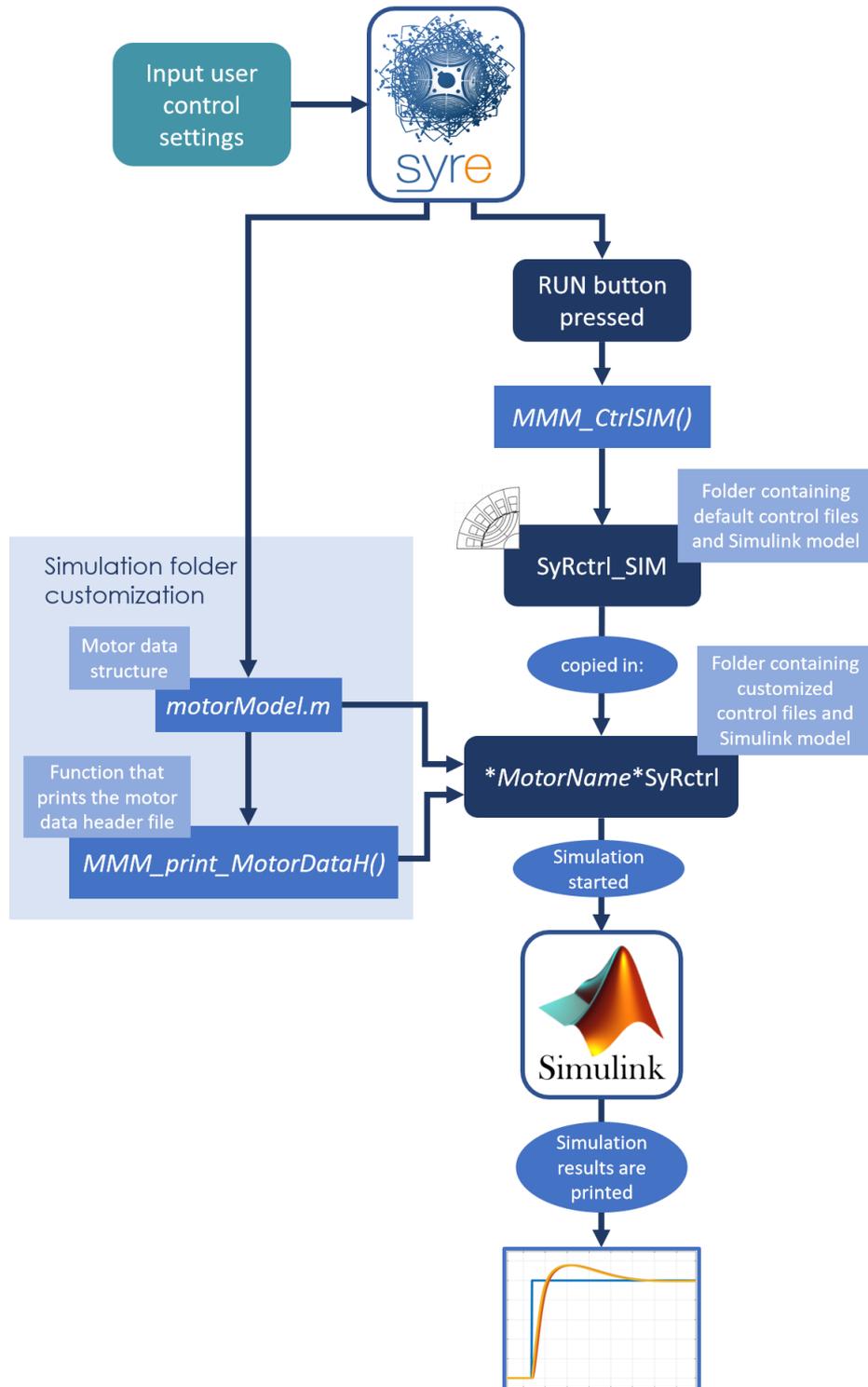


Figure 4.3: The operating principle *syreDrive*.

Thus, the user can evaluate the selected control performances from the plots, especially sensorless control implementation, but also further tune the control parameters on the selected SyR machine for a more in-depth dedicated analysis or employ the automatically generated control files in a digital control for a practical application or test.

4.1.1 Simulink Model Template

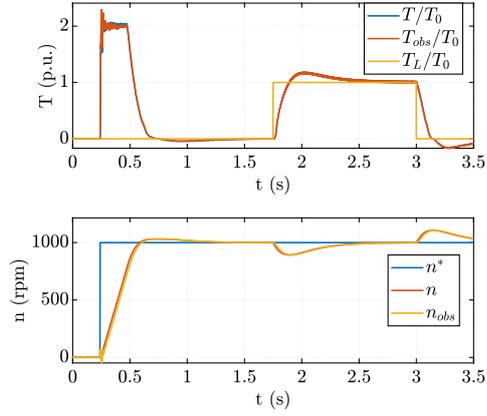
The model of the whole system is implemented in Simulink to perform simulations and assess the performances of the different control methods addressed in Chapter 3.

The Simulink model, depicted in Fig. 4.5, comprises the motor model (green block), along with an inverter model (yellow block), which is not discussed here, and the digital control (blue block) presented in Section 2.2.1.

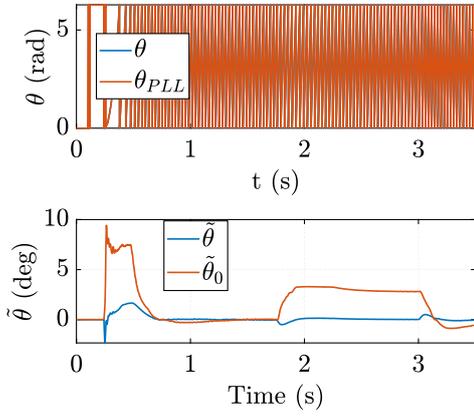
The C-files of the control are stored in the S-function contained in the digital control block and will be further described in Section 4.1.2.

The motor model block is made up of the electromagnetic and the mechanical models, respectively shown in Fig. 2.3 and 2.4.

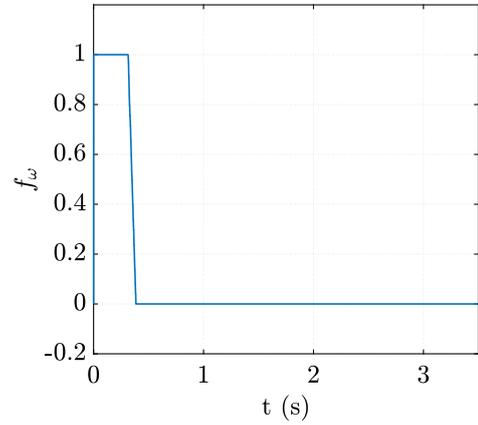
The content of the motor model block is depicted in Fig. 4.6. The quantity *FMapsModel*, contained in the yellow blocks shown here, stands for the information of the desired flux maps model to be used, either *dq* or *dqt*. The two switches in orange select the desired flux maps model. The MATLAB code inside the *dqt Flux Map LUT* blocks, for torque and currents, interpolates the *dqt* inverse model torque and currents maps, which admit as inputs the *d*- and *q*-fluxes and the position, to retrieve the corresponding value of torque or *d* and *q* stator currents.



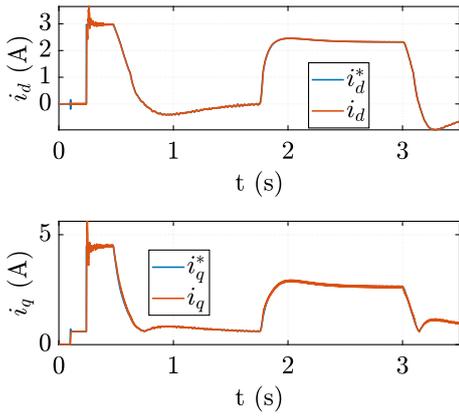
(a) Torque and speed.



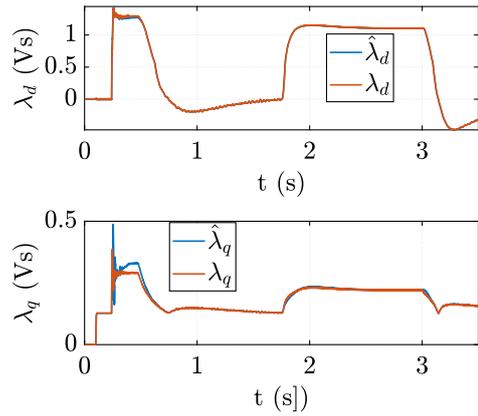
(b) Position and position error.



(c) Fusion coefficient.



(d) d and q currents.



(e) d and q fluxes.

Figure 4.4: The quantities plotted by *syreDrive* when the control simulation ends.

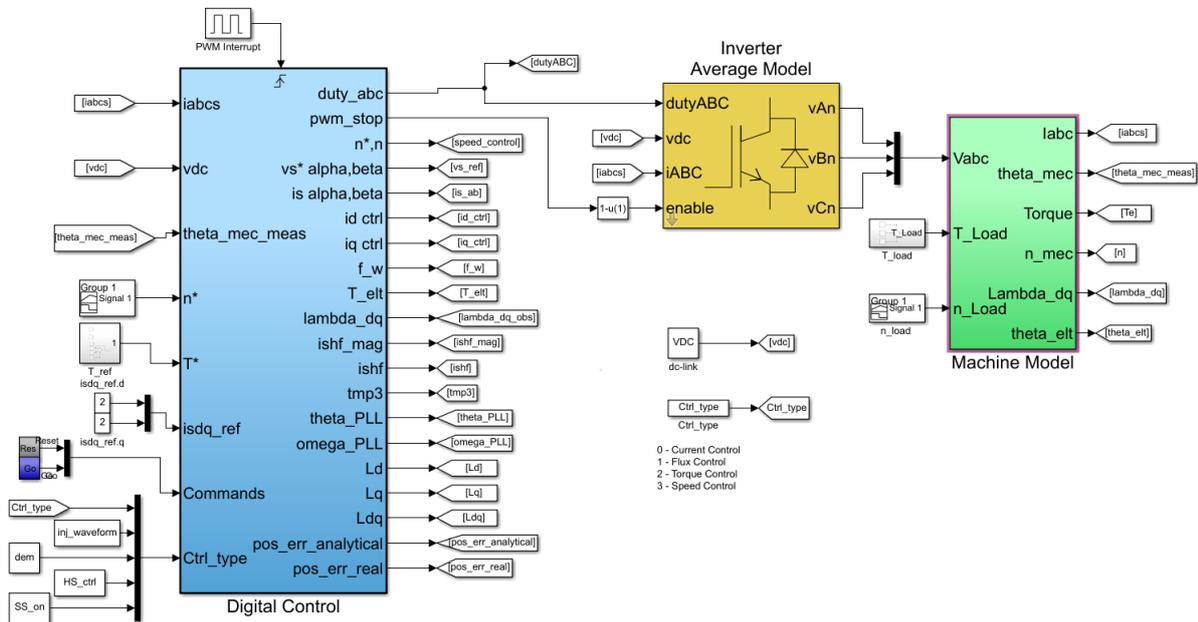
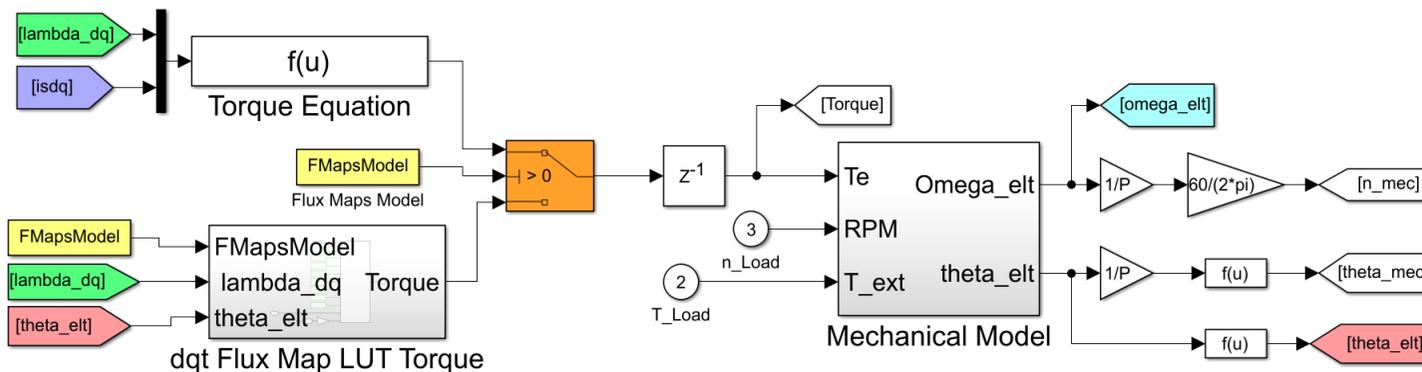
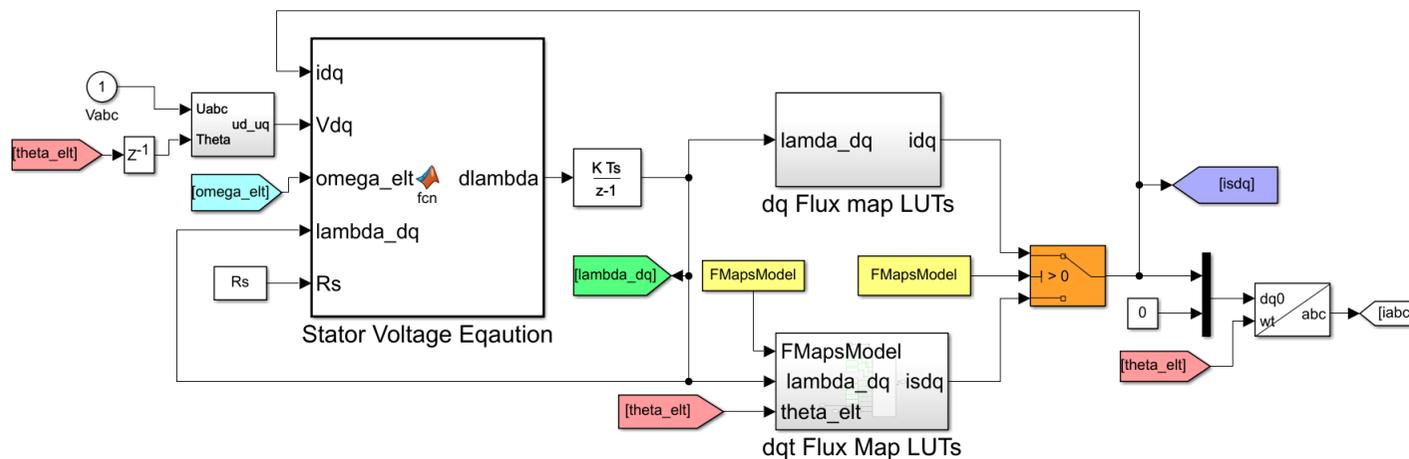


Figure 4.5: The Simulink model of the whole system, comprising the digital control (blue block), the inverter model (in yellow) and the motor model (in green).



