POLITECNICO DI TORINO

Bachelor of Science in Electrical Engineering

Master Thesis Dissertation

Flux Polar Control of ac Motor Drives

Supervisors
Prof. Radu Bojoi
Dr. Sandro Rubino

Candidate
Luisa Tolosano

September 2021

Summary

Transport electrification is leading to an impressive development of electric drives using permanent magnet synchronous motors (PMSMs) and induction motors (IMs). According to this trend, impressive research efforts to develop torque control schemes characterized by high performance, easy to tune, and able to deal with saturated machines have been made in recent years. Moreover, these control solutions must guarantee a linear torque regulation for the entire speed range, including the deep flux-weakening (FW) operation with maximum torque per volt (MTPV). Currently, the control of the torque is mostly based on vector control schemes using inner current control loops whose references are in many cases provided by multi-dimensional calibrated maps to linear the torque regulation. However, the performance of the current loops depends on the motor inductance, i.e., the machine operating point, thus requiring demanding tuning procedures.

Therefore, the goal of the thesis is to develop an unified torque controller for ac motors using inner flux- and load angle- control loops since they are both independent of the machine inductance. Hence, their performance is only limited by the sampling frequency of the digital controller, thus guaranteeing the same torque regulation performance in deep FW with MTPV. Moreover, the torque linearity is guaranteed from zero up to the maximum speed reaching FW with MTPV using a single calibrated load-angle map that allows the maximum torque production under inverter current and voltage constraints.

This control technique can be applied either on synchronous motors (PMSMs) and induction motors (IMs) guaranteeing optimal performance in all the speed range. However, the main obstacles to use of this strategy are the creation and usage of the load angle map in the control code. In fact, the load angle map is generated accordingly to the motor flux maps and the control constrains introduced by the optimal exploitation of the machine potential, i.e., the MTPA and MTPV locus besides the inverter current limit. The procedure that must be adopted to generate this map is described in detail. Given the machine flux maps, the MTPA and MTPV locus are derived and used with the

inverter current limit to obtain a preliminary load-angle map. After a process of normalization that involves the reference values of torque and flux, the serviceable version of the load-angle map is attained, ready to be used in the control algorithm.

The load angle map can be used in the control code only when the pu values of torque and flux requested are known. Given the value of torque, the amplitude of flux can be straightforward obtained using the MTPA law. This value can be eventually redefined with a model-based flux-weakening law, depending on the type of machine controlled. The flux-weakening law takes in consideration both the speed and the inverter voltage limit to define the maximum exploitable flux in the machine in the specific conditions. Once the value of flux is known, the maximum torque reachable can be properly obtained using a 1D LUT that stores the information related with the torque limit of the machine. Both flux and torque values must be normalized over their nominal values to be used in the interpolation process involving the 2D LUT load angle map to get the exact value of load angle that must be imposed into the machine to satisfy the torque requests. Finally, the load angle and the amplitude of the flux must be used as reference values in two control loops implemented to define the voltage set point in (dq) coordinates.

Despite the generation of the load angle control map requests some care, the proposed method can be considered in a way easier than the solution currently used, as the DFVC, DTC, CVC ones, because of the simplicity of the strategy adopted. All the machine characteristics, including the non-linear behavior due to the saturation of the magnetic material, are reflected in the control map generated. Non-additional care must be used in the tuning process of the control loops as it is requested in the mentioned strategies that are highly affected by the non-linear behavior of the machine.

The simulation results and the experimental ones confirm the robustness of this control technique that gives excellent results both on the control of non-linear synchronous motors and induction ones.

Contents

List of Tables

List of Figures

List of Symbols

AM	Asynchronous Machine
SM	Synchronous Machine
IM	Induction Motor machine
IPM	Internal Permanent Magnet machine
SPM	Surface-mounted Permanent Magnet machine
SYR	Synchronous Reluctance machine
PMSM	Permanent Magnet Synchronous Motors
FPC	Flux Polar Control
FOC	Field Oriented Control
DFVC	Direct Flux Vector Control
DTC	Direct Torque Control
MTPA	Maximum Torque Per Ampere
MTPV	Maximum Torque Per Volt

Chapter 1

Introduction

Electric machine torque controller has become of primary importance in a world in which the spread of electric vehicles is continuously growing. In fact, this type of application requires electrical drives able to exploit the electric machine capability in the most effective way to get the best performance. The potentiality to have a torque control system able to deal with all the type of electric machine can become a reference solution, applicable also in industrial high-performance process.

The main purpose of all the different control strategies that can be adopted is the capability of control of machine in the whole different working conditions that belong to the machine operating range. Knowing what the maximum achievable speed is and inverter- and machine -limits it is possible to understand where are placed the control-boundary in terms of usable operating points. A good control scheme should be able to exploit the machine performance in the complete area described, without wasting any working point that can be useful for the final application. In fact, having a machine that can be used up to a certain maximum speed but limited from the control to a lower one, represents a poor use of it.

Many control strategies have been introduced until today. These approaches are different one from each other, but the conventional one are united by the fact that they are all based on the imposition of at least one inner current loop reference according to a torque set point given by the user or imposed by an external speed loop. The main problem that must be faced with these strategies is the behavior of the control when the machine presents non-linear magnetic characteristics and so the control is highly affected by the saturated performance of the machine. Here is showed a list of control strategies that are currently adopted with their advantages and disadvantages from the implementation and performance point of view.

1.1 Field Oriented Control (FOC)

FOC based control strategy is named after the orientation of (dq) rotor reference according to the magnetic flux position. FOC control is based on the definition of i_{dq} reference currents following the constrains of the control, i.e., MTPA and MTPV locus besides inverter's maximum voltage, that ensure a control strategy that provide, given the current module imposed, the maximum torque exploitable from the machine in the specific operating point and gives stability in all the operation conditions. Torque map of the machine is created starting from flux maps and current values, using the well known torque law in (dq) rotor reference showed in (1.1). The definition of the MTPA and MTPV trajectory on the current (or flux) plane is then obtained from the torque map of the machine compared with the iso-current (iso-flux) curves.

$$T = \frac{3}{2} p(\lambda_d i_q - \lambda_q i_d) \tag{1.1}$$

In (1.1) both the component of current are needed to manage the machine's torque production, so two inner current control loops are essential to define reference voltage values in (dq) coordinates. Different techniques have been proposed to define current reference values in order to be able to control the machine respecting all the constraints that guarantee stability and controllability.

The value of i^*_{dq} can be retrieved using different LUT-approaches. The simpler requires MTPA locus points that define $(i_d, i_q)_{MTPA}$ coordinates besides the base speed of the machine. When the speed overcomes this value have to be changed accordingly to the adopted flux-weakening law reaching at the end the deep field weakening region, i.e., MTPV locus, where the the torque production is maximized given the maximum voltage available. All the mentioned constrains are applied to the control reading different LUTs, following the instantaneous machine operating point, to guarantee stability and, at the end, gives the value of (i^*_d, i^*_q) currents that must be forced into the machine. Being the strategy based on the control of currents, whichever is the way used to define their values, two inner regulator depending on the operating point, i.e., on the inductance values, are needed and their tuning procedure is always demanding.

The approach presented in [1], whose control scheme is showed in Figure 1.1,requires a 4D LUT to define reference currents. In this case the (i^*_d, i^*_q) set points depends on four variables of the control: reference torque T^*, temperature Θ, dc voltage V_{dc} and mechanical speed ω_m. No additional management of the values obtained from the interpolation process is needed: all the operating conditions are already modeled and the current set point can be directly applied.

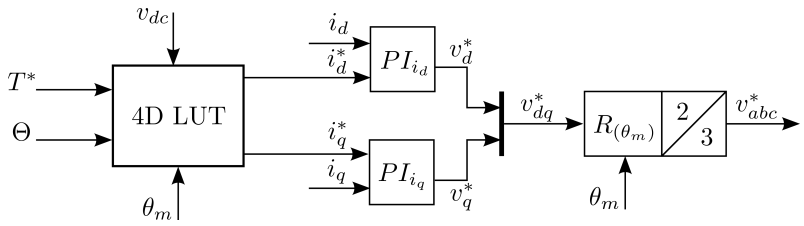


Figure 1.1: Field Oriented Control using 4D LUT scheme

Another applicable approach presented in [2], schematized in Figure 1.2 requires several LUTs to define (i_d^*, i_q^*) set points. Starting from reference torque T^* its value can be reduced using a LUT that stores the information of the MTPS profile and is read using the actual value of flux in the machine. When flux is nominal, at a speed lower than the base one, the torque value is limited at its maximum reachable value in standard condition. As the flow value decreases, i.e., the speed increases, the torque limit applied goes down constantly until minimum flux value, or the mechanical maximum speed, is reached.
Flux reference is obtained starting form the voltage available (the residual from the inverter limit reduced with the actual v_{dq}^*) divided by the speed of the machine. A regulator is used to define the actual flux needed that change following the speed, i.e. the operating point. Torque reference value, together with flux one are then used as inputs in two 2D LUTs that provide (i_d^*, i_q^*) set points forced into the machine control loops.

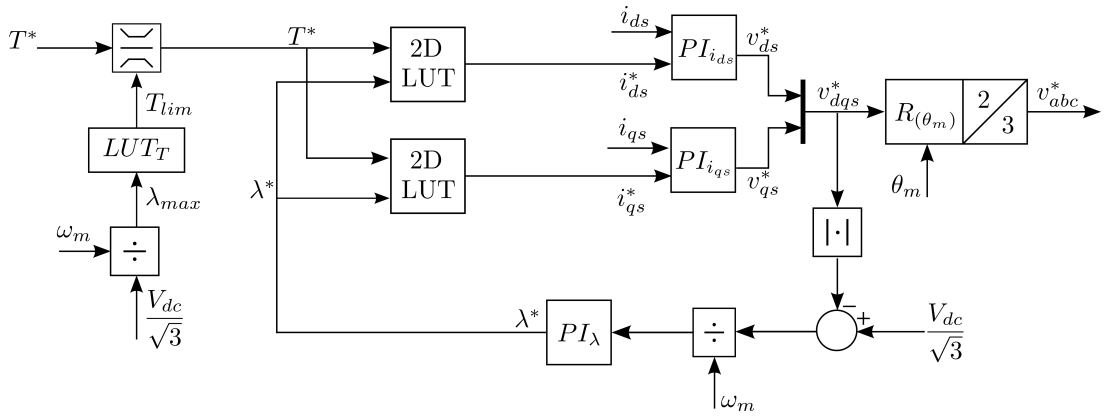


Figure 1.2: Field Oriented Control scheme with external flux weakening law

The main advantages of these type of controls are:

- Direct control of the current components via LUTs information.
- Rotation angle is measured with a position transducer: no extra flux observer

structure is needed, as in the control strategy that manage directly the flux vector, i.e., DFVC.

However, this control strategy presents some critical disadvantages:

- Torque regulation's linearity cannot be obtained due to saturation phenomena that affects the current-to-torque relation:
  - Below base speed the current references are obtained from the MTPA LUTs
  - Over base speed also the flux-weakening operation must be considered to generate the reference values of current for the regulation's linearity. Also, the MTPV operation must be implemented if needed to avoid machine pull-out at high speed.
- Torque linearity in flux-weakening region can be obtained only with one of the following:
  - An outer voltage loop that directly imposes the current reference to avoid voltage goes over the limit. It requires another regulator that must be tuned and affect the dynamic performance of the torque regulation.
  - An interpolation procedure involving at least two 3D LUT that require as input the speed, the dc-link voltage, and the reference torque to produce the $\{i_d^* i_q^*\}$ references for the current loops.
- The dynamic performance of the two PI current regulators depends on the working point of the machine:
  - The voltage limits of the PI regulator must be adaptive following the value of back-emf that they must compensate that depends on the machine's speed.
  - The bandwidth of each control axis follows the value of the differential inductance of the machine. To avoid instability phenomena the regulator gain must be adaptive with the machine operating point.

1.2 Direct Flux Vector Control (DFVC)

The DFVC is a control strategy based on the direct control of the flux amplitude and the torque-producing current component, namely i_{qs} stator component as showed in the control scheme in Figure 1.3 extracted from [3]. This type of control emulates the one of a dc motor in which the flux amplitude is regulated via the i_{ds} component, and the i_{qs} current impose separately the torque.

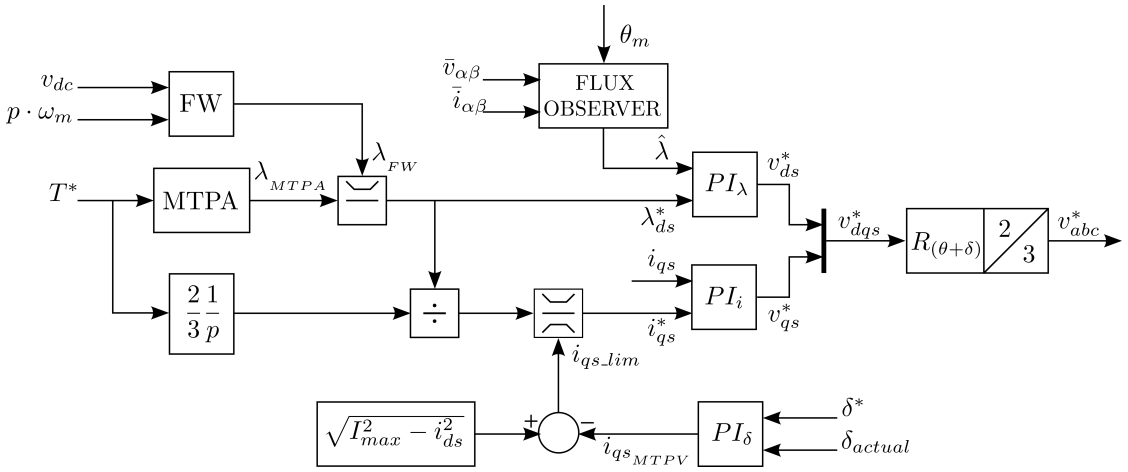


Figure 1.3: Direct Flux Vector Control scheme

Being the stator flux a non-measurable quantity, to impose the amplitude of flux using a PI regulator the actual value of flux must be known. It is obtained from a flux observer that uses both the current and the voltage model of the machine to retrieve the actual value of flux. The flux observer is implemented in ($\alpha\beta$) frame. From the flux vector component the angle of stator flux vector respect to $\alpha = a$ can be evaluated and used as rotation angle in the ($\alpha\beta$) $\leftrightarrow$ (dq) coordinates rotation. The reference value of flux needed comes from a 1D LUT that couple the torque value with the flux amplitude along the MTPA locus. Given the desired value of torque the interpolation produces the value of flux that is necessary. Further, this value can be reduced following the adopted flux-weakening law.

The reference value of i_{qs} current is easily obtained using the inverse torque relation (1.2) that exploit the orientation of the stator reference along the direction of flux, i.e., the d-axis.

$$i_{qs} = \frac{T}{\frac{3}{2} p \lambda_{ds}} \tag{1.2}$$

This value can be again adjusted following the inverter current limit and the MTPV

limitation that must be applied to ensure controllability even at high speed. First, the value of i_{qs} should not overcomes the maximum current amplitude reduced by thei_{ds} component to avoid a value of current over the nominal one of the machine. In accordance with the MTPV limitation i_{qs} can be further reduced after the output of a PI regulator that manage the load angle in MTPV condition. The relation between load angle and observed flux amplitude is stored into a 1D LUT that gives the value of maximum admissible load angle related to the actual value of the flux. The difference between the reference- and actual -load angle is the error provided to the PI regulator that impose at the output the admissible variation extremes for the i_{qs}. Finally, the two inner control loops are used: the flux regulator manage the v_d^* and the quadrature current one the v_q^*. From these two the v_{abc}^* reference is retrieved and used to obtain the duty values.
This type of control presents the following advantages:

- Below base speed the flux reference is obtained from a 1D LUT knowing what is the torque reference .
- Over base speed the flux reference is reduced following the flux-weakening law.
- The reference current value is calculated adopting the inverse torque relation without any LUT guaranteeing linearity in all the operating range.
- The output limitation of the two PI regulators is unbalanced: the d-axis impose the time derivative of stator flux amplitude that is very limited, while the q-axis dispose of almost all the voltage range to manage all the back-emf of the machine.
- The flux loop of the machine is insensitive to the operating point being independent from the machine inductance value.

The main drawbacks of this type of control can be summarized in the following:

- The value of the flux amplitude is known only adopting a flux observer in (dq_s) stator coordinates that also gives the value of the stator angle used in the frame rotation while in FOC the position from the transducer is sufficient (in this case the control works in (dq_s) rotor coordinates).
- The i_{qs} limitation to avoid loss of control in MTPV operation is retrieved from the output of a PI regulator that requires a demanding tuning procedure.
- The bandwidth of the PI regulator adopted for the management of i_{qs} depends on the value of the differential inductance. Therefore, the gain of the regulators should be adapted following the machine operating point to avoid instability phenomena.

1.3 Flux Polar Control (FPC)

The FPC torque controller [4] is based on the regulation of stator flux amplitude and the machine's load angle accordingly to the machine load angle map as represented in Figure 1.4.

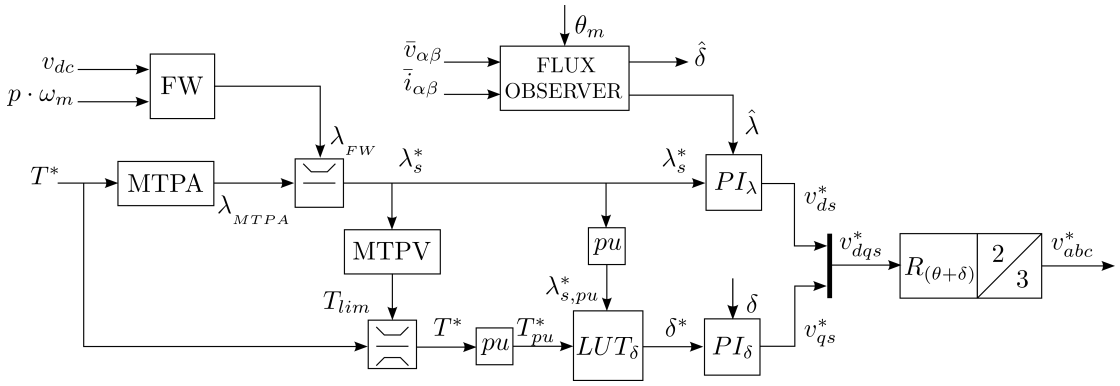


Figure 1.4: Polar Flux Control scheme

This controller uses the *d*-axis regulator to control the machine flux amplitude, as in the DFVC approach, while the *q*-axis component is used to regulate the load angle instead of the i_q current component. In fact, the approach is like the one adopted for the DFVC where flux reference is obtained from the torque reference using the MTPA flux values related to the torque. This flux value, valid only in MTPA condition, can be again changed following the flux-weakening law. Knowing the final value of flux, the torque, if higher, is limited to the maximum value achievable in the specific working condition. The *pu* value of flux and torque must be calculated to be used as input in the interpolation process involving the load angle maps that produces the reference value of load angle that must be forced into the machine to retrieve the desired torque.

Finally, the reference values are provided to two inner loops that generate the reference voltage in the (*dq*) frame. Again, in this case, like happened in the DFVC, the output of flux loop is limited only by the small voltage drop on the stator plus inverter resistance, that means the flux time derivative is limited. All the remaining voltage, the larger part of all the available voltage, is used to regulate the load angle of the machine. The main advantages that this control strategy presents are the following:

- As in the DFVC the d-axis regulator that manage the flux amplitude is insensible to the machine's operating point and so the tuning of the regulator is done easily.
- In this case also the q-axis regulator acting on the load angle is independent from the operating point, and, for this reason, the PI regulator gains can be the same as

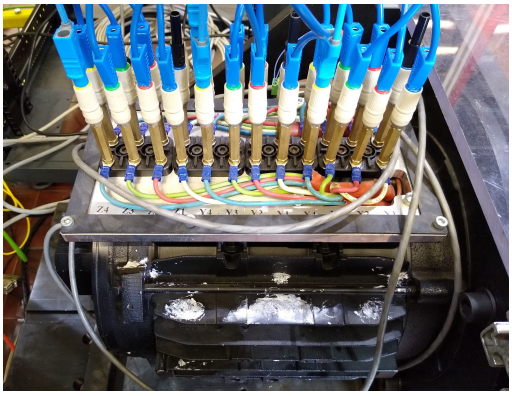
the one used for the other axis normalized with the flux amplitude. No additional tuning is required.

- The MTPV operation regulation is obtained directly from the load angle map that already consider this working condition and limit the machine load angle reference automatically. No more than the two inner PI regulators having the same dynamic performance are needed to run this control.

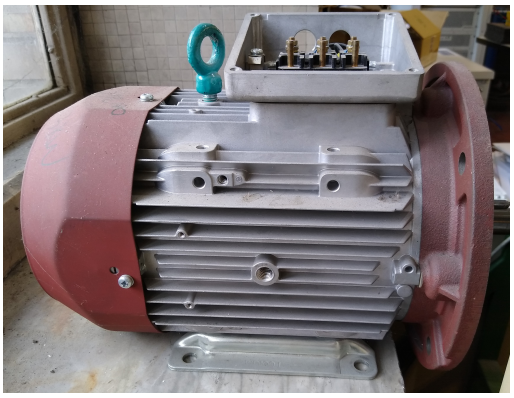
The disadvantages of this torque controller are:

- As for the DFVC also in this case a flux observer structure is needed to monitor the amplitude of actual flux in the machine.
- With this strategy the torque's regulation linearity is losses because in this case the value of the reference load angle cannot be retrieved with an analytical relation as can be done for the reference value of current in the DFCV approach, but only using a LUT.

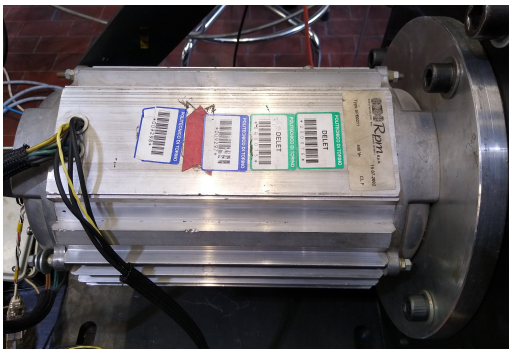
The main goal of this thesis is the validation of the FPC torque controller on both synchronous- and asynchronous -electric machine. The torque controller has been tested in simulation on four different type of real motor: an induction motor (IM) and three different types of synchronous motors (SyR, SPM and IPM) represented in Figure 1.5. Experimental test have been carry out in the Power Electronics Innovation Center (PEIC) of Politecnico di Torino on IM and IPM. The results obtained will be presented here and also used in articles and publications to prove the feasibility, dynamic performance and robustness of this torque controller.



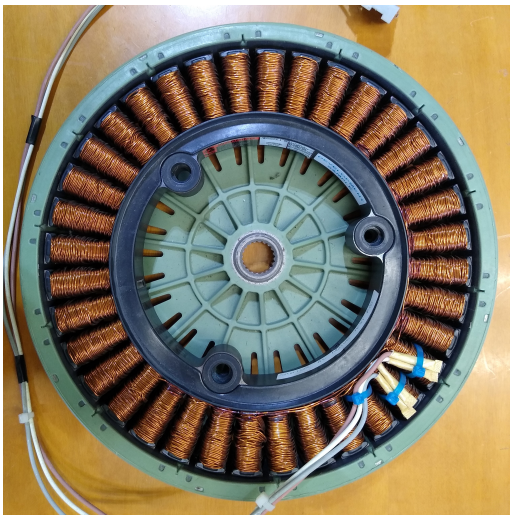
(a) *IM*



(b) *SYR*



(c) *IPM*



(d) *SPM*

Figure 1.5: Testing motors for FPC simulation/experimental validation

Chapter 2

ac Electric Machine Modeling

The proposed control strategy exploit the machine's equations in different frame, i.e. ($\alpha\beta$) and (dq_s), in order to use the properties that are available in the specific reference considered. The electric and magnetic equations of the IM machine and of the SMs are here showed and commented. For the SM machine electric and magnetic equations are the same for different type of machine considered. The adaptation for the specific type of motor must be done considering the characteristics of each of them.

2.1 Asynchronous machine

2.1.1 Dynamic model in ($\alpha\beta_s$) stator frame

The equations of the AM can be expressed in different frame. Starting from the original frame (abc) in which the flux linkage equations that characterize the induction motor are rather complex due the magnetic interaction between stator and rotor, using a coordinates transformation we can move them into ($\alpha\beta_s$) stator reference where the magnetic information is still stored but more accessible. Both the stator and rotor voltage and magnetic equations of a generic AM are showed.

Voltage Equation

Stator and rotor voltage equation in ($\alpha\beta_s$) frame are:

$$\bar{v}_{s,\alpha\beta} = R_s \bar{i}_{s,\alpha\beta} + \frac{d}{dt}\bar{\lambda}_{s,\alpha\beta} \tag{2.1}$$

$$\bar{v}_{r,\alpha\beta} = 0 = R_r \bar{i}_{r,\alpha\beta} + \frac{d}{dt}\bar{\lambda}_{r,\alpha\beta} - \boldsymbol{J} p \omega_m \bar{\lambda}_{r,\alpha\beta} \tag{2.2}$$

In the equations Rs and Rr are respectively the resistance of the stator and rotor winding, p is the number of pole pairs of the machine and ω_n is the mechanical rotational speed that appears in rotor voltage equation because of . The rotor's winding is short-circuited, and, for this reason, its voltage is null.

Flux Linkage Equation

$$\bar{\lambda}_{s,\alpha\beta} = L_s\bar{i}_{s,\alpha\beta} + L_m\bar{i}_{r,\alpha\beta} \tag{2.3}$$

$$\bar{\lambda}_{r,\alpha\beta} = L_m\bar{i}_{s,\alpha\beta} + L_r\bar{i}_{r,\alpha\beta} \tag{2.4}$$

In the equations L_s is the stator inductance that account the leakage inductance $L_{\sigma s}$ and the magnetizing one L_m, the same happens for L_r rotor inductance that is the sum of the leakage inductance $L_{\sigma r}$ and the magnetizing. Substituting $L_s = L_{\sigma s} + L_m$ and $L_r = L_{\sigma r} + L_m$ in (2.3) and (2.4) the equations become:

$$\bar{\lambda}_{s,\alpha\beta} = L_{\sigma s}\bar{i}_{s,\alpha\beta} + L_m(\bar{i}_{s,\alpha\beta} + \bar{i}_{r,\alpha\beta}) \tag{2.5}$$

$$\bar{\lambda}_{r,\alpha\beta} = L_{\sigma r}\bar{i}_{r,\alpha\beta} + L_m(\bar{i}_{s,\alpha\beta} + \bar{i}_{r,\alpha\beta}) \tag{2.6}$$

In this case the leakage inductance of both the stator and rotor are only excited by the current of the same winding and no magnetic coupling is present between the two, while magnetizing inductance is subject to both the stator and rotor currents thus showing the coupling. Moreover, the behavior of these inductance of the machine is different: while the leakage ones are quite constant when current increase, proving immunity from saturation phenomena, the value of the magnetizing inductance is highly influenced by the current level inside the machine and decrease when the current rises. This occurrence must be taken in consideration when simulating the machine performance: the current of rotor and stator should be known in order to forecast correctly which will be the inductance value of the machine in the specific working point.

An equivalent expression of stator flux can be obtained staring from (2.3). Substituting $\bar{i}_r$ with its expression retrieved from (2.4) stator flux equation become:

$$\bar{\lambda}_{s,\alpha\beta} = L_s\bar{i}_{s,\alpha\beta} + L_m\left(\frac{\bar{\lambda}_r - L_m\bar{i}_{s,\alpha\beta}}{L_r}\right) \tag{2.7}$$

Stator flux can be written as:

$$\bar{\lambda}_{s,\alpha\beta} = k_r\bar{\lambda}_{r,\alpha\beta} + \sigma L_s\bar{i}_{s,\alpha\beta} \tag{2.8}$$

where the term $k_r = \frac{L_m}{L_r}$ and $\sigma = 1 - \frac{L_m^2}{L_r L_s}$. This expression put in evidence how the stator flux is composed: both by the stator current and the rotor flux contribute to its value.

Furthermore, an equivalent magnetizing flux term $\bar{\lambda}_m$ can be expressed as function of the magnetizing current and inductance:

$$\bar{\lambda}_m = L_m \bar{i}_{m,\alpha\beta} = L_m(\bar{i}_{s,\alpha\beta} + \bar{i}_{r,\alpha\beta}) \tag{2.9}$$

In this way the flux linkage equations (2.5) and (2.6) become:

$$\bar{\lambda}_{s,\alpha\beta} = L_{\sigma s}\bar{i}_{s,\alpha\beta} + \bar{\lambda}_m \tag{2.10}$$

$$\bar{\lambda}_{r,\alpha\beta} = L_{\sigma r}\bar{i}_{r,\alpha\beta} + \bar{\lambda}_m \tag{2.11}$$

Stator and rotor currents are then calculated using (2.10) and (2.11): the leakage inductance, as a first approximation, can be considered immune from saturation.

$$\bar{i}_{s,\alpha\beta} = \frac{\bar{\lambda}_{s,\alpha\beta} - \bar{\lambda}_m}{L_{\sigma s}} \tag{2.12}$$

$$\bar{i}_{r,\alpha\beta} = \frac{\bar{\lambda}_{r,\alpha\beta} - \bar{\lambda}_m}{L_{\sigma r}} \tag{2.13}$$

Substituting (2.12) and (2.13) into the voltage equations (2.1) and (2.2) generate two state equation that represents the behavior of rotor and stator voltage of the machine with respect to the fluxes:

$$\bar{v}_{s,\alpha\beta} = \frac{R_s}{L_{\sigma s}}(\bar{\lambda}_{s,\alpha\beta} - \bar{\lambda}_m) + \frac{d}{dt}\bar{\lambda}_{s,\alpha\beta} \tag{2.14}$$

$$\bar{v}_{r,\alpha\beta} = 0 = \frac{R_r}{L_{\sigma r}}(\bar{\lambda}_{r,\alpha\beta} - \bar{\lambda}_m) + \frac{d}{dt}\bar{\lambda}_{r,\alpha\beta} - \boldsymbol{J} p\omega_m \bar{\lambda}_{r,\alpha\beta} \tag{2.15}$$

2.1.2 Dynamic model in (dq) frames

Coordinates transformation between (abc) and ($\alpha\beta_s$) has been done referring to stationary stator coordinates. Another transformation can be done from this frame to a rotating one: the choice that is made in FOC-based control strategy is to rotate both (2.1) and (2.2) in (dq) coordinates where the d-axis is oriented along the direction of the rotor flux linkage vector $\bar{\lambda}_r$ of the machine and q-axis is 90° in advance. To perform this rotation the angular position of rotor flux is retrieved using a flux observer structure.