(a) Mechanical model



(b) Electromagnetic model

Figure 4.6: The motor model as implemented in Simulink, comprising the mechanical (a) and the electromagnetic (b) model.

4.1.2 Torque and Speed Control Scheme

The sensorless techniques presented in Chapter 3 are implemented in the Field Oriented Control described in Section 2.2.1 to derive the rotor position thanks to the position error estimation methods and dismiss the encoder.

Hence, when the sensorless control is enabled, the Field Oriented Control scheme becomes the one depicted in Fig. 4.7.

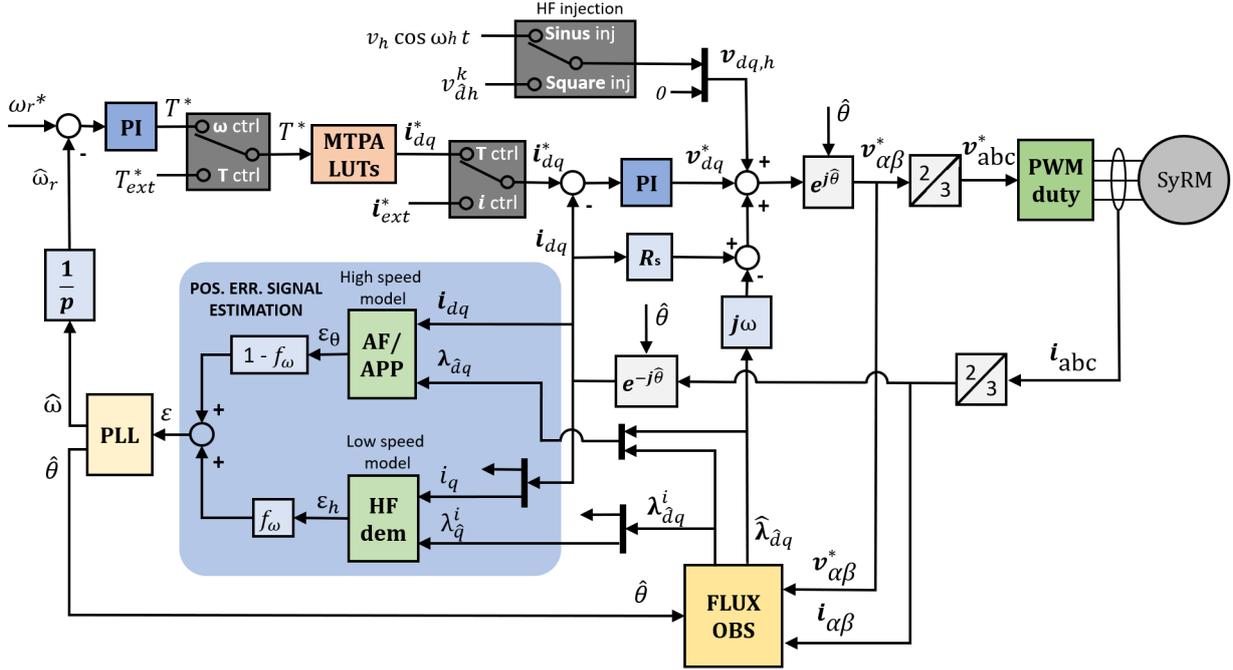


Figure 4.7: Scheme of the Field Oriented sensorless control implemented.

The green block in Fig. 4.7 marked with **AF/APP** inside the position error signal estimation box contains the position error estimation methods described in 3.5. Inside, active flux or APP can be selected. The same applies for the green block marked with **HF dem**, as it contains the demodulation schemes of the high frequency voltage signal injected to determine the position error signal for the low speed operation. These two blocks are followed by the fusion system presented in 3.6. The resulting value ϵ is processed by the PLL block, whose output are the estimated rotor position and speed used instead of the measurements from the encoder.

The control is implemented in the form of C-files, suitable for digital control, which

are contained in the Simulink model in the S-function inside the control block in Fig. 4.5 (blue block).

Besides the main C-file, where the core of the control is, other files are included, such as the *User_functions.c* containing the functions used in the control and other header files which stock the motor data, the constants used, the variables, the macros, etc.

The main file contains the available control types, current, torque and speed control, along with the selectable sensorless position error estimation methods.

4.2 Simulation Results for the Encoder-based Control

Before testing sensorless control, the traditional encoder-relying Field Oriented Control is tested to verify its functioning.

To perform this simulations, the selected SyR machine is the *Bari* motor, whose data is reported in table 4.1.

Two simulated sequences of torque and speed are reported below, using torque control in Fig. 4.8a and speed control in Fig. 4.8b.

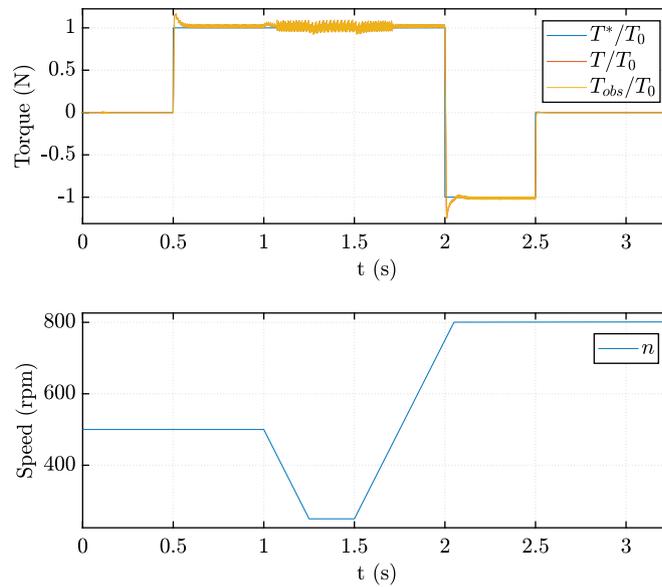
The control behaves as expected, since both the torque and speed reference are carried out with a good accuracy, as shown in Fig. 4.8.

It is clear from Fig. 4.8 that the flux observer described in Section 2.2.1 and implemented in the control works fine, as the torque computed with the observed values \hat{T} perfectly matches the torque computed in the mechanical model of the motor depicted in Fig. 4.6a.

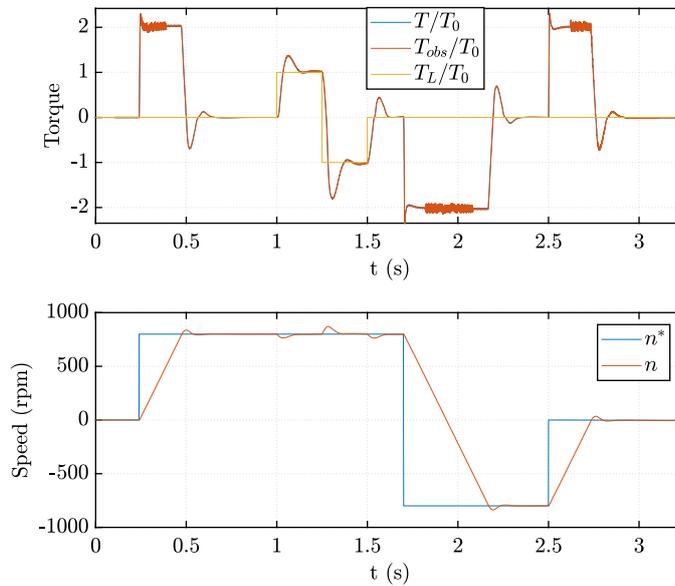
$$\hat{T} = \frac{3}{2}p \left(\hat{\lambda}_\alpha i_\beta - \hat{\lambda}_\beta i_\alpha \right) \quad (4.1)$$

The speed control shows a good response to torque loads, as shown in Fig. 4.8b.

The results of two simulations in torque control at 500 *rpm* performed with the SyR *Bari* motor are reported in Fig. 4.9, where a zoomed-in particular in steady state condition of the reference torque T^* , the real torque T and the observed torque \hat{T} is shown. Fig. 4.9a shows the torque and the reference torque from a simulation



(a) Torque control



(b) Speed control

Figure 4.8: Two torque and speed simulated sequences with the position information coming from the encoder. The simulation reported in (a) is torque-controlled, while the one in (b) is controlled in speed. The torque values are in p.u. relatively to the rated torque of the motor T_0 .

which employed the dq flux model, while the torque graph in Fig. 4.9b presents the same quantities of the former plot, but it is taken from a simulation performed using the dqt model presented in Section 2.1.2.

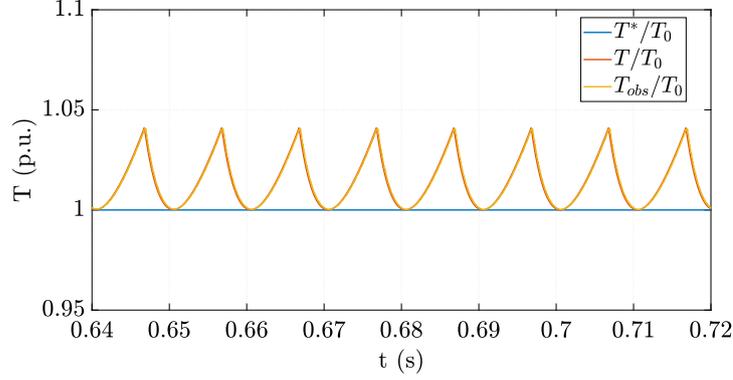
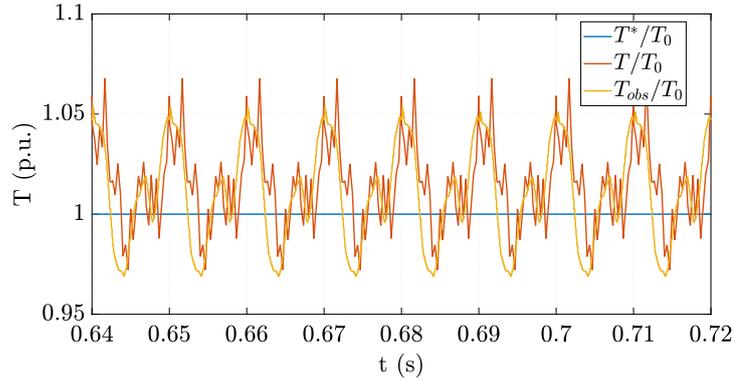
(a) dq model(b) dqt model

Figure 4.9: Two graphs representing the same torque control sequence. In the simulation in (a) the dq model was employed, while in (b) the dqt model.

It is evident from Fig. 4.9 that the dqt model can better represent the torque ripple due to the influence of the slots. In fact, dq - θ model accounts for the secondary space and slot harmonics which result in the torque ripple visible in Fig. 4.9b, more prominent than the one in Fig. 4.9a, where dq model is employed, better representing the real behaviour of the machine.

dqt model can be embedded in the control algorithm to realize a regulation that counteracts the influence of the slots and the resulting torque ripple to obtain a smoother torque response.

Before switching to full-sensorless control, the PLL is tested along with the encoder. The observed speed \hat{n} is plotted in Fig. 4.10. As shown in the graph, the observed speed is close to the real speed n , proving the fact that the sensorless tools such as the PLL and the flux observer are well functioning and ready to back fully sensorless control.

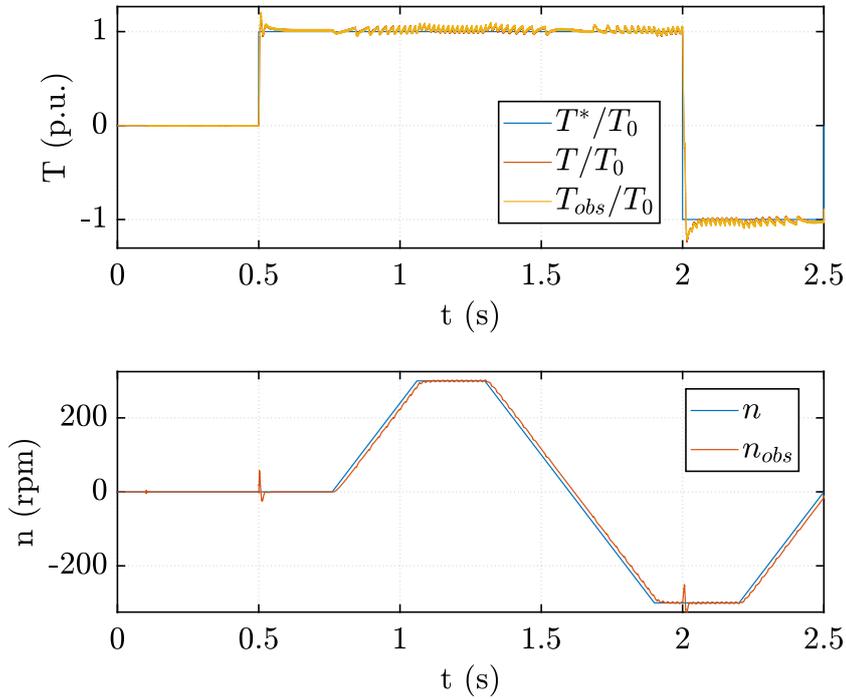


Figure 4.10: A torque and speed plot of a simulation to test the computation of the observed speed n_{obs} .

4.3 Automatic Calibration Rules

syreDrive is meant to lay the groundwork for SyR motor control for general purpose operations. Hence, the tuning of the control parameters aims at giving good performances for any SyR motor, without reaching the outermost limit, to ensure acceptable dynamic responses for a broad range of machines of different sizes. A

thorough calibration is left to the final user, after obtaining a first approach tuning from *syreDrive*.

Three different motors of different sizes are simulated to guarantee that the control parameters are suitable for most SyR motors. The ratings of the motors under test are reported in the following table:

| | | | <i>Bari</i> | <i>Raw-P</i> | <i>Electro Adda</i> |
|-------------------------|-----------|-------------------|-------------|--------------|---------------------|
| Rated power | P_0 | (kW) | 1.1 | 4.4 | 2.2 |
| Rated torque | T_0 | (N) | 7.3 | 17 | 14 |
| Rated speed | n_0 | (rpm) | 1500 | 2500 | 1500 |
| Maximum speed | n_{max} | (rpm) | 3000 | 6000 | 3000 |
| Rated current | i_0 | (A) | 3.0 | 15 | 7.0 |
| Maximum current | i_{max} | (A) | 8.0 | 30 | 20 |
| Phase resistance | R_s | (Ω) | 4.5 | 0.46 | 3.5 |
| DC-link voltage | V_{DC} | (V) | 565 | 565 | 520 |
| Pole pairs | p | (-) | 2 | 3 | 2 |
| Inertia | J | kg·m ² | 0.004 | 0.008 | 0.005 |

Table 4.1: Ratings of the motors under test.

The parameters chosen resulted in good performances for all of the tested motors, as shown in Section 4.4, and are defined as in the following.

4.3.1 Current loop

The proportional and integral gains of the current loop are adapted to the incremental inductances computed online as in equation (2.8). Each of the d and q axes has its own PI regulator, and therefore the gains are separately defined as in the following

$$k_{p,d}^I = \Omega_I \cdot l_d(\mathbf{i}_{dq0}) \quad k_{i,d}^I = \frac{\Omega_I^2}{10} \cdot l_d(\mathbf{i}_{dq0}) \quad (4.2a)$$

$$k_{p,q}^I = \Omega_I \cdot l_q(\mathbf{i}_{dq0}) \quad k_{i,q}^I = \frac{\Omega_I^2}{10} \cdot l_q(\mathbf{i}_{dq0}) \quad (4.2b)$$

Where the superscript I denotes the gains of the current loop. The bandwidth of the current loop Ω_I is set to $2\pi \cdot 75 \text{ rad/s}$.

4.3.2 Speed loop

The proportional and integral gains of the PI regulator of the speed loop are defined to take into account the inertia of the motor as

$$k_p^\omega = 2\Omega_\omega \cdot J \quad k_i^\omega = \Omega_\omega^2 \cdot J \quad (4.3)$$

The superscript ω stands for the speed loop. The bandwidth Ω_ω is set to $2\pi \cdot 1 \text{ rad/s}$ and its tuning will be further discussed in Section 4.3.8 by comparing the performance of the three SyR motors under test.

4.3.3 PLL

The proportional and integral gains of the PLL are tuned for a critically damped response by placing the two poles at $s = -\Omega_{PLL}$, where Ω_{PLL} is the bandwidth.

$$k_p^{PLL} = 2\Omega_{PLL} \quad k_i^{PLL} = \Omega_{PLL}^2 \quad (4.4)$$

In the case of sinusoidal pulsating voltage injection the bandwidth of the position and speed estimation loop needs to be at least three times smaller than the cut-off frequency of the low pass filter employed to demodulate the signal. [14].

$$\Omega_{PLL} \leq \frac{1}{3} \Omega_{0,LPF} \quad (4.5)$$

This prevents the PLL bandwidth from increasing freely and since the cut-off frequency of the filter is set to 50 Hz to sufficiently reduce the noise of the high frequency component, the PLL bandwidth for sinusoidal injection is $\Omega_{PLL}^{sin} = 2\pi \cdot 10 \text{ rad/s}$, in order to guarantee the relation in (4.5).

While in the case of high frequency square wave pulsating voltage injection, Ω_{PLL} is not restrained by the presence of the filter as the LPF is not necessary for demodulating such a signal. Hence the bandwidth of the tracking loop in this case is $\Omega_{PLL}^{sw} = 2\pi \cdot 25 \text{ rad/s}$, where the superscript *sw* stands for square wave.

4.3.4 Hybrid flux observer and fusion coefficient

The flux observer gain is set to $g = 2\pi \cdot 10 \text{ rad/s}$. It also coincides with the cross-over frequency of the linear fusion coefficient f_ω defined in Eq. (3.26), while the span

of transition ω_g is set to $2\pi \cdot 4 \text{ rad/s}$ to ensure a smooth transit between a position estimation method to the other.

4.3.5 High Frequency Voltage Injection Parameters

Regarding the high frequency voltage signal injected, the parameters to be set up are its frequency and magnitude.

The switching frequency used for all the simulations performed is $f_s = 10 \text{ kHz}$ and the DC-link voltages of each of the motors simulated are reported in the motor ratings listed in table 4.1.

The amplitude of the sinusoidal pulsating voltage signal is set to $v_h^{sin} = 100 \text{ V}$, while for the square wave signal it is $v_h^{sw} = 125 \text{ V}$, where the superscript *sin* stands for sinusoidal injection and *sw* for square wave injection. Considering the DC-link voltages of the motors, the following relations hold for the two cases:

$$v_h^{sin} \simeq \frac{1}{5.5} V_{DC} \quad (4.6a)$$

$$v_h^{sw} \simeq \frac{1}{4.5} V_{DC}. \quad (4.6b)$$

To be accurately represented, the sinusoidal injected signal should have a frequency f_h at least ten times smaller than the switching frequency.

$$f_h^{sin} \leq \frac{1}{10} f_s \quad (4.7)$$

To be absolutely certain that the relation in Eq. (4.7) holds, a further safety factor of 2 is utilized. Thus, the frequency of the sinusoidal signal is set to $f_h^{sin} = f_s/20 = 500 \text{ Hz}$.

While the frequency of the square wave voltage signal is set to half the switching frequency $f_h^{sw} = f_s/2 = 5 \text{ kHz}$, its maximum possible value, making the low-pass filter unnecessary to demodulate the high frequency signal and thus increasing the bandwidth of the position tracking loop.

4.3.6 No-Load Operation

The no-load operation of each motor is investigated. The necessity of setting a minimum current value was already introduced in Section 3.4. When imposing a

minimum i_d , a problem raises, as at high i_d the saliency becomes weak again due to saturation of d-axis. So it is complicated to select a stable $i_{d,min}$ in the automated simulation without manual fine-tuning. This justifies the choice of minimum i_q . As anticipated in Section 3.4, when operating with sensorless high frequency injection methods, it is convenient to impose a minimum q -current, rather than d -current, to optimize the saliency ratio when operating at zero torque. This allows to have a good saliency value at no-load and not lose the control.

Fig. 4.11 shows the contour plots of the saliency of the three motors under test plotted against the d - and q -currents in relative values. From the graphs, a value of saliency greater than 6 is obtained for all three motors by setting a minimum q -current of 20% the rated current. A minimum saliency of 6 is a safe value to operate at when running at no-load, this ensures the correct functioning of the saliency-based sensorless algorithms at start-up and at low-speeds operation.

4.3.7 High-Load operation

The high-load functioning is a critical operation for saliency-based sensorless techniques, as well. The maximum torque capability of this type of control is tested for all the three motors.

The simulations reported in the following plots were performed solely employing the low-speed saliency-based method of injecting a sinusoidal high frequency voltage signal and demodulating the stator q -current component to retrieve the position error for the sensorless control. The overload operation is affected by the cross-saturation position error which characterizes the current demodulation method, this must be considered when analyzing the following simulations. The starting of the motors and the low-speed operation are intended to be tested, hence the external speed imposed is set constant to 100 *rpm*, to be certain that the sensorless control solely runs on the low-speed saliency-based position error estimation method.

The bandwidth of the PLL is set to $2\pi \cdot 25$ *rad/s*. This should be kept in mind, as the overload operation is sensitive to this parameter.

The plots in Fig. 4.12 show three simulations in torque control mode, one for each of the three motors under test. The same simulation was run for every machine,

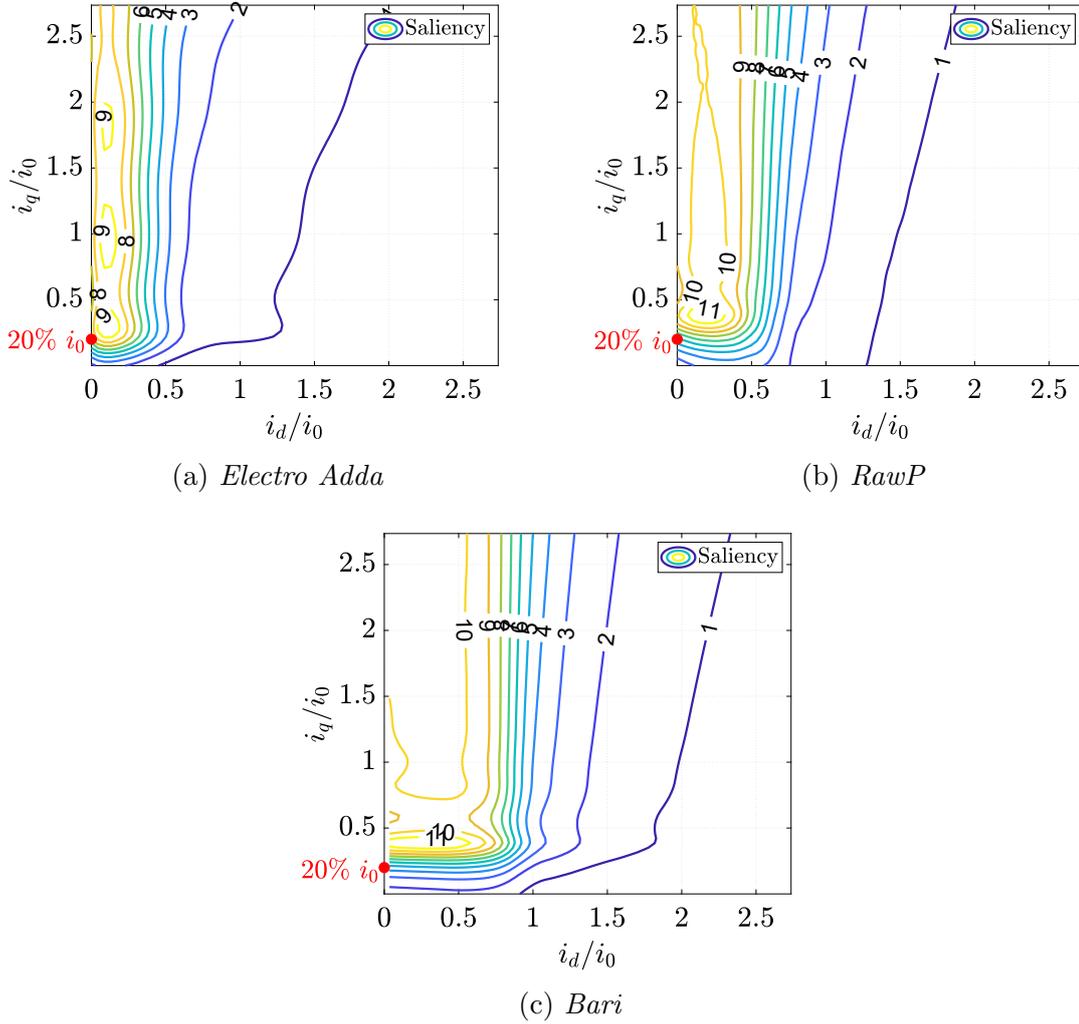
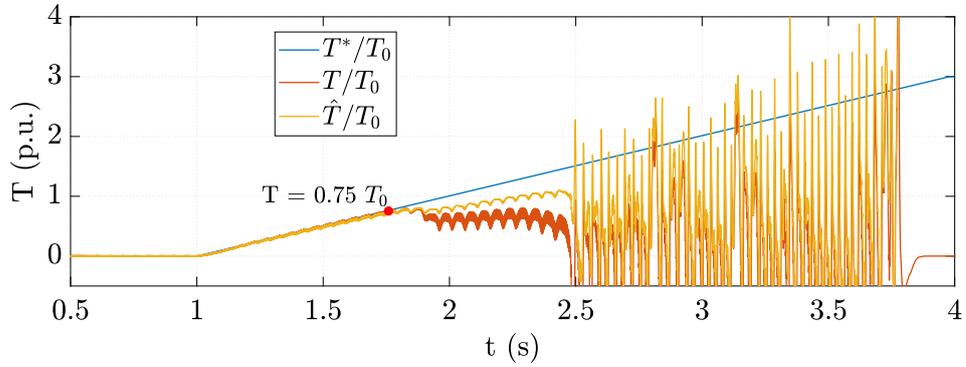


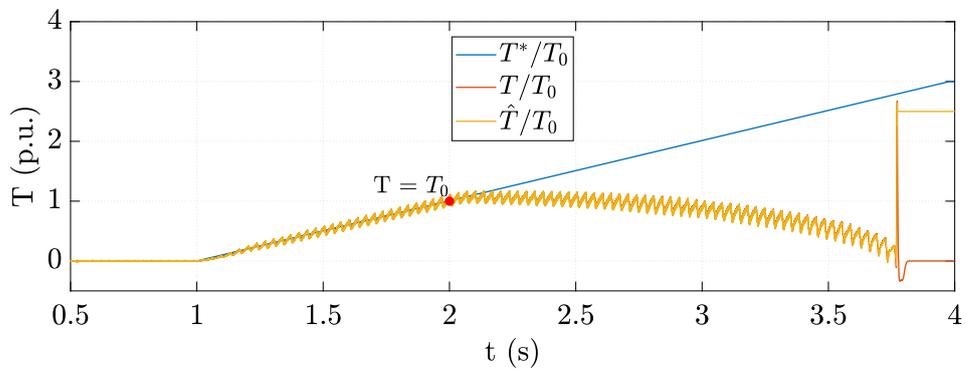
Figure 4.11: Saliency maps of the three motors under test. A red marker is placed at 20% of the rated current value on the i_q axis.

consisting in a torque ramp varying from zero to four times the rated torque of each motor. These plots provide the information on the maximum load limitation of the low-speed methods, as it is clear from the graphs at which torque value the real torque T and the observed torque \hat{T} deviate from the reference torque ramp. These points are highlighted with a red dot in every plot and labeled with the approximate value of torque corresponding to that point relative to the rated torque of the motor.