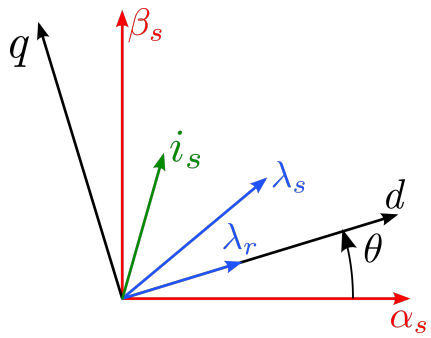


Figure 2.1: (dq) rotor flux linkage orientation (FOC IM)

Knowing the rotor flux vector components $\bar{\lambda}_{r,\alpha}$ and $\bar{\lambda}_{r,\beta}$ the amplitude is simply evaluated and the sine and cosine of θ are calculated as:

$$\sin(\theta) = \frac{\lambda_{r,\beta}}{|\bar{\lambda}_r|} \tag{2.16}$$

$$\cos(\theta) = \frac{\lambda_{r,\alpha}}{|\bar{\lambda}_r|} \tag{2.17}$$

Another rotation transformation that can be performed is the one from $(\alpha\beta_s)$ to (dq_s) frame in which the d_s-axis is oriented along the stator flux $\bar{\lambda}_s$ vector as in Figure 2.2. In this frame the voltage equations of the machine are expressed in a form that is the one on which the control loops of the FPC control are based on.

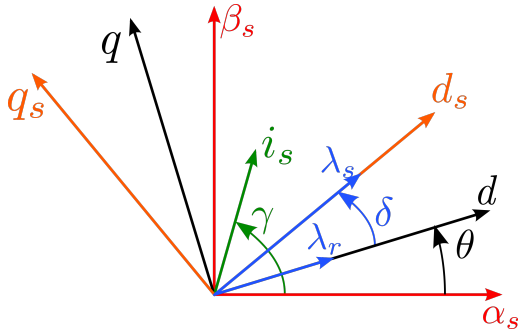


Figure 2.2: (dq_s) stator flux linkage orientation

Also in this case the rotation angle can be known only implementing a flux-observer structure in $(\alpha\beta_s)$ reference from which the stator flux is retrieved and so its angle $(\delta + \theta)$ respect to α_s is used as rotation angle from $(\alpha\beta_s)$ to (dq_s).

$$\sin(\delta + \theta) = \frac{\lambda_{s,\beta}}{|\bar{\lambda}_s|} \tag{2.18}$$

$$\cos(\delta + \theta) = \frac{\lambda_{s,\alpha}}{|\bar{\lambda}_s|} \tag{2.19}$$

Voltage Equation

Voltage equation of the machine in (dq) rotor flux-linkage coordinates are:

$$\bar{v}_{s,dq} = R_s \bar{i}_{s,dq} + \frac{d}{dt}\bar{\lambda}_{s,dq} + \omega \boldsymbol{J} \bar{\lambda}_{s,dq} \tag{2.20}$$

$$\bar{v}_{r,dq} = 0 = R_r \bar{i}_{r,dq} + \frac{d}{dt}\bar{\lambda}_{r,dq} + (\omega - p\omega_m)\boldsymbol{J}\bar{\lambda}_{r,dq} \tag{2.21}$$

The term $(\omega - p\omega_m)\boldsymbol{J}\bar{\lambda}_{r,dq}$ represents the electrical slip between stator and rotor coordinates that characterize the AM and can be also written as $\omega_{slip}\boldsymbol{J}\bar{\lambda}_{r,dq}$

For control purpose voltage equations can be also written in (dq_s) stator flux-linkage coordinates. In this frame they become:

$$\bar{v}_{s,dqs} = R_s \bar{i}_{s,dqs} + \frac{d}{dt}\bar{\lambda}_{s,dqs} + \boldsymbol{J}\frac{d}{dt}(\theta + \delta)\bar{\lambda}_{s,dqs} \tag{2.22}$$

$$\bar{v}_{r,dqs} = 0 = R_r \bar{i}_{r,dqs} + \frac{d}{dt}\bar{\lambda}_{r,dqs} + \boldsymbol{J}\frac{d}{dt}(\theta + \delta)\bar{\lambda}_{r,dqs} - \boldsymbol{J}p\omega_m\bar{\lambda}_{r,dqs} \tag{2.23}$$

In particular the (dq_s) component of stator voltage equation (2.22) can be expressed as:

$$\begin{cases} v_{ds} = R_s i_{ds} + \dfrac{d}{dt}|\bar{\lambda}_s| & (2.24) \\ v_{qs} = R_s i_{qs} + |\bar{\lambda}_s|\dfrac{d}{dt}\delta + \omega_r|\bar{\lambda}_s| & (2.25) \end{cases}$$

where ω_r is the electrical rotational speed of the rotor flux.

Flux Polar Control strategy is based on voltage equations in (2.24) and (2.25). The regulation of stator flux amplitude in done via d-axis stator voltage while the q-axis one manage the variation of amplitude of the flux vector load angle.

Flux Linkage Equation

In (dq) rotor flux linkage coordinates, flux equation can be written as:

$$\bar{\lambda}_{s,dq} = L_s \bar{i}_{s,dq} + L_m \bar{i}_{r,dq} \tag{2.26}$$

$$\bar{\lambda}_{r,dq} = L_m \bar{i}_{s,dq} + L_r \bar{i}_{r,dq} \tag{2.27}$$

Substituting $\bar{\lambda}_r = (|\bar{\lambda}_r| + j0)$ in (2.8) that is valid in all frames, stator flux linkage components are:

$$\begin{cases} \lambda_{s,d} = k_r \lambda_r + \sigma L_s i_d & (2.28) \\ \lambda_{s,q} = \sigma L_s i_q & (2.29) \end{cases}$$

To exploit the AM model in (dq) as is done for the SM one, rotor flux must be expressed as a simple function of an inductance and current. To gain this results $\bar{i}_{r,dq}$ expression from flux linkage rotor (2.27) is used in voltage equation (2.21) obtaining:

$$\begin{aligned} 0 &= \frac{R_r}{L_r}\bar{\lambda}_{r,dq} - R_r k_r \bar{i}_{s,dq} + \frac{d}{dt}\bar{\lambda}_{r,dq} + \omega_{slip}\boldsymbol{J}\bar{\lambda}_{r,dq} \\ &= \frac{1}{\tau_r}\bar{\lambda}_{r,dq} - R_r k_r \bar{i}_{s,dq} + \frac{d}{dt}\bar{\lambda}_{r,dq} + \omega_{slip}\boldsymbol{J}\bar{\lambda}_{r,dq} \end{aligned} \tag{2.30}$$

The d- and q -axis components of this equation can be separated:

$$\begin{cases} 0 = \dfrac{1}{\tau_r}\lambda_r + \dfrac{d}{dt}\lambda_r - R_r k_r i_{s,d} & (2.31) \\ 0 = -R_r k_r i_{s,q} + \omega_{slip}\lambda_r & (2.32) \end{cases}$$

Dynamic behavior of rotor flux is driven by the d-axis component of the stator current while the q-axis component affect the slip of the machine. Equation (2.31) can be still be elaborated:

$$\begin{aligned} 0 &= \lambda_r + \tau_r \frac{d}{dt}\lambda_r - \tau_r R_r k_r i_{s,d} \\ &= \lambda_r + \tau_r \frac{d}{dt}\lambda_r - \frac{L_r}{R_r} R_r \frac{L_m}{L_r} i_{s,d} \\ &= \lambda_r + \tau_r \frac{d}{dt}\lambda_r - L_m i_{s,d} \end{aligned} \tag{2.33}$$

Moving in Laplace domain: (2.33) become:

$$0 = \lambda_r + s\tau_r\lambda_r - L_m i_{s,d} \tag{2.34}$$

Finally the equation of the dynamic behavior of the rotor flux is showed:

$$\lambda_r = \frac{L_m i_{s,d}}{1 + s\tau_r} \tag{2.35}$$

A step variation of d-axis stator current cause a change in rotor flux with some delay given by the rotor time constant τ_r that slow down the variation, affecting also the torque performance that is dependent from the rotor flux value (see eq. (2.42)). For this reason, to guarantee good dynamic behavior, in AM is usually performed a preliminary phase of flux build in which the rotor flux is bring to its nominal value and, during the normal operation, only q-axis current is change to modify the torque value.
In steady-state conditions, using $\lambda_r = L_m i_d$ in (2.28) leads to an equivalent expression

of the flux linkage of the asynchronous motor that resembles the one of the synchronous one:

$$\begin{cases} \lambda_{s,d} = L_s i_d & (2.36) \\ \lambda_{s,q} = \sigma L_s i_q & (2.37) \end{cases}$$

2.1.3 Torque Equation

The torque equation can be retrieved from the balance of the power entering the machine:

$$P_{el} = P_{stator} + P_{rotor} = \frac{3}{2}\bar{v}_s\bar{i}_s + \frac{3}{2}\bar{v}_r\bar{i}_r \tag{2.38}$$

Substituting the voltage expression (2.1) and (2.2) we obtain:

$$P_{el} = \frac{3}{2}R_s|\bar{i}_s|^2 + \frac{3}{2}\left(\frac{d}{dt}\bar{\lambda}_s\right)\bar{i}_s + \frac{3}{2}R_r|\bar{i}_r|^2 + \frac{3}{2}\left(\frac{d}{dt}\bar{\lambda}_r\right)\bar{i}_r - \frac{3}{2}\boldsymbol{J}p\omega_m\bar{\lambda}_r\bar{i}_r \tag{2.39}$$

The term $\left[\frac{3}{2}R_s|\bar{i}_s|^2 + \frac{3}{2}R_r|\bar{i}_r|^2\right]$ represents stator and rotor Joule losses.
The term $\left[\frac{3}{2}\left(\frac{d}{dt}\bar{\lambda}_s\right)\bar{i}_s + \frac{3}{2}\left(\frac{d}{dt}\bar{\lambda}_r\right)\bar{i}_r\right]$ the magnetizing powers of rotor and stator that in steady-state conditions are null.
Finally, last term represents the mechanic power produced by the machine.

$$P_{mecc} = -\frac{3}{2}\boldsymbol{J}p\omega_m\bar{\lambda}_r\bar{i}_r \tag{2.40}$$

The expression of the electromagnetic torque produced by the machine is:

$$T = \frac{P_{mecc}}{\omega_m} = -\frac{3}{2}\boldsymbol{J}p\bar{\lambda}_r\bar{i}_r \tag{2.41}$$

Exploiting the properties of scalar and vector product, the equation (2.41) can be rewritten:

$$T = \frac{3}{2}p(\bar{i}_r \wedge \bar{\lambda}_r) \tag{2.42}$$

Different form of this equation can be retrieved substituting rotor current or rotor flux with their expression.

- Substituting i_r with its expression retrieved from (2.4) torque become:

$$T = \frac{3}{2}p\left(\frac{\bar{\lambda}_r - L_m\bar{i}_s}{L_r}\right) \wedge \bar{\lambda}_r = \frac{3}{2}pk_r\bar{\lambda}_r \wedge \bar{i}_s \tag{2.43}$$

where $k_r = \frac{L_m}{L_r} = \frac{L_m}{L_m + L_{\sigma r}}$ is close to the unit ($L_{\sigma r} << L_m$). This expression put in evidence the contribute of stator current and rotor flux to torque production.

- An equivalent expression of torque can be retrieved from (2.43) substituting rotor flux with its expression obtained inverting (2.8). Torque is:

$$\begin{aligned} T =& \frac{3}{2} p \left(\bar{\lambda}_s - \sigma L_s \bar{i}_s \right) \wedge \bar{i}_s \\ =& \frac{3}{2} p \bar{\lambda}_s \wedge \bar{i}_s \end{aligned} \tag{2.44}$$

- Torque expression that better fits our control purpose is (2.45), obtained substituting in (2.44) the expression of the stator current function of the rotor flux retrieved from (2.8)

$$T = \frac{3}{2} p \bar{\lambda}_s \wedge \left(\frac{\bar{\lambda}_{s,\alpha\beta} - k_r \bar{\lambda}_{r,\alpha\beta}}{\sigma L_s} \right) \tag{2.45}$$

$$T = \frac{3}{2} p \frac{k_r}{\sigma L_s} \left(\bar{\lambda}_r \wedge \bar{\lambda}_s \right) \tag{2.46}$$

that can be also written putting in evidence the non-linear dependence of torque from the load angle δ:

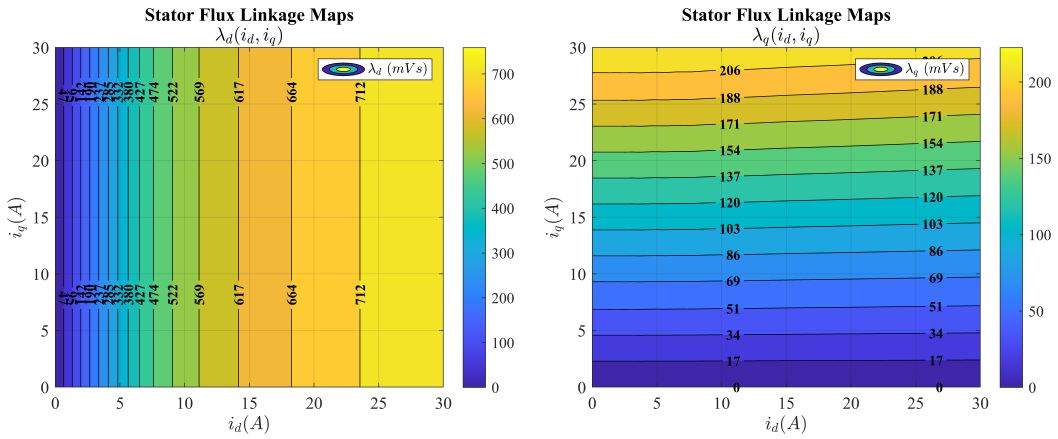
$$T = \frac{3}{2} p \frac{k_r}{\sigma L_s} |\bar{\lambda}_r| |\bar{\lambda}_s| \sin(\delta) \tag{2.47}$$

2.1.4 Flux Maps

Starting from experimental data obtained in locked rotor and no load procedure on the machine under test, the magnetic behavior of the machine is known. Given the value of currents along d-axis and q-axis of the motor and corresponding inductance both in d- and q -axis, direct flux maps of AM are built using (2.36) and (2.37) to obtain two matrices that stores the magnetic behavior of the machine. Final maps resemble the one adopted for the magnetic modeling of SM. The general relation are showed in (2.48) and (2.49).

$$\begin{cases} \lambda_d = f(i_d, i_q) & (2.48) \\ \lambda_q = f(i_d, i_q) & (2.49) \end{cases}$$

Direct flux maps of induction motor are showed.



(a) $\lambda_d(id, iq)$ (b) $\lambda_q(id, iq)$

Figure 2.3: Direct flux maps of Induction Motor

As can be seen in Figure 2.3a and 2.3b the behavior of fluxes along the two axis is different. Along the q-axis the machine seems quite linear, on the contrary, in d-axis direction the behavior is highly saturated. This is due to the inductance that these two axis see that are the one retrieved in section 2.1.2 and is highlighted in Figure 2.4.

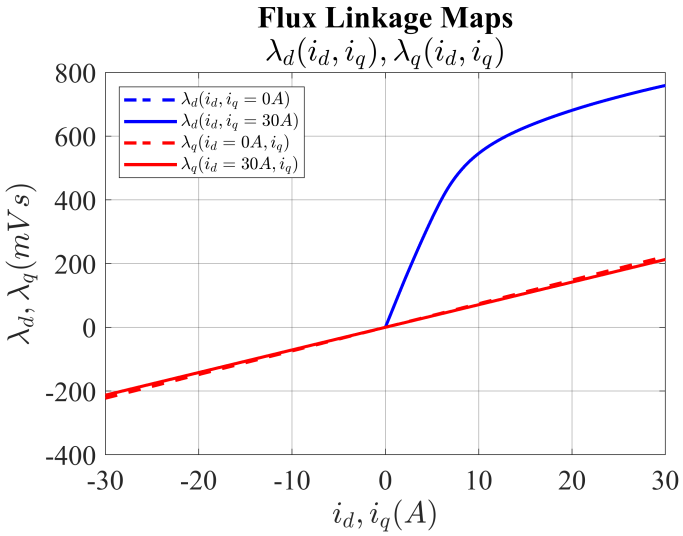


Figure 2.4: Flux linkage maps of Induction Motor

Starting from direct magnetic maps of the machine, indirect flux maps can be also retrieved. They will be used in the generation of control limit, i.e., MTPV locus. To obtain an indirect map from a direct one the inverse relation that links current to fluxes must be found.

To obtain this relations, an interpolation function on direct maps data is performed using an interpolator available in MATLAB environment, called `scatteredInterpolant`, able to deal with scattered data as ours. Thanks to this function, the relation that links current to flux can be retrieved from direct flux maps and applied to build the indirect version of these maps. Direct maps, both in term of fluxes and currents, are aligned in form of vectors and, with them, the interpolation rule is created. Subsequently, the regular grid of flux is build, knowing the extreme values, set equal to those representing the boundaries of the direct maps. The dimension of this grid np, is usually the same as the one of current adopted for direct maps.

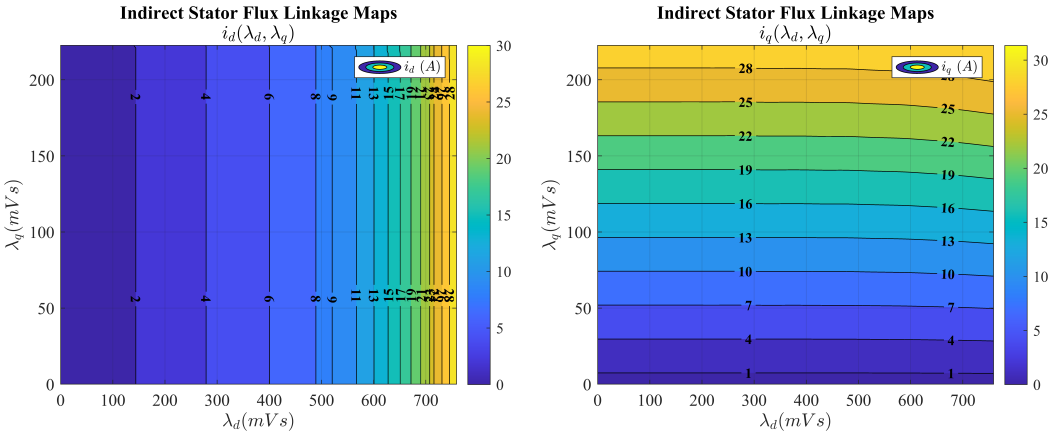
Finally, the current values correspondent to grid flux coordinates can be obtained. In this way, the inverse maps, described by (2.50) and (2.51), are created.

$$\begin{cases} i_d = f(\lambda_d, \lambda_q) & (2.50) \\ i_q = f(\lambda_d, \lambda_q) & (2.51) \end{cases}$$

An example of the code adopted is here showed: all the variables named "x_old" are the vector-form of the original direct maps.

```
IntD = scatteredInterpolant(Fd_old,Fq_old,Id_old,'natural','linear');
IntQ = scatteredInterpolant(Fd_old,Fq_old,Iq_old,'natural','linear');
fD = linspace(Fd_min, Fd_max, np);
fQ = linspace(Fq_min, Fq_max, np);
iD = IntD(fD,fQ);
iQ = IntQ(fD,fQ);
```

Indirect flux maps of induction motor are here represented.



(a) $i_d(\lambda_d, \lambda_q)$ (b) $i_q(\lambda_d, \lambda_q)$

Figure 2.5: Indirect flux maps of Induction Motor

2.1.5 Iron Losses model

A final review on the modeling of the iron losses in the machine is required. The experimental test that can be carry out on the machine are the short-circuit and the no-load tests. The first is used to obtain the values of the longitudinal parameters: stator and rotor resistance (R_s and R_r) and leakage inductance ($L_{\sigma s}$ and $L_{\sigma r}$). The second provides the value of the transversal parameters: the equivalent iron resistance and the magnetizing inductance (R_{fe} and L_m).

The presence of the parasitic elements means that not all the power provided in input to the machine is used in the electromagnetic conversion involving the production of torque. A small part is losses in these element. To take into account this phenomena, the circuit of the machine in Figure 2.6, with the main parameters, is build to identify the exact value of the voltage and current that are actually used for the active part of the machine. For the implementation in Simulink, an equivalent Thevenin model is created.

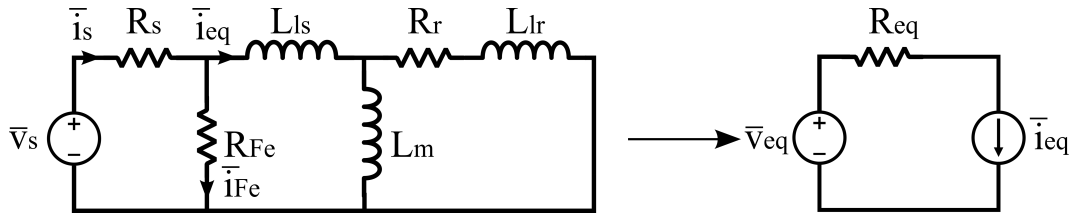


Figure 2.6: Equivalent circuit of the electric machine for iron losses evaluation

The model is fed with $\bar{v}_s$ but the useful current is only $\bar{i}_{eq}$, part of the total $\bar{i}_s$, while $\bar{i}_{Fe}$ is the current that represent the iron losses into the R_{Fe}. In the second circuit, where the Thevenin model is showed, $\bar{V}_{eq}$ and R_{eq} are the equivalent voltage and resistance. Their expression are in Equation (2.52) and (2.53).

$$\begin{cases} \bar{V}_{eq,dq} = \bar{V}_{s,dq} \cdot \dfrac{R_{Fe}}{R_{Fe} + R_s} & (2.52) \\ R_{eq} = \dfrac{R_{Fe} \cdot R_s}{R_{Fe} + R_s} & (2.53) \end{cases}$$

Knowing the expression of the total current feeding the machine, the equivalent current is obtained:

$$\bar{i}_s = \frac{\bar{v}_s}{R_s + R_{fe}} + \bar{i}_{eq} \cdot \frac{R_{fe}}{R_s + R_{fe}} \Rightarrow \bar{i}_{eq} = \left[\bar{i}_s - \frac{\bar{v}_s}{R_s + R_{fe}} \right] \cdot \frac{R_s + R_{fe}}{R_{fe}} \tag{2.54}$$

In the Simulink model, to take into account the iron losses, the stator voltage and magnetic equations are used. Starting from the standard equations of the machine, adopting the actual circuital variables, Equation (2.55) and (2.56) are obtained:

$$\begin{cases} \bar{v}_s = R_s(\bar{i}_{Fe} + \bar{i}_{eq}) + \dfrac{d}{dt}|\bar{\lambda}_s| & (2.55) \\ \bar{\lambda}_s = L_{\sigma s}\bar{i}_{eq} + L_m(\bar{i}_{eq} + \bar{i}_r) = L_{\sigma s}\bar{i}_{eq} + \bar{\lambda}_m & (2.56) \end{cases}$$

Finally, the electromagnetic model is obtained in Equation (2.57) in which the expression of equivalent current $\bar{i}_{eq}$, retrieved from (2.56), has been substituted in (2.55):

$$\bar{v}_s = R_s \cdot \left[\bar{i}_{Fe} + \frac{\bar{\lambda}_s - \bar{\lambda}_m}{L_{\sigma s}}\right] + \frac{d}{dt}|\bar{\lambda}_s| \tag{2.57}$$

In the Simulink model of the machine this expression is used to obtain the stator flux vector $\bar{\lambda}_s$, used in the magnetic model of the machine, beside the torque expression and to obtain, together with rotor flux, the amplitude of load angle δ.

2.2 Synchronous machine

As for the AM, also for the synchronous one voltage and flux linkage equations can be written in $(\alpha\beta_s)$ stator frame. From this stationary frame we can move the set of equations in (dq) rotor- and (dq_s) stator - flux linkage oriented frames using the appropriate rotational angles.

2.2.1 Dynamic model in (dq) frames

Voltage and flux linkage equation of SM can be written in two different rotational frame that are represented in Figure 2.7.

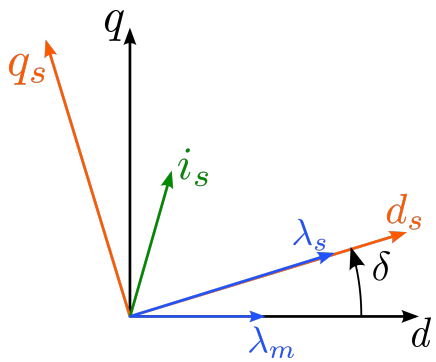


Figure 2.7: (dq) rotor and (dq_s) stator flux linkage orientations - PM style

Usually, the definition of the (dq_s) stator flux linkage oriented frame is unique, following the direction of stator flux vector, while different configuration of (dq) rotor

frame are possible after the adopted convention. Typically, there are two main type of (dq) rotor frame conventions that can be adopted. The d-axis can be:

- aligned with the magnet flux vector of the machine, following the so-called "PM-style" convention. Having the magnet a magnetic permeance μ_0 similar to air, with this choice the value of d-axis inductance is lower than the q-axis one.
- Aligned with the direction of maximum permeance of the machine, following the so-called "SyR-style" convention. In this case being the d-axis the one with minimum reluctance the d-axis inductance is higher than the q-axis one.

The choice of which type of convention should be adopted is made at will, by selecting the one that offers the greatest advantages according to the user's needs [5]. Typically, a wide spread criteria is to adopt one or the other following the nature of the torque produced by the machine: the percentage that prevails over the other wins and determine the convention. Anyway, both of them are valid, it is only a matter of how the equation are written. In practical nothing change choosing one or the other, but the adopted convention must be known in order to be able to control exactly the machine behavior.

For simplicity in terms of code necessary for the generation of control maps, the first choice done is to make no distinction between the different type of motors and to impose always the "PM-style" convention even to SyR motor without magnets: in this case the d-axis is placed along the direction of minimum of permeance that is the one where the flux barriers of the motor are placed [3]. In Figure 2.8 extracted from [3] the choice done is represented on different type of motors. In simulation the need of the conventional (dq) reference for the reluctance machine has been putted in evidence by the unstable behavior of the machine while running the control code. Thus the adopted frame for reluctance machine is the conventional ones.

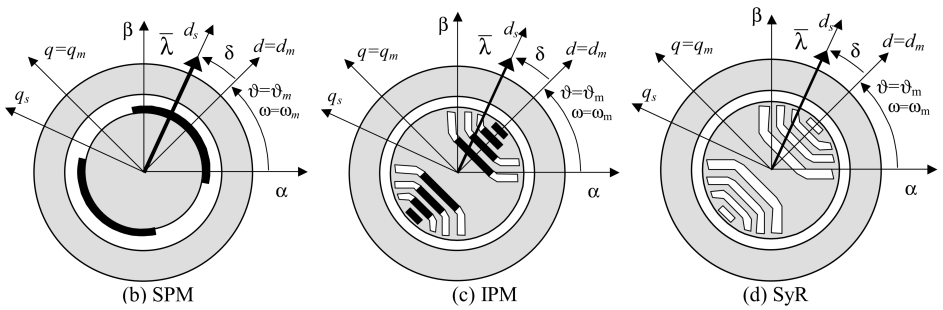


Figure 2.8: Definition of reference frames for ac SM machines

Voltage Equation

According to the literature [6] the electric (dq) model of the machine is computed as:

$$\bar{v}_{dq} = R_s \bar{i}_{dq} + \frac{d}{dt}\bar{\lambda}_{dq} + p\omega_m \boldsymbol{J}\bar{\lambda}_{dq} \tag{2.58}$$

Starting form the voltage equation (2.58) in (dq) rotor frame in which the d-axis is oriented on the magnets flux linkage, the two components of the equation can be retrieved:

$$\begin{cases} v_d = R_s i_d + \dfrac{d}{dt}\lambda_d - p\omega_m \lambda_q & (2.59) \\ v_q = R_s i_q + \dfrac{d}{dt}\lambda_q + p\omega_m \lambda_d & (2.60) \end{cases}$$

The reference can still be rotated aligned to stator flux-linkage frame.

In (dq_s) the ds-axis corresponds to the position of the stator flux linkage vector and the deviation angle between the ds-axis and the d-axis corresponds to the machine's load angle δ. To move the electromagnetic model from d-axis to ds-axis a rotational transformation is applied using the machine's load angle δ as the rotational angle.

$$\bar{v}_{dqs} = R_s \bar{i}_{dqs} + \frac{d}{dt}\bar{\lambda}_{dqs} + \boldsymbol{J}\frac{d}{dt}(\theta + \delta)\bar{\lambda}_{dqs} \tag{2.61}$$

The two components of the equation can be retrieved:

$$\begin{cases} v_{ds} = R_s i_{ds} + \dfrac{d}{dt}|\bar{\lambda}_s| & (2.62) \\ v_{qs} = R_s i_{qs} + |\bar{\lambda}_s|\dfrac{d}{dt}\delta + p\omega_m |\bar{\lambda}_s| & (2.63) \end{cases}$$

The d-axis component of voltage manage the amplitude of flux vector and the quadrature one operates on the value of load angle. Given the reference values of flux amplitude and load angle, two control loops can be generated to define $\{v^*_{ds}, v^*_{qs}\}$ voltages: this is the core on which Flux Polar Control strategy is based on.

Flux Linkage Equation

In (dq) rotor frame the flux linkage equations that represent the magnetic model, assuming the machine (dq) inductances are depending on (dq) currents, are:

$$\bar{\lambda}_{dq} = \boldsymbol{L}\bar{i}_{dq} + \bar{\lambda}_m \tag{2.64}$$

The d-axis and q-axis expression of the magnetic model can be expressed as:

$$\begin{cases} \lambda_d = L_{dd}(i_d, i_q) \cdot i_d + L_{dq}(i_d, i_q) \cdot i_q + |\bar{\lambda}_m| & (2.65) \\ \lambda_q = L_{qd}(i_d, i_q) \cdot i_d + L_{qq}(i_d, i_q) \cdot i_q & (2.66) \end{cases}$$

where L_{dd} and L_{qq} are the apparent self-inductances of the machine along the d-axis and q-axis, respectively, while L_{dq} and L_{qd} are the apparent cross-inductances of the machine along the d-axis and q-axis, respectively; these last used to model the cross saturation phenomena. Finally, λ_m represents the amplitude of the PM flux linkage vector, whose value is assumed constant and not dependent on the PM temperature for simplicity.

In (dq_s) frame the flux linkage equation become:

$$\bar{\lambda}_{dqs} = \boldsymbol{L}\bar{i}_{dqs} + \bar{\lambda}_m \tag{2.67}$$

The ds-axis and qs-axis expression of the magnetic model can be expressed again assuming the machine (dq_s) inductances are depending on (dq_s) currents, as:

$$\begin{cases} \lambda_{ds} = L_{dds}(i_{ds}, i_{qs}) \cdot i_{ds} + L_{dqs} i_{ds}, i_{qs}) \cdot i_{qs} + |\bar{\lambda}_m| \cos(\delta) & (2.68) \\ \lambda_{qs} = L_{qds}(i_{ds}, i_{qs}) \cdot i_{ds} + L_{qqs}(i_{ds}, i_{qs}) \cdot i_{qs} - |\bar{\lambda}_m| \sin(\delta) & (2.69) \end{cases}$$

2.2.2 Torque Equation

For SM, like is done for asynchronous one, the torque equation can be retrieved starting from power balance of the machine. Electric power in input is expressed as:

$$P_{el} = P_{stator} = \frac{3}{2}\bar{v}_s\bar{i}_s \tag{2.70}$$

Substituting the (dq) stator voltage expression (2.58) in (2.70) we obtain:

$$P_{el} = \frac{3}{2}R_s|\bar{i}_{dq}|^2 + \frac{3}{2}\left(\frac{d}{dt}\bar{\lambda}_{dq}\right)\bar{i}_{dq} + \frac{3}{2}\left(p\omega_m \boldsymbol{J}\bar{\lambda}_{dq}\right)\bar{i}_{dq} \tag{2.71}$$

The term $\left[\frac{3}{2}R_s|\bar{i}_{dq}|^2\right]$ represents stator Joule loss.