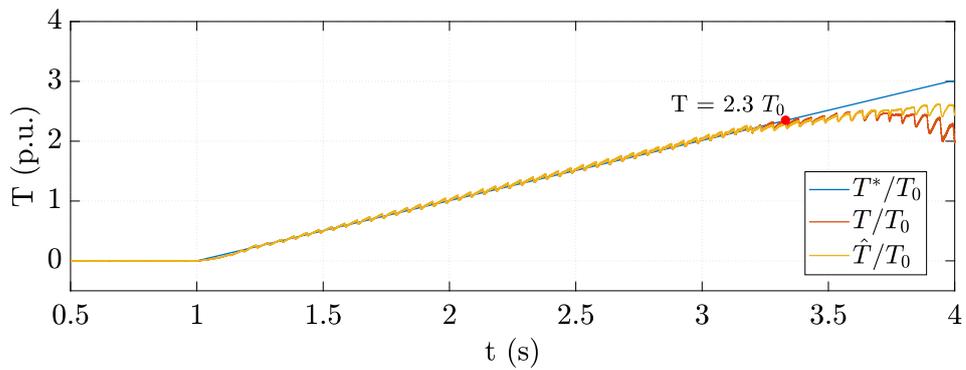
As clear from Fig. 4.12a, the low-speed sensorless control cannot achieve reference



(a) *Electro Adda*



(b) *RawP*



(c) *Bari*

Figure 4.12: Three simulations of the tested motors in torque control mode. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 rpm.

torque values greater than the 75% of the rated torque T_0 in the case of *Electro Adda*.

The same applies to the case of *RawP* reported in Fig. 4.12b. Here the reference torque which was not endured by the control is equal to the rated torque T_0 .

Fig. 4.12c reports the simulation of *Bari* motor. Here the first sign of subsidence of the saliency-based position error estimation appears around 2.3 times the rated torque, a much higher value than the previous two motors.

The torque value at which the control fails the reference torque in each plot of Fig. 4.12 can be explained with the contour plots in Fig. 4.13. The three images represent the saliency contour plot, the MTPA trajectory (in red) and the torque map in *p.u.* (in black) of the motors under test plotted against the *d*- and *q*-currents in relative values with respect to the rated current. From these images, the saliency at each operating point of the MTPA curve can be evaluated and the corresponding torque value can be defined. It is important to remind that the already poor saliency in regions such as the ones where the high-load operation takes place is further deteriorated by the cross-saturation error which affects the current demodulation method.

Fig. 4.13a is related to *Electro Adda*. It shows that to the operating point matching with 75% of the rated torque, it corresponds a saliency slightly lower than 4. This value is considerably low and results in the loss of control when operating with sensorless high frequency injection techniques, which highly rely on saliency to work. This explains the failure of the simulation in Fig. 4.12a at a reference torque value equal to $0.75 \cdot T_0$. The same considerations apply to *RawP* and *Bari* motors, whose saliency and torque maps are reported in Fig. 4.13b and 4.13c, respectively. When a reference torque equal to the rated torque is imposed to the *RawP* motor, the saliency of the machine is around 5; but as the reference torque increases beyond this value, the control loses the reference, as in the simulation reported in Fig.4.12b.

The *Bari* motor exhibits considerably higher values of saliency compared to the other motors, as shown in Fig. 4.13c, even when operating at twice the rated torque, where the *Bari* motor still has a saliency between 4 and 5, a border-line value for high frequency injection saliency-based sensorless methods. Beyond a

reference torque of $2.3 \cdot T_0$ the torque control results in an unsteady chasing of the reference torque followed by the failure of the control if the reference value further increases, as proven in Fig. 4.12c, due to a drop in saliency below 4.

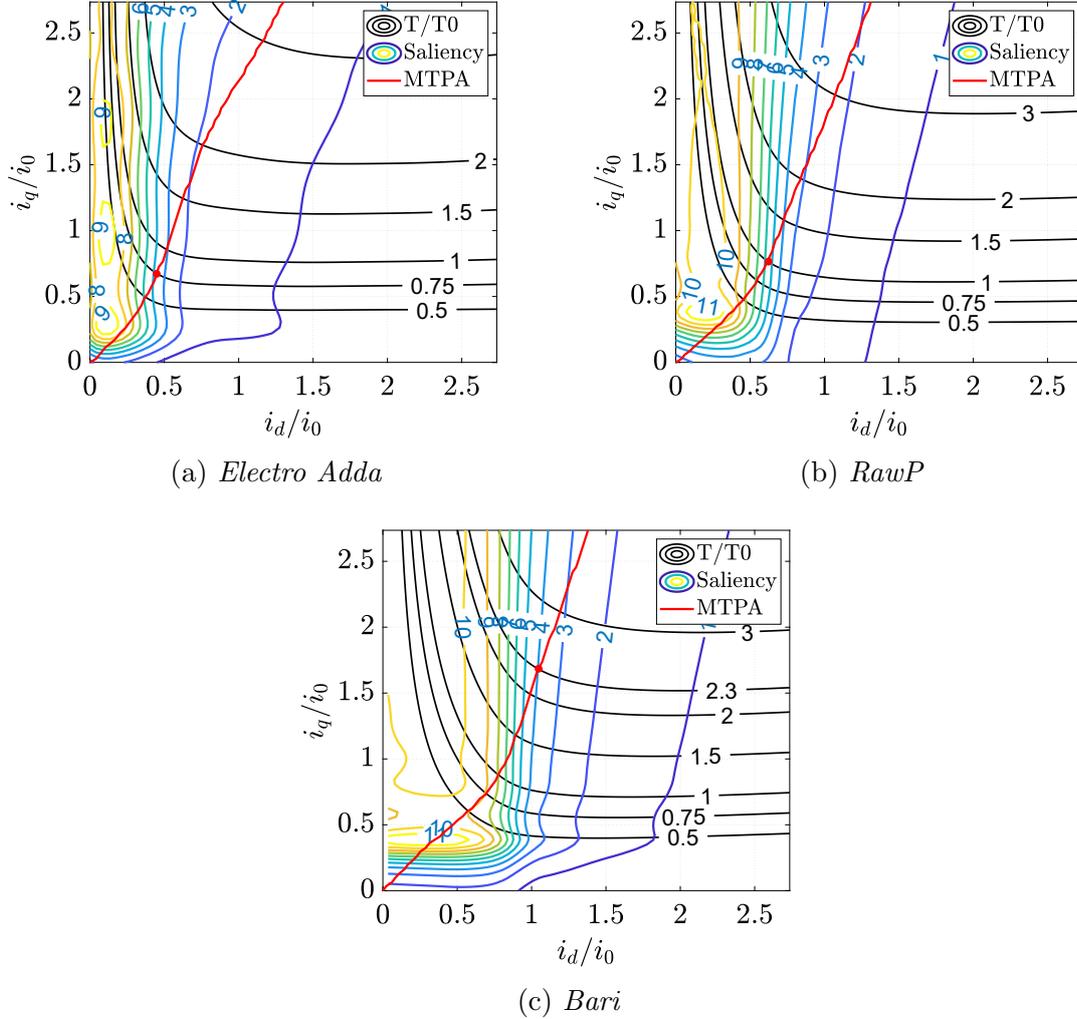


Figure 4.13: Saliency maps, MTPA trajectory (in red) and torque map in *p.u.* (in black) of the three motors under test.

The loss of control is partially due to the cross-saturation effect, as well. As torque increases, the position error due to cross-saturation also increases, leading to the loss of control for high values of torque, where the position error is too big to be endured. If the torque ramp simulation is run with flux demodulation technique, which overcomes the problem of cross-saturation, it can be noticed that

the torque value at which the control is lost is slightly higher than the case of current demodulation. In Fig. 4.14, it is reported a simulation in flux demodulation with the *Electro Adda* motor. It is visible that the control is lost around a torque value equal to the rated torque, a higher value than for the case of current demodulation in Fig. 4.15, where torque deviates from the reference value around $T = 0.75 T_0$. Also, it is shown that the position error remains almost zero in Fig. 4.14 until the loss of control, while in Fig. 4.15 the position error keeps increasing as the torque ramps up. So the premature loss of control with current demodulation in Fig. 4.15 respect to the case with flux demodulation in Fig. 4.14, is due to the cross-saturation position error that pushes the operation into poor saliency regions.

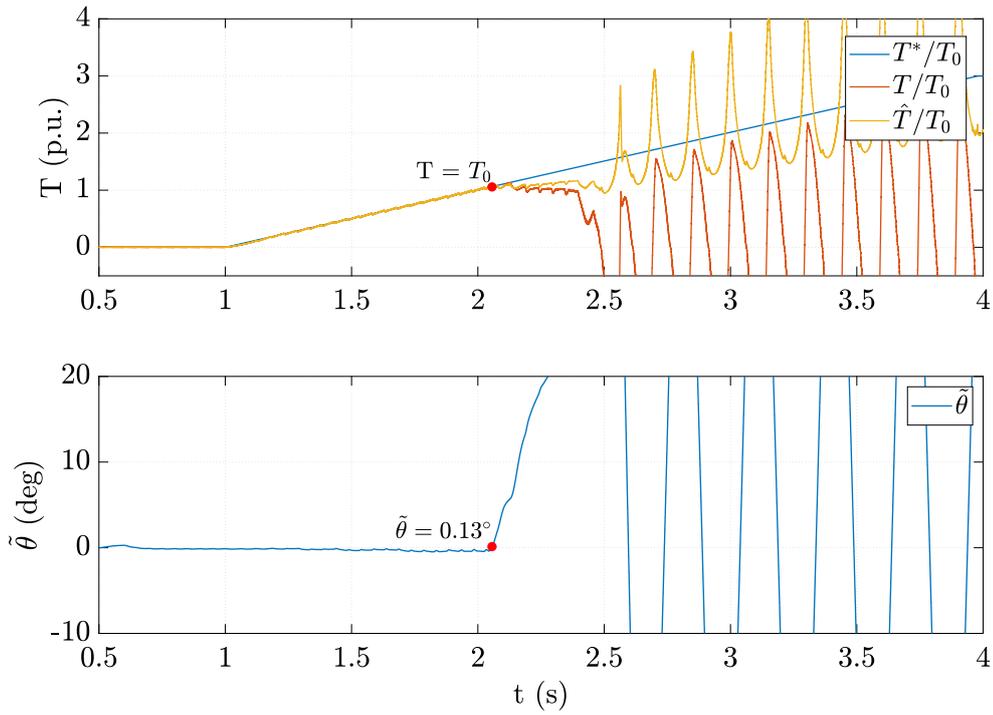


Figure 4.14: A simulation of the *Electro Adda* motor in torque control mode employing flux demodulation and a PLL bandwidth of $2\pi \cdot 25 \text{ rad/s}$. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 rpm . The reference torque is lost around $T = T_0$.

As mentioned at the beginning of this section, overload operation is sensitive

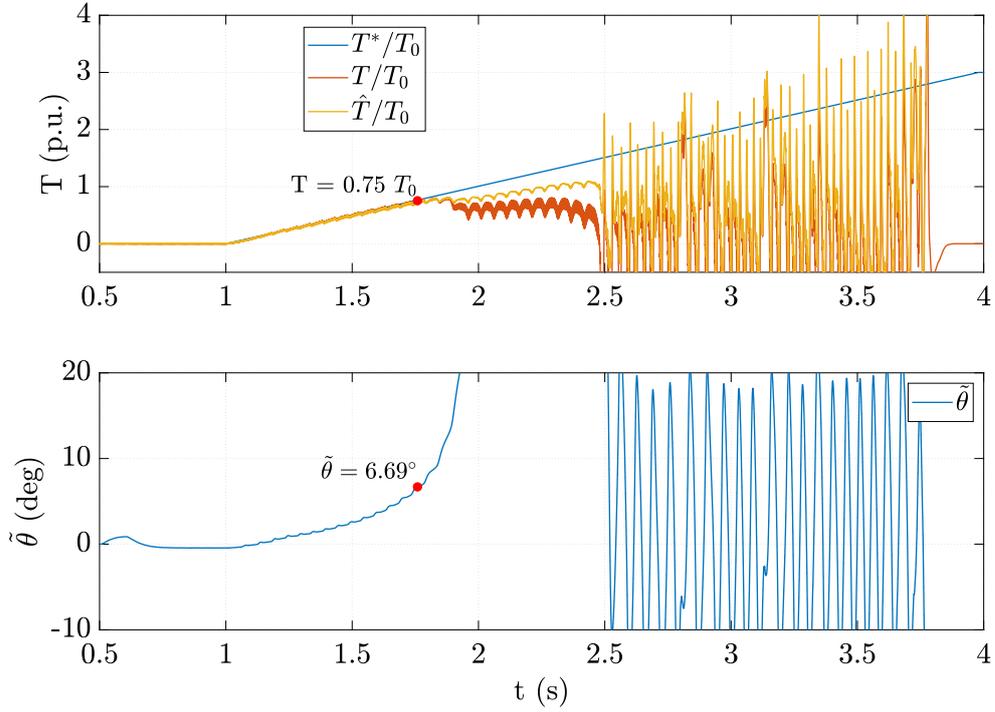


Figure 4.15: A simulation of the *Electro Adda* motor in torque control mode employing current demodulation and a PLL bandwidth of $2\pi \cdot 25 \text{ rad/s}$. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 rpm . The reference torque is lost around $T = 0.75 T_0$.

to the bandwidth of the PLL, which was set to $2\pi \cdot 25 \text{ rad/s}$ for the simulations reported above. If the PLL bandwidth is lowered, the reference torque achieved increases, as for the case of the *Electro Adda* motor, shown in Fig.4.16 and 4.16a, where the PLL bandwidth is set to $2\pi \cdot 15 \text{ rad/s}$ and $2\pi \cdot 10 \text{ rad/s}$ and the maximum torque fulfilled becomes $1.5T_0$ and $1.76T_0$, respectively.