The term $\left[\frac{3}{2}\left(\frac{d}{dt}\bar{\lambda}_{dq}\right)\bar{i}_{dq}\right]$ is the magnetizing powers of stator that in steady-state conditions is null.

Finally, last term represents the mechanic power produced by the machine.

$$P_{mecc} = \frac{3}{2}\left(p\omega_m \boldsymbol{J}\bar{\lambda}_{dq}\right) \cdot \bar{i}_{dq} \tag{2.72}$$

The expression of electromagnetic torque produced by the machine is:

$$T = \frac{P_{mecc}}{\omega_m} = \frac{3}{2}\left(p\boldsymbol{J}\bar{\lambda}_{dq}\right) \cdot \bar{i}_{dq} \tag{2.73}$$

that is equivalent to:

$$T = \frac{3}{2} p(\bar{\lambda}_{dq} \wedge \bar{i}_{dq}) \tag{2.74}$$

- Using (dq) rotor reference frame leads to (2.75) in which the machine torque is proportional to the outer product between the stator flux linkage vector and stator current vector. Therefore, it depends only on the shift angle between the two vectors regardless of the considered reference frame:

$$T = \frac{3}{2} \cdot p \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \tag{2.75}$$

  Using flux linkage expression (2.65) and (2.66) in (2.75) the different contribution to torque production are in evidence (for simplicity the cross term in flux equation are neglected):

$$\begin{aligned} T &= \frac{3}{2} \cdot p \cdot [(L_{dd} \cdot i_d + |\bar{\lambda}_m|) \cdot i_q - (L_{qq} \cdot i_q) \cdot i_d] \\ &= \frac{3}{2} \cdot p \cdot [(L_{dd} - L_{qq}) \cdot i_d \cdot i_q + |\bar{\lambda}_m| \cdot i_q] \end{aligned} \tag{2.76}$$

  The first term is the reluctance torque of the machine, proportional to both the currents components, while the latter represents the torque produced by the magnetic flux of the machine and its value depends only on the quadrature current i_q. By following the nature of the machine, both or only one of this two terms contribute to the final torque value produced in output.

- Using (dq_s) stator flux reference the expression is:

$$\begin{aligned} T &= \frac{3}{2} p(\bar{\lambda}_{dqs} \wedge \bar{i}_{dqs}) \\ &= \frac{3}{2} p|\bar{\lambda}_{dqs}| i_{qs} \end{aligned} \tag{2.77}$$

- Substituting stator current expression retrieved from (2.67) in torque equation (2.77) leads to:

$$T = \frac{3}{2} \cdot p \cdot \bar{\lambda}_{dqs} \wedge \left(\frac{\bar{\lambda}_{dqs} - \bar{\lambda}_m}{\boldsymbol{L}} \right) \tag{2.78}$$

With the hypothesis of a isotropic machine the torque equation become:

$$T = \frac{3}{2} \cdot p \frac{1}{Ls} \cdot \left(\bar{\lambda}_m \wedge \bar{\lambda}_{dqs}\right) \tag{2.79}$$

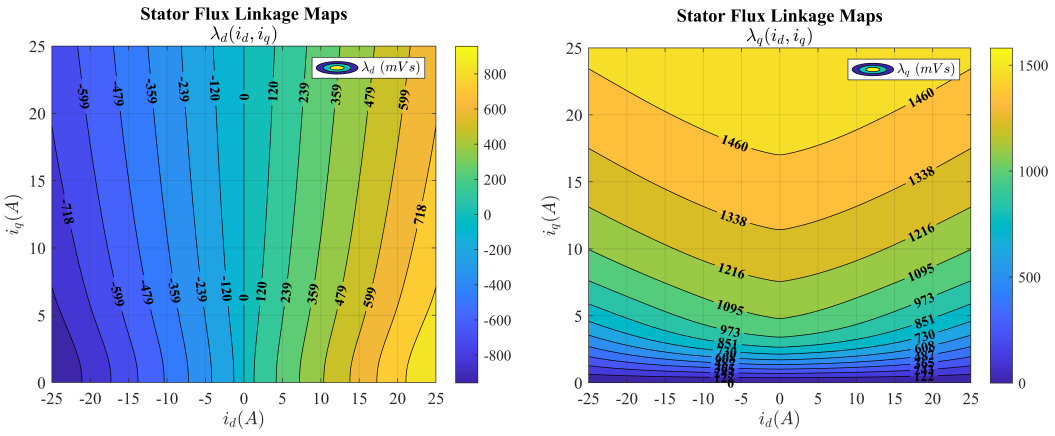
that can also be written in the form exploited in Flux Polar Control:

$$T = \frac{3}{2} p \frac{1}{Ls} |\bar{\lambda}_m| |\bar{\lambda}_{dqs}| sin(\delta) \tag{2.80}$$

It can be noticed the linear dependence of the torque value from the magnetic flux and the non-linear one from the load angle δ.

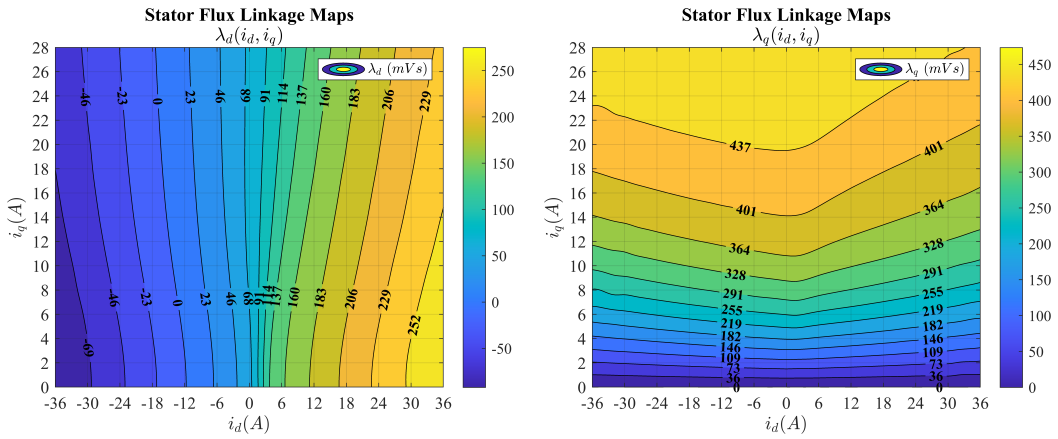
2.2.3 Flux Maps

Following the experimental identification procedure of the magnetic model of SM in [7] and [8] the flux maps of the three SMs adopted for the validation of the proposed control strategy are generated. They can be generally described by flux linkage equations (2.65) and (2.66) that take into account both the contribute of magnets flux and current to the total flux of the machine. In Figure 2.9,2.10 and 2.11 are showed respectively the direct maps of SyR, IPM and SPM machines used in simulation.



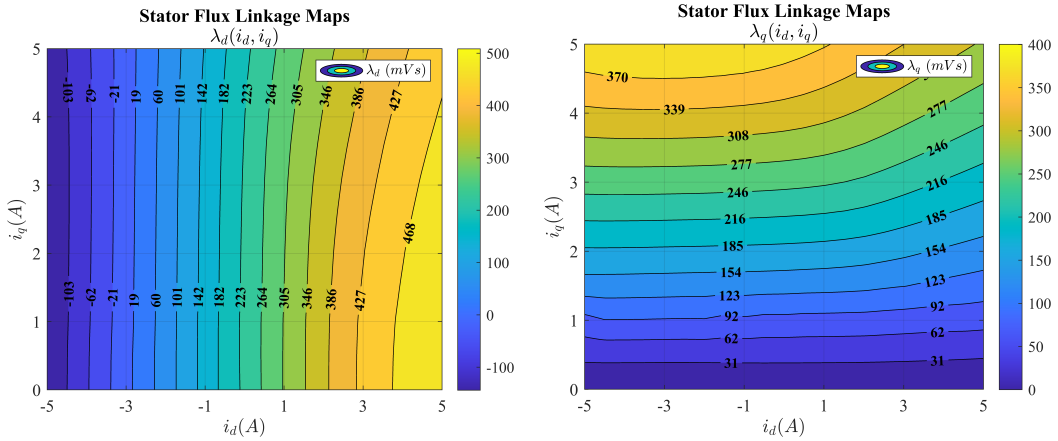
(a) $\lambda_d(id, iq)$ (b) $\lambda_q(id, iq)$

Figure 2.9: Direct flux maps of Synchronous Reluctance Motor



(a) $\lambda_d(id, iq)$ (b) $\lambda_q(id, iq)$

Figure 2.10: Direct flux maps of Internal Permanent Magnets Motor



(a) $\lambda_d(id, iq)$ (b) $\lambda_q(id, iq)$

Figure 2.11: Direct flux maps of Surface Permanent Magnet Motor

Each machine has its own characteristic magnetic map that represents its behavior with current. Given a pair of current values (i_d, i_q) the (λ_d, λ_q) flux values obtained from the magnetic map are the real ones that are in the machine in the specific operating point, with saturation phenomena already included in the model. To represent the (λ_d, λ_q) flux behavior of the machine in a clearer way, in Figure 2.12, 2.13 and 2.14 are showed the flux linkage maps of the three SMs under analysis, adopting the "PM-style" axis convention.

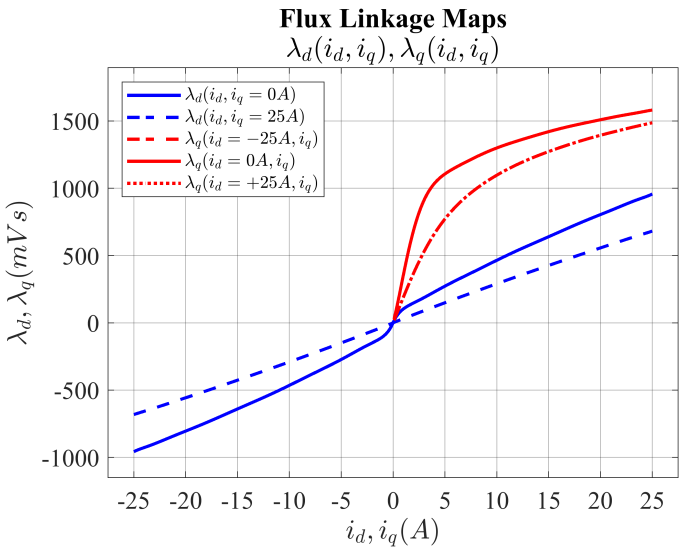


Figure 2.12: Flux linkage maps of Synchronous Reluctance Motor

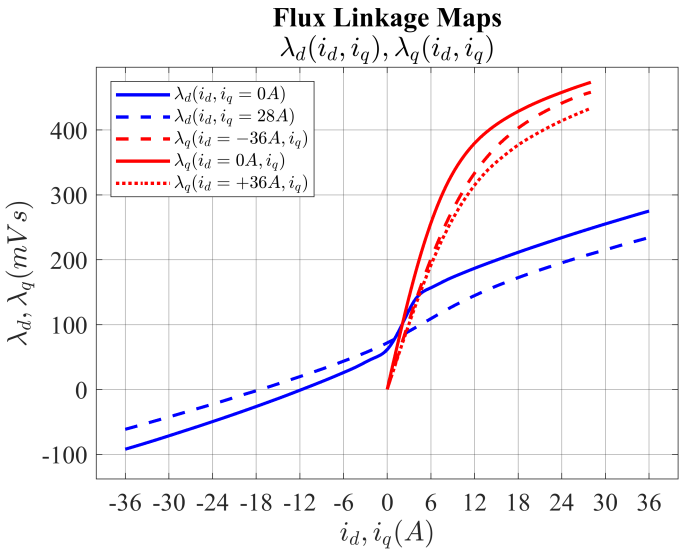


Figure 2.13: Flux linkage maps of Internal Permanent Magnets Motor

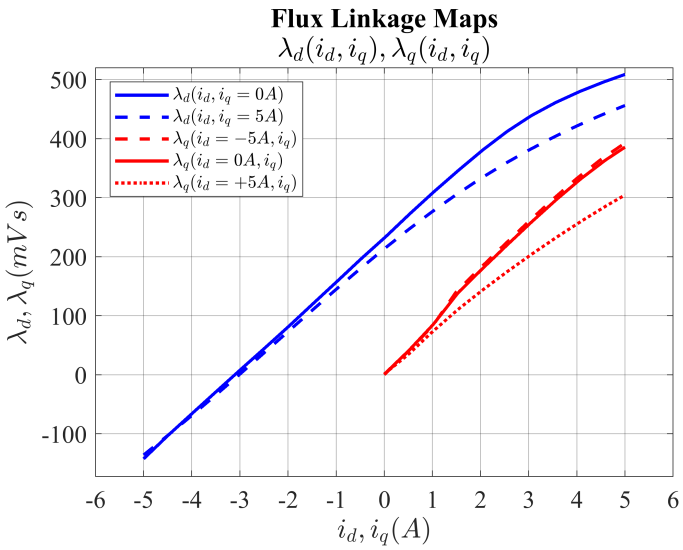
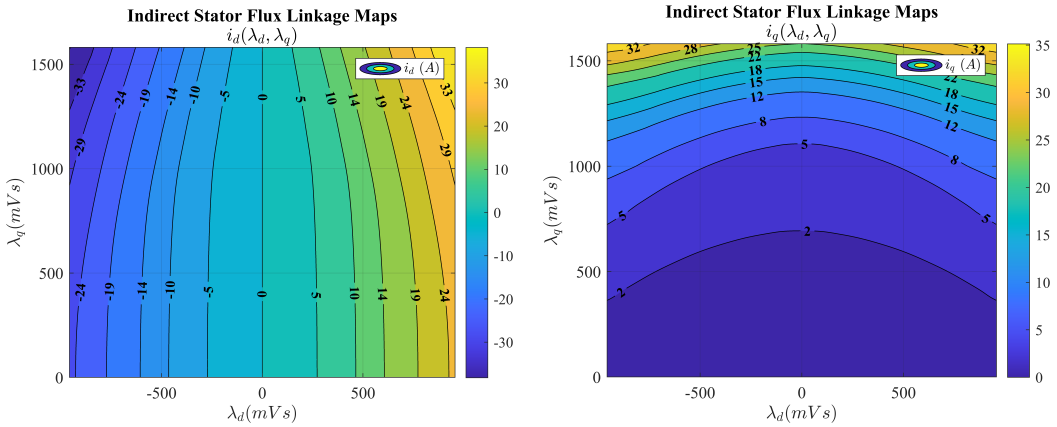


Figure 2.14: Flux linkage maps of Surface Permanent Magnet Motor

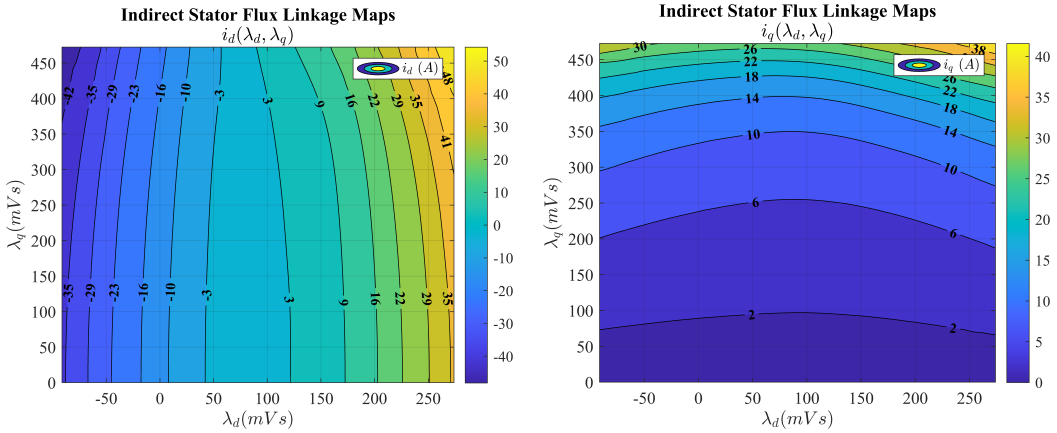
Being the SyR a magnetless motor the d-axis and the q-axis flux linkage have in common the zero-current point, while for IPM and SPM ones the d-axis characteristics is shifted in the origin ($i_d = 0A, i_q = 0A$) along the flux axis of the value magnet flux linkage ("PM-style" convention). Moreover, all the motor present an evident saturated behavior along the q-axis: the value of differential inductance on which the voltage equations of the motor are based on (see Equation (2.65) and (2.66)), is highly changing with current. If the control strategy is based on an inner current loop, like in FOC or in DFVC, this behavior cause a demanding tuning process of the adopted proportional-integral regulators.

As explained in 2.1.4 given the direct flux maps, the inverse ones, showed in 2.15,2.16 and 2.17, can be retrieved and used for control purpose, in particular for the generation of MTPV limit on $\{i_d, i_q\}$ plane to define the operating region of the machine.



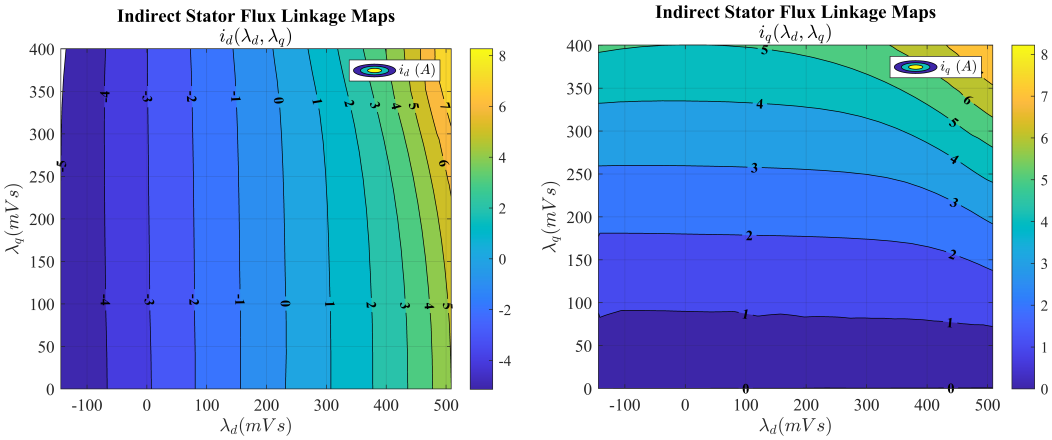
(a) $i_d(\lambda_d, \lambda_q)$ (b) $i_q(\lambda_d, \lambda_q)$

Figure 2.15: Indirect flux maps of Synchronous Reluctance Motor



(a) $i_d(\lambda_d, \lambda_q)$ (b) $i_q(\lambda_d, \lambda_q)$

Figure 2.16: Indirect flux maps of Internal Permanent Magnets Motor



(a) $i_d(\lambda_d, \lambda_q)$ (b) $i_q(\lambda_d, \lambda_q)$

Figure 2.17: Indirect flux maps of Surface Permanent Magnet Motor

2.2.4 Iron Losses model

Being the definition of the equivalent resistance of the iron of a SM quite difficult, the strategy adopted is to forecast a reasonable high value for this parameter and use it in the machine model. The main consequence of this choice is that, in the Simulink model of the machine, the iron losses are taken into account in a very simple way. As done for the AM in Section 2.1.5 the equivalent Thevenin model is created and its parameter are evaluated in the same way as for the AM.

These two parameter are directly used in the machine voltage Equation (2.81), substituting the original $\bar{v}_{dq}$ and the R_s terms. With this simply procedure the iron losses are modeled.

$$\bar{v}_{eq,dq} = R_{eq}\bar{i}_{dq} + \frac{d}{dt}\bar{\lambda}_{dq} + p\omega_m \boldsymbol{J}\bar{\lambda}_{dq} \tag{2.81}$$

Chapter 3

Flux Polar Control

The proposed torque controller allows the regulation of torque for an electric machine in alternating current three-phase, guarantying high dynamic performance in all the usable speed- and torque -range, respecting the machine and power converter constrains, i.e. maximum speed, current and voltage. This solution has no constrains in term of machine to be controlled, that can be synchronous and asynchronous, or on the working conditions, motor or generator.

Flux Polar Control is a torque control strategy implemented in a way that enable the user, the control of different types of electric machine with an approach "plug and play". It means that the regulators that perform flux polar control, i.e. the simultaneous regulation of the its amplitude and relative load angle, are always the same, as is their tuning that depends only on the sampling frequency of the digital controller. The only thing that should be changed, when passing from a machine to another, are the motor data.

The proposed torque controller scheme is no more difficult than ones for other control structure, as DFVC and the FOC-based control strategy ones, but with the advantages of being insensitive to the working point, i.e. the inductance value that are always influenced by magnetic saturation phenomena, and being able to exploit the machine performance in the deep flux-weakening region without additional regulation. This makes the proposed control algorithm particularly suitable for managing magnetically saturated machines such as electric / hybrid traction motors or those used in industrial applications with high performance and efficiency.

A general scheme of the control is in Figure 3.1 where all the novelty are enclosed in the Flux Polar Oriented blocks: "Load angle reference generation" and "Polar flux regulators".

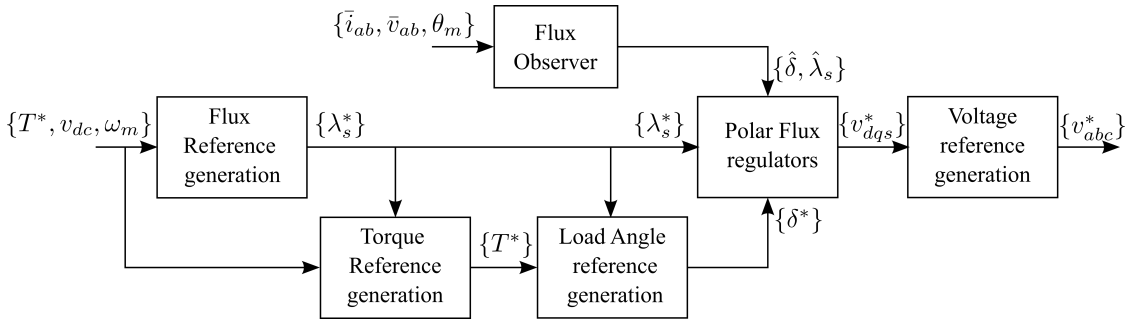


Figure 3.1: Flux Polar Control block scheme

All the block of the proposed controller will be described in detail with their functionality to torque management in the next sections.

3.1 Improvement on conventional solutions

Currently, different torque controller solutions have been introduced and used in the management of high performance three phase machine in automotive application but none of them is able to face different type of machine without modify any part of the implemented torque code, except the proposed one. As shown in section 1.1 and 1.2 the conventional solutions adopted up to now (CVC[6],DTC[6],DFVC [3]) have in common the main problem of being sensitive to the instantaneous working point, regulating at least one current component of the machine. This make them unsuitable for handling high-saturated machines unless additional adjustments are made tfo avoid loss of control in extreme situations. In this control strategy, LUTs are deeply used to set the main control constrain, in addition to the common one like current and voltage limits of the converter, and their dimension can progressively increase:

- current to max torque (MTPA) or flux to max torque (MTPV) in 1D-LUT;
- current to flux in two 2D-LUT in the flux observer structure;
- multi-dimensional LUT to manage all the possible working point of the machine depending on the number of considered control variables such as: mechanical speed or position, temperature, dc-link voltage and control set-point, i.e. torque reference.

As the LUT becomes more complicated, the internal control scheme tends to be simpler, because all the control constrain have been already take into account, but at the price of high amount of data to be stored and interpolated at every switching period of the control.

Nevertheless, the problem of the dependency of control on the working point is not overcomes: if the output of the most complete and accurate multi-dimensional LUT are the two reference currents (i_d^*, i_q^*) evaluated for the inner control loops, the problem of the tuning of the regulators is present and still needs to be solved.

Flux Polar Control overcomes these limit by introducing a unique control map able to handle the machine in all possible working conditions. Given the torque set-point, linear regulation is guaranteed by the definition of the flux amplitude and load angle:

- flux amplitude reference is retrieved from a 1D-LUT storing the relation between torque value and flux, in MTPA condition. This value can still be adjusted following a model-based flux weakening (FW) law, independent on the machine type, that take into account the vdc voltage, the machine speed and the actual power delivered.
- Torque and flux target values are the input of a single 2D-LUT providing the reference load angle value that must be forced into the machine. The interpolation process is performed on a map defined on a non-regular flux-torque domain that must be handle with a specific algorithm able to linearize it. This process is carry out in the preparation stage of the map necessary for the control and, during the code execution, no particular care must be given to it. Hence the real-time interpolation can be easily done on commercial discrete-time controllers.

Most important, tuning process of the two inner regulators is independent on the machine type and operating point, but only limited by the discretization frequency of the digital control that performs it.

All the previous features lead to some benefits in term of performance on the one achievable with other control strategy:

- the flux weakening region is exploited thanks to a model-based law, without any outer regulator, enabling high dynamic performance in all the speed range, especially at high speed in the flux-weakening region. The MTPV locus is already included in the exploitable working region of the motor present into the load angle control map. On the contrary an outer regulator acting on the q-axis voltage limit, require a demanding tuning procedure and means also a significant slowing. The same applies to a regulator that acts on the load angle of the machine to fulfill the MTPV operation.
- A single 2D load angle LUT already take into account all the limitation of the control, i.e. torque reference- and maximum -value, mechanical speed, voltage and

current power converter limits. No more that one LUT is necessary, contrary to what is required in LUT-based control algorithms, in which they are usually several and large.

- Polar flux control has unbalanced voltage limits for the inner regulator, as the DFVC strategy. In steady-state conditions the two axis are almost decoupled:
  - the d-axis voltage loop, acting on the amplitude of flux, has a value that is only the one due to phase resistance voltage drop and voltage error of the power converter (dead time and voltage threshold of the various components);
  - the q-axis voltage loop, acting on the phase of flux vector, should face all the back EMF of the machine and, for this reason, it has all the residual voltage of the converter, that is the larger part.
- As said before, the behavior achievable from this torque controller is always the same independently on the type and on the saturation of the machine on which it is applied: being the regulators always the same, permits to compare the machine performance only on the base of its characteristics, without any limitation from the regulator itself.

3.2 Control scheme

The Flux Polar Control scheme in Figure 3.2, already presented in 1.3 is now analyzed in all its component to give an exhaustive explanation of all the different blocks that compose it. An important part of the proposed structure is in the same as others torque controller schemes, such as DFVC, but some blocks that perform characteristic tasks have been specifically developed.

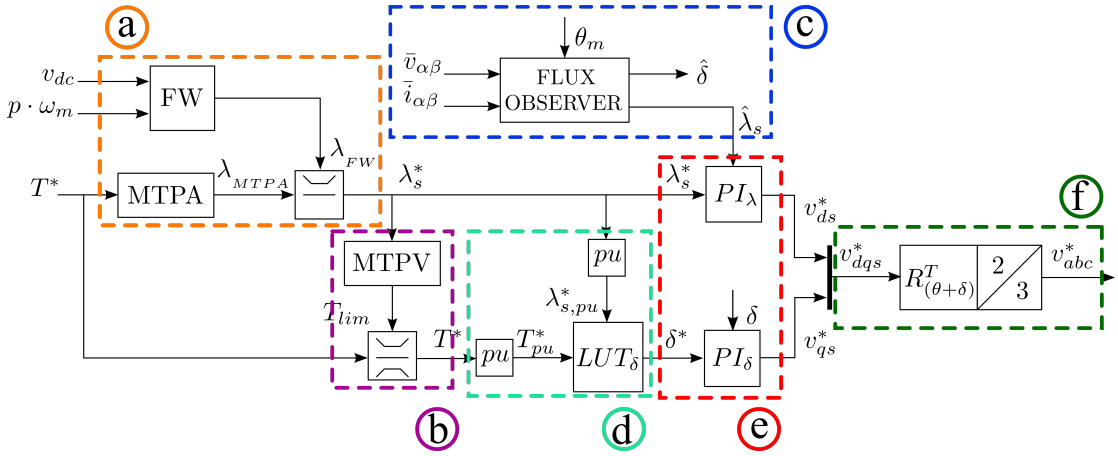


Figure 3.2: Flux Polar Control blocs scheme

In particular blocks ***a***,***b***,***c***,***e*** and ***f*** are quite common between the different control structure, implementing some basic functions such as: the definition of flux reference amplitude in ***a***, torque limitation following MTPV locus in ***b***, a flux observer structure in ***c*** and, independently from the tuning procedure, two regulators in ***e*** that give the voltage reference for ***f***. On the contrary blocks ***d*** represent the real novelty for the proposed control structure both in term of content of the 2D-LUT used to define the load angle value and of its usage.

3.2.1 Input signal management

To perform the Flux Polar Control a number of variables have to measured from the system. The value of phase currents, the dc link voltage and the mechanical position angle from encoder must be known in order to run this control.

The value of the three phase currents of the machine is measured at every ISR call. From the instantaneous value is removed the sensor offset, evaluated in a previous dedicated stage of the control. In this way the value of $\bar{i}_{abc}$ is known.

The mechanical angle value is obtained, at each ISR call, from the position transducer placed on the machine shaft. To obtain the real mechanical position of the machine, it is necessary to perform an alignment operation in advance, to identify the mechanical mounting offset of the device. The offset value is then removed from the measured position, to get the real one. Starting from this data, two different evaluation can be performed:

- the electric angle of the machine θ_{elt} is evaluated as pole pairs time the mechanical angle. This operation is useful only for SM in which the measured angle represent

one of the p possible position of the magnet-aligned-axis, in our convention the d-axis.

- the speed of the machine can be evaluated using a PLL (Phase Locked Loop) structure like the one in Figure 3.3, that receive as inputs the mechanical angle θ. Using a PLL enable to obtain the speed value ω and its filtered version ω_{filt} without any other type of transducer then the position one, and also a filtered version of the mechanical position value $\hat{\theta}$.

  Usually, rather then feeding directly the PLL with the difference between measured and filtered angle, the sine approximation of this difference is used. In this way any discontinuity resulting from the difference between angles is avoided. The gain of the adopted PI(Proportional Integral) regulator, are set choosing the desired bandwidth ω_b and phase margin φ_m adopting the following equations:

$$k_p = \frac{\omega_b \tan(\varphi_m)}{\sqrt{1 + \tan^2(\varphi_m)}} \tag{3.1}$$

$$k_i = \frac{\omega_b^2}{\sqrt{1 + \tan^2(\varphi_m)}} \tag{3.2}$$

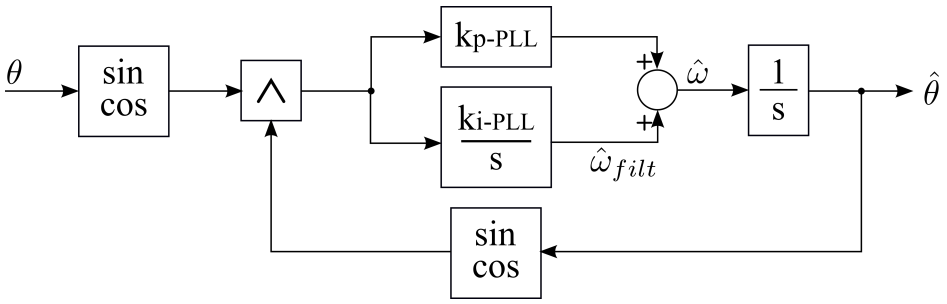


Figure 3.3: Phase Locked Loop scheme

The measured dc link voltage is filtered with a high-band low pass filter (500 Hz) in order to remove the high-frequency noise. This value is one of the main input to reconstruct the stator phase voltages in phase coordinates $\bar{v}_{abc,ideal}$ from the duty-cycles d_{abc} of the converter.

$$\begin{cases} v_{a,ideal} = \dfrac{V_{dc}}{3}(2d_a - d_b - d_c) & (3.3) \\ v_{b,ideal} = \dfrac{V_{dc}}{3}(2d_b - d_c - d_a) & (3.4) \\ v_{c,ideal} = \dfrac{V_{dc}}{3}(2d_c - d_a - d_b) & (3.5) \end{cases}$$

A subsequent compensation of converter voltage errors using techniques is performed to

obtain $\bar{v}_{abc,real}$. The voltage error introduced by the converter is obtained using a LUT that stores the p.u. value of voltage error depending on the current value in the device. For each dc voltage level and switching frequency adopted in the control, a LUT must be created following the procedure proposed in [9]. An example of the content of the inverter DT LUT, obtained experimentally carrying out the described procedure on the Induction Motor, is showed in Figure 3.4.

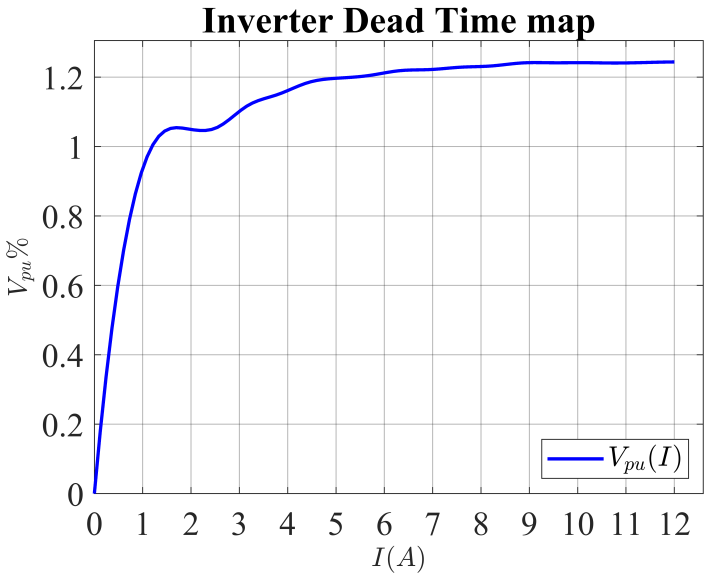


Figure 3.4: Experimental Inverter Dead Time of IM (Vdc=600V, fsw=4 kHz)

The value of voltage error of generic phase x is evaluated as:

$$v_{x,DT} = sign(i_x) \cdot v_{x,DT\_pu}(|i_x|) \cdot v_{dc} \tag{3.6}$$

The final value of the real phase voltage is showed:

$$\begin{cases} v_{a,real} = v_{a,ideal} - v_{a,DT} & (3.7) \\ v_{b,real} = v_{b,ideal} - v_{b,DT} & (3.8) \\ v_{c,real} = v_{c,ideal} - v_{c,DT} & (3.9) \end{cases}$$

Finally, the stator phase voltage and current in stationary coordinates ($\alpha\beta$) have been evaluated applying the invariant amplitude Clarke transform.

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & -\sin(30) & -\sin(30) \\ 0 & \cos(30) & -\cos(30) \end{bmatrix} \cdot \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \tag{3.10}$$

3.2.2 Flux reference generation

Being the proposed controller a "Flux Polar Control", the knowledge of the amplitude of reference flux that must be imposed into the machine to satisfy a specific torque request, is of primary importance. The approach here adopted to define the adaptive flux value is the same used in DFVC: the amplitude of flux is generated starting from the reference torque value, previously saturated between its extremes value, thanks to a LUT that stores the torque to flux relationship along MTPA locus.

The "Flux reference generation" scheme, highlighted with the letter ***a*** in Figure 3.2 is detailed in Figure 3.5 here showed.

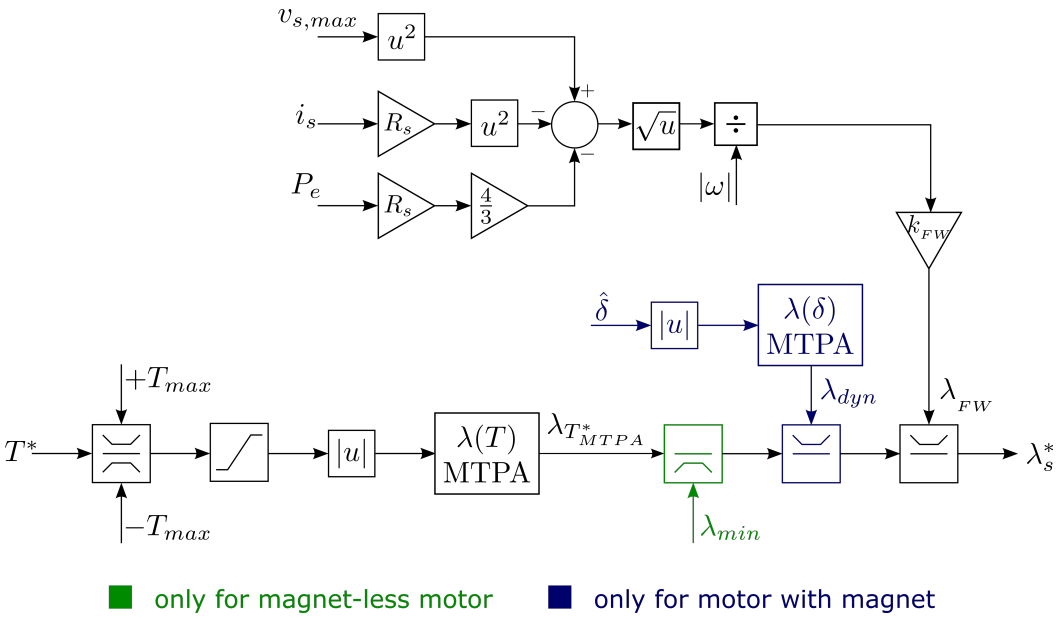


Figure 3.5: Flux reference generation in FPC scheme

The input torque value is the reference one imposed by an external command to the control: it can be an outer speed loop requiring torque to satisfy a specific set-point or, directly, a torque value needed for a specific application.

First of all, to this set point are performed two different operation:

- saturation: the reference torque cannot exceed the maximum exploitable value define for the specific machine. Given the nominal torque, an overload one T_{max} can be set as maximum reachable value, corresponding to the extreme point of *MTPA* defined by the current limit imposed by the control. It must not be overcomes in all different working condition: this is forbidden by the control structure;

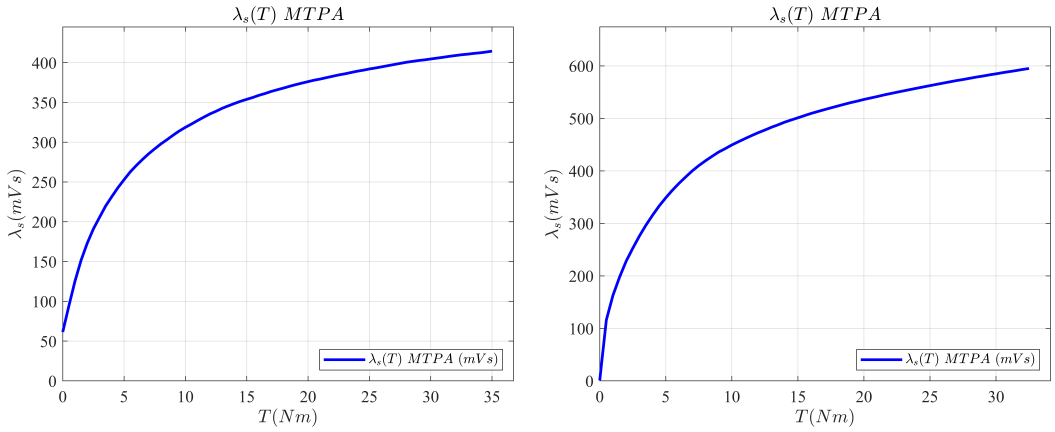
- slew-rate limitation: to avoid step change in the torque reference value, a limit to

slope the variation of the reference value is imposed by the control structure. In this way, uncontrolled change in the reference value, are avoided. The value is gradually changed form the starting value to reach the final one with with a predetermined dynamic behavior.

Following the two preceding step, the torque value is used to define the starting flux reference value. The first 1D-LUT is used for this purpose: it stores the relation between torque value and machine flux amplitude in *MTPA*. The curves that describe this behavior is a characteristic of the machine: for magnet-less machine the curve start from zero torque with no flux, while the ones with magnet have already present the magnet flux, also with zero torque, along MTPA. Two examples of this LUT, for an IPM and an IM are showed in Figure 3.6.

The MTPA locus is obtained in (i_d, i_q) plane by the derivative of torque equation with respect to current load angle γ. MTPA locus can be also expressed in (λ_d, λ_q) plane adopting an interpolation process involving the direct flux maps. Given the current value $(i_{d,MTPA}, i_{q,MTPA})$ the relative flux $(\lambda_{d,MTPA}, \lambda_{q,MTPA})$ are so obtained.

The flux amplitude in *MTPA* condition $\lambda_{S,MTPA}$ is the square root of this two components squared. With the data (T_{MTPA},$\lambda_{S,MTPA}$) the LUT between them is created, permitting the definition of a preliminary flux value from which starts the accurate definition of its actual value. Note that, being the *MTPA* and the *MTPV* locus symmetrical along the d-axis, only one side of them must be memorized into the LUT, halving the required memory.



(a) *IPM* (b) *IM*

Figure 3.6: $\lambda_s(T)$ in *MTPA* of IPM and IM

The LUT is read using a specific function that has been already implemented, and obtained by the writer from a function archive (see Appendix). The absolute reference

torque value T^* is given as input and the correspondent flux amplitude $\lambda_{T^*,MTPA}$ is calculated and provided as output of the function.

Three different saturation of the so obtained value can be performed now:

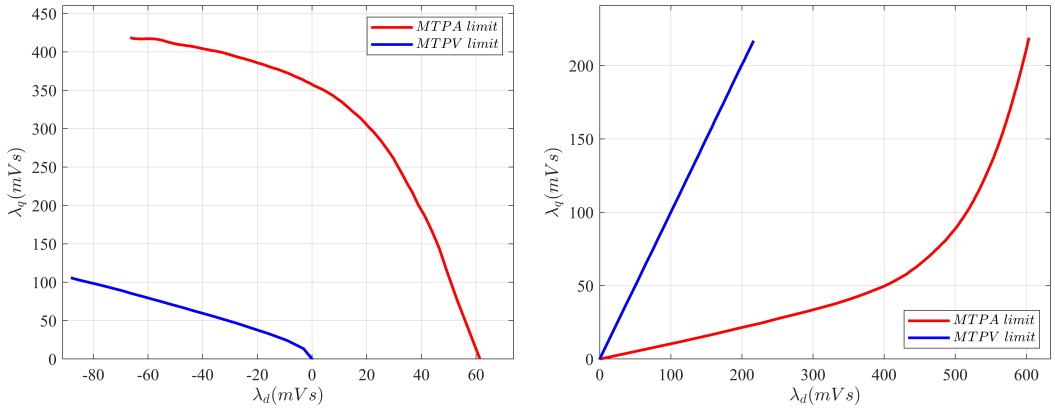
- a saturation to the minimum flux value that must be guarantee in the machine, only for the magnet-less ones;
- a dynamic saturation of the flux reference based on the observed load angle, following the *MTPA* relation between load angle and flux. It is particularly suitable for machine with high anisotropy ratio, but is not mandatory;
- a saturation to the maximum flux value in the high-speed working range, following a model-based flux-weakening law, for all kind of machine.

Former saturation is reserved for those machines without magnets, such as the AM and, between the SM, the pure reluctance one. In these case, a minimum value of flux must be always guarantee to preserve the machine from complete cancellation of the flux. For this reason, minimum flux reference is always set to λ_{min} and then, eventually, redefined applying following saturation. In fact, for these two machine a preliminary stage of flux build is required to set the machine flux to λ_{min} and to be ready for the control. The minimum value of desired flux into the machine can be choose arbitrarily by the coder, following the machine characteristics. A typical value of flux, can be the one that corresponds to the nominal torque of the machine.

The second saturation is performed to ensure respect of *MTPA* locus limitation in case of torque reference inversion. In fact, when the reference torque value is inverted from a positive to a negative value, wherever the actual working point is placed, the *MTPA* flux limit must be respected. Reversing the torque reference maintaining its amplitude, means only changing the sign of load angle value, maintaining both its and flux amplitudes constant at their previous value. This issue must be faced only for motors whose control maps are build in the second quadrant ($i_d < 0, i_q > 0$) where the q-coordinates of the *MTPA* locus are higher then the one of *MTPV*. In this case a variation of flux over the value defined by the *MTPA* can causes instability phenomena that can leads to loss of control. If the control maps are in the first quadrant ($i_d > 0, i_q > 0$) this problem is naturally overcome, because the q-coordinates of *MTPA* locus are lower then the one of *MTPV* and so, no problem rises.

This is true for SM with magnets in which the two *MTPV* locus (motor and generator one) are placed in-between the *MTPA* motor- and generator -locus. For what concern the AM and the synchronous reluctance one, this problem is avoided by the favorable positioning of the *MTPA* and *MTPV* locus on the (dq) current and flux plane. In fact, in this case the two side of the *MTPA* locus can be directly connected and are lower then the

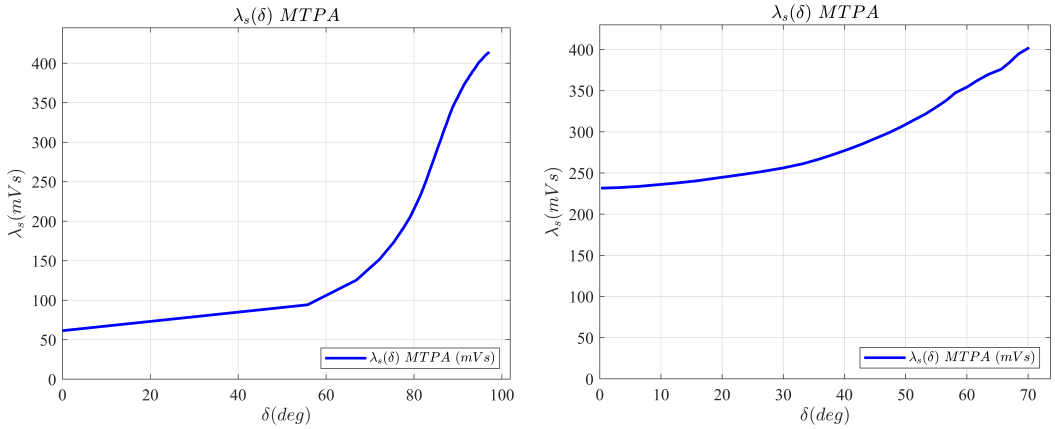
MTPV one. For this reason no dynamic flux-weakening limit is required. The *MTPA* and *MTPV* of these two type of motors, whit the different positioning, are showed in Figure 3.7



(a) *IPM* (b) *IM*

Figure 3.7: Comparison of positioning of *MTPA* and *MTPV* of IPM and IM

To impose the dynamic limitation a 1D LUT is used. It contains the information of flux and load angle value along *MTPA* (δ_{MTPA},$\lambda_{S,MTPA}$). Given the actual value of estimated load angle $\hat{\delta}$, obtained from the flux observer structure, its absolute value is provided to the LUT that produces the correspondent flux value $\lambda(\delta)_{MTPA}$. This is used to perform the first saturation of the flux reference value. Two examples of $\lambda_{MTPA}(\delta)$, for an IPM and a SPM are in Figure 3.8.



(a) *IPM* (b) *SPM*

Figure 3.8: $\lambda(\delta)$ in *MTPA* of IPM and SPM

Latter saturation is done following actual mechanical speed ω_m and dc voltage v_{dc}

of the machine, to perform flux weakening operation, enabling working in the high-speed region. Indeed, if the machine operates below the base speed, the reference flux determined after this first saturation is the one that effectively corresponds to the value assumed as the reference by the flux amplitude control loop. Otherwise, a model-based flux-weakening law must be used to limit the flux amplitude, when the speed overcomes the base one. To obtain this law, the machine voltage equations are required. They can be written as follows:

$$\begin{cases} v_d = R_s i_d + \dfrac{d}{dt}\lambda_d - \omega\lambda_q & (3.11) \\ v_q = R_s i_q + \dfrac{d}{dt}\lambda_q + \omega\lambda_d & (3.12) \end{cases}$$

where the term ω is equal to: $\omega = p\omega_m$ for SM and $\omega = p\omega_m + \omega_{slip}$ for AM. Anyway, summing the two equations squared leads to the following:

$$v_d^2 + v_q^2 = v_s^2 = \left(R_s i_d + \frac{d}{dt}\lambda_d - \omega\lambda_q\right)^2 + \left(R_s i_q + \frac{d}{dt}\lambda_q + \omega\lambda_d\right)^2 \tag{3.13}$$

The two flux time-derivative terms can be neglected considering that, the weakening dynamic evolves in seconds, while the derivative take into account phenomena that change in millisecond. The equation become:

$$\begin{aligned} v_s^2 &= \left(R_s i_d - \omega\lambda_q\right)^2 + \left(R_s i_q + \omega\lambda_d\right)^2 \\ &= R_s^2(i_d^2 + i_q^2) + \omega^2(\lambda_d^2 + \lambda_q^2) - 2R_s i_d \omega\lambda_q + 2R_s i_q \omega\lambda_d \\ &= R_s^2 i_s^2 + \omega^2\lambda_s^2 + 2R_s\omega(\lambda_d i_q - \lambda_q i_d) \\ &= R_s^2 i_s^2 + \omega^2\lambda_s^2 + \frac{2}{3}2R_s\frac{3}{2}p(\lambda_d i_q - \lambda_q i_d)\omega_m \\ &= R_s^2 i_s^2 + \omega^2\lambda_s^2 + \frac{4}{3}R_s T_e \omega_m \\ &= R_s^2 i_s^2 + \omega^2\lambda_s^2 + \frac{4}{3}R_s P_e \end{aligned} \tag{3.14}$$

where the electromagnetic torque equation has been substituted by its symbol T_e and the electric power term P_e appears as product of torque and mechanical speed $P_e = T_e\omega_m$. Finally, the maximum admissible flux value in the machine, given the instantaneous working condition, can be evaluated reversing the equation (3.14) as:

$$\lambda_{s,max} = k_{FW} \cdot \frac{\sqrt{v_{max}^2 - R_s^2 i_s^2 - \dfrac{4}{3}R_s P_e}}{|\omega|} \tag{3.15}$$

where v_{max} is the amplitude limit of the phase voltages, whose value usually corresponds to the sinusoidal voltage limit of the inverter ($v_{dc}/\sqrt{3}$), i_s represents the current amplitude in the machine and R_s and P_e are respectively the resistance and the electric machine power. The value of P_e is evaluated at every control cycle using directly voltage and current:

$$P_e = \frac{3}{2} \cdot (v_\alpha i_\alpha + v_\beta i_\beta) \tag{3.16}$$

The implementation of the resistive terms in (3.15) is justified only for low-power motors, as they are usually characterized by a significant value of the phase resistance. The term "k_{FW}" $(0.85 - 0.95)$, that appears in front of the law, is a conservative term, that allows the regulator of the load angle to have always sufficient tension margin in all conditions.

Finally, it is noted how (3.15) leads to a straightforward FW regulation, without using any outer voltage or load angle regulator (as for the conventional FOC or DFVC scheme [2], [10]) whose side effect is often to compromise the dynamic of the torque regulation in the whole speed range.

After these two saturation the real reference value of flux λ^* that must be used to adjust the torque reference value and to set the load angle value, is determined.

3.2.3 Torque reference management

Given the amplitude of flux λ^* obtained following the previous prescription, the original value of reference torque T^* can still be changed after the machine current limit and MTPV locus limitation imposed by the flux to torque relation saved in another 1D LUT, generated thanks to motor control maps.

The "Torque reference management" scheme, highlighted with the letter ***b*** in Figure 3.2 is detailed in Figure 3.9 here showed.

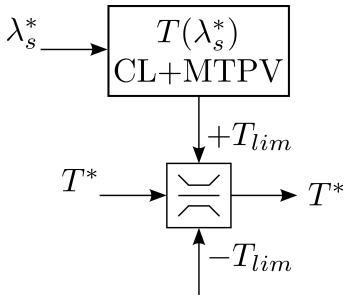


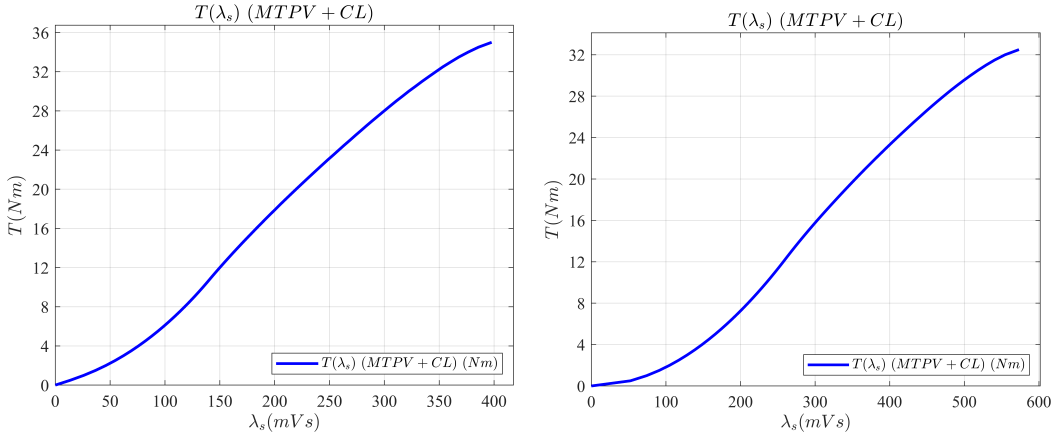
Figure 3.9: Torque reference generation in FPC scheme

In this case, the considered relation is the one between flux and maximum torque amplitudes along the Current Limit and MTPV locus of the machine. As explained in the

previous Section, above base speed, the flux value that can be requested in the machine is the one of *MTPA* locus reduced, using the adopted flux-weakening law, to the maximum reachable value in the specific working condition. The same happens for the reference torque: up to base speed the only limit that is applied to the reference value requested by the user, is the extreme value of mechanical torque that can be obtained from the machine. When the speed rises above this threshold, this maximum limit is progressively reduced, first following the current limit of the machine and then reaching the *MTPV* one.

To perform this limitation, the LUT storing the admissible torque value related to flux is used. Providing as input the flux reference, the corresponding value of maximum reachable torque is produced. This value is applied as a symmetrical limit to the torque reference, covering both the motor and generator cases (torque maintain its value with sign that is taken into account for the generation of load angle reference).

Two examples of $T(\lambda_s)$ in *MTPV+CL* are in Figure 3.10. This limit must be applied as soon as MTPA is left to continue on the limit defined first by the CL and then by the MTPV.



(a) *IPM* (b) *IM*

Figure 3.10: $T(\lambda_s)$ along *MTPV+CL* of IPM and SPM

After this saturation stage the final reference torque value T^* is settled.

3.2.4 Flux observer structure

Once the desired flux value λ_s^* is known, the actual one must be evaluated. To do this, a flux observer structure has been implemented to obtain both the estimated amplitude $\hat{\lambda}_s$ and load angle $\hat{\delta}$ values of the flux vector present in the machine, that are the two feedback for the inner control loops in FPC.

Different type of flux observer can be adopted following the desired accuracy and complexity of the implemented structure. The Flux Observer used to estimate the stator flux component, has been created in ($\alpha\beta$) reference and it is a simple version of the so-called Gopinath observer [11]. It combine the output of two different flux estimator: one is the rotor model (or current model) ($I\theta$) and the other is the stator model (or voltage model) (VI). The global output between the two is their frequency-weighted average.

Asynchronous machine

The "Flux observer structure" scheme, highlighted with the letter ***c*** in Figure 3.2 is detailed in Figure 3.11 for AM.

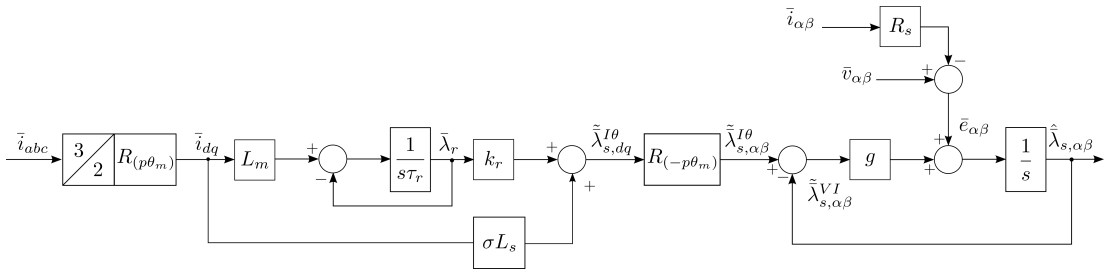


Figure 3.11: Flux observer scheme for Asynchronous machine

The ($I\theta$) rotor model is created adopting the rotor voltage equation in (dq) rotor coordinates. Equation (2.21) can be rewritten setting to zero the motional term, being in this case ($\omega == p\omega_m$), and therefore no coupling among the two components of the equation exist. Moreover the rotor current can be substitute by its expression retrieved from (2.6):

$$\begin{aligned}\bar{v}_{r,dq} = 0 &= R_r\bar{i}_{r,dq} + \frac{d}{dt}\bar{\lambda}_{r,dq} \\ &= R_r\left(\frac{\bar{\lambda}_r - L_m\bar{i}_{s,dq}}{L_r}\right) + \frac{d}{dt}\bar{\lambda}_{r,dq} \\ &= \frac{1}{\tau_r}\bar{\lambda}_r - \frac{1}{\tau_r}L_m\bar{i}_{s,dq} + \frac{d}{dt}\bar{\lambda}_{r,dq}\end{aligned} \tag{3.17}$$

Moving in Laplace domain the equation become:

$$0 = \bar{\lambda}_r - L_m\bar{i}_{s,dq} + s\tau_r\bar{\lambda}_{r,dq} \tag{3.18}$$

from which the rotor flux can be written as:

$$\bar{\lambda}_r = \frac{L_m \bar{i}_{s,dq}}{1 + s\tau_r} \tag{3.19}$$

Finally, the stator flux estimated with $(I\theta)$ model in $(\alpha\beta)$ coordinates $\bar{\tilde{\lambda}}^{I\theta}_{s,\alpha\beta}$, can be obtained, rotating with $\{\theta_m\}$ the stator flux $\bar{\tilde{\lambda}}^{I\theta}_{s,dq}$ obtained adopting Equation (2.8):

$$\begin{aligned} \bar{\tilde{\lambda}}^{I\theta}_{s,\alpha\beta} &= [R(-\theta_m)]\bar{\lambda}_{s,dq} \\ &= [R(-\theta_m)] \cdot (k_r \bar{\lambda}_{r,dq} + \sigma L_s \bar{i}_{s,dq}) \end{aligned} \tag{3.20}$$

The (VI) stator model is created using the voltage stator equation in $(\alpha\beta)$ stator reference (2.1). Estimated stator flux $\bar{\lambda}^{VI}_{s,\alpha\beta}$ can be expressed as integral of the back EMF of the machine $\bar{e}_{\alpha\beta}$ evaluate using the real phase voltages $\bar{v}_{\alpha\beta,real}$ previously calculated:

$$\bar{\tilde{\lambda}}^{VI}_{s,\alpha\beta} = \frac{\bar{v}_{\alpha\beta,real} - R_s \bar{i}_{\alpha\beta}}{s} \tag{3.21}$$

Synchronous machine

The "Flux observer structure" scheme for SM is detailed in Figure 3.12.

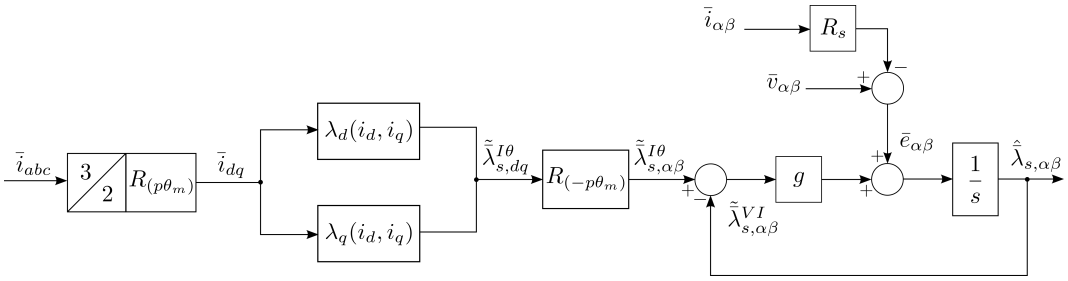


Figure 3.12: Flux observer scheme for Synchronous machine

The $(I\theta)$ current model is created exploiting the direct flux map of the machine. The current in $(\alpha\beta)$ coordinates are rotated in (dq) rotor frame using a direct rotational transform. Here the two direct maps $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ are used to define the two component of flux that are then bring back in $(\alpha\beta)$ frame, obtaining $\bar{\tilde{\lambda}}^{I\theta}_{s,\alpha\beta}$. In this case, all the magnetic behavior of the machine is already into the maps and thus the magnetic equation are avoided in favor of this simpler method.

The (VI) voltage model is the same adopted for the AM, using Equation (3.21). The integral of the back EMF is used to define the estimated flux $\bar{\tilde{\lambda}}^{VI}_{s,\alpha\beta}$.

Using the ($I\theta$) model or the (VI) one alone can cause some disadvantages. The ($I\theta$) model works well at low speed and standstill and it is immune from voltage error due to dead time. The main problems are that, requiring rotor position and current value, these are discretized at high speed and so, the model can fail. Moreover it is highly sensitive to parameter variation (L_m, τ_r) for asynch. one. On the contrary the (VI) model is reliable only at high speed when the back EMF is sufficiently high and take into account also the iron losses of the machine. Its drawbacks are that the dead time measurement must be correct to evaluate the phase voltages and, at low speed, the integral doesn't works.

The evident solution is to combine these two together to get the best from each of them. To do this a ($VI\theta$) model is settle, introducing a gain "g" that represent a frequency-threshold between the use of one or the other of the two proposed estimator. For both type of machine, the observed stator flux $\hat{\bar{\lambda}}^{VI\theta}_{s,\alpha\beta}$ can be finally evaluated:

$$\hat{\bar{\lambda}}^{VI\theta}_{s,\alpha\beta} = \tilde{\bar{\lambda}}^{I\theta}_{s,\alpha\beta} \frac{1}{1+\frac{s}{g}} + \tilde{\bar{\lambda}}^{VI}_{s,\alpha\beta} \frac{s}{s+g} \tag{3.22}$$

The frequency behavior of the combination of these two estimator is showed in Figure 3.13.