Since these considerations only apply for saliency-based sensorless methods for rotor position estimation, employed at zero- to low-speeds, particular attention must be paid in overloading the the motor at start-up when such techniques are used and the capability of each motor should be assessed for each case.

This is the reason why the speed control dynamic response to torque load step, as in the case of Fig. 4.24, is rather relaxed, to avoid harsh torque peaks and to allow any motor to be controlled with the automatically generated parameters,

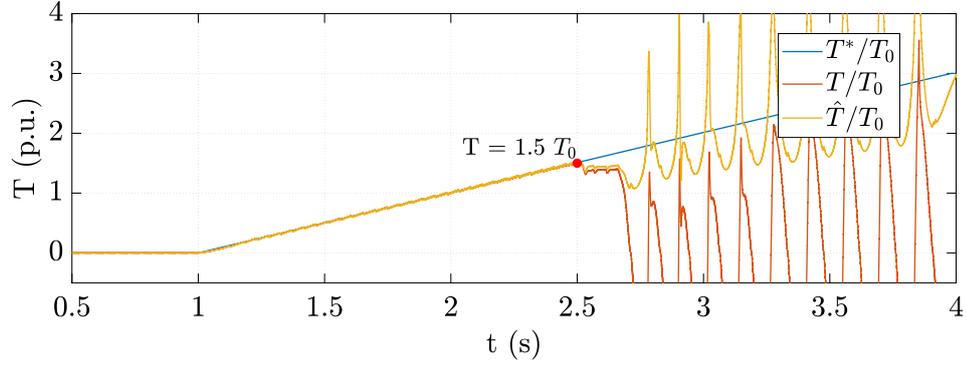
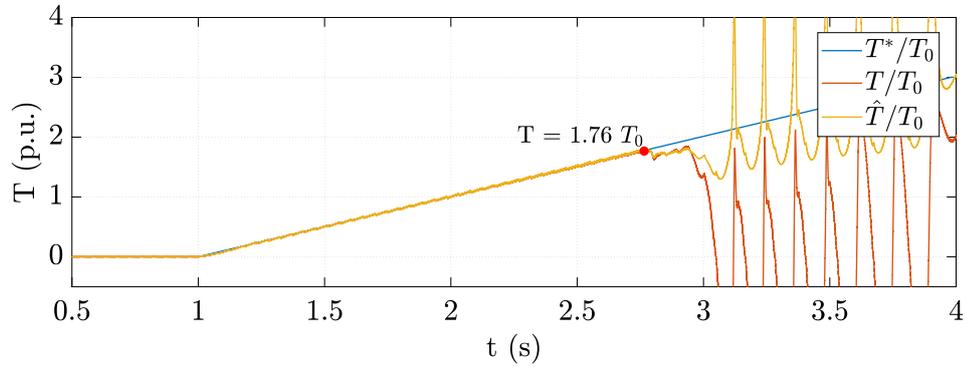
(a) *Electro Adda*(b) *RawP*

Figure 4.16: Two simulations of the *Electro Adda* motor in torque control mode employing flux demodulation. The torque reference is varied linearly from zero to four times the reference torque T_0 . The external speed imposed is set at 100 *rpm*. In the plots in (a) the bandwidth of the PLL is set to $2\pi \cdot 15$ *rad/s*, while in (b) it is set to $2\pi \cdot 10$ *rad/s*.

even the ones that exhibit poor saliency at high loads, for which the saliency-based position estimation methods would struggle to keep the control, or the ones that are particularly sensitive to the control parameters. This influenced the calibration of the bandwidth of the speed loop.

4.3.8 Calibration Optimization

After obtaining the first-approach results from *syreDrive*, the user will be able to easily edit the C-files to further adjust the control parameters for a dedicated tuning for the motor under test.

The control parameters are all stored in the *User_Constants.h* header file. These quantities are reported in Fig. 4.17, as they appear in the header file. *KOBS* is the hybrid flux observer gain. *OMEGA_BW* and *OMEGA_BI* are the bandwidths of the speed and current loop, respectively. *OMEGA_0_INJ* is the cut-off frequency of the low-pass filter needed for the demodulation of the high frequency signal in the case of sinusoidal voltage injection. *WB_PLL_SIN* and *WB_PLL_SQUARE* are the bandwidths of the PLL for the case of sinusoidal and square wave voltage injection, defined separately. *OMEGA_G* is the speed span ω_g of the fusion scheme presented in Section 3.6. The user only needs to change these values and run a simulation from *syreDrive* to evaluate the performance of the machine with the new setups of the control parameters. An example of calibration optimization of the control

```

45
46 // Flux Observer
47 #define KOBS          TWOPI*10.0f // 300 rpm
48
49 // PI Reg
50 #define OMEGA_BW      TWOPI*1.0f
51 #define OMEGA_BI      TWOPI*75.0f
52 #define OMEGA_0_INJ  TWOPI*50.0f
53 #define WB_PLL_SIN    TWOPI*10.0f
54 #define WB_PLL_SQUARE TWOPI*25.0f
55
56 // Control fusion
57 #define OMEGA_G        TWOPI*4.0f // 120 rpm
58

```

Figure 4.17: The control parameters as they appear in the *User_Constants.h* header file.

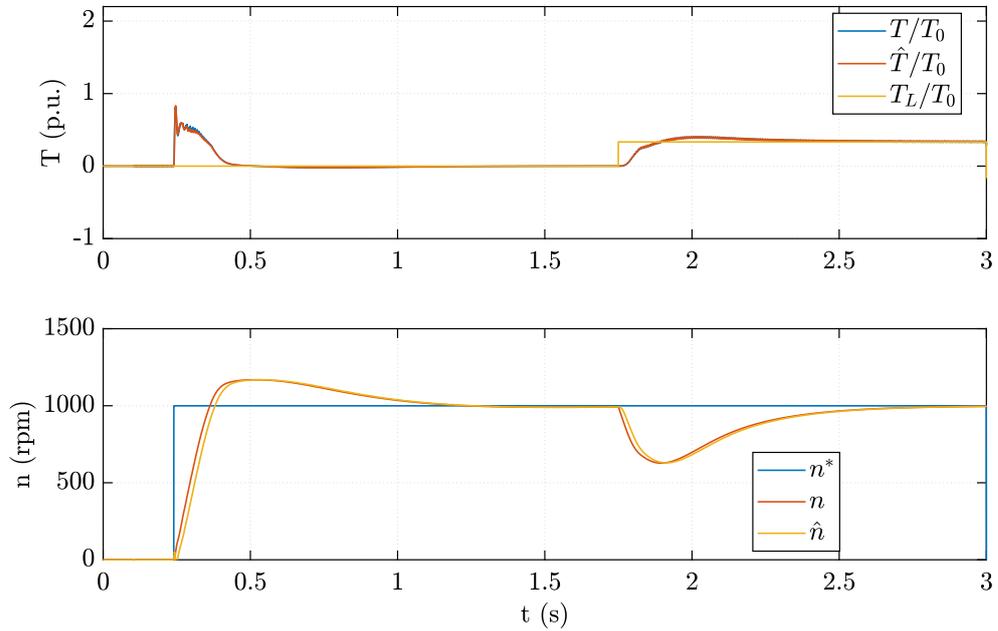
parameters is provided in Fig. 4.18. In this simulation the bandwidth of the speed loop was modified to improve the performance of the *Bari* motor. As shown in Section 4.3.7, this machine presents a much higher saliency per Newton-meter ratio than the other two motors studied, as visible comparing its saliency and torque map in Fig. 4.13c to the maps of the two other motors in Fig. 4.13a and 4.13b. Hence, a more performing response can be claimed from *Bari* motor, as the saliency-based sensorless methods employed at start-up and at low speeds can make the most out of the sufficiently high values of saliency that this motor exhibits even when it is

forced to work at high loads. In the simulation in Fig. 4.18, the bandwidth of the speed loop Ω_ω is increased from $2\pi \cdot 1 \text{ rad/s}$, in Fig. 4.18a, to $2\pi \cdot 4 \text{ rad/s}$, in Fig. 4.18b, to have a faster response to speed steps. The initial acceleration is visibly quicker in the case of $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$ and the response to the torque step is faster, as well. This simple example of ad-hoc tuning of the control of the *Bari* motor was easily achieved just changing a single value in the *User_Constants.h* header file.

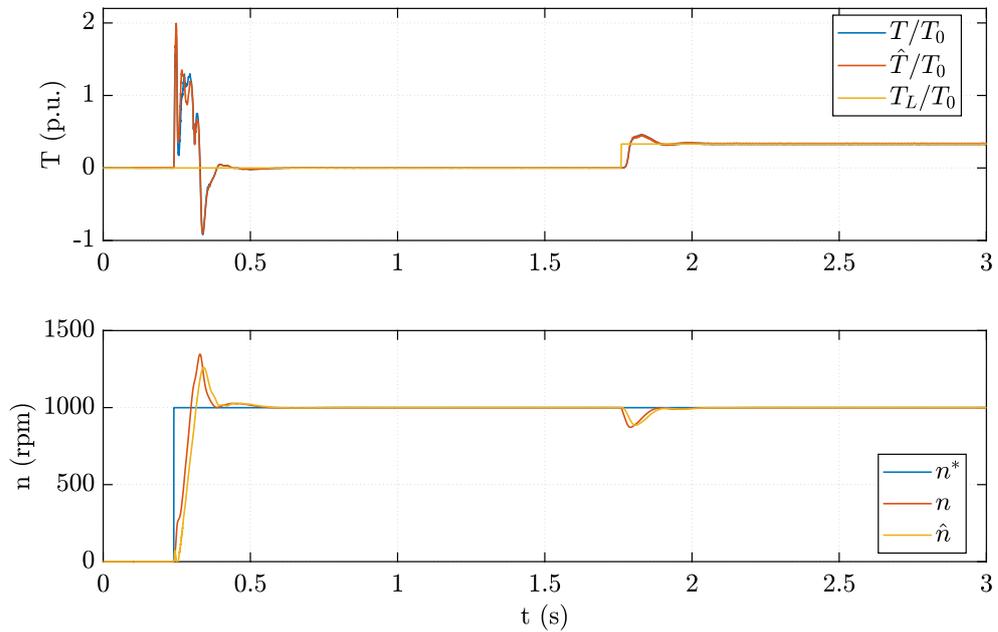
The same test is performed for the *Electro Adda* and the results are reported in Fig. 4.19. The bandwidth $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, selected as general value to be used in *syreDrive*, proved to work fine with *Electro Adda*, as shown in Fig. 4.19a. But when a bandwidth of $2\pi \cdot 4 \text{ rad/s}$ is applied for the speed control of *Electro Adda*, the results are not as promising as they were for *Bari* motor. As shown in Fig. 4.19b, the control cannot endure the initial speed step with such a harsh dynamic and ends up failing the reference speed value. The best that could be done out of the *Electro Adda* motor was to increase the speed loop bandwidth up to $2\pi \cdot 1.75 \text{ rad/s}$. In this case the maximum torque achieved by the control is $T = 0.75T_0$, the maximum torque value that the low-speed saliency-based sensorless control techniques can withstand, as shown in Fig. 4.12a. Increasing the bandwidth further beyond this value results in the failure of the control. The torque and speed plots of this simulation are reported in Fig. 4.21, where it can be seen that the response of the motor is improved with respect to Fig. 4.19a, where a lower bandwidth was employed.

The same applies to the *RawP* motor. The results of a simulation in speed mode with a bandwidth of the speed loop of $2\pi \cdot 1 \text{ rad/s}$ are reported in Fig. 4.20a. The response is slow, but the control fulfills the reference speed value, even when a torque load step is applied. In Fig. 4.20b, instead, a bandwidth of $2\pi \cdot 4 \text{ rad/s}$ is used. This bandwidth, which provided better results for the *Bari* motor, results in the control failure in the case of the *RawP* motor, as visible in Fig. 4.20b, as it did for the *Electro Adda* motor. The highest speed loop bandwidth that prevented the control of *RawP* to fail was $\Omega_\omega = 2\pi \cdot 1.8 \text{ rad/s}$, as it resulted in the maximum torque that the saliency-based sensorless method could withstand at start-up, namely $T = T_0$ for the case of *RawP*, as visible in Fig. 4.12b

This proves that the general validity of the calibration of the speed loop described

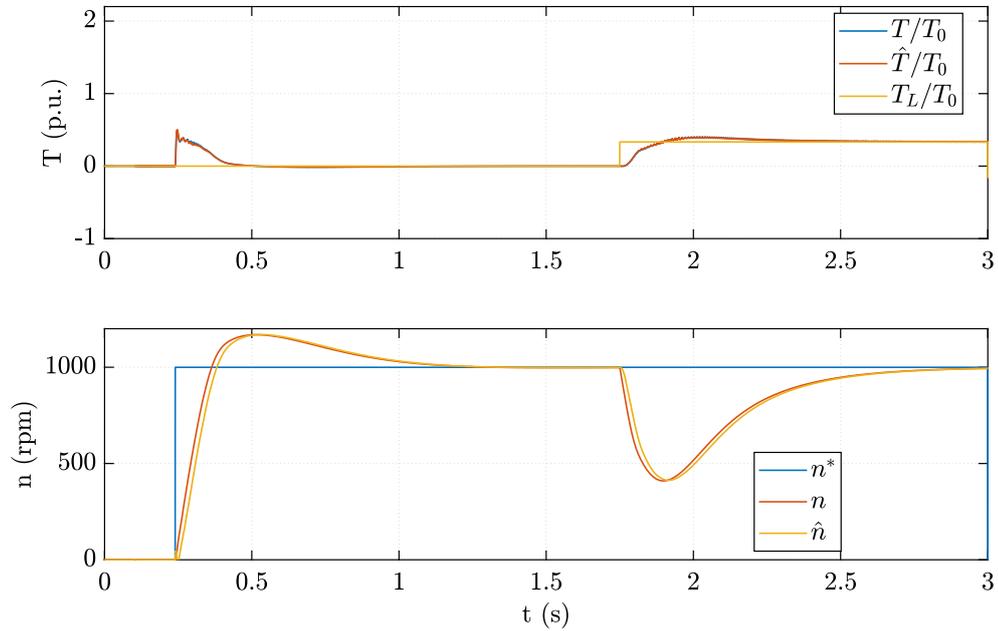


(a) $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$.

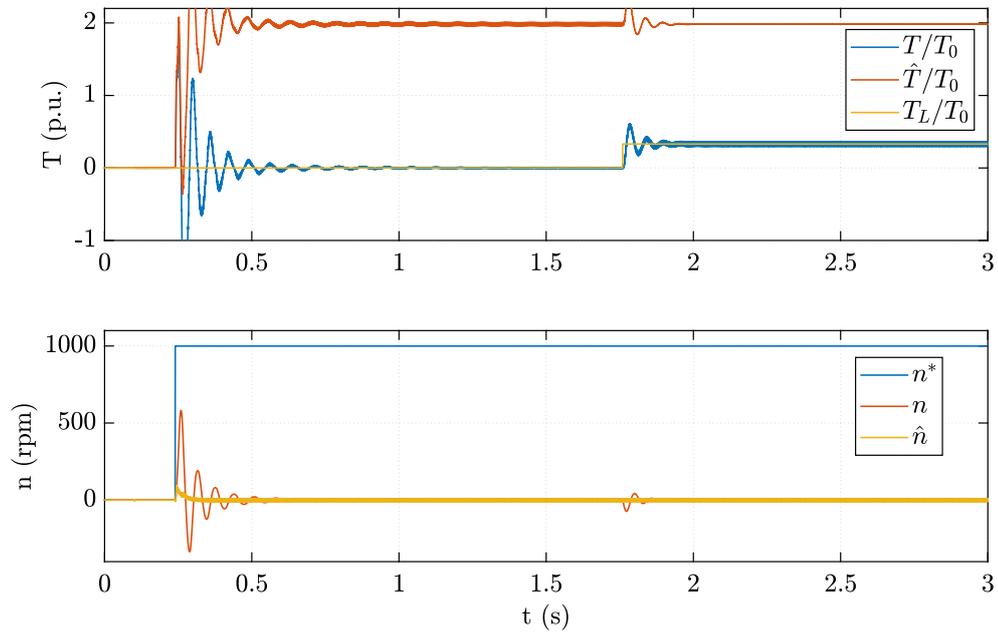


(b) $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$.

Figure 4.18: Two simulations in speed control mode of the *Bari* motor. The one in (a) was performed using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, while the one in (b) using $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$.

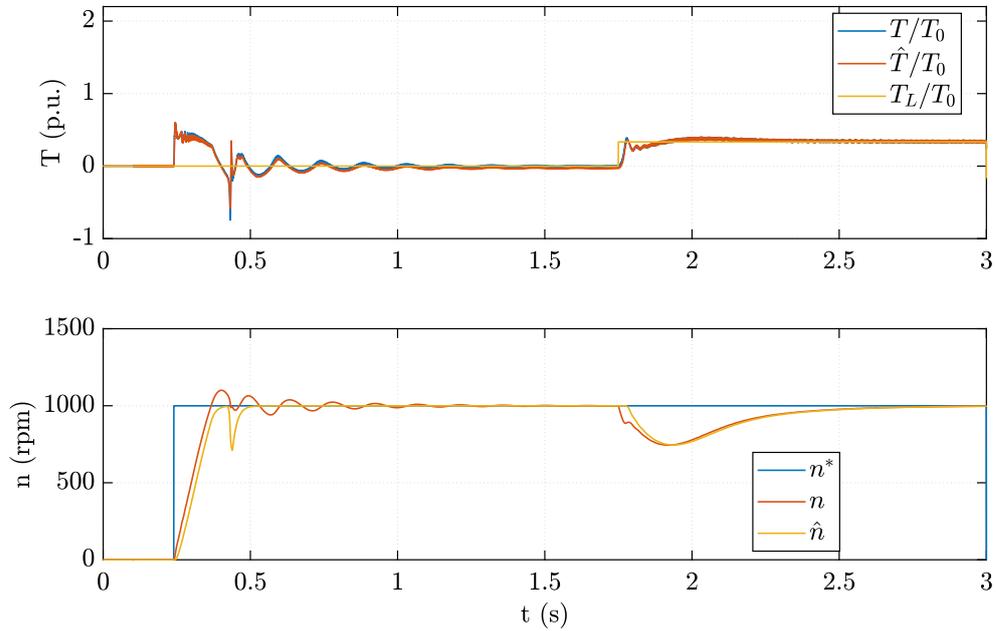


(a) $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$.

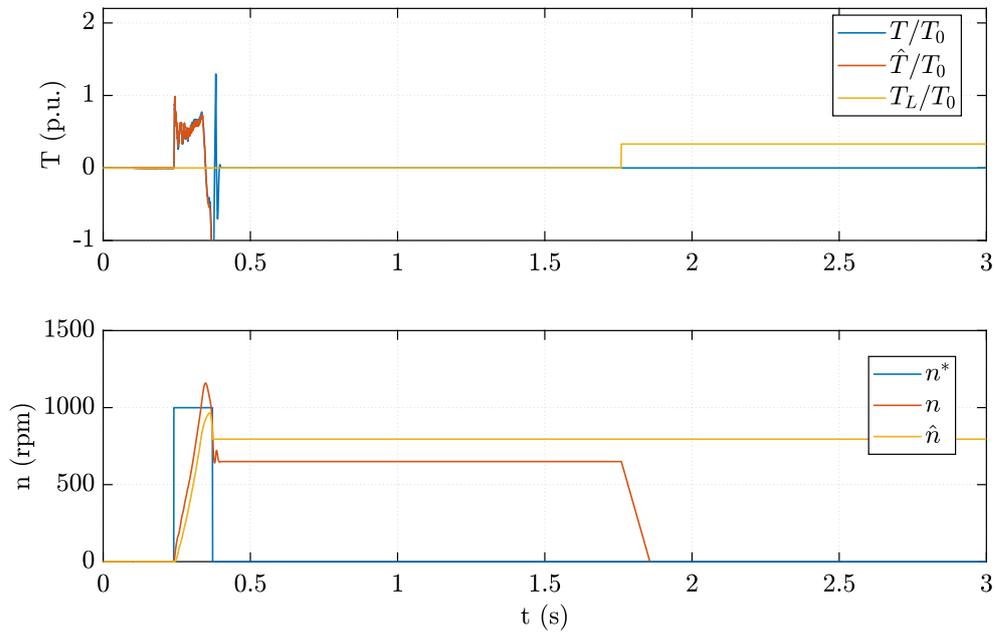


(b) $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$.

Figure 4.19: Two simulations in speed control mode of the *Electro Adda* motor. The one in (a) was performed using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, while the one in (b) using $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$.



(a) $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$.



(b) $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$.

Figure 4.20: Two simulations in speed control mode of the *RawP* motor. The one in (a) was performed using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1 \text{ rad/s}$, while the one in (b) using $\Omega_\omega = 2\pi \cdot 4 \text{ rad/s}$.

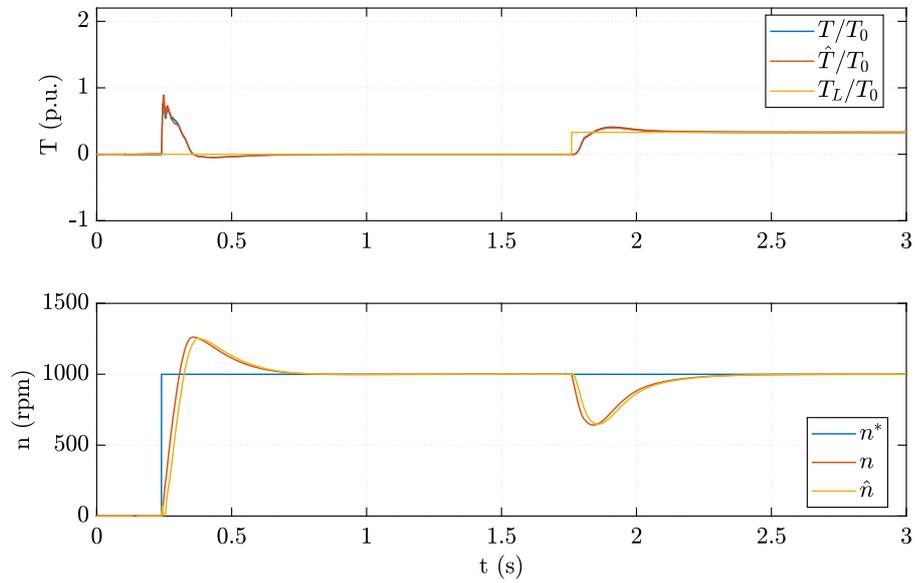


Figure 4.21: A simulation in speed control mode of the *Electro Adda* motor, using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1.75 \text{ rad/s}$.

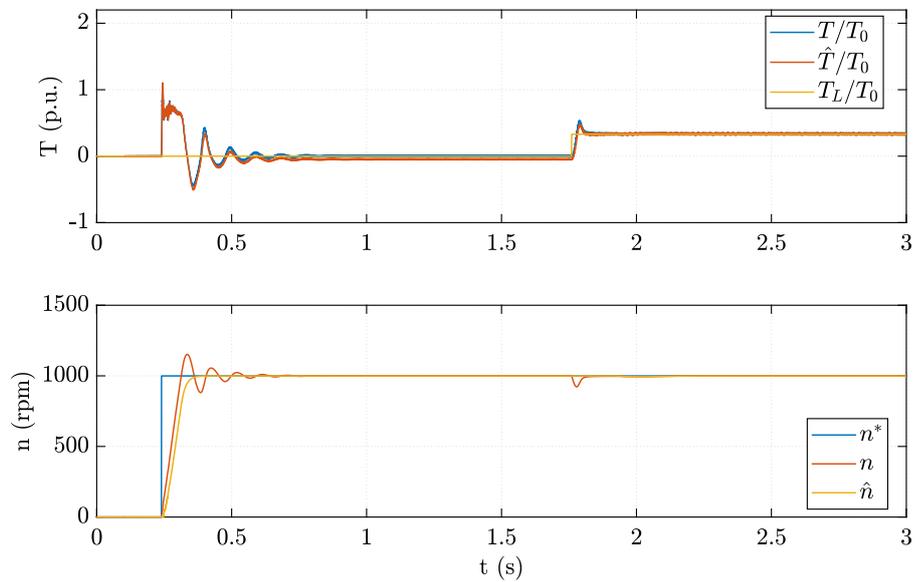


Figure 4.22: A simulation in speed control mode of the *RawP* motor, using a bandwidth of the speed loop $\Omega_\omega = 2\pi \cdot 1.8 \text{ rad/s}$.

in Section 4.3.2 holds for most SyR motors, as a more daring tuning of the control parameters would not be suitable for machines with poor saliency when operating at high torque, such as the *Electro Adda* motor. For this reason, the possibility for further dedicated tuning is left to the user, after the generation of the generally valid control with *syreDrive*.

4.4 Simulation Results for the Sensorless Control

The position error estimation techniques presented in Chapter 3.1 have been analyzed to determine their capability and constraints and to assess their performances. First, the low-speed range sensorless position estimation methods are tested, then the high-speed techniques, along with the fusion scheme are examined.

The following simulations were performed on the *Bari* SyR motor, whose ratings are reported in table 4.1.

4.4.1 Low-Speed Range

The low-speed position estimation techniques are based on the injection of a high-frequency voltage signal. The multiple options for this method presented in Section 3.1 are here analyzed and their functioning is verified.

First the correct operation of the demodulation procedure is assessed. A simulation using sinusoidal high frequency voltage signal injection and demodulating q -current component is performed.

Analyzing the following plots in this chapter, the demodulation appears to operate as expected. The position error signal is obtained by multiplying the demodulated and filtered magnitude of the high frequency current component by the coefficient k_e , as in Eq. (3.6).

The demodulation was verified for all the high frequency injection methods and performed well in all the cases.

In Fig. 4.23, four pairs of graphs are presented, one for each combination of the

low-speed position error estimation technique tested, which are:

- sinusoidal injection & stator current demodulation;
- square wave injection & stator current demodulation;
- sinusoidal injection & stator current-model flux demodulation;
- square wave injection & stator current-model flux demodulation.

All the results of these simulations are in torque control mode. Each graph is made up of three subplots, respectively for torque, speed and position error. The first one shows the reference torque T^* , the actual torque realized by the motor T and the torque computed with the observed quantities \hat{T} ; all three values are reported in p.u. relative to the rated torque of the motor. The second one presents the speed sequence performed during the simulation in *rpm* and the observed speed. Since this analysis deals with low-speed position error estimation methods, the external speed imposed reaches a maximum absolute value of 300 *rpm*. The last subplot of the series depicts the position error $\tilde{\theta}$ and the expected steady-state position error $\tilde{\theta}_0$ due to cross-saturation, both in degrees. $\tilde{\theta}_0$ is computed as

$$\tilde{\theta}_0 = -\frac{1}{2} \tan^{-1} \frac{l_{dq}}{l_{\Delta}} \quad (4.8)$$

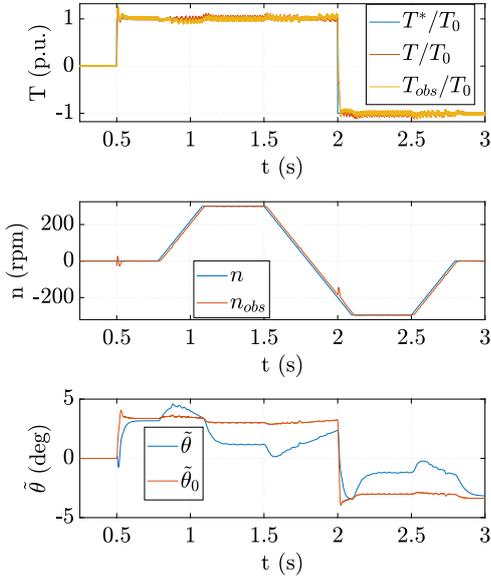
where $l_{\Delta} = \frac{l_d - l_q}{2}$.