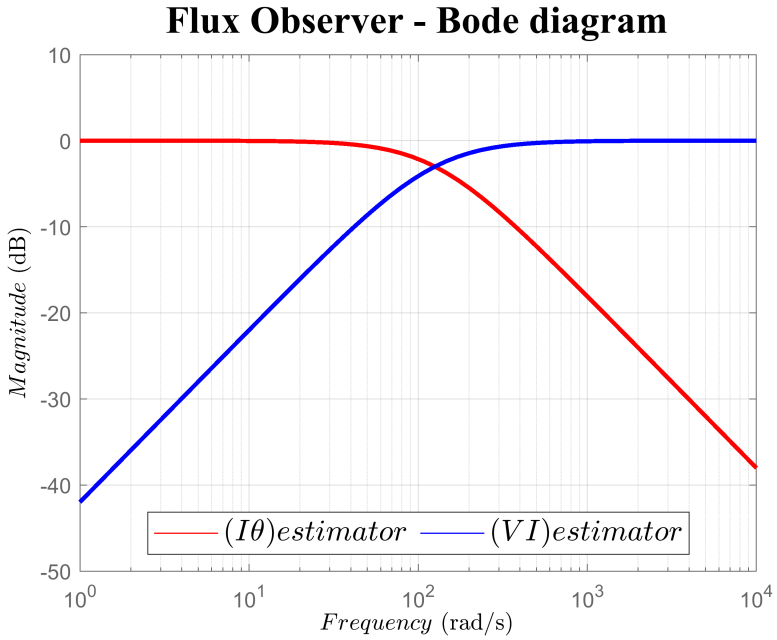


Figure 3.13: Flux observer: frequency behavior

In the end, the main output of the flux observer are the estimated flux amplitude and

load angle evaluated as:

$$\begin{cases} |\hat{\bar{\lambda}}_s| = \sqrt{\hat{\bar{\lambda}}_{s,\alpha}^2 + \hat{\bar{\lambda}}_{s,\beta}^2} & (3.23) \\ \delta = \arctan(\cos(\delta), \sin(\delta)) & (3.24) \end{cases}$$

Being δ the phase-shifting between stator and flux vector, its value can be evaluated as the difference between the angle of the two fluxes:

$$\begin{cases} \cos(\delta) = \cos(\theta_s - \theta_r) = \cos(\theta_s)\cos(\theta_r) + \sin(\theta_s)\sin(\theta_r) & (3.25) \\ \sin(\delta) = \sin(\theta_s - \theta_r) = \sin(\theta_s)\cos(\theta_r) - \cos(\theta_s)\sin(\theta_r) & (3.26) \end{cases}$$

- for the AM, the observed flux $\hat{\bar{\lambda}}_s$ is calculated in $(\alpha\beta)$ reference, so to calculate the load angle value both stator and rotor flux angles are needed.

$$\begin{cases} \cos(\theta_s) = \dfrac{\hat{\bar{\lambda}}_{s,\alpha}}{|\hat{\bar{\lambda}}_s|} & \sin(\theta_s) = \dfrac{\hat{\bar{\lambda}}_{s,\beta}}{|\hat{\bar{\lambda}}_s|} & (3.27) \\ \cos(\theta_r) = \dfrac{\bar{\lambda}_{r,\alpha}}{|\bar{\lambda}_r|} & \sin(\theta_r) = \dfrac{\bar{\lambda}_{r,\beta}}{|\bar{\lambda}_r|} & (3.28) \end{cases}$$

- for SM, once the observed flux $\hat{\bar{\lambda}}_{s,\alpha\beta}$ is known, (3.27) is performed. Moreover, its components are moved from $(\alpha\beta)$ reference to (dq) , using a rotational transform with argument the measured electric angle θ_{elt}. Finally, the load angle is simply evaluated as:

$$\delta = \arctan(\hat{\bar{\lambda}}_{s,q}, \hat{\bar{\lambda}}_{s,d}) \tag{3.29}$$

To get the synchronous speed of rotation ω_s of stator flux, $(\cos(\theta_s)$,$\sin(\theta_s))$ obtained with (3.27) can be give to a PLL, like the one presented in Section 3.2.1. Only the integral part of the internal PI regulator is used as reference electrical synchronous speed of the machine. This value is useful for phase-advancing of reference voltages, introduced later.

The knowledge of the position of the stator flux vector θ_s, that is the position of the d_s-axis of the rotating frame (dq_s) at time step (k), enable the evaluation of variable in this coordinates. An electric quantity that will be useful in this frame, is the stator current $\{i_{s,dq_s}\}$. The rotational transformation of the measured current, already in $(\alpha\beta)$, is the following:

$$\begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} = \begin{bmatrix} \cos(\theta_s) & \sin(\theta_s) \\ -\sin(\theta_s) & \cos(\theta_s) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{3.30}$$

3.2.5 Load angle reference generation

The heart of the Flux Polar Control structure has been reached. Knowing the values of torque T^* and flux λ_s^*, load angle reference value δ^*, can be obtained from the previous ones using the procedure that will be explained in detail. This permits to obtain the q-axis reference value δ^*, that, together with the d-axis one, that is the flux amplitude λ_s^*, will be used as input for the two inner regulators of the system which determine the $\{v_{ds}^*, v_{qs}^*\}$ voltage values.

The "Load angle reference generation" scheme, highlighted with the letter ***d*** in Figure 3.2 is detailed in Figure 3.14 here showed.

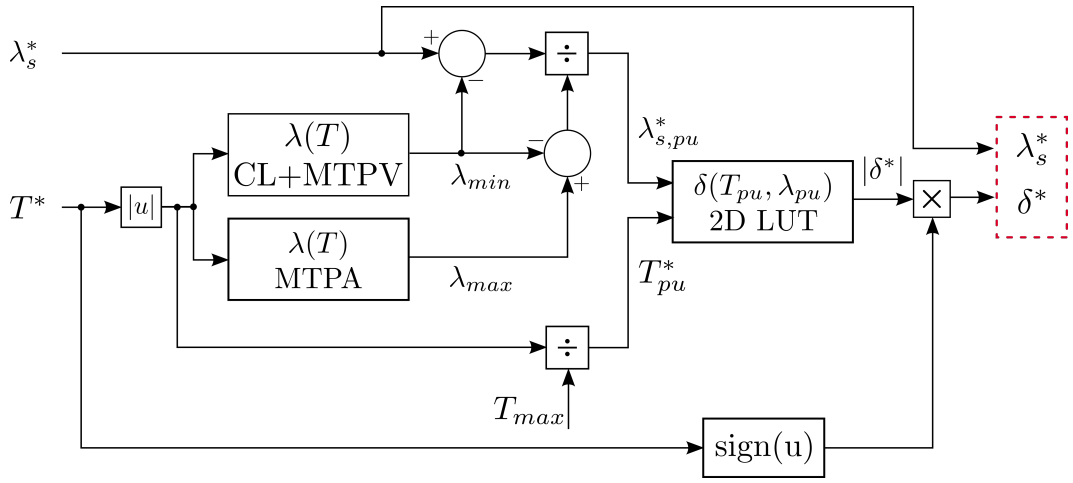


Figure 3.14: Load angle reference generation

The p.u. values of reference torque T_{pu}^* and flux $\lambda_{s,pu}^*$ are the input of the 2D LUT that stores the relation between load angle and values of torque and flux. To obtain these values several step must be done. This procedure reflect the one adopted in the generation of control map that will be explained in Section 3.3.

- **Flux reference in pu** $\lambda_{s,pu}^*$

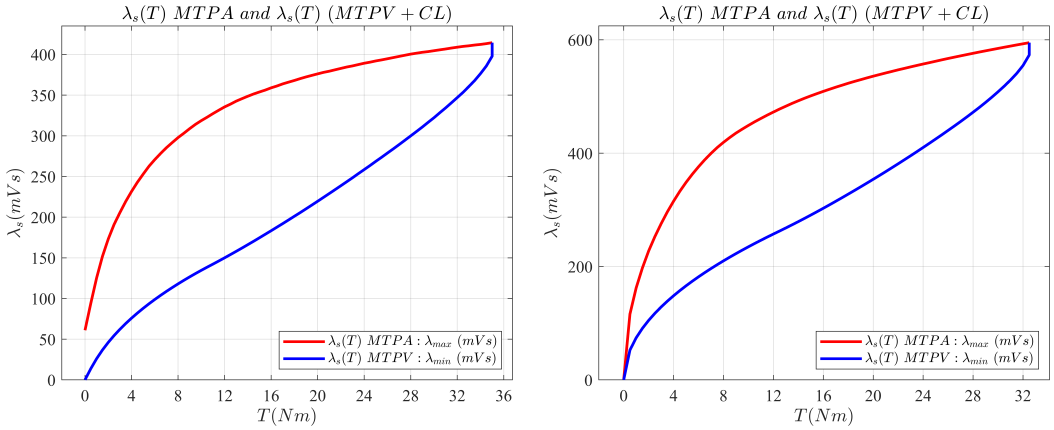
  A normalization is required to obtain the reference value of flux in p.u. starting from the absolute one, as expressed in Equation (3.31):

$$\lambda_{s,pu}^* = \frac{\lambda_s^* - \lambda_{min}(|T^*|)}{\lambda_{max}(|T^*|) - \lambda_{min}(|T^*|)} \tag{3.31}$$

  where the minimum and maximum flux values are defined using two different LUTs:

- $\lambda_{min}(|T^*|)$ is obtained from the 1D LUT that stores the relation between flux and torque along the Current Limit (CL) and the *MTPV* locus. This LUT is the inverse of the one previously used in Section 3.2.3 to define the torque limit value T_{lim}, depending on flux amplitude in the machine, in the same condition. Given the current reference torque, the extreme value of flux are correctly evaluated to obtain the p.u. value valid for interpolation in the load angle control map.
- $\lambda_{max}(|T^*|)$ is obtained from the 1D LUT that stores the relation between flux and torque along the the *MTPA* locus. In this condition the flux in the machine is at its maximum value, representing the upper value that can be reached with the current torque value. This 1D LUT is exactly the same used in Section 3.2.2, where, knowing the current torque reference value T^*, the correspondent flux along *MTPA* locus is obtained and used as the starting point from which the exact value is defined.

An example of the content of this LUTs is in Figure 3.15 for a SM with magnets and for an AM.



(a) *IPM* (b) *IM*

Figure 3.15: $T(\lambda_s)$ along *MTPV+CL* of IPM and SPM

- **Torque reference in pu T^*_{pu}**

  The process is this case is way more easier: T^*_{pu} is the absolute of the ratio between the actual value of reference torque T^* and its maximum T_{max}, as expressed in Equation 3.32.

$$T^*_{pu} = \frac{T^*}{T_{max}} \tag{3.32}$$

The resulting value is in the range [0,1]. The value that is used as T_{max} is the extreme point of *MTPA*, corresponding to the maximum overload torque of the machine.

These normalization produce $(\lambda^*_{s,pu}, T^*_{pu})$ values that are the input of the load angle control map. Using a function able to read 2D LUT, obtained by the writer, as the 1D-version, from a function archive, the absolute value of correspondent load angle $|\delta^*|$ is evaluated.

Being the flux amplitude used as d-axis reference value a naturally absolute value, to obtain from the machine the desired torque value, the sign of reference torque should be applied to load angle, q-axis reference value. To do this the sign of torque is evaluated, using a simply function, and applied to load angle to obtain, in the end, δ^*.

Finally, the reference value of the FPC control are know: (λ^*_s, δ^*).

3.2.6 Polar flux regulators

The two regulators adopted to define the values of $\{v^*_{ds}, v^*_{qs}\}$, represent the main advantage of all the proposed structure. They are at all insensitive to the working point, and so, to the machine type: given the switching frequency of the adopted power converter, the bandwidth of the regulators must be sufficiently lower. This is the only real imposition that must be satisfy, nothing else must be taken into account.

The "Polar flux regulators" scheme, highlighted with the letter ***e*** in Figure 3.2 is detailed in Figure 3.16 here showed.

The control loop are base on the voltage equation of the machine in (dq_s) coordinates. For all type of machines the (dq_s) frame corresponds to the one in which the stator flux linkage vector $\bar{\lambda}_s$ is aligned with the d_s-axis of the rotational frame. In this case the voltage equation in time-domain, valid for both AM and SM, are again showed:

$$\begin{cases} v_{ds} = R_s i_{ds} + \dfrac{d}{dt}\lambda_s & (3.33) \\ v_{qs} = R_s i_{qs} + \lambda_s \dfrac{d}{dt}\delta + p\omega_m \lambda_s & (3.34) \end{cases}$$

The d_s-axis control loop is the one that manage the variation of flux amplitude, while the q_s-axis one, regulate the load angle value. Moving in Laplace domain the Equation become:

$$\begin{cases} V_{ds} = R_s I_{ds} + s\lambda_s & (3.35) \\ V_{qs} = R_s I_{qs} + \lambda_s s\delta + p\omega_m \lambda_s & (3.36) \end{cases}$$

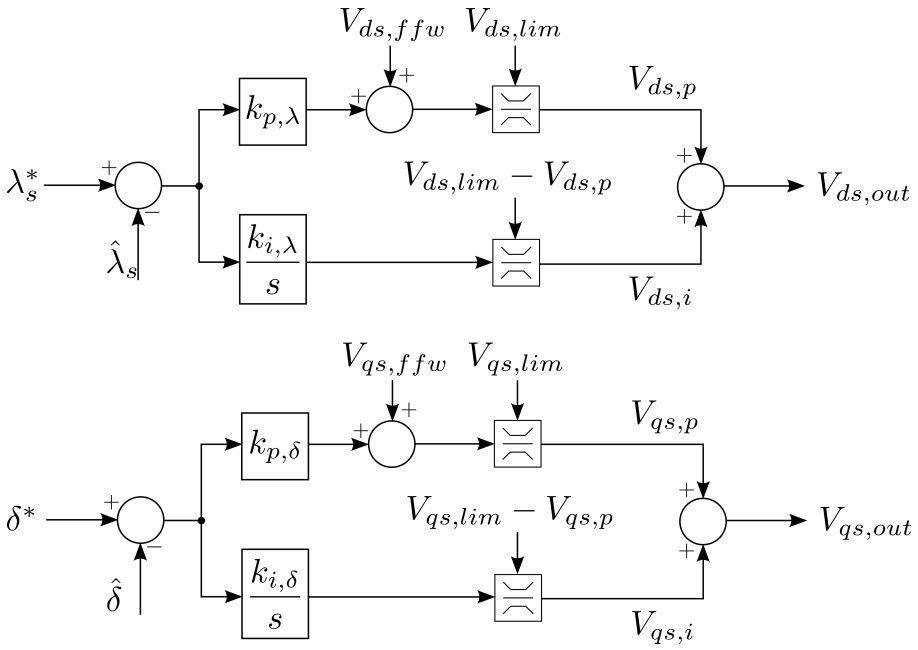


Figure 3.16: (dq_s) reference voltages generation in FPC scheme

In discrete-time, the derivative term can be replaced by its equivalent discrete form:

$$\begin{cases} V_{ds} = R_s I_{ds} + [\lambda_s(k) - \lambda_s(k-1)] f_s & (3.37) \\ V_{qs} = R_s I_{qs} + \lambda_s [\delta(k) - \delta(k-1)] f_s + p\omega_m \lambda_s & (3.38) \end{cases}$$

Given this couple of equation, the controller can be properly set to manage the flux and load angle errors in order to produce the reference values of voltages necessary to null them.

As can be seen in Figure 3.16 the input of the two controllers are the difference between the reference value of flux and load angle and their feedback, that are the actual values, obtained by the Flux Observer structure. The two PI (Proportional Integral) regulator provide the (V_{ds}^*, V_{qs}^*). Let's consider a generic variable u(t) in discrete form U(k). The desired value is the one of the actual step $U^* = U(k)$ while the feedback is the value at the previous time step $U_{fbk} = U(k-1)$.

Usually, to improve the behavior of the regulator, a feed-forward term is introduced to properly set the output. By adding this term, even when the state variable does not change between two consecutive discrete step, the output of the regulator is evaluated correctly, taking into account the residual term of the state equation.

In our case, the two feed-forward term are:

$$\begin{cases} V_{ds,ffw} = R_s I_{ds} & (3.39) \\ V_{qs,ffw} = R_s I_{qs} + p\omega_m \lambda_s & (3.40) \end{cases}$$

where the d-axis take into account only the resistive voltage drop that is usually negligible for high-power machine, while the q-axis, beyond the resistive voltage drop term, has the back EMF produced by the machine, which rotates at ω_m.

The maximum limit for the output of the regulator are set, exploiting the voltage unbalance between the two axis. This limits are:

$$\begin{cases} V_{ds,lim} = R_s I_{max} & (3.41) \\ V_{qs,lim} = \sqrt{\frac{V_{dc}^2}{3} - V_{ds,out}^2} & (3.42) \end{cases}$$

The output of the two PI regulators are build as follow.

d-axis _ flux regulator:

$$\begin{cases} V_{ds,p}(k) = [\lambda_s(k) - \lambda_s(k-1)] \cdot k_{p,\lambda} + V_{ds,ffw} & \text{sat: } |V_{ds,p}| < V_{ds,lim} \\ V_{ds,i}(k) = V_{ds,i}(k-1) + [\lambda_s(k) - \lambda_s(k-1)] \cdot k_{i,\lambda} & \text{sat: } |V_{ds,i}| < (V_{ds,lim} - V_{ds,p}) \\ V_{ds,out}(k) = V_{ds,p}(k) + V_{ds,i}(k) & \end{cases}$$

q-axis _ load angle regulator:

$$\begin{cases} V_{qs,p}(k) = [\delta(k) - \delta(k-1)] \cdot k_{p,\delta} + V_{qs,ffw} & \text{sat: } |V_{qs,p}| < V_{qs,lim} \\ V_{qs,i}(k) = V_{qs,i}(k-1) + [\delta(k) - \delta(k-1)] \cdot k_{i,\delta} & \text{sat: } |V_{qs,i}| < (V_{qs,lim} - V_{qs,p}) \\ V_{qs,out}(k) = V_{qs,p}(k) + V_{qs,i}(k) & \end{cases}$$

The proposed PI regulators are quite standard. Except for the limits, which must be choose according to the controlled variables, the other main parameters are the gain of the two controllers. Both have a proportional and an integral gain that have to be properly identified, given the desired performance.

To understand how these gain should be set, the state equations can be written as:

$$\begin{cases} \lambda_s = \frac{V_{ds} - R_s I_{ds}}{s} & (3.43) \\ \delta = \frac{1}{\lambda_s} \cdot \frac{V_{qs} - R_s I_{qs} - p \cdot \omega_m \cdot \lambda_s}{s} & (3.44) \end{cases}$$

As can be seen, the dynamic behavior of the controlled variables is not limited by any time constant, as happens for current loop in which the stator time constant always appears

in the equation. For this reason the tuning of the proportional parts of the regulators is only limited by the need to maintain an adequate distance (at least 10 times) between the loop band and the control band.

Thus, the tuning of the gain is the following:

$$\begin{cases} k_{p,\lambda} = 2 \cdot \pi \cdot f_{bw} & k_{i,\lambda} = \dfrac{2 \cdot \pi \cdot f_{bw}}{N} \quad (3.45) \\ k_{p,\delta} = k_{p,\lambda} \cdot \hat{\lambda_s} & k_{i,\delta} = k_{i,\lambda} \cdot \hat{\lambda_s} \quad (3.46) \end{cases}$$

No machine-dependent parameters appears in the tuning rule: they can always be left the same for each type of machine on which the control is applied. The gains used to adjust the load angle must be adapted at every period of execution of the control code, following the evolution of flux. In fact, the correction adopted is the current amplitude of flux $\hat{\lambda_s}$, obtained from the flux observer structure.

3.2.7 Voltage reference and duty cycle generation

The last step to be performed is the evaluation of the v^*_{abc} reference values that are needed to calculate the duty cycles for the inverter. From the reference in (dq_s) stator frame a rotation transform is performed to get the $(\alpha\beta)$ values and, in the end, the Clarke transform is used to move in stator frame (abc). The v^*_{abc} value are used in the generation of the duty cycles d^*_{abc}, using the chosen technique, and translated in instantaneous command to the inverter control section.

The "Voltage reference generation" scheme, highlighted with the letter ***f*** in Figure 3.2 is detailed in Figure 3.17 here showed.

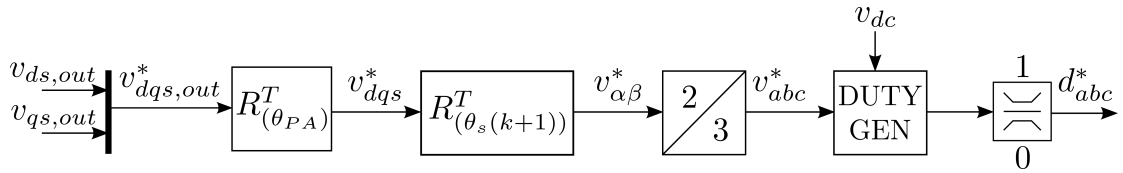


Figure 3.17: Duty cycles generation from reference voltage

The input of the last part of control structure are the output of the two controller performing flux and load angle regulation: $(v_{ds,out}, v_{qs,out})$. To be able to control the machine, the duty cycles d^*_{abc} must be evaluated to produce the command for the inverter. In this case, the main problem is to understand which angle should be used for the rotational transform from (dq_s) to $(\alpha\beta)$.

Using the current position, at time step (k), for the rotation of commands to apply at time step $(k + 1)$,is incorrect. In this way, we are evaluating the command to apply

to the next step, but using the same reference that we already use for the current one. The problem is not only philosophic, but also physics. While we are running our control code, the motor runs at ω_m and so are the rotational frames and electric quantities that we can possibly use for the control.

The use of θ_s from (dq_s) to $(\alpha\beta)$ represents an alignment error with the moving frame, thus an error on references.

Another cause of error is the delay introduced by the power converter to apply the exact value of reference voltages to the machine. A delay of half a switching period is always present between the production of the command and the application on the real system.

The solution adopted to prevent these error is made of two consecutive rotational transform: one to prevent the power converter delay and the other to take into account the actuation one.

By exploiting the Eulerian integration, the $(VI\theta)$ flux observer is able to predict the vector of flux at next sampling time $\hat{\bar{\lambda}}_{s,\alpha\beta}(k+1)$, thus enabling the knowledge of the position of (dq_s) reference at time $(k+1)$. Performing (3.27) with this forecasted flux vector, produce the sine and cosine of the position of the flux vector at the next step $(\cos(\theta_s(k+1)),\sin(\theta_s(k+1)))$. Using this angle instead of $\theta_s(k)$ for rotational transform of reference voltage values between (dq_s) and $(\alpha\beta)$, improves the control performance, especially at high speed. In fact, in this way we can know where the (dq_s) frame is correctly positioned when the commands are actually applied to the machine, taking into account part of the implementation delay.

This cause a major effect: the phase-advancing technique that is used to compensate the delay caused of the power converter and the actuation of command by the digital control, has only a coefficient of "0.5". This is useful to compensate the PWM delay of power converter, while the actuation one is already canceled by the use of $\theta_s(k+1)$. Otherwise, without this care, the coefficient must be "1.5" to compensate both the cited phenomena, like in FOC-based control strategy.

The phase advancing angle is evaluated as the angular space covered by the rotating system in half a switching period T_s:

$$\theta_{PA} = 0.5 \cdot T_s \cdot \omega_s \tag{3.47}$$

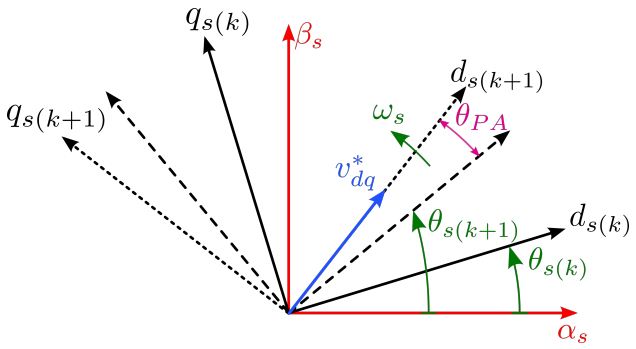


Figure 3.18: Predictive behavior of flux observer and phase-advancing

Taking advantage of these information, the $(v_{ds,out}, v_{qs,out})$ values undergo two rotational transform. The former is done using as rotation angle θ_{PA} to prevent the power converter delay, while the latter uses $\theta_s(k+1)$ to take into account the actuation delay. At the end, with these precautions, the (v^*_α, v^*_β) values applied at time step $(k+1)$ should have been correctly evaluated.

$$\begin{bmatrix} v^*_\alpha \\ v^*_\beta \end{bmatrix} = \begin{bmatrix} \cos(\theta_s(k+1)) & -\sin(\theta_s(k+1)) \\ \sin(\theta_s(k+1)) & \cos(\theta_s(k+1)) \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_{PA}) & -\sin(\theta_{PA}) \\ \sin(\theta_{PA}) & \cos(\theta_{PA}) \end{bmatrix} \cdot \begin{bmatrix} v_{ds,out} \\ v_{qs,out} \end{bmatrix} \tag{3.48}$$

Finally, the v^*_{abc} are calculated using the inverse Clarke transform:

$$\begin{bmatrix} v^*_a \\ v^*_b \\ v^*_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\sin(30) & \cos(30) & 1 \\ -\sin(30) & -\cos(30) & 1 \end{bmatrix} \cdot \begin{bmatrix} v^*_\alpha \\ v^*_\beta \end{bmatrix} \tag{3.49}$$

Lastly, the duty cycles that are used to define the instantaneous states of power converter must be obtained from the current voltage reference in (abc) and from the v_{dc} link voltage value. They can be computed using any of the pulse-width modulation (PWM) techniques showed in the literature [12]. The adopted one is the carrier-based 'MinMax' modulation technique. This technique guarantees, in linear condition, a peak voltage output of $v_{dc}/\sqrt{(3)}$, that is $\sqrt{(3)}/2 = 0.866$ higher then the value reachable without the zero-sequence injection, equal to $v_{dc}/2$). The comparison between the two modulation is in Figure 3.19 both in term of peak voltage and obtained output duty cycles. The ones with 'MinMax' have the characteristic double hump, a consequence of common mode injection.

Given the v^*_{abc} values, the instantaneous value of common-mode voltage is calculated at every ISR call. This represent the zero-sequence component that must be injected in the inverter output voltage in order to exploit at its best the available v_{dc} link voltage.

The zero sequence voltage v_{zs} can be computed as the average between the instantaneous maximum and minimum voltages, or half of the middle voltage, between the desired three. In every case, v^*_{abc} are normalized over v_{dc} to obtain $v^*_{abc,norm}$ and then sorted, to correctly attribute these three values.

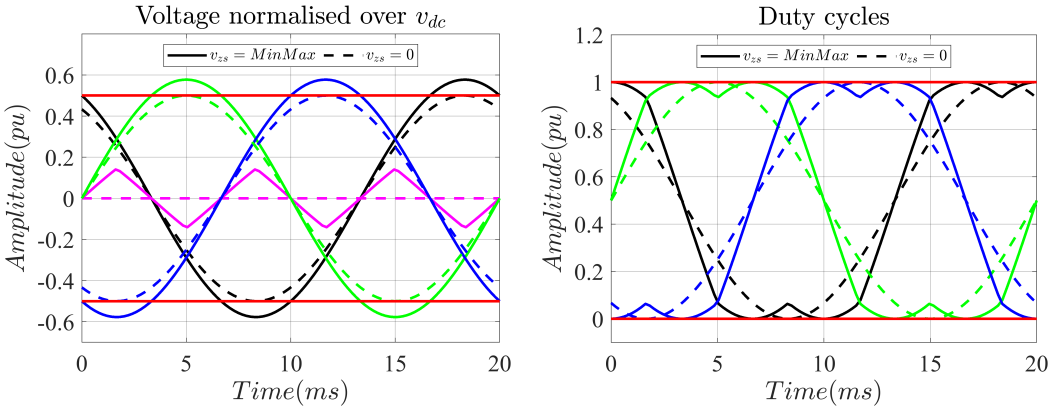
$$v_{zs} = -\frac{v_{max}(v^*_{a,norm}, v^*_{b,norm}, v^*_{c,norm}) + v_{min}(v^*_{a,norm}, v^*_{b,norm}, v^*_{c,norm})}{2}$$

$$v_{zs} = \frac{v_{mean}(v^*_{a,norm}, v^*_{b,norm}, v^*_{c,norm})}{2} \tag{3.50}$$

The duty cycles d^*_{abc} are calculated:

$$\begin{aligned} d_a &= \frac{1}{2} + v^*_{a,norm} + v_{zs} \\ d_b &= \frac{1}{2} + v^*_{b,norm} + v_{zs} \\ d_c &= \frac{1}{2} + v^*_{c,norm} + v_{zs} \end{aligned} \tag{3.51}$$

In any case, to avoid the duties exceed the extremes [0,1] a saturation between these two values is performed before using them.



(a) *Reference and common mode voltages* (b) *Duty cycles*

Figure 3.19: Comparison between maximum voltage reachable in linear range with and without zero sequence injection

3.3 Control Maps for FPC

In the Flux Polar Control scheme several LUTs, both 1D that 2D, are used for the evaluation of reference variables that are directly used for the control or to define limits. In the scheme in Figure 3.20 is highlighted all the time that one LUT is used and, in the following, is explained how all the different LUTs are created to manage the system while it is running. The amount or the size of the adopted LUTs is not so much different from the one in other conventional approaches [1],[2], thus the adopted control strategy, from this point of view, is not critical.

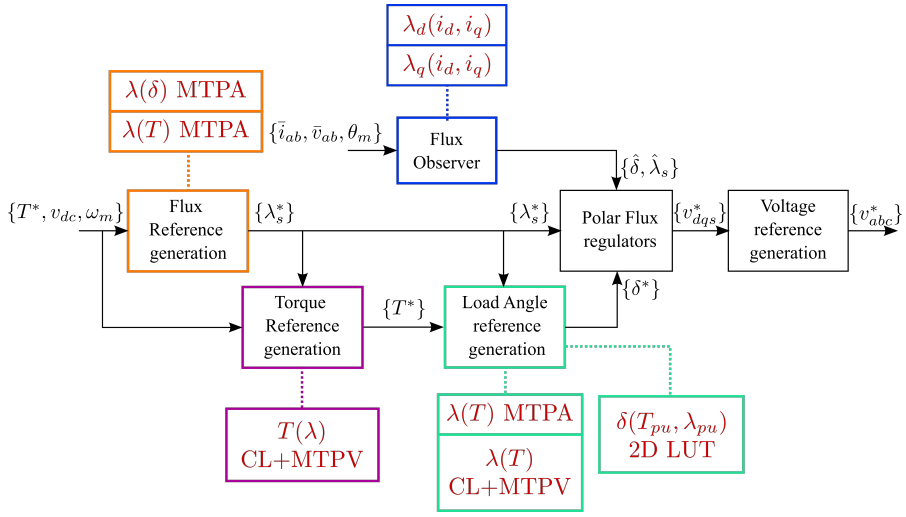


Figure 3.20: Detail of LUT used in FPC scheme

The starting point from which all control limit and maps are created, are the machine direct flux maps. As explained in Section 2.1.4 for AM and in Section 2.2.3 for SM, the flux maps are conventional maps usually build from data obtained in experimental procedure [8]. Thus this control does not require extra experimental procedure to obtain specific set of data, but simply using the classical and well known flux maps, it is able to exploit all the information contained and using it for control purpose.

The first action to be taken to obtain precise control map, is the extension of Direct Maps to a size that makes them sufficiently accurate and detailed. As this maps are the product of an experimental procedure, they can not be as detailed, in the beginning, as this would require a massive testing phase, which can not be handled easily and takes a long time to be perform. The solution is to carry out an experimental test which points cover all the desired map area, in our case in term of currents coordinates, and are as close as allow the accuracy-time trade-off, from each other. To improve the density of point, so the accuracy of the map, a post-experimental procedure is performed on the initial flux

maps. This is done in Matlab environment, taking advantage of some functions offered by the product.

The desired dimension of the final map, is set choosing the parameter N_{map}. From the original one, the current extreme in d-axis and q-axis are found and used to create a new equally-spaced vectors with the same extremes as the original ones but more dense, with dimension N_{map}. To reproduce the starting map, but with a high level of detail, the original relation must be known and used as rule to create, with the newer current vector, the high-definition map. Two rule are created, one for the d-axis flux and the other for the q-axis one. The Matlab function used is `griddedInterpolant`, in its 2D version: it receives a grid of regular point defined by current with coordinates (i_d, i_q) , that are the base on which the map is build, and original direct map, that has the flux information. This function require also to choose what kind of interpolation and extrapolation methods should be used to link the data: the former is for data that are in the range, while the latter for data out-of-range. All the information about this function can be easily found in the Help of Matlab. Once the rules are ready, the new maps can be created adopting it.

An example of code is here showed.

```
IntD = griddedInterpolant(Id_old,Iq_old,Fd','spline','spline');
IntQ = griddedInterpolant(Id_old,Iq_old,Fq','spline','spline');
Fd = IntD(Id_new,Iq_new)';
Fq = IntQ(Id_new,Iq_new)';
```

The same procedure can be run with the Indirect Flux Maps. The original grid, in this case, is based on flux information. The extreme of both component of fluxes are found and used to create the dense version of vector with dimension N_{map}. The rule that links current to flux is created and, using the new vector, the high-density maps are obtained.

```
IntD = griddedInterpolant({fD_old,fQ_old},iD','spline','spline');
IntQ = griddedInterpolant({fD_old,fQ_old},iQ','spline','spline');
iD = IntD({fD_new,fQ_new})';
iQ = IntQ({fD_new,fQ_new})';
```

3.3.1 Torque map

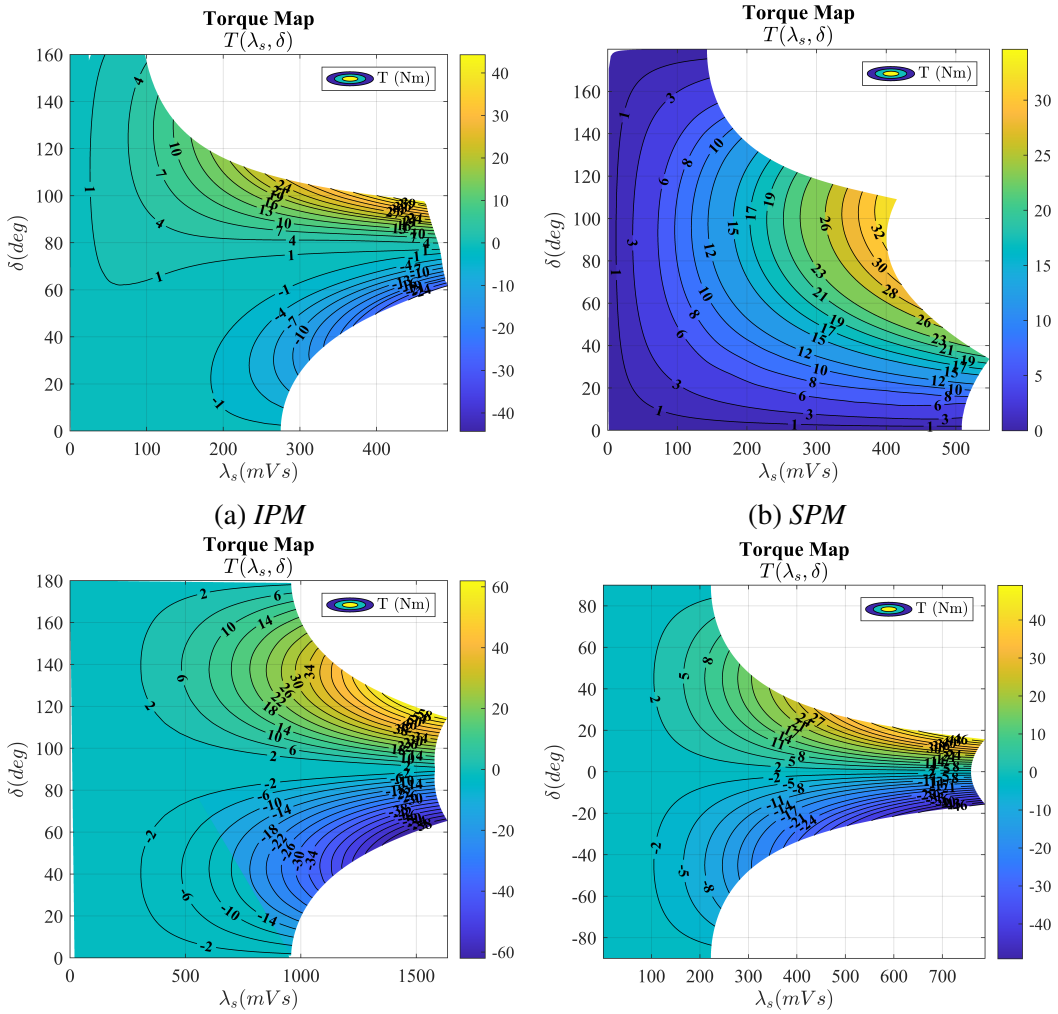
First map that can be directly created from direct map is the Torque Map of the machine. It represents the torque characteristic that can be obtained from the machine for each possible combination of flux and load angle. The equation that is used to calculate

the torque is the one generally adopted in (dq) coordinates, see Equation 2.75 and uses only information available in Direct Flux Map.

```
Te = 3/2 * p * (Fd.*Iq - Fq.*Id);
```

where "p" is the number of pole pairs of the machine and `Fd,Fq,Id,Iq` are the matrices containing the direct flux map and the current grid.

The torque map of the four motor used for the validation of the control are in Figure 3.21



(a) *IPM* (b) *SPM*

(c) *SYR* (d) *IM*

Figure 3.21: Torque maps of IPM, SPM, SYR and IM machines

3.3.2 MTPA and MTPV locus

Having the flux map enable the creation of the preliminary control limits, i.e. MTPA and MTPV locus, that must be used in the control of the machine.

MTPA locus

The MTPA (Maximum Torque Per Ampere) represent the set of points both on the (i_d, i_q) or on the (λ_d, λ_q) plane that should be followed in order to get the highest torque from the machine given the current amplitude imposed into the machine. Formally MTPA locus is obtained from the derivative of the torque expression with respect to the angle of the current vector γ.

Torque expression of induction motor in Equation (2.44) in (dq) coordinates becomes:

$$\begin{aligned} T &= \frac{3}{2} p (L_s i_d i_q - \sigma L_s i_q i_d) \\ &= \frac{3}{2} p (L_s - \sigma L_s) i_d i_q \\ &= \frac{3}{2} p L_s (1 - \sigma) |I| \cos(\gamma) |I| \sin(\gamma) \\ &= \frac{3}{2} p L_s (1 - \sigma) |I|^2 \cos(\gamma) \sin(\gamma) \\ &= \frac{3}{2} p L_s (1 - \sigma) |I|^2 \frac{1}{2} \cos(2\gamma) \end{aligned} \tag{3.52}$$

To obtain the maximum value of torque given the amplitude of stator current, the derivative of the torque expression with respect to the current angle γ must be set to zero.

$$\left. \frac{\partial T}{\partial \gamma} \right|_{I=cost} = \frac{\partial}{\partial \gamma} \left[\frac{3}{2} p L_s (1 - \sigma) |I|^2 \frac{1}{2} \cos(2\gamma) \right] = 0 \tag{3.53}$$

The same procedure can be applied to SM, starting from torque expression in Equation (2.76):

$$\begin{aligned} T &= \frac{3}{2} \cdot p \cdot [(L_{dd} - L_{qq}) \cdot i_d \cdot i_q + |\bar{\lambda}_m| \cdot i_q] \\ &= \frac{3}{2} \cdot p \cdot [(L_{dd} - L_{qq}) \cdot |I| \cos(\gamma) |I| \sin(\gamma) + |\bar{\lambda}_m| \cdot |I| \sin(\gamma)] \\ &= \frac{3}{2} \cdot p \cdot \left[(L_{dd} - L_{qq}) \cdot |I|^2 \frac{1}{2} \cos(2\gamma) + |\bar{\lambda}_m| \cdot |I| \sin(\gamma) \right] \end{aligned} \tag{3.54}$$

The derivative with respect to current angle is:

$$\left.\frac{\partial T}{\partial \gamma}\right|_{I=cost} = \frac{\partial}{\partial \gamma}\left[\frac{3}{2}\cdot p\cdot\left((L_{dd}-L_{qq})\cdot|I|^2\frac{1}{2}\cos(2\gamma)+|\bar{\lambda}_m|\cdot|I|\sin(\gamma)\right)\right]=0 \quad (3.55)$$

Once the value of current angle γ_{MTPA} that represent the *MTPA* locus is obtained from (3.53) and (3.55), knowing the amplitude of current, the components $\{i_d, i_q\}_{MTPA}$ are simply retrieve using the sine and cosine of the angle.

MTPV locus

The *MTPV* (Maximum Torque Per Volt) represent the set of points both on the (i_d, i_q) or on the (λ_d, λ_q) plane that should be followed in order to get the highest torque from the machine given the flux amplitude imposed, i.e., the voltage available from the supply. Formally *MTPV* locus is obtained from the derivative of the torque expression with respect to the angle of the flux vector δ, using quantities in (dq) rotor reference. For SM, torque equation must be express as a function of the flux, retrieving(i_d, i_q) current expression functions of (λ_d, λ_q) fluxes from (2.28) and (2.29) and substituting in (2.44):

$$\begin{aligned}
T &= \frac{3}{2}p\left(\lambda_d\frac{\lambda_q}{\sigma L_s}-\lambda_q\frac{\lambda_d}{L_s}\right)\\
&= \frac{3}{2}p\left(\frac{1}{\sigma L_s}-\frac{1}{L_s}\right)\lambda_d\lambda_q\\
&= \frac{3}{2}p\left(\frac{1}{\sigma L_s}-\frac{1}{L_s}\right)|\lambda|\cos(\delta)|\lambda|\sin(\delta)\\
&= \frac{3}{2}p\left(\frac{1}{\sigma L_s}-\frac{1}{L_s}\right)|\lambda|^2\cos(\delta)\sin(\delta)\\
&= \frac{3}{2}p\left(\frac{1}{\sigma L_s}-\frac{1}{L_s}\right)|\lambda|^2\frac{1}{2}\cos(2\delta)
\end{aligned} \quad (3.56)$$

To obtain the maximum value of torque given the amplitude of flux the derivative of torque expression with respect to flux angle δ must be set to zero.

$$\left.\frac{\partial T}{\partial \delta}\right|_{\lambda=cost} = \frac{\partial}{\partial \delta}\left[\frac{3}{2}p\left(\frac{1}{\sigma L_s}-\frac{1}{L_s}\right)|\lambda|^2\frac{1}{2}\cos(2\delta)\right]=0 \quad (3.57)$$

The same can be performed with torque expression of SM in Equation (2.76), substituting current with its expression in term of flux, Equation (2.65) and (2.66) neglecting the cross terms:

$$
\begin{aligned}
T &= \frac{3}{2} \cdot p \cdot \left(\lambda_d \cdot \frac{\lambda_q}{L_{qq}} - \lambda_q \cdot \frac{\lambda_d - \lambda_m}{L_{dd}} \right) \\
&= \frac{3}{2} \cdot p \cdot \left[\left(\frac{1}{L_{qq}} - \frac{1}{L_{dd}} \right) \cdot \lambda_d \lambda_q + \lambda_q \cdot \frac{\lambda_m}{L_{dd}} \right] \\
&= \frac{3}{2} \cdot p \cdot \left[\left(\frac{1}{L_{qq}} - \frac{1}{L_{dd}} \right) \cdot |\lambda| \cos(\delta) |\lambda| \sin(\delta) + |\lambda| \sin(\delta) \cdot \frac{\lambda_m}{L_{dd}} \right] \\
&= \frac{3}{2} \cdot p \cdot \left[\left(\frac{1}{L_{qq}} - \frac{1}{L_{dd}} \right) \cdot |\lambda|^2 \frac{1}{2} \cos(2\delta) + |\lambda| \sin(\delta) \cdot \frac{\lambda_m}{L_{dd}} \right]
\end{aligned}
\tag{3.58}
$$

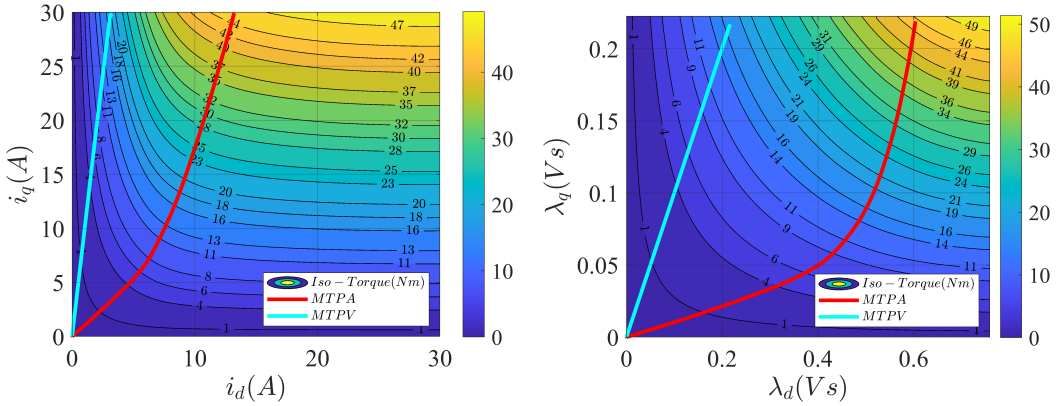
The maximum value of torque given the amplitude of flux, i.e. the MTPV locus, is the derivative of torque expression with respect to flux angle δ, set to zero.

$$
\left. \frac{\partial T}{\partial \delta} \right|_{\lambda = cost} = \frac{\partial}{\partial \delta} \left[\frac{3}{2} \cdot p \cdot \left[\left(\frac{1}{L_{qq}} - \frac{1}{L_{dd}} \right) \cdot |\lambda|^2 \frac{1}{2} \cos(2\delta) + |\lambda| \sin(\delta) \cdot \frac{\lambda_m}{L_{dd}} \right] \right] = 0 \tag{3.59}
$$

Once the value of flux angle δ_{MTPV} that represent the MTPV locus is obtained from (3.57) and (3.59), knowing the amplitude of flux, the components $\{\lambda_d, \lambda_q\}_{MTPV}$ are simply retrieve using the sine and cosine of the angle.

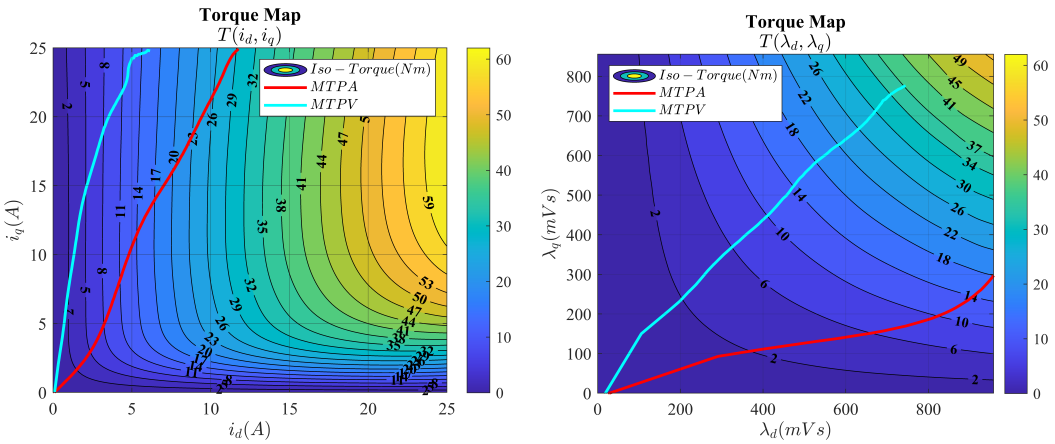
The *MTPA* and *MTPV* locus of all the machines used for simulation are showed from Figure 3.22a to Figure 3.25b. To represent these control locus both on (i_d, i_q) and (λ_d, λ_q) planes, experimental direct and inverse flux maps are adopted, to obtain $\{\lambda_d, \lambda_q\}_{MTPA}$ from $\{i_d, i_q\}_{MTPA}$ and $\{i_d, i_q\}_{MTPV}$ from $\{\lambda_d, \lambda_q\}_{MTPV}$.

The results obtained for different kind of machines allow an expert user to understand which type is, simply by its *MTPA/MTPV* profiles.



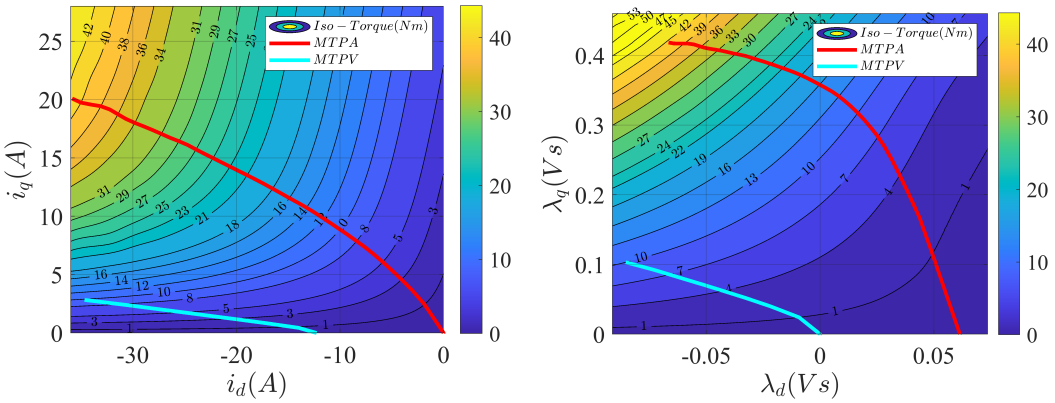
(a) $(i_d, i_q) plane$ (b) $(\lambda_d, \lambda_q) plane$

Figure 3.22: MTPA and MTPV locus of Induction Motor



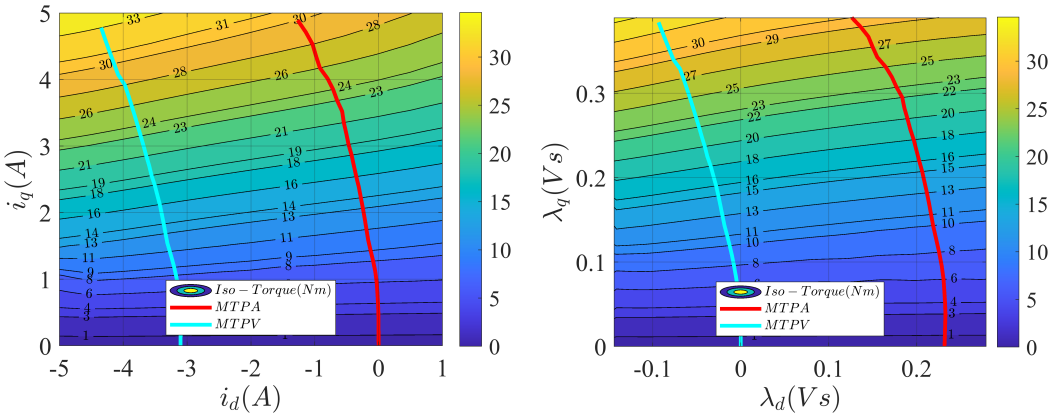
(a) $(i_d, i_q)plane$ (b) $(\lambda_d, \lambda_q)plane$

Figure 3.23: MTPA and MTPV locus of Synchronous Reluctance Motor



(a) $(i_d, i_q)plane$ (b) $(\lambda_d, \lambda_q)plane$

Figure 3.24: MTPA and MTPV locus of Internal Permanent Magnets Motor



(a) $(i_d, i_q)plane$ (b) $(\lambda_d, \lambda_q)plane$

Figure 3.25: MTPA and MTPV locus of Surface Permanent Magnet Motor

The distances of the *MTPA* locus from the q-axis, expressed as γ_{MTPA}, change following the type of machine. As the percentage of magnets in the machine decreases, i.e. with the increase of anisotropy contribution to torque, the *MTPA* locus tend to move. From a position close to the q-axis typical of motor with a high percentage of torque from magnets, it rotates close to the negative d-axis when the considered motor has no more torque from magnets but only from the anisotropy structure.

The starting point of the *MTPA* locus on the (i_d, i_q) plane is always the zero-current point that corresponds to a λ_d equal to the magnet flux λ_m, if present, on the (λ_d, λ_q) plane. If not different specified, the end corresponds with the end of the map. Usually a Current Limit (typical of the machine or of the power converter) is settle, imposing the extreme value of current for which the *MTPA* locus must be create.

On the contrary, the ending point of the *MTPV* locus on (λ_d, λ_q) plane is always the zero-flux point (extreme of flux weakening region with no more flux in the machine) that corresponds to a i_d equal to the characteristic current of the machine $i_{d,sc}$ that is the one able to null the magnet flux of the machine, if present:

$$i_{d,sc} = \frac{\lambda_m}{L_{eq,d}} \tag{3.60}$$

For the *MTPV*, the starting point can be either the extreme of the map or the limit imposed by Current Limit, that, on the current plane, is a circumference centered in the origin of axis. The Current Limit (CL) cuts both the *MTPA* and *MTPV* locus defining a transition between the end of *MTPA* and the start of *MTPV* proceeding with increasing speed.

The *MTPA* and *MTPV* locus, together with the CL that is the thermic limit, are used to define the operation region of the motor, that is the exploitable area in which the working point can be placed for stable machine operation.

Code for generation

The procedure adopted to build up the *MTPA* and *MTPV* locus, starting from the flux maps of the machine, has been carried out in Matlab exploiting some of its function. First, being the *MTPA* locus build on the intersection between the iso-torque curves and the iso-current circumferences on current plane, as represented in Figure 3.26a, the iso-torque curves must be found. To do this the Matlab function `contourc` is used: it receive as input the current grid on which the map has been crated, the map that must be analyzed and the number of levels that we want to found in the final output. This function is able to investigate the torque map to find the iso-level curves on it, and produce as output the group of point on the current plane (i_d, i_q, coordinates) that represent that curves. The output of the function is manipulated in order to dived the information related to each level: $(i_d, i_q)_{TO}$, $(i_d, i_q)_{T1}$, ... , $(i_d, i_q)_{Tn}$.

From the definition we know that a point of *MTPA* is found as the one with the highest torque, given the current amplitude. In other word it can be expressed as the point of the iso-torque curve that can be realized with the minimum value of current. Following the last definition, the amplitude of current is obtained from the square root of (dq) components squared. For each level the minimum is found (and its relative (dq) components too), which represents the $I_{s,MTPA} = I_{d,MTPA} + jI_{q,MTPA}$ on the current plane.

Using the 2D interpolation procedure, the correspondent point on the flux plane $F_{s,MTPA} = F_{d,MTPA} + jF_{q,MTPA}$ are found. The function needs the original data `(Id,Iq,Fd)` , stored in the high-definition Direct Map previously extended, and the coordinates of *MTPA* point `(Id_mtpa,Iq_mtpa)` , beside the method that should adopt. The amplitude of flux is so evaluated. The use of a Direct Map non sufficiently detailed influence the accuracy of the control locus just created.

Moreover the load angle value in *MTPA*, can be easily evaluated as the arc tangent of the (dq) flux components.

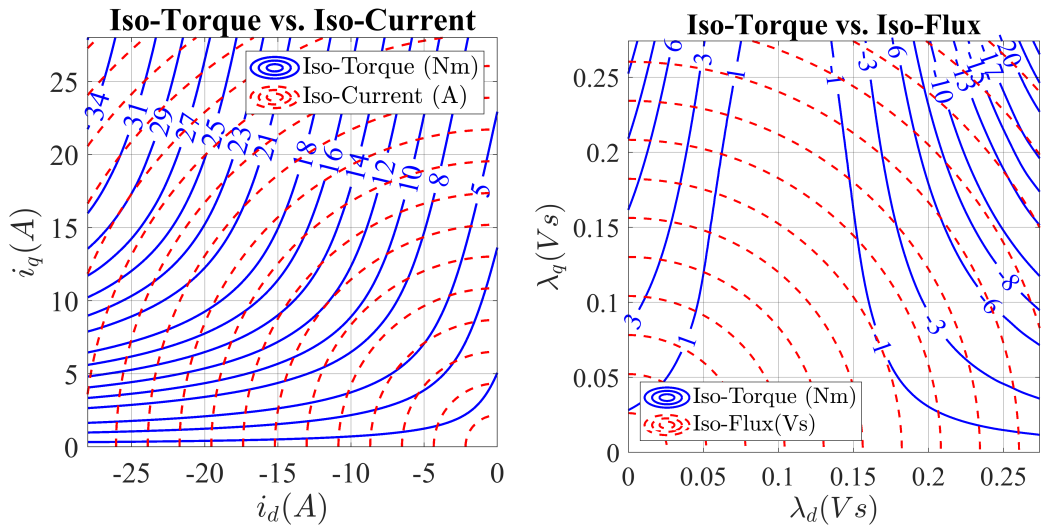
An extract of the salient part of the code is here showed:

```
T_iso = contourc(Id(1,:),Iq(:,1),Te,T_levels);
... Founding the coordinates for each level:
id_value = T_iso(1,((pointer+1):(step+pointer)));
iq_value = T_iso(2,((pointer+1):(step+pointer)));
... Evaluating the current amplitude:
is_value = hypot(id_value,iq_value);
... Finding the minimum for each level:
[ , index] = min(is_value,[],2);
```

```
Id_mtpa(i) = id_value(i,index(i));
Iq_mtpa(i) = iq_value(i,index(i));
Is_mtpa(i) = is_value(i,index(i));
... Evaluating the MTPA locus on flux plane:
Fd_mtpa = interp2(Id,Iq,Fd,Id_mtpa,Iq_mtpa,'spline');
Fq_mtpa = interp2(Id,Iq,Fq,Id_mtpa,Iq_mtpa,'spline');
Fs_mtpa = hypot(Fd_mtpa,Fq_mtpa);
... Evaluating the load angle value along MTPA:
La_mtpa = atan2(Fq_mtpa,Fd_mtpa);
```

For what concern the *MTPV* locus, the same procedure must be carry out, but this time on the flux plane using the information of Inverse Map. In fact, when looking for the *MTPV*, the intersection between iso-torque and iso-flux curves must be found to locate the maximum-torque per flux locus, as represented in Figure 3.26b. Also in this case the high-definition version of the Indirect Map is needed to create a decent *MTPV*. The significant part of the code is here showed:

```
T_iso = contourc(fD(1,:),fQ(:,1),tE,T_levels)
... Founding the coordinates for each level:
fd_value = T_iso(1,((pointer+1):(step+pointer)));
fq_value = T_iso(2,((pointer+1):(step+pointer)));
... Evaluating the flux amplitude:
fs_value = hypot(fd_value,fq_value);
... Finding the minimum for each level:
[ , index] = min(fs_value,[],2);
Fd_mtpv(i) = fd_value(i,index(i));
Fq_mtpv(i) = fq_value(i,index(i));
Fs_mtpv(i) = fs_value(i,index(i));
... Evaluating the MTPV locus on current plane:
Id_mtpv = interp2(fD,fQ,iD,Fd_mtpv,Fq_mtpv,'spline')
Iq_mtpv = interp2(fD,fQ,iQ,Fd_mtpv,Fq_mtpv,'spline')
Is_mtpv = hypot(Id_mtpv,Iq_mtpv);
... Evaluating the load angle value along MTPv:
La_mtpv = atan2(Fq_mtpv,Fd_mtpv);
```



(a) *Current plane* (b) *Flux plane*

Figure 3.26: Iso-torque on current and flux plane - IPM machine

3.3.3 Control Maps

The *MTPA* and the *MTPV*, together with the Direct Flux Map, are the base data needed to generate the machine Control Maps. This maps are used to define the value of variables of the control, following the constrains applied when the map are build, that represent the working constrains of the machine. Other conventional control solutions, like the LUT-based FOC ([1],[2]), require at least one 4D map or more 2D maps that represent the behavior of the machine in all the working point of interest for the application.

In our case the Control Map that will be created is only one 2D Load Angle Control Map that stores the load angle data in relation with torque and flux amplitude values. As already presented, it is used to define the load-angle value of the flux vector needed in accordance with the desired torque and flux.

The building procedure is carry out completely in Matlab environment, taking advantage of some function available in this software, like for the generation of the *MTPA* and *MTPV*.

The first operation that must be performed is, as done for *MTPA*, the generation of the iso-torque curves with respect to current values. The grid of current, the Torque Map and the number of desired torque levels are provided as input to the `contourc` function and the set of (i_d, i_q) points corresponding to different torque levels, is obtained as output. The first two map (`Id_cm` and `Iq_cm`) are created storing for each row the coordinates points of the specific torque level. From these, the correspondent variables can be evaluated:

- current amplitude as square root of the (dq) components squared (`Is_cm`) ;
- (dq) flux components, using the Direct Map information on current data (`Fd_cm` and `Fq_cm`) ;
- the amplitude of flux, from (dq) components (`Fs_cm`) ;
- the preliminary load angle map, as the arc tangent of the ratio between the q - and d -flux values (`La_cm`) .

The so obtained maps represent the behavior of the main variables of the machine in all possible working point defined up to the current extremes on the Direct Map. They are not useful for control purpose: they are hard to read, and, in the end, no significant information is stored inside them. Some more work is required to set the control map in a proper way.

Application of constraints

To obtain significant map, some constraints must be applied: the definition of the working quadrant, Current Limit, together with MTPA and MTPV limits.