By comparing the four pairs of plots, no major difference in torque response is noticeable, though the torque ripple due to the high frequency voltage injection is clearly visible in all of the plots.

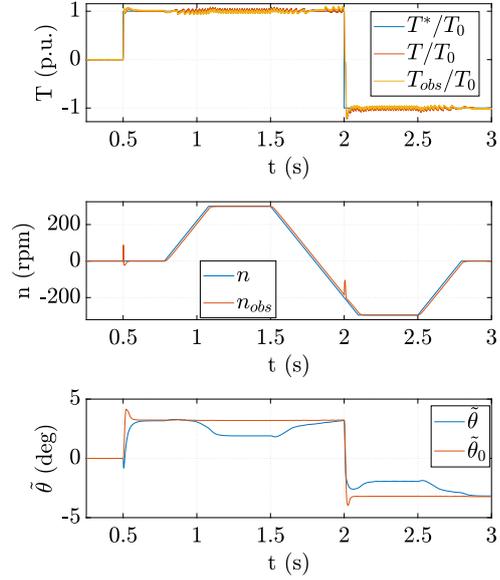
While it is evident that demodulating current-model flux (Fig. 4.23c and 4.23d) overcomes the problem of cross-saturation, as the position error $\tilde{\theta}$ is close to zero during steady-state, while in the case of stator current demodulation (Fig. 4.23a and 4.23b) the position error largely matches the expected steady-state position error $\tilde{\theta}_0$, except during the speed transient.

4.4.2 High-Speed Range

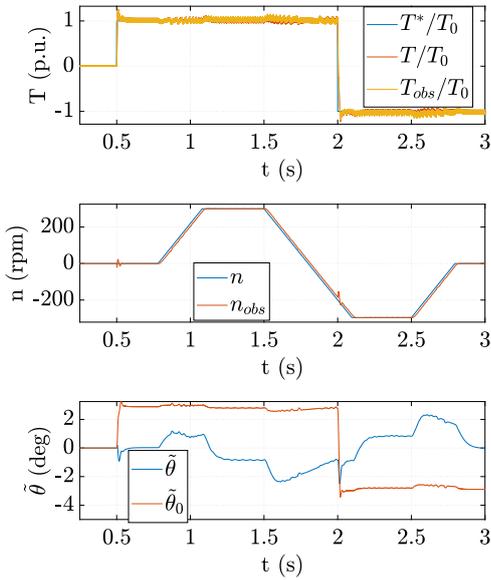
The high-speed position error estimation techniques presented in Section 3.5 are tested to verify their functioning.



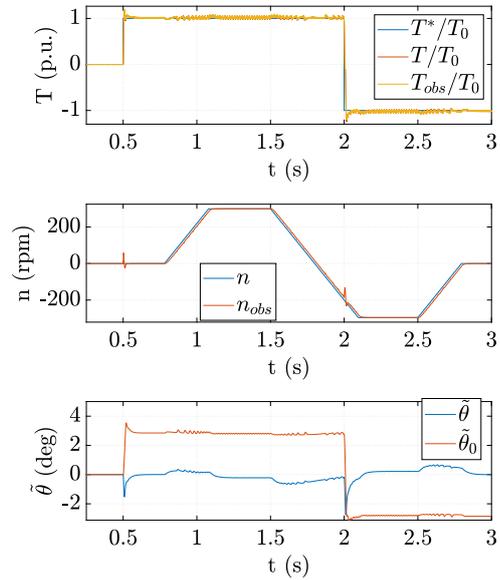
(a) Sinusoidal injection & current demodulation



(b) Square wave injection & current demodulation.



(c) Sinusoidal injection & flux demodulation.



(d) Square wave injection & flux demodulation.

Figure 4.23: Four torque and speed plots of torque control simulations using different low-speed position error estimation methods.

The high-speed range fundamental model-based methods considered are the active flux and the adaptive projection vector for position error estimation.

The plots in Fig. 4.24 are taken from a simulation performed in speed control mode, employing the full-speed sensorless control scheme described in Section 3.6. The low-speed position error estimation method used is high frequency injection of square wave voltage and the demodulated quantity to retrieve the position error signal is current-model flux. While for the high-speed operation, APP is employed.

The reference speed is perfectly achieved in Fig. 4.24. The observed quantities such as speed \hat{n} and torque \hat{T} are accurately reproduced and the control shows a good response to torque load steps.

The fusion scheme, synthesized by the fusion coefficient f_ω , proves to work fine, as a smooth linear transition is fulfilled around 300 *rpm* and no disturbance in speed can be noticed in the span of transition.

Also, it can be seen how torque is more affected by ripple when the sensorless control operates with high frequency injection methods at low speeds, i.e. when the fusion coefficient f_ω is 1. The torque smooths out when the control runs at high speeds and APP technique takes over, when f_ω assumes a null value. The torque ripple is due to the high frequency component present in the currents, especially the *d*-axis component, where injection takes place.

The rotor position is accurately estimated by the position observer, as shown in Fig. 4.24. The position error $\tilde{\theta}$ is mainly null during steady state and remains below 4° during accelerations and decelerations, with minor oscillations when torque load steps are applied.

A similar simulation to the one of Fig. 4.24 is performed; the only difference between the two is that active flux is employed for estimating the position error in high-speed region instead of APP. The results are reported in Fig. 4.25. A minimum torque load is constantly applied during the whole simulation and the control performs well in this situation, as it can be seen from Fig. 4.25, keeping the position error below 1°.

If this torque load is removed, the active flux control fails when no load is applied to

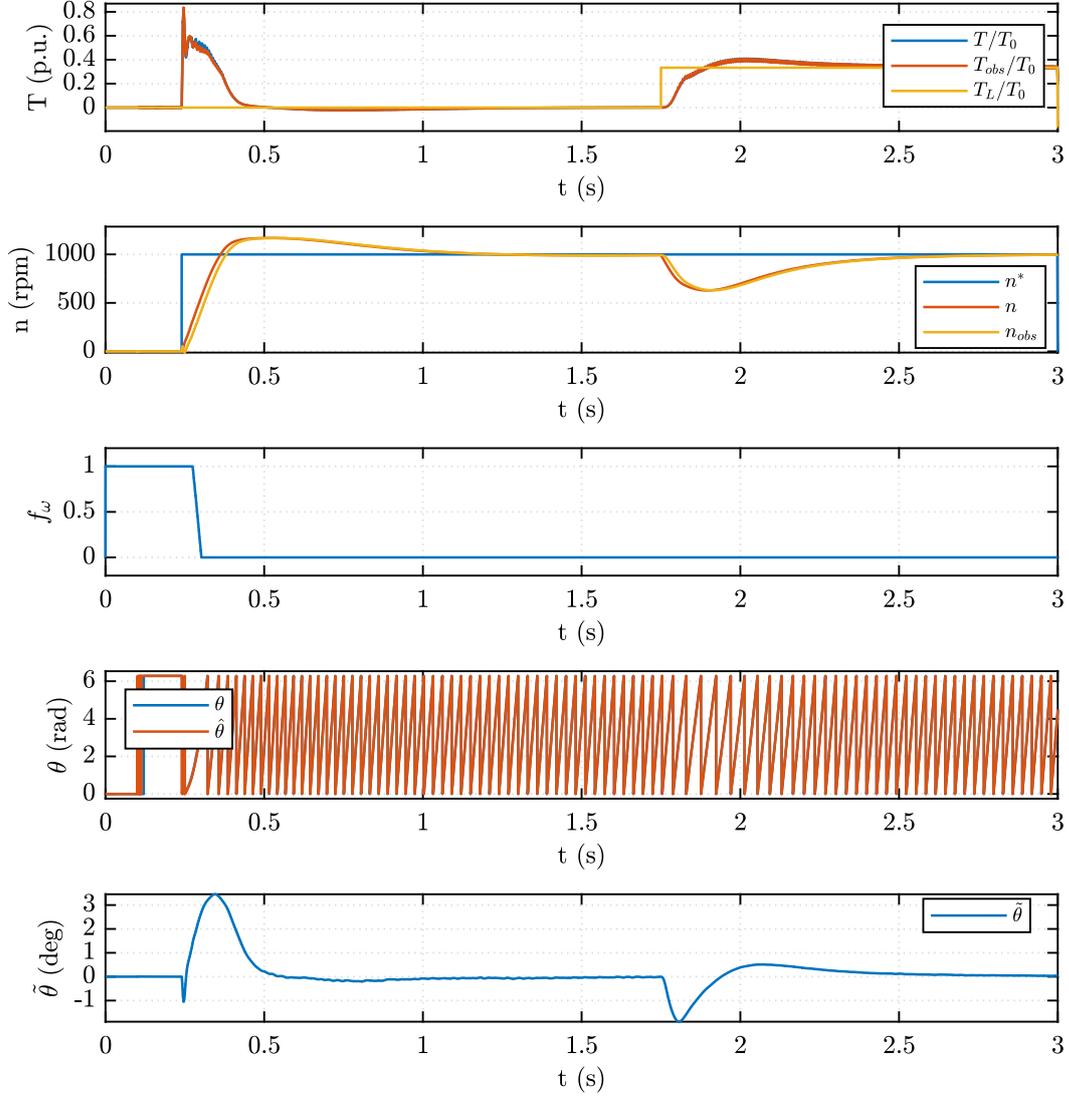


Figure 4.24: From top to bottom: the actual torque, the observed torque and the load torque in $p.u.$; the reference speed, the actual speed and the observed speed in rpm ; the fusion coefficient f_ω ; the real position θ and the observed position $\hat{\theta}$ in rad ; the position error $\tilde{\theta}$ in degrees. The plots are taken from a simulation in speed control employing high frequency voltage injection and flux demodulation in the low-speed region, and APP during the high-speed operations.

the motor. This is the case of Fig. 4.26. By comparing the results with APP in the torque plot in Fig. 4.24 to the ones with active flux in Fig. 4.26, it is visible that the effectiveness of active flux at no-load is worsened by the fact that a minimum current is set on the q -axis instead of d -axis. The reason for doing so is that saliency-based position error estimation methods used at low-speeds require a minimum value of saliency to perform well, as asserted in Section 3.4. Hence, the user must be wary of using active flux technique at no load. Alternatively, a minimum d -axis current imposition permits no load operation, however, compromising the ease of automated calibration for low speeds region.

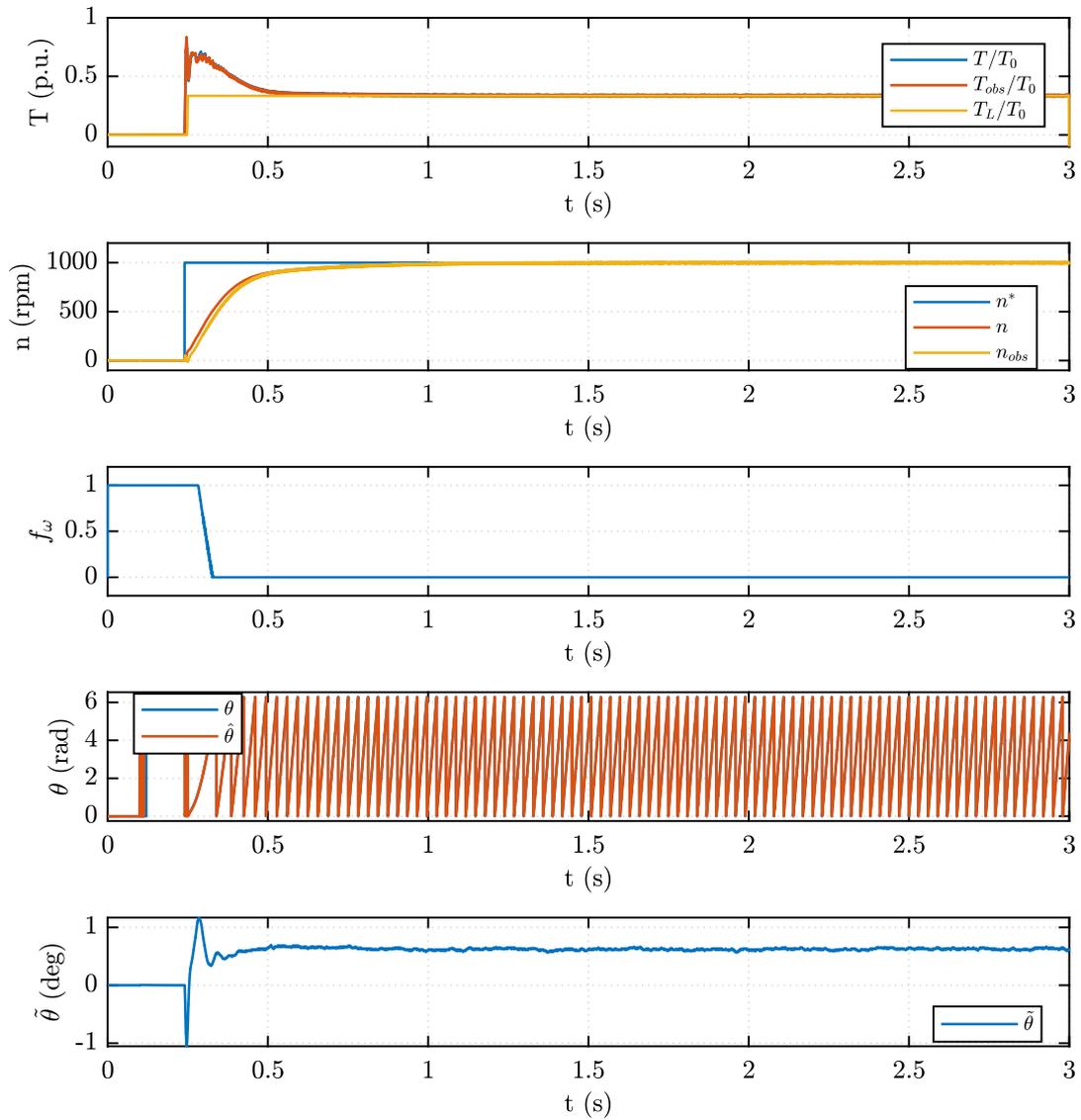


Figure 4.25: From top to bottom: the actual torque, the observed torque and the load torque in *p.u.*; the reference speed, the actual speed and the observed speed in *rpm*; the fusion coefficient f_ω ; the real position and the observed position; the position error. The plots are taken from a simulation in speed control employing high frequency voltage injection and flux demodulation in the low-speed region, and active flux during the high-speeds operation. Notice that a minimum torque load is applied through out the whole simulation.