- **Working quadrant**

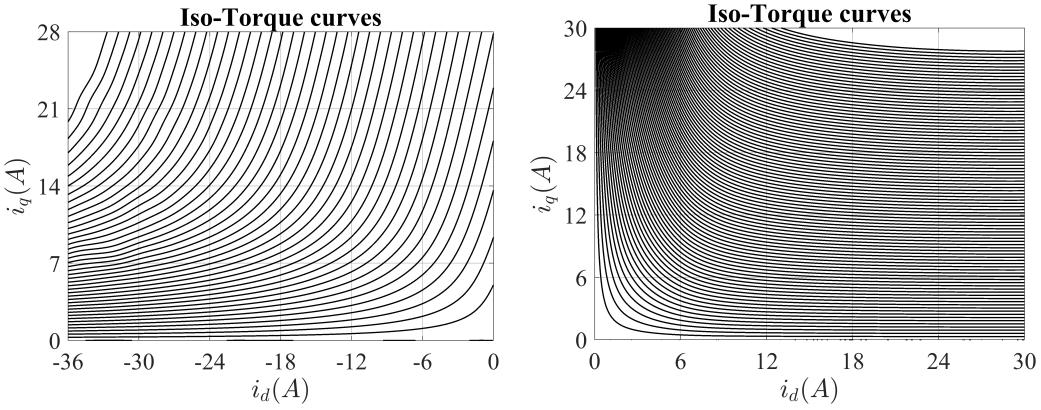
  The choice of working quadrant comes from the adopted convention. For SM, as exposed in Section 2.2.1, the adopted choice is to refer always to the same pair of (dq) rotor coordinates that are the one where the d-axis is aligned with the magnet position, even if it is not present, like in pure reluctance machine. From these choice all the *MTPA* and *MTPV* locus of the machine in Section 3.3.2 are placed in the second quadrant, with $(i_d < 0, i_q > 0)$ for positive torque value. To invert the torque, it is sufficient to reverse the q current component, moving to the third quadrant.

  In first place, this has been applied also to the synchronous reluctance machine, resulting into a uncontrolled behavior of it when torque approach zero Moreover the *MTPA* and *MTPV* locus has been obtained into the first quadrant, so, to overcome this problem a very easy solution has been found: for pure reluctance machine its conventional axis convention has been again adopted. The d-axis is aligned with the direction of minimum of reluctance, while the q-axis, in quadrature, with the direction of the flux barriers. In this way the working quadrant is the first with $(i_d > 0, i_q > 0)$ for positive torque values, while, when torque inverts, the working quadrant become the fourth, reversing the q-axis current.

For what concern the AM, the convention adopted is directly the one with positive torque in the first quadrant and negative in the third, as imposed by torque expression.

Summing up, for the machines here adopted: the ones with magnet, even in minimal percentage, (IPM and SPM) use the "PM style" convention and work in II°, III°quadrants, while the magnet-less ones (SyR), including the Induction machine (IM), in I°,VI°quadrants.This information permits to chose what quadrant of the machine save and and which to delete.

The first control boundary is applied removing from all the maps previously created, the positive (IPM and SPM) or negative (SyR and IM) part of d-axis current, working on `(Id_cm)` map. Moreover, whichever type of machine is treated, is reasonable reduce the area of interest on which the control map is created, to the positive side of the q-axis. In this way, the accuracy of the map is increased, exploiting for its use the specular behavior for positive and negative torque values. In Figure 3.27 the maps of IPM and IM after the application of the first constrain.



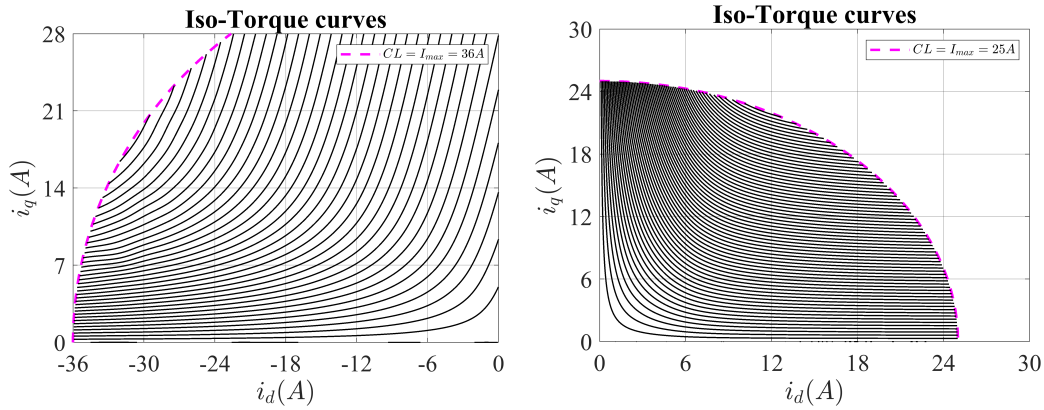
(a) *IPM* (b) *IM*

Figure 3.27: Control map build: working quadrant constrain

- **Current Limit**

  The second limit is a very common one: the Current Limit. It can represent both the thermal limit of the power converter or of the machine but, the choice of a power converter with maximum current limit lower that the one of machine, is unusual. The application of this limit is very easy. Given the map with current amplitude values `(Is_cm)` all the values out of the circumference defined by I_{max} value are unacceptable, and must be eliminated. The limit is again applied to all

maps. In Figure 3.28 the maps of IPM and IM after the application of the current limit constrain.



(a) *IPM* (b) *IM*

Figure 3.28: Control map build: current limit constrain

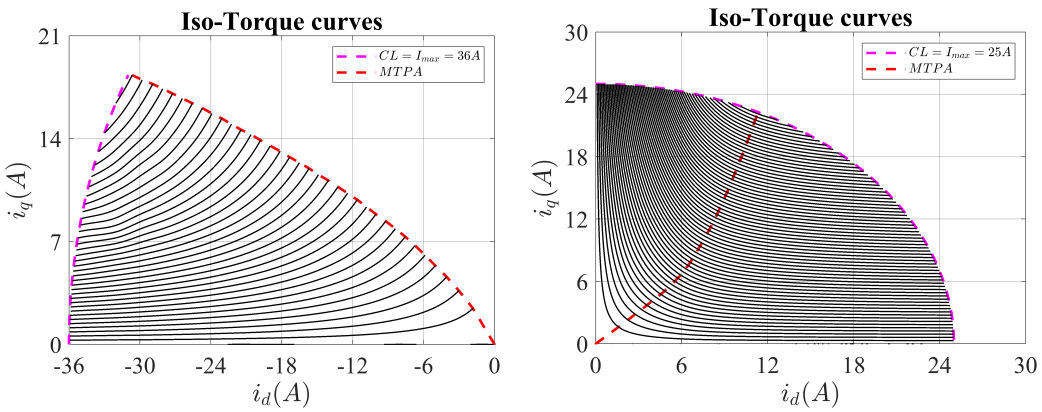
- **MTPA Limit**

  This limit is again a machine-dependent one. The *MTPA* is used to limit the amplitude of flux to a finite value, in whichever working condition: no flux value higher than the one in *MTPA* condition can be present in the machine. Here is the distinction: for (IPM and SPM) it is the fundamental boundary while for (SyR and IM) it is not. This is due to the convention.

  Following the "PM-style" convention, the *MTPA* locus is placed over the *MTPV*, thus having a flux $F_{s,MTPA}$ that is higher respect to $F_{s,MTPV}$. In this case it is a real limit for the operating region of the machine.

  For (SyR and IM) it is not a real problem. With the adopted convention the *MTPA* locus is the one close to d-axis, so its flux is certainly lower than the one of *MTPV*. The *MTPA* limitation in this case is unnecessary. On the contrary, letting all the area under the *MTPA* improves the dynamic performance of the machine in high speed working condition when performing rapid inversion of torque reference.

  To apply the *MTPA* limit where necessary, the flux amplitude maps `(Fs_cm)` is needed. Only the point with flux amplitude lower then the one along $F_{s,MTPA}$ are useful, while the others must be eliminated from all the maps. In Figure 3.29 the maps of IPM and IM after the application of the MTPA limit, where needed.



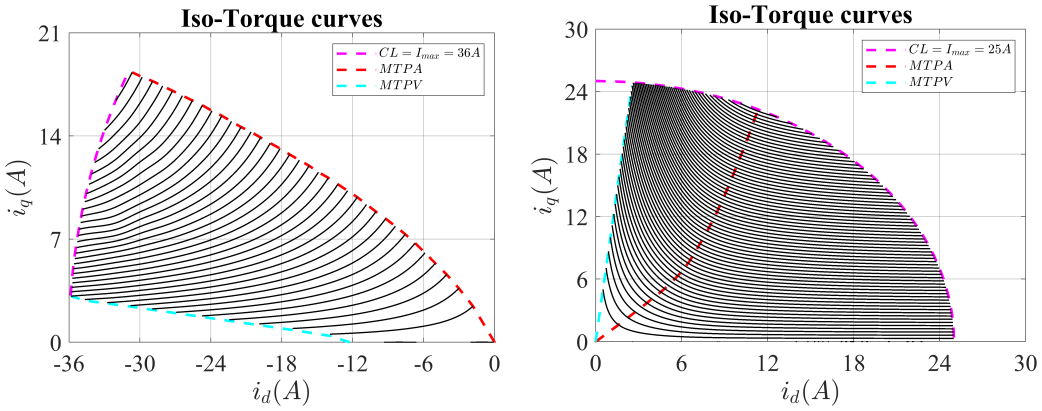
(a) *IPM* (b) *IM*

Figure 3.29: Control map build: MTPA limit

- **MTPV Limit**

  The last limit useful to build up the control map is the one of *MTPV*. All the point beyond this locus are point in which the loss of control of the machine increase dramatically, because here the flux amplitude is very limited. It is important to set properly this limit, in a way that we are sure that there is no working condition in which this barrier can be overcome.

  Both AM and SM are interested by this limit that is applied employing the `(Id_cm)` together with the $I_{d,MTPV}$. All the points with a d-axis current value lower then the ones along the *MTPV* locus, are removed from the maps. Only the remaining belong to the final working area. In Figure 3.30 the maps of IPM and IM after the application of the MTPV limit.



(a) *IPM* (b) *IM*

Figure 3.30: Control map build: MTPV limit

Finally, after the application of all the constraints, the un-normalized version of the control maps are ready.

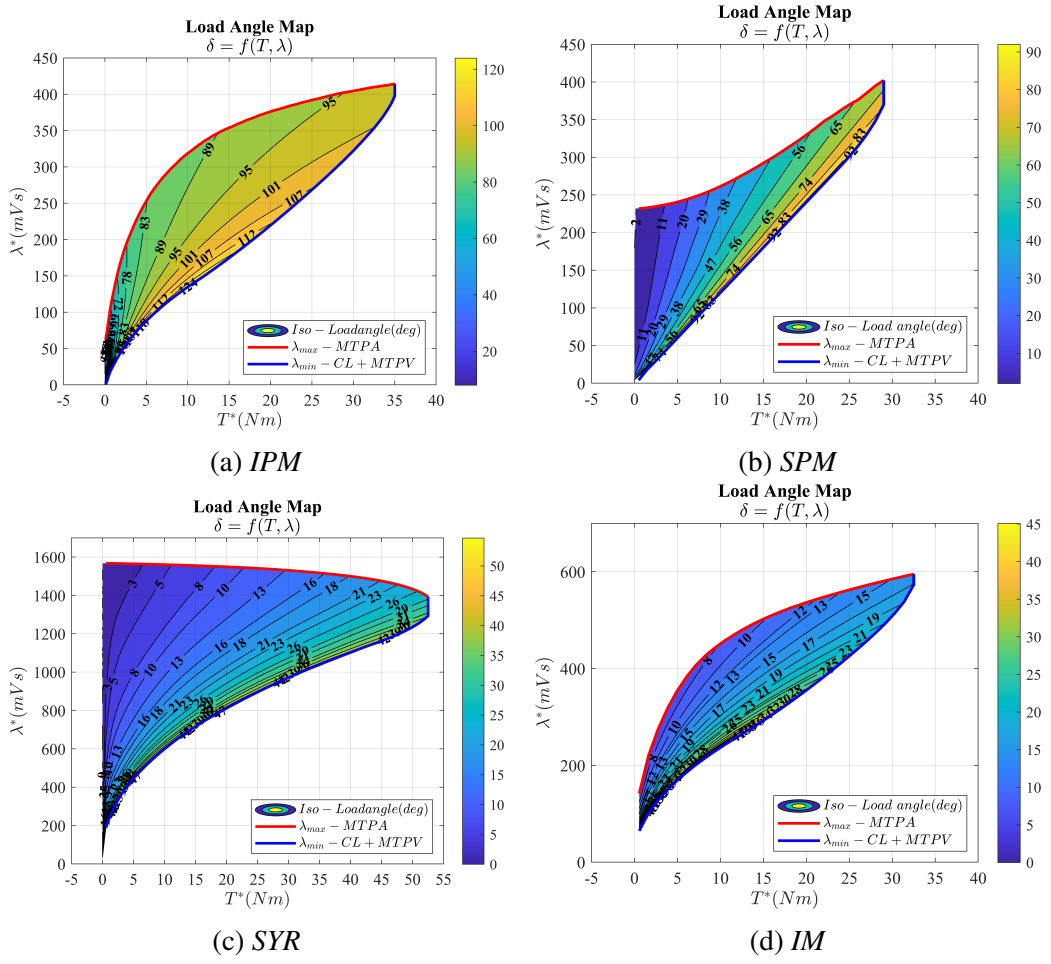
At this point, it is advisable to evaluate the new maximum value of torque that can be reached, T_{max}. Currently, it is the highest value in the original Torque Map, but, with the application of the control constrains, it is reduced to a lower one. The maximum correspond to the point at the intersection between the current and *MTPA* limits. It can be obtained investigating the maximum in the latest version of the Torque Map (`Te_cm`), evaluated with the torque expression in Equation (2.76).

Map Normalization

Now, the variables that must be adopted to normalize the map must be chose. Being the proposed solution a torque controller, based on polar flux control, the choice of normalization variables is already done: torque and flux amplitude.

To better understand how the following process works, a more precise explanation of the structure of the matrices is required. Each matrix is composed by a number of row per columns [RxC]. The number of row is the number of the torque levels, from zero to the maximum reachable value. In each row, along the columns, we find the map-variable values along the torque curve of the specific level: some levels are full, being the ones with the longest curves; some other have only few values, being an extreme torque value with less points. This scattered structure not allow the direct use of the maps, thus requiring the cited normalization procedure.

The normalization process will be applied on the Load Angle map. As said in the previous, this map is the one needed to define the flux-angle value, in order to reach a specific value of desired torque. As can be seen in the Figure 3.31 the form in which are these maps, does not allow their use in an interpolation process: the limits are highly non-regular, being a consequence of the application of the cited constrains. Moreover, each maps have its own extremes value, both in torque and flux amplitude. This requires a custom solution to interpreter the maps, representing an obstacle to the desired "plug and play" behavior from proposed controller.



(a) *IPM* (b) *SPM*

(c) *SYR* (d) *IM*

Figure 3.31: Original version of Load Angle Control Map of IPM, SPM, SYr and IM

- **Flux normalization**

  In order to perform the normalization with flux values, each row of the control maps must be sorted as the correspondent one of the `(Fs_cm)` matrix. This is not a specific for the control but it is needed for the interpolator in Matlab.

  Having ordered the flux amplitude of each torque level in ascending way, enables the knowledge of the minimum and maximum amplitude of flux that realize that torque level. The first and the last value of each row represent the λ_{min} and λ_{max}, that are, respectively, the flux point on *MTPV* and *MTPA* locus for the specific torque value. This information is the one used in the evaluation of the value of flux in p.u, both in the following procedure, but also in control scheme, as explained in Section 3.2.5.

  The interpolation process can be finally performed one row at the time, on all the

matrices:

- the value of flux $\lambda_{s,pu}$ in p.u. is created: is a vector of [*NF*] terms that space in the range [0,1];
- the i-*th* row of `(Fs_cm)` $\lambda_{s\_cm}(i)$, is normalized using its correspondent extreme values ($\lambda_{min}(i)$, $\lambda_{max}(i)$)

$$\lambda_{s,norm}(i) = \frac{\lambda_{s\_cm}(i) - \lambda_{min}(i)}{\lambda_{max}(i) - \lambda_{min}(i)} \tag{3.61}$$

- the 1D interpolation function is applied to the i-*th* row of all the other control maps. This function require as input the original relations between data, for instance ($\lambda_{s,norm}(i)$, $\delta_{cm}(i)$) and the vector in p.u. on which the resulting output must be produced, $\lambda_{s,pu}$. The result is the $\delta_{cm,norm}(i)$ vector composed by [*NF*] elements.

In Figure 3.32 is represented an example for IPM machine of the Load Angle map after the normalization with flux. The extremes of flux, λ_{min} and λ_{max} are showed, but in *pu* limiting the load angle map up and down.

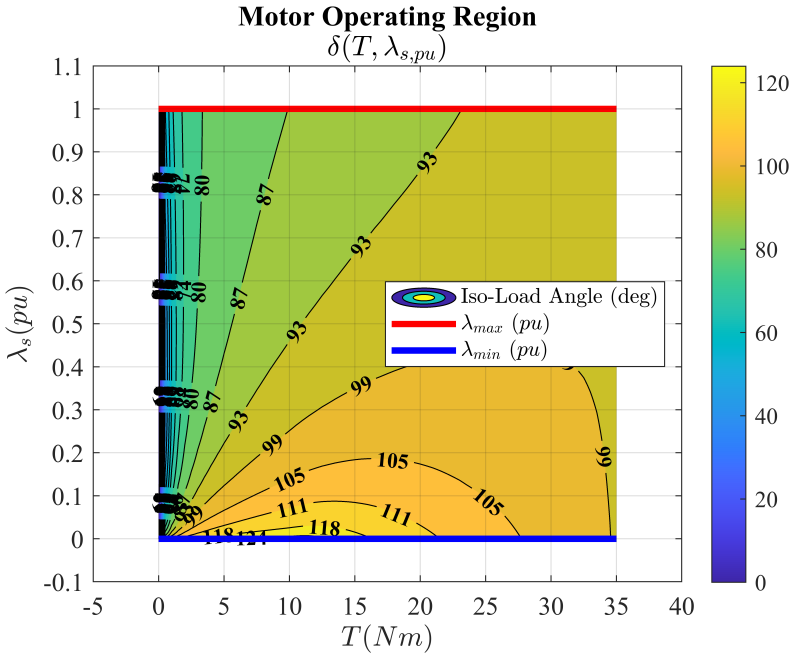


Figure 3.32: Load Angle Control Map of IPM - Flux Normalization

- **Torque normalization**

The second, and last, normalization is on torque. The interpolation process in this case can be performed directly using a 2D interpolation function. The reference torque vector T_{pu} in p.u. must be created with $[NT]$ elements and spans the range [0,1].

On the contrary of what happens for the fluxes, the normalized reference vector of torque that represent the original data, is not adaptive. In fact, all the element of all the rows are in the same torque range between the minimum and maximum level of torque, namely T_{max}. The vector is:

$$T_{norm} = \frac{T_{e\_cm}(:,1)}{T_{max}} \tag{3.62}$$

The input to the 2D interpolation function are the original relation $(\lambda_{s,pu}, T_{norm})$, the map that must be normalized $(\delta_{cm,norm})$ and the new relation $(\lambda_{s,pu}, T_{pu})$. The function produces as output the final form of the control map $(\delta_{cm,pu})$, in which, finally, the data are in a form that is compatible with the one needed in the control code. The *pu* form of the Load Angle map is showed in Figure 3.33. Both flux amplitude and torque are in *pu*, enabling the unified use of the map of whatever type of motor in the control code, without any particular care. The knowledge of the extremes of flux and torque is the only ancillary data required to the interpretation of this map.

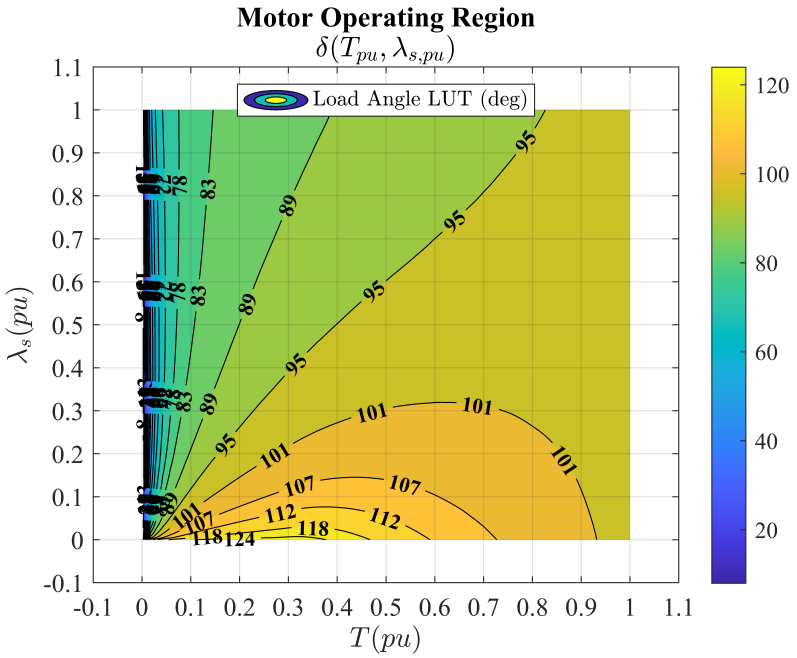


Figure 3.33: Load Angle Control Map of IPM - Flux and Torque Normalization

Extracts of the main parts of the code, from Control Maps generation for an IPM machine, are here showed.

1) original maps generation:

```
T_iso = contourc(Id(1,:),Iq(:,1),Te,T_levels_mtp);
```

... from `T_iso` *the* `id_cm_or,iq_cm_or` *maps are obtained*

```
is_cm_or = hypot(id_cm_or,iq_cm_or);
fd_cm_or = interp2(Id,Iq,Fd,id_cm_or,iq_cm_or,'spline');
fq_cm_or = interp2(Id,Iq,Fq,id_cm_or,iq_cm_or,'spline');
fs_cm_or = hypot(fd_cm_or,fq_cm_or);
La_cm_or = atan2(fq_cm_or,fd_cm_or);
```

2) Application of control constrains to all the matrices, here represented by the generic `x_cm_or`

... selection of the working quadrant

```
id_cm_log = id_cm_or<=0;
x_cm = x_cm_or.*id_cm_log;
```

... application of the current limit

```
is_cm_log = is_cm<=I_max;
x_cm = x_cm.*is_cm_log;
```

... application of MTPA limit

```
fs_cm_log = fs_cm<=Fs_mtpa;
x_cm = x_cm.*fs_cm_log;
```

... application of MTPV limit

```
id_cm_log = id_cm>I_mtpv;
x_cm = x_cm.*id_cm_log;
```

3) Finding the new torque extreme

```
T_max = max(max(te_cm));
```

4) Normalization of the maps

... sort of λ *rows, and application of the sorting to all the matrices*

```
[ , IX] = sort(fs_cm(i,:));
x_cm(i,:)  = x_cm(i,IX);
```

... research of torque extreme for each row: (fs_min and fs_max)

```
fs_min(i,1) = min(fs_cm(i,:));
fs_max(i,1) = max(fs_cm(i,:));
```

4.1) Flux interpolation

... generation of p.u. reference vector

```
fs_vect_pu = linspace(0,1,NF);
```

... evaluation of un-normalized flux values

```
fs_pu_unorm = (fs_cm(i,:)  - fs_min(i)) / (fs_max(i) - fs_min(i));
```
... normalization of the matrix along flux-direction
```
x_pu_f(i,:)  = interp1(fs_pu_unorm,x_cm(i,:),fs_vect_pu, 'spline');
```
4.2) Torque interpolation
... generation of p.u. reference vector
```
tq_vect_pu = linspace(0,1,NT)';
```
... evaluation of un-normalized torque values
```
tq_unorm = (T_levels_cm-T_levels_cm(1))/(T_levels_cm(end)-T_levels_cm(1));
```
... normalization of the matrix along torque-direction
```
x_pu = interp2(fs_vect_pu,tq_unorm,x_pu_f,fs_vect_pu,tq_vect_pu,'linear');
```

Usually, to verify that the the process of interpolation preserve all the information stored into the matrices, a reconstruction is performed. The original control maps is compared with the one just build. This last map, to be comparable with the former, must be bring to its absolute form reversing the normalization process done. As can be seen in Figure 3.34 the two iso-load angle curves are perfectly overlapped one to each other, proving that the interpolation does not corrupts the data.

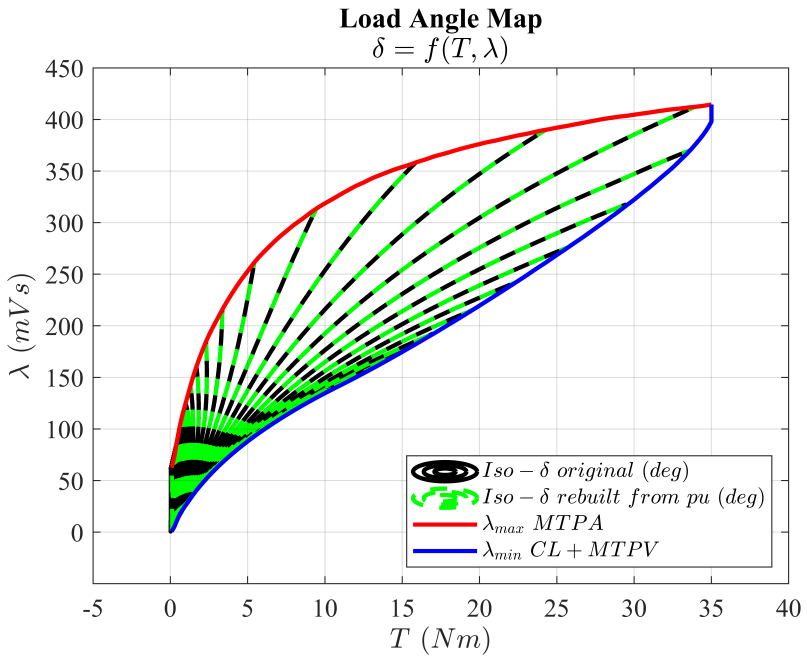


Figure 3.34: Correspondence of Load Angle Control Map before and after normalization of IPM

Montecarlo accuracy test

To analyze the performance of the load angle control map, ($\delta_{cm,pu}$) in determining the load angle reference value, and therefore, the accuracy to define the torque in the machine, a test must be performed. It reproduce the control procedure of FPC in the motor.

After the definition of a working condition (v_{dc}, ω_r, T^*), the test is performed. All the control variables are properly set and the results are obtained. At the end, the reference torque value T^* is compared with the one obtained applying the torque expression, and the error between this two value is evaluated.

To properly analyze the performance of the load angle control map, a unique test is not meaningful. Multiple runs of the described procedure must be performed, with different input $(\omega_r(i), T^*(i))$ representing different working conditions. In this way it is possible to obtain a set of results in the whole working range, up to the maximum speed and the maximum torque level that can be used by the machine.

To be able to analyze all the possible working area, the Montecarlo procedure is run. Few variables must be set: the v_{dc} voltage that is unique, and the coordinates on the torque vs. speed plane (ω_r, T^*) belonging to the working area. To be sure to explore all the usable area, a random generation of the test point has been performed.

Before starting, the base speed of the machine must be found, in order to distinguish the low speed range in which the *MTPA* characteristics is still valid, from the one at high speed in which the flux weakening is working. The value of ω_{base} is evaluated as the ratio between the maximum available voltage V_{max} and the flux amplitude at the end on the *MTPA* locus, using $\lambda_{max,end}$:

$$\omega_{base} = \frac{V_{max}}{\lambda_{max,end}} = \frac{v_{dc}/\sqrt{3}}{\lambda_{max,end}}$$

The test is then ready to be performed:

- A random pair of coordinates is chose: $(\omega_r(i), T^*(i))$
- Flux management is performed

  The upper limit of flux value is evaluated as the ratio between maximum voltage V_{max} and actual electrical speed $\omega_e(i) = p \cdot \omega_r(i)$.

  $$\lambda_{lim}(i) = \frac{V_{max}}{\omega_e(i)}$$

  The starting flux amplitude is evaluated usin the *MTPA* values, stored in a 1D LUT.

The inputs of the interpolator are the levels of torque and the *MTPA* flux vector, beside the reference torque value $T^*(i)$. The value obtained $\lambda^*(i)$ is valid only if the working point is in under ω_{base}. Otherwise, in the high speed range, the value is saturated to $\lambda_{lim}(i)$.

$$\lambda^*(i) \leq \lambda_{lim}(i)$$

- Torque management is performed

  If the working point belong to the speed region over ω_{base}, the torque reference value $T^*(i)$ is limited using the torque profile that represent the maximum reachable torque along *CL+MTPV* locus, that limits the working area. To find the upper torque limit $T_{lim}(i)$ another 1D interpolation is performed. It receives in input the flux along *CL+MTPV* and the torque level vector, and the actual flux value $\lambda^*(i)$. If the current torque value $T^*(i)$ is higher then the one in output, the reference is limited to this value.

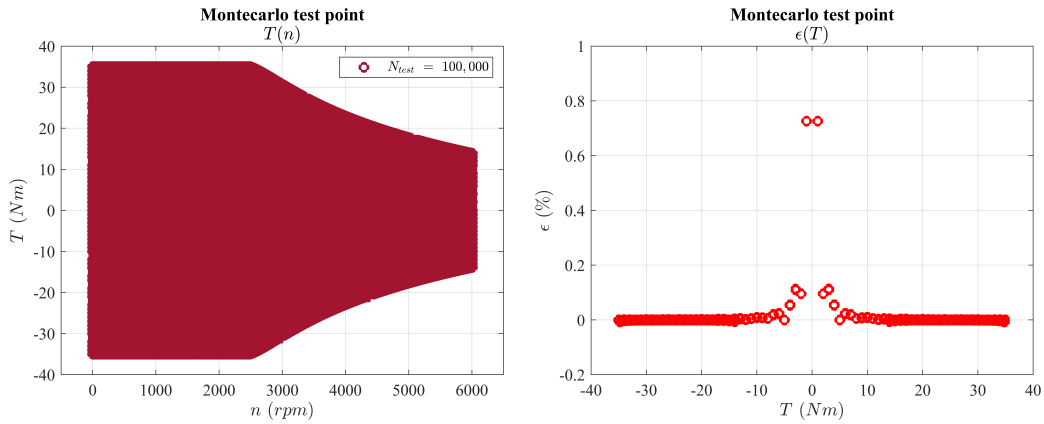
  $$T^*(i) \leq T_{lim}(i)$$

- The p.u. value of both torque and flux are evaluated, in the same way as the one explained for the generation of control map in Equation (3.61) and (3.62). These values are used as input in the reading of Control Maps, using 2D interpolation function, to produce the values of $(\lambda_d, \lambda_q, i_d, i_q, \delta)$ of this working point.

- With the torque equation that is exploited by Flux Polar Control strategy, in Equation (2.47) for AM and (2.80) for SM the calculated value of torque T_{calc} is obtained. It is directly compared with the reference one $T^*(i)$ to evaluate the percentage deviation as

  $$\epsilon(i) = \frac{T^*(i) - T_{calc}}{T^*(i)} \cdot 100$$

The Montecarlo accuracy test has been carry out on the IPM machine with 100,000 test points in all the speed-torque range. As showed in Figure 3.35, the percentage error between the desired torque value and the one calculated with the current and flux values of the specific working point, is always under the 1%, under 0.2% if neglecting the zero torque point.



(a) *Test points* (b) *Error*

Figure 3.35: Montacarlo Test

Extracts of the main parts of the code, from Montecarlo test, are here showed.