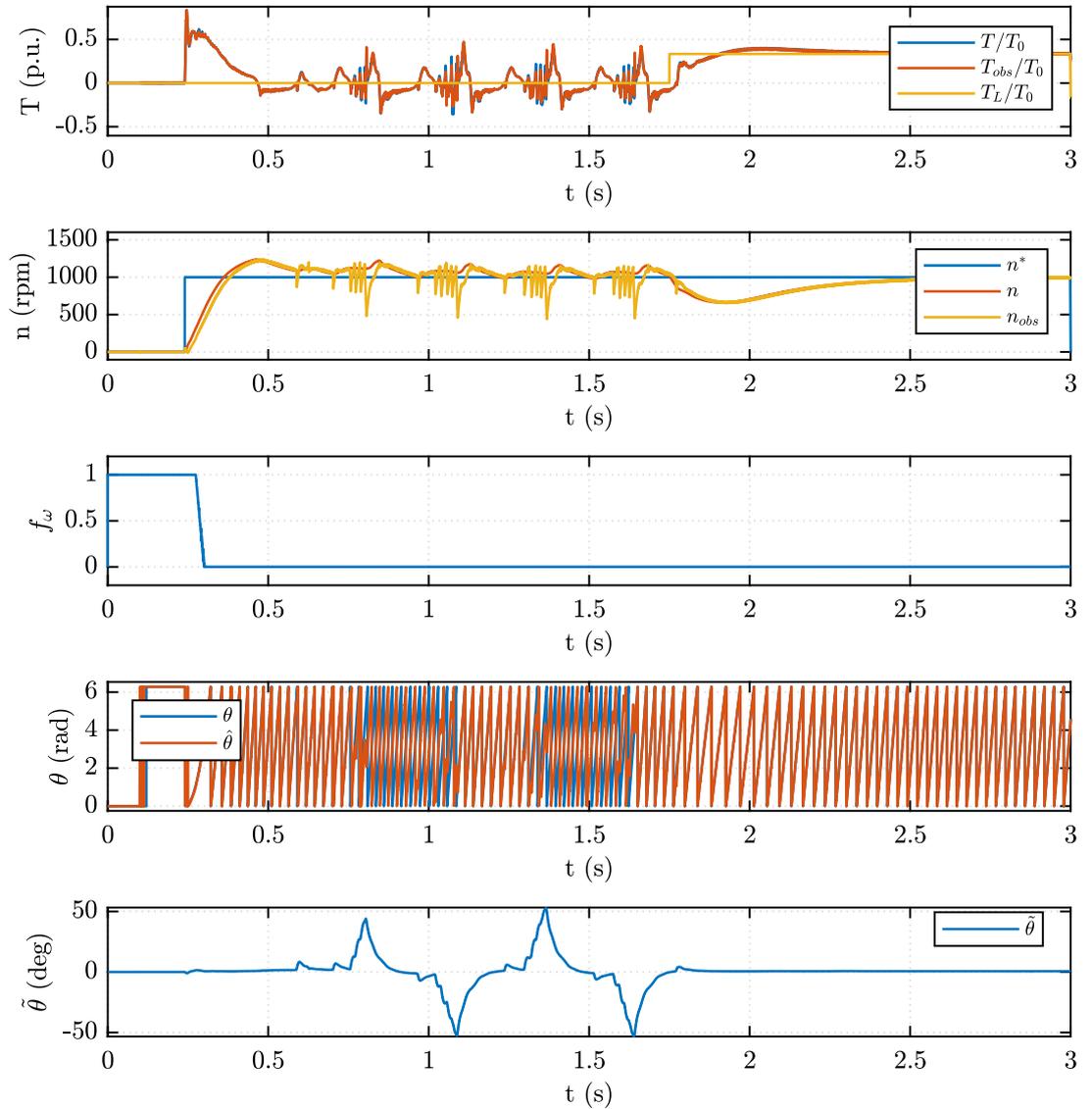


Figure 4.26: From top to bottom: the actual torque, the observed torque and the load torque in $p.u.$; the reference speed, the actual speed and the observed speed in rpm ; the fusion coefficient f_ω ; the real position and the observed position; the position error. The plots are taken from a simulation in speed control employing high frequency voltage injection and flux demodulation in the low-speed region, and active flux during the high-speeds operation. No torque load is applied in this case.

Chapter 5

Conclusions

The aim of this work was to provide a tool for fast and easy evaluation of control strategy for synchronous reluctance motors, given solely the machine data. In this thesis work, a method for automated generation of control files and a Simulink model for any given synchronous reluctance motor is introduced and implemented in the software *SyR-e* in the form of the new add-on, *syreDrive*.

An introduction to synchronous reluctance motors and their control strategy is provided, the mathematical model of the SyR machine is reported and the field oriented control scheme for this kind of motor is described. The *dq-theta* magnetic model is presented and a simplified version of it, that allows to reduce the computational time, is introduced.

An analysis of technical literature on sensorless control techniques follows; these methods are described in Chapter 3 of this text, where several sensorless position error estimation methods are presented. These techniques are discerned into low-speed and high-speed techniques. The former are saliency-based methods, which employ high frequency voltage injection and stator current demodulation or stator current-model flux demodulation to retrieve the position, while the latter are fundamental model-based algorithms that rely on stator and rotor equations and on back-EMF. The two systems are fused together to form a full-speed sensorless control scheme.

In Chapter 4, the implementation procedure to constitute the *syreDrive* software

tool is shown. The full-speed sensorless control scheme presented, with several selectable position error estimation techniques, is implemented in *syreDrive*. The operating principle of the automatic generation of the files for the control and the simulation model is shown. Here, both the Simulink model for the simulation and the control structure implemented in C-files are thoroughly described. The calibration of the control parameters, that allows the control to fit to any SyR motor, is discussed and assessed. The results of the three motors simulated are presented and the performance of the different control methods are compared. An analysis of the saliency of the motors under test is performed and some considerations on simulating motors of different saliency is carried out, paying attention to the operating points which are critical for the sensorless control techniques, no-load operation and torque overloading, in particular. The need to impose a minimum current arises from this study. Following the necessary analysis, it is determined the convenience of setting a minimum i_q , rather than i_d , to satisfy the requirements of the low-speed saliency-based position error estimation methods. However, the active flux becomes unobservable at no load due to the unexcited d -axis. Hence, the no load operation with active flux position observer at high speeds is not conducive with the imposition of minimum q -axis current. An example of optimization of the control parameters produced in output from *syreDrive* is provided, to show the simplicity of dealing with dedicated control calibration for a specific motor.

syreDrive allows to easily integrate sensorless control for a selected SyR motor by evaluating the performance of the machine under sensorless mode from the results of the automatically produced simulation. This software tool is meant to be used as first-approach control strategy evaluation for a freshly designed SyR motor, for a machine for which the control has not been realized yet or for a motor for which a different control is desired, especially for considering position sensorless control implementation. *syreDrive* may be used as an instrument of assessment and an overall framework to lay the groundwork for a more in-depth analysis and tuning of the control for any more specific applications. The control files generated are also suitable to be implemented in a digital control and utilized for practical tests. The final version of *syreDrive* is presented as it appears to the user and an example of results is provided.

The automatic generation of the control proved to deliver reliable results for all the three different-sized motors simulated, evidence of the fact that the calibration of the control is suitable for most SyR machines, which was the goal of this work. Future work may consist in the consolidation of this software feature by examining more SyR motors and by experimentally testing the results obtained at simulation stage or expand the number of control strategies selectable in the software.

Bibliography

- [1] <https://sourceforge.net/projects/syr-e/>.
- [2] <http://www.femm.info>.
- [3] G. Pellegrino, R. Bojoi, P. Guglielmi, "*Unified Direct-Flux Vector Control for AC Motor Drives*," IEEE Transaction on Industry Applications, vol. 47, no. 5, September-October 2011.
- [4] S. A. Odhano, P. Pescetto, H. A. A. Awan, M. Hinkkanen, G. Pellegrino and R. Bojoi, "*Parameter Identification and Self-Commissioning in AC Motor Drives: A Technology Status Review*," in IEEE Transactions on Power Electronics, vol. 34, no. 4, pp. 3603-3614, April 2019.
- [5] E. Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino and M. Pastorelli, "*Experimental Identification of the Magnetic Model of Synchronous Machines*," in IEEE Transactions on Industry Applications, vol. 49, no. 5, pp. 2116-2125, September-October 2013.
- [6] M. Hinkkanen, P. Pescetto, E. Mölsä, S. E. Saarakkala, G. Pellegrino and R. Bojoi, "*Sensorless Self-Commissioning of Synchronous Reluctance Motors at Standstill Without Rotor Locking*," in IEEE Transactions on Industry Applications, vol. 53, no. 3, pp. 2120-2129, May-June 2017.
- [7] G. Pellegrino, E. Armando, P. Guglielmi, "*Direct-Flux Vector Control of IPM Motor Drives in the Maximum Torque Per Voltage Speed Range*", in IEEE

Transactions on Industrial Electronics, vol. 59, no. 10, October 2012.

- [8] A. Yousefi-Talouki, G. Pellegrino, "Sensorless direct flux vector control of synchronous reluctance motor drives in a wide speed range including standstill", in XXII International Conference on Electrical Machines (ICEM), vol. , no. , September 2016.

- [9] A. Yousefi-Talouki, P. Pescetto, G. Pellegrino, "Sensorless Direct Flux Vector Control of Synchronous Reluctance Motors Including Standstill, MTPA and Flux Weakening", IEEE Transactions on Industry Applications, vol. 53, no. 4, 2017.

- [10] G. Pellegrino, E. Armando, P. Guglielmi, "Field oriented control of IPM drives for optimal constant power operation", European Conference on Power Electronics and Applications, 2-5 September 2007

- [11] S. Kang, J. Kim, S. Sul, "Position Sensorless Control of Synchronous Reluctance Motor Using High Frequency Current Injection", IEEE Transactions on Energy Conversion, vol. 14, no. 4, December 1999.

- [12] M. J. Corley, R. D. Lorenz, "Rotor Position and Velocity Estimation for a Salient-Pole Permanent Magnet Synchronous Machine at Standstill and High Speeds", IEEE Transactions on Industry Applications, vol. 34, no. 4, July-August 1998.

- [13] J. Jang, J. Ha, M. Ohto, K. Ide, S. Sul, "Analysis of Permanent-Magnet Machine for Sensorless Control Based on High-Frequency Signal Injection," IEEE Transactions on Industry Applications, vol. 40, no. 6, November-December 2004.

- [14] A. Yousefi-Talouki, P. Pescetto, G. Pellegrino "Combined Active Flux and High-Frequency Injection Methods for Sensorless Direct-Flux Vector Control of Synchronous Reluctance Machines," IEEE Transactions on Power Electronics, vol. 33, no. 3, March 2018.

- [15] A. Piippo, M. Hainnken, J. Luomi, "Analysis of an Adaptive Observer for Sensorless Control of Interior Permanent Magnet Synchronous Motors", IEEE

- Transactions on Industrial Electronics, vol.55, no. 2, February 2008.
- [16] T. Tuovinen, M. Hinnkanen "Adaptive Full-Order Observer With High-Frequency Signal Injection for Synchronous Reluctance Motor Drives", IEEE Journal of Emerging and Selected Topics in Power Electronics, vol. 2, no. 2, June 2014.
- [17] C. Hwang, Y. Lee, S. Sul, "Analysis on Position Estimation Error in Position-Sensorless Operation of IPMSM Using Pulsating Square Wave Signal Injection," IEEE Transactions on Industry Applications, vol. 55, no. 1, January-February 2019.
- [18] Y. Yoon, S. Sul, S. Morimoto, K. Ide "High-Bandwidth Sensorless Algorithm for AC Machines Based on Square-Wave-Type Voltage Injection," IEEE Transactions on Industry Applications, vol. 47, no. 3, May-June 2011.
- [19] I. Boldea, M. C. Paicu, and G. Andreescu, "Active flux concept for motionsensorless unified AC drives," IEEE Transaction on Power Electronics, vol. 23, no. 5, pp. 2612–2618, September 2008.
- [20] A. Varatharajan, G. Pellegrino, E. Armando, "Sensorless Synchronous Reluctance Motor Drives: Auxiliary Flux based Position Observer," IEEE Journal of Emerging and Selected Topics in Power Electronics, 2020.
- [21] M. Hinkkanen, S. E. Saarakkala, H. A. A. Awan, E. Mölsä, and T. Tuovinen, "Observers for Sensorless Synchronous Motor Drives: Framework for Design and Analysis," IEEE Transactions on Industry Applications, vol. 54, no. 6, pp. 6090–6100, 2018.
- [22] A. Varatharajan, G. Pellegrino, "Sensorless Synchronous Reluctance Motor Drives: A General Adaptive Projection Vector Approach for Position Estimation," IEEE Transactions on Industry Applications, 2020.
- [23] A. Varatharajan, G. Pellegrino, "Sensorless Synchronous Reluctance Motor Drives: A Full-Speed Scheme using Finite-Control-Set MPC in a Projection Vector Framework," IEEE Transactions on Industry Applications, 2020.
- [24] J. Lee, Y. Kwon, and S. Sul, "Experimental identification of IPMSM flux-linkage considering spatial harmonics for highaccuracy simulation of IPMSM drives", Proc. IEEE Energy Conversion Congress Expo., pp. 5804-5809, September 2018.

- [25] H. Cho, Y. Kwon, and S. Sul, "*Optimal Current Trajectory Control of IPMSM for Minimized Torque Ripple*", IEEE Transportation Electrification Conference and Expo (ITEC), Detroit, MI, USA, 2019.