1) Generation of the set of working points:

```
V_max = Vdc/sqrt(3)
N_test = 1e5;
speed_rnd = randi([0,ceil(wm_max)],N_test,1);
torque_rnd = randi([te_min,te_max],N_test,1);
wm_base = V_max/(fs_max(end)*pp);
```

2) Selection of one working point:

```
te_act = torque_rnd(i,1);
we_act = pp*speed_rnd(i,1);
```

... evaluation of flux amplitude

```
fs_lim = V_max/we_act;
F_in = interp1(T_levels,fs_max,abs(te_act),'linear');
if(F_in>fs_lim) F_in = fs_lim;
```

... evaluation of torque amplitude

```
te_lim = interp1(fs_min,T_levels,F_in);
if(te_act>te_lim) te_act = te_lim;
```

... evaluation of torque and flux in p.u.

```
T_in_pu = (te_act-te_min)/(te_max-te_min);
f_min = interp1(T_levels,fs_min,abs(te_act),'linear');
f_max = interp1(T_levels,fs_max,abs(te_act),'linear');
F_in_pu = (F_in-f_min)/(f_max-f_min);
```

3) Reading of CM to retrieve variables

```
fd = interp2(fs_vect_pu, tq_vect_pu, fd_cm_pu, F_in_pu, T_in_pu);
```

```
fq = interp2(fs_vect_pu, tq_vect_pu, fq_cm_pu, F_in_pu, T_in_pu);
id = interp2(fs_vect_pu, tq_vect_pu, id_cm_pu, F_in_pu, T_in_pu);
iq = interp2(fs_vect_pu, tq_vect_pu, iq_cm_pu, F_in_pu, T_in_pu);
La = interp2(fs_vect_pu, tq_vect_pu, La_cm_pu, F_in_pu, T_in_pu);
```

4) Evaluation of torque with the values obtained

```
te_calc = 3/2*pp*(fd*iq - fq*id);
T_err = (te_act - te_calc)/te_act *100;
```

3.3.4 MTPA, MTPV and Load Angle Map derivated control limits

All the vectors of machine control variables has been created with *MTPA* and *MTPV* locus. Moreover, during the generation of Control Maps further control limits has been set. Now, relations between pairs of variable must be created in order to exploit them in the control code for settting the limits.

As represented in Figure 3.20 the limits are adopted in three main blocks: "Flux reference generation", "Torque reference generation" and "Load Angle reference generation". Let's analyze one LUT at the time.

1D LUT : $\lambda(T)MTPA$

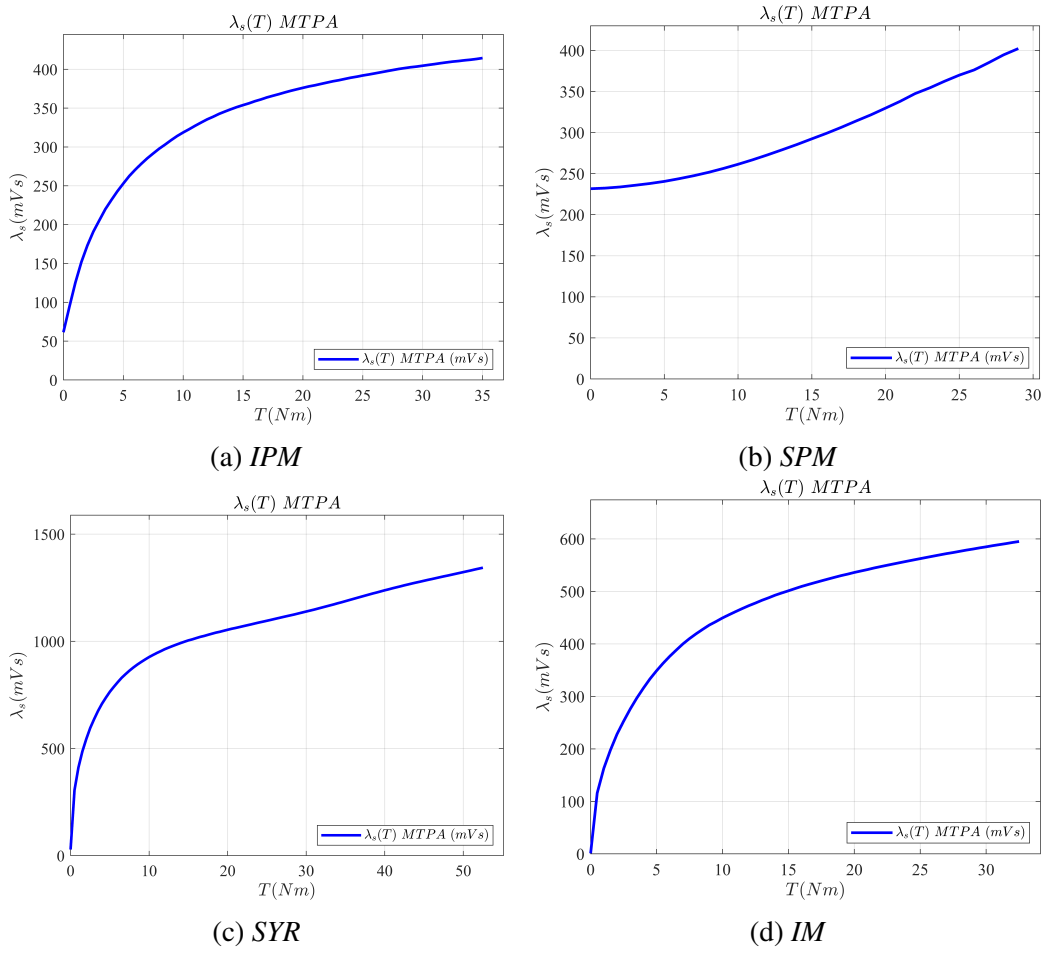
This is the first LUT used in the control code to define the preliminary amplitude of flux, before, eventually, apply further constrains. Providing the actual torque reference the correspondent flux in *MTPA* is evaluated.

The generation of this LUT is straightforward. The values of flux and torque along the *MTPA* locus are known (λ_{MTPA}, T_{MTPA}). The desired dimension of the final LUT, N_{LUT} is user-defined and must be set.

The adopted LUT-function-reader require that the LUT is based on a regular set of data. In this case, the torque vector must be a regular vector, in which every element is equally spaced from the previous and the following. For this reason, if the base vector does not satisfy this requirement, a regular version of it must be created, in the same range of the starting one. For this purpose the vector $T_{reg,MTPA}$ ranging between [0,T_{max}], with N_{LUT} elements, one away from the other DT is created.

The correspondent values of the flux vector are generated adopting the Matlab interpolation procedure. Providing as input the original vectors (λ_{MTPA}, T_{MTPA}) and the regular one T_{reg}. The output, $\lambda_{reg,MTPA}$ is stored into an header file, together with: ($T_{min}, T_{max}, DT, DT^{-1}$). This additional set of data are useful for interpolation purpose of the LUT, as explained in Appendix.

In Figure 3.36 the $\lambda(T)MTPA$ limits for all the machine are showed.



(a) *IPM* (b) *SPM*

(c) *SYR* (d) *IM*

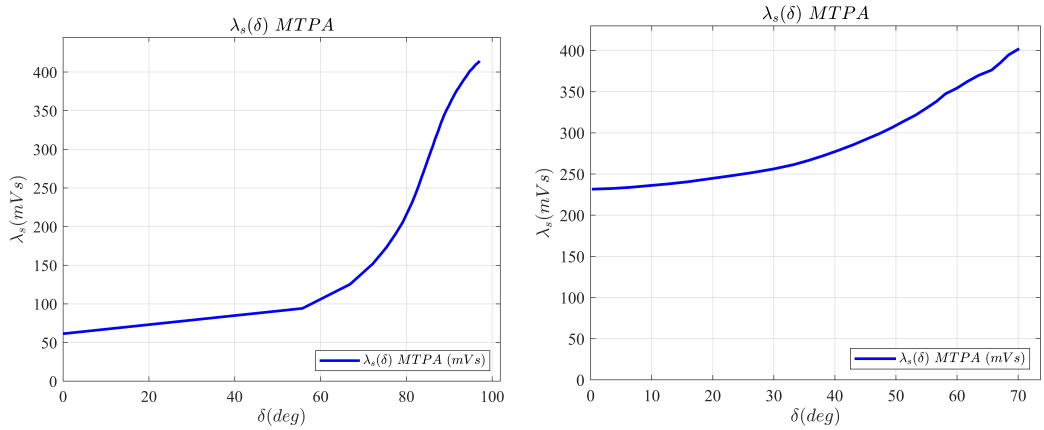
Figure 3.36: LUT $\lambda(T)MTPA$ for IPM, SPM, SYR and IM

1D LUT : $\lambda(\delta)MTPA$

This is the so-called "Dynamic-flux" used only for SM with (dq) coordinates defined following "PM-style" convention. It is used to limit the amplitude of flux to the one of *MTPA* during fast torque reverse at high speed. The adoption of this LUT ensure the improvement of dynamic performance.

The vector interested to generate this LUT are: $(\lambda_{MTPA}, \delta_{MTPA})$. The regular vector of load angle value with N_{LUT} elements must be created $\delta_{reg,MTPA}$, with its ancillary information $(\delta_{min}, \delta_{max}, D\delta, D\delta^{-1})$. The regular version of flux vector is obtained providing $(\lambda_{MTPA}, \delta_{MTPA})$ as original relation and $\delta_{reg,MTPA}$ as new base. The $\lambda_{reg,MTPA}$ is obtained.

In Figure 3.37 the $\lambda(\delta)MTPA$ limits for the SM with magnet (IPM and SPM) are showed.



(a) *IPM* (b) *SPM*

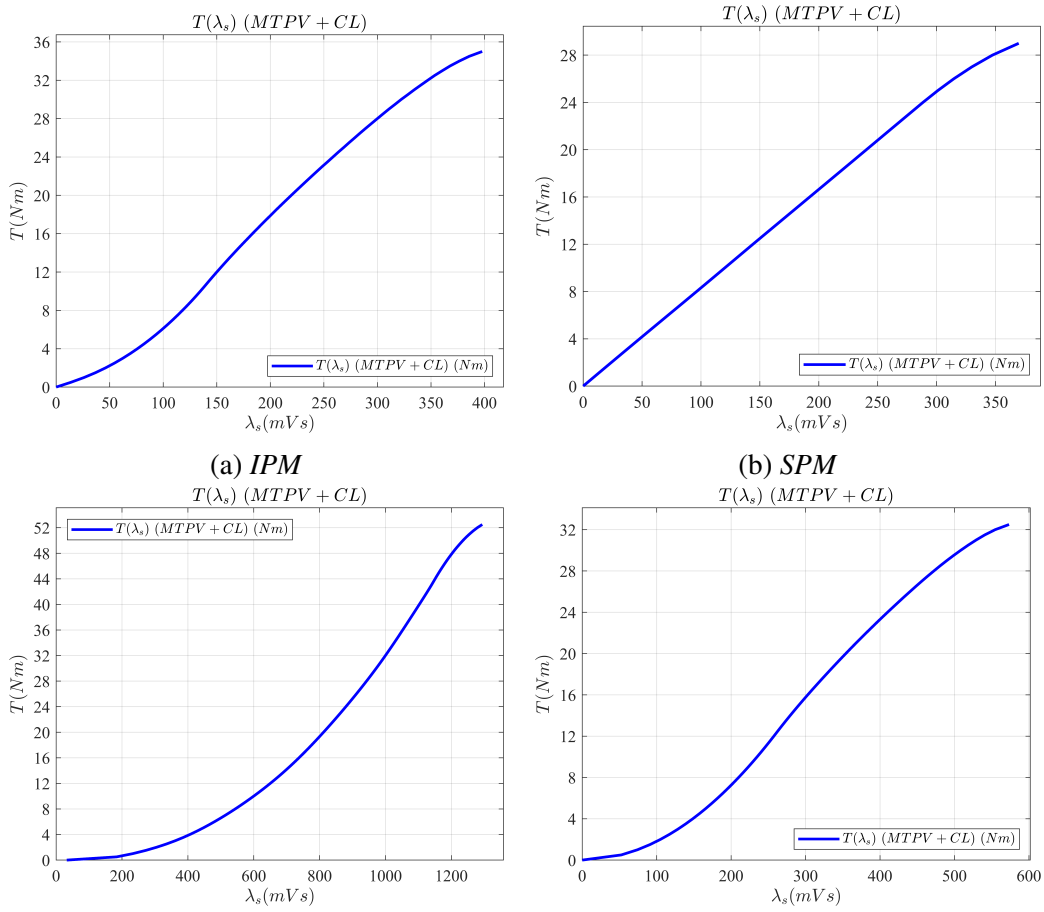
Figure 3.37: LUT $\lambda(\delta)MTPA$ for IPM and SPM

1D LUT : $T(\lambda)CL + MTPV$

This vector is the one that stores the maximum torque profile exploitable above ω_{base}, thus after the end of *MTPA* locus. As known, the maximum torque under ω_{base} is the nominal of the machine, while increasing the speed above this limit, means reducing progressively the highest torque obtainable. The real torque profile can be known only when the I_{max} Current Limit is set. The variable used to interpolate the vector is the flux amplitude because of its speed-variable amplitude and central role in the developed control.

The vector interested to generate this LUT are: $(T_{CL+MTPV}, \lambda_{CL+MTPV})$. The regular vector of flux value must be created $\lambda_{reg,CL+MTPV}$, with its ancillary information $(\lambda_{min}, \lambda_{max}, D\lambda, D\lambda^{-1})$. The regular version of torque vector with N_{LUT} elements is obtained providing $(T_{CL+MTPV}, \lambda_{CL+MTPV})$ as original relation and $\lambda_{reg,CL+MTPV}$ as new base to the interpolator. The $T_{reg,CL+MTPV}$ is obtained.

In Figure 3.38 the $T(\lambda)CL + MTPV$ limits for all the machine are showed.



(a) *IPM* (b) *SPM*

(c) *SYR* (d) *IM*

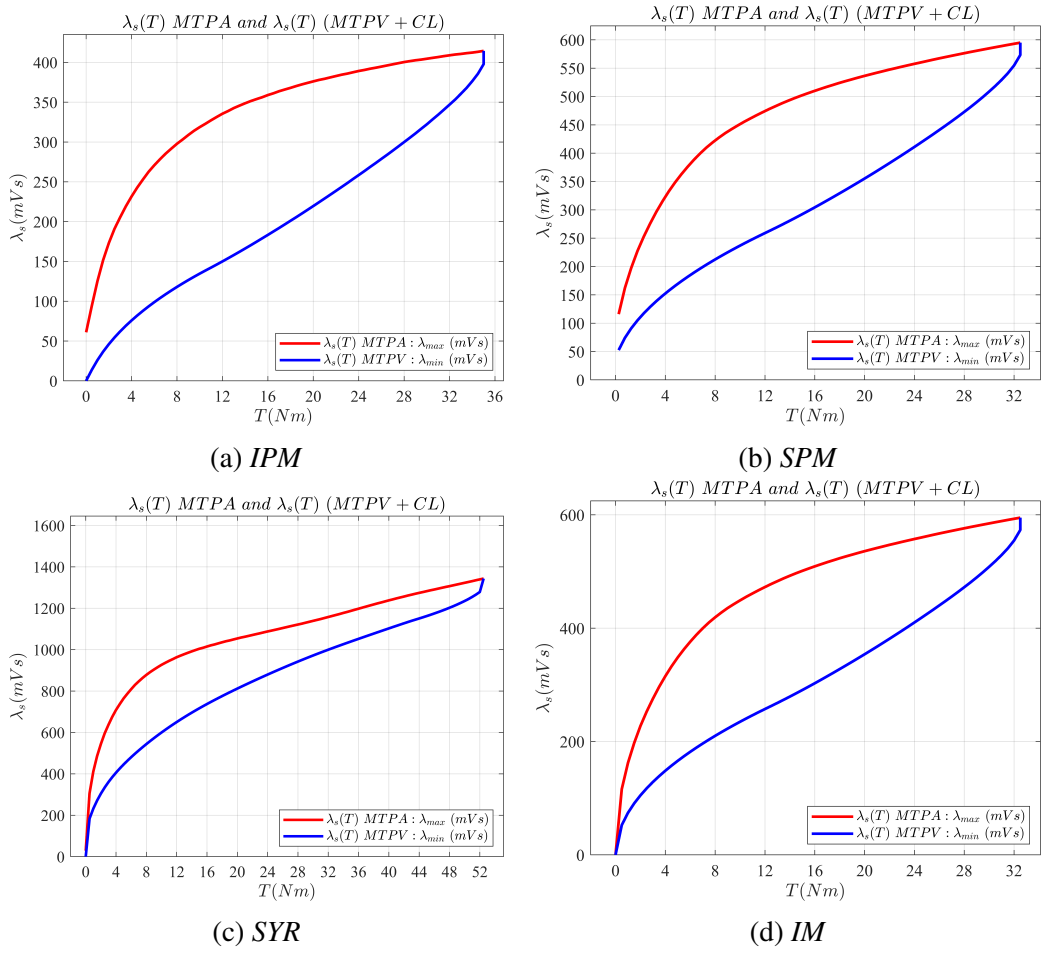
Figure 3.38: LUT $T(\lambda)CL + MTPV$ for IPM, SPM, SYR and IM

1D LUT : $\lambda(T)CL + MTPV$ **and** $\lambda(T)MTPA$

These two LUT are used for the evaluation of the value in p.u. of the flux necessary to read the 2D Load Angle LUT. The first, to which we also referred by calling it λ_{min} in Section 3.2.5 and in Section 3.3.3, represent the inverse of the relation explained in the previous paragraph dedicated to $[T(\lambda)CL + MTPV]$. In fact, it contains the flux values correspondent to torque along *(CL+MTPV)* locus. The second, also called λ_{max} in the same Sections previously cited, is exactly the same showed in the first paragraph of this list: the flux along *MTPA*.

Moreover, in this case the procedure is the same followed for $[\lambda(\delta)MTPA]$. The obtained vectors, with N_{LUT} elements, are store in an header file.

In Figure 3.39 the $\lambda(T)CL + MTPV$ and $\lambda(T)MTPA$ limits for all the machine are showed.



(a) *IPM* (b) *SPM*

(c) *SYR* (d) *IM*

Figure 3.39: LUT $\lambda(T)CL + MTPV$ and $\lambda(T)MTPA$ for IPM, SPM, SYR and IM

2D LUT : $\lambda_d(i_d, i_q)$ and $\lambda_d(i_d, i_q)$

These two are only used in the Flux Observer structure of SM in which the Magnetic Model is implemented. They are the experimental Direc Flux Maps of the machine.

To store a matrix, the rule that have to be followed are the same as the one for vector but in two dimension. The matrices are based on current vectors that must be equally-spaced. The regular vector of current $(I_{d,reg}, I_{q,reg})$ are generated with the same extremes of the original vector and with N_{LUT} elements. The ancillary information are, in this case, doubled: $(I_{d,min}, I_{d,max}, DI_d, DI_d^{-1}, I_{q,min}, I_{q,max}, DI_q, DI_q^{-1})$

The correspondent value of flux are obtained adopting the usual 2D interpolation function in Matlab, with the original relation (I_d, I_q, λ) and the new points in which evaluate the function $(I_{d,reg}, I_{q,reg})$.

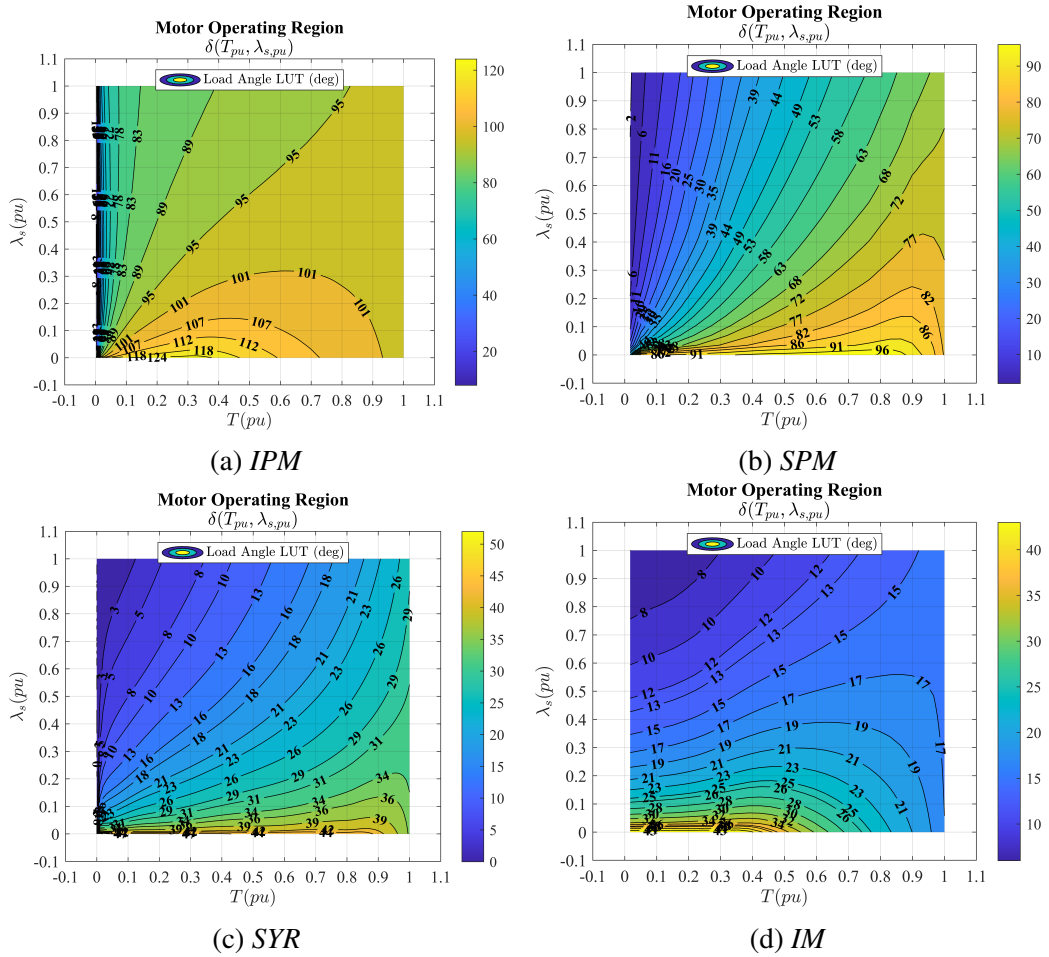
Finally, the matrices $(\lambda_{d,reg}, \lambda_{q,reg})$ are memorized into the header file, moving one

column at the time. Note that, during the generation of this file, the column of the flux matrix became the row in the file. This must be kept in mind during the use of this maps in the control code.

2D LUT : $\delta(T_{pu}, \lambda_{pu})$

Finally, the Load Angle Control Map, is generated. Being a 2D LUT, the two regular vectors in p.u. with N_{LUT} elements, ranging between [0,1], are created: $(T_{p.u,reg}, \lambda_{p.u.,reg})$ .The ancillary information are: $(T_{min}, T_{max}, DT, DT^{-1}, \lambda_{min}, \lambda_{max}, D\lambda, D\lambda^{-1})$. The Load Angle Control Map based on these regular data is created. The input in this case are the $(T_{p.u,old}, \lambda_{p.u.,old}, \delta)$ and the actual points $(T_{p.u,reg}, \lambda_{p.u.,reg})$. The map for the control code δ_{reg} is saved into the header file.

In Figure 3.40 the $\delta(T_{pu}, \lambda_{pu})$ maps for all the machine are showed.



(a) *IPM* (b) *SPM*

(c) *SYR* (d) *IM*

Figure 3.40: 2D LUT $\delta(T, \lambda)$ for IPM, SPM, SYR and IM

Chapter 4

Validation of FPC control

In the previous chapter the main characteristics of Flux Polar Control have been explained in detail, together with the explanation of how to generate the control maps. Now, the control code to run this torque controller, first in Simulink environment and then in real experimental tests, must be settle down. Applying step by step all the procedure explained in the previous chapter, the FPC code is build.

The control code has been implemented for simulation purpose with a common editor to produce the C and H file. In fact, these two type of files are the base on which whatever type of code is wrote. It is required a C file (.c) that contains the source code and a number of H (header) file (.h) with all the ancillary information related to the control. Both the C and the H files can be wrote in whichever type of editor, because, in the end, the merge of all the file is done in Matlab or in another type of aggregator. Having the control code in the C file and the simulation environment ready in Simulink, or the test bech available, are the only requirement to run the FPC simulation.

In this case both the validation in simulation of the control and the experimental test have been carry out. The starting control code is exactly the same in the two cases but, the only thing that change is the environment in which it is used. In this last case it must receive information and produce command on real component, instead of simulate all of these. The compatibility of the simulation and experimental environment is only a matter of correspondence of the name associated to the control variables.

Now the control code structure is investigated in order to understand the discrete implementation of a torque controller.

4.1 Control code writing

The core of the control code is certainly the implementation of the FPC part, but a series of step are required to bring the system in the right condition to work properly. To do this a self-made state machine is required. The one implemented is composed by eight states and is represented in Figure 4.1.

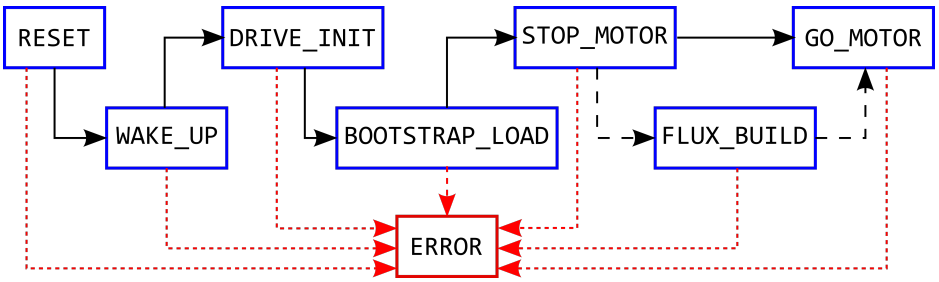


Figure 4.1: State machine of the developed control

- `RESET(-5)`: in this state the duty cycle are kept all equal at their medium value `da=db=dc=0.5`, so the voltage output of the system is null. The PWM is disabled. If the `START` button is pressed, the state machine move to `WAKE_UP`.
- `WAKE_UP(-4)`: in this state the dc voltage is checked to evaluate that the dc link capacitor have been charged to a proper value. If the desired value has been reached, the state machine move to next state: `DRIVE_INIT`.
- `DRIVE_INIT(-3)`: in this state the current sensor offset is evaluated in order to be properly compensated while the control code is running. The procedure is scanned by a counter, and, when it reaches the final value, the state machine move to the `BOOTSTRAP_LOAD` state.
- `BOOTSTRAP_LOAD(-2)`: here the duty are kept all equal `da=db=dc=0.5` for a proper time in order to charge the bootstrap capacitor that gives the reference voltage for the floating drain of the upper switch of the inverter leg. When the counter reaches the target value the machine can move to `STOP_MOTOR` state.
- `STOP_MOTOR(-1)`: this is a waiting state for the machine. All has been properly prepared in order to run the control, and the trigger signal to change the state, is awaited. When the `GO` button is pressed, the reset and the initialization function of the main control variables are performed. After that the state machine can move directly to `GO_MOTOR` or pass before to `FLUX_BUILD` state if needed.

- `FLUX_BUILD(1)`: in this state the flux build of the machine is executed. It's only needed for those type of machines without magnets that must be fluxed in order to be ready to perform with the highest dynamic the FPC control. The procedure is explained in the following. When the target flux value has been reached the state machine can move to`GO_MOTOR`.
- `GO_MOTOR(0)`: finally the real control code is here executed, in order to satisfy the request of the user. This state is maintained permanently, unless there are events that cause the transition to the `ERROR` state.
- `ERROR(2)`: it is call by an event in the system that can cause damage, so, to prevent any possible consequence, the machine is putted in a safe state. In fact, here, the duty are all equal `da=db=dc=0.5`, so the voltage output of the system is null. Moreover, the PWM modulation is disabled. The event that can cause it are over-current in one of the three phase, dc over-voltage or over-speed while the machine is in `GO_MOTOR` state.

The writing of the code require the discrete time form of all the equation needed to run FPC. In digital control, time is marked by the switching frequency *fs* that can be chose by the user. In simulation whichever value of frequency can be adopted, while, during experimental test, the upper limit is settle accordingly to the available technology that sets the boundary. This has a major effect: the value of the variables can be known only in precise time step, defined by the switching frequency and identified by the distance of the sampled value at the current step (k), by an integer number of steps.

To better understand the discrete form of an equation a basic example is provided. Let's consider a integral of a generic function *x(t)* in the time domain:

$$y(t) = \int x(t)\, dt \tag{4.1}$$

Moving in Laplace domain the equation become:

$$y(s) = \frac{x(s)}{s} \tag{4.2}$$

Finally, in discrete time form, distinguishing between the actual value, described by (k) and the previous one, identified by (*k-1*), the equation become:

$$\begin{aligned} y(k) &= y(k-1) + Ts \cdot x(k) \\ y(k) &+= +Ts \cdot x(k) \end{aligned} \tag{4.3}$$

where the current value of *y(k)* variable is obtained as the sum of the one at the previous

step *y(k-1)* plus the integral of the *x(k)* variable over Ts. The transformation in discrete time form of the equations is done to all the ones that compose the code. Note that, only the equations with an integral in time domain need this type of treatment. For all the other it is sufficient to write it down in the conventional form, as in a normal calculation.

As said before, the code of the FPC applies what have been previously described: starting from the inputs, they must be processed in order to be used in the control: voltages, currents and mechanical information must be obtained. Given the torque reference, all the LUTs are used to define the final value of references to obtain voltages that must be applied to the machine to get the desired behavior. The code lines needed to do that are the same for all the types of machines and, in the end, they are actually few. All the code related to the *VI*θ flux observer structure, must be also used to get the essential information that are needed to run the code.

Before the explanation of the core of the code, a preparatory part, required only by some kind of machines, is needed. The AM and the SYR, since they have no magnet in their structure, require a preliminary flux build to prepare them to run properly from the very first moments in FPC control. If the dynamic behavior of the machine is of primary importance from the first time instant, this is a non-avoidable step.

This can be clearly understand looking at the rotor flux equation of IM in Equations (2.35). In this case the amplitude of flux is controlled by the $i_{d,s}$ current component, but its dynamic behavior is limited by the rotor time constant τ_r, that controls the evolution of flux amplitude given the variation of d-axis current. A behavior like this one, in which a torque-producing flux varies with a delay with respect to the command, is unacceptable when a high-dynamic performance is required.

To avoid this type of phenomena a minimum value of flux λ_{min} must be in the machine before the start of the control, and the variation of flux during the FPC execution is always kept under control to avoid the flux goes under this minimum level, if not required by the positioning of the workin point in flux-weakening region.

Note that, this type of treatment is not a peculiarity of the FPC, thus making it more difficult to implement, but a general procedure carry out on this type of machine, whatever type of control is used.

In the following, the main parts of the FPC control code and the flux build code are showed.

4.1.1 Input Signals management

The physical signals obtained from the simulation environment or the experimental bench test are the same: current, dc voltage and mechanical position. They must be treated in order to be usable in the control code.

The code adopted is showed:

1) Acquisition of currents U(0:2), voltage U(3) and mech. position U(4)

```
input.ch0 = U(0);
input.ch1 = U(1);
input.ch2 = U(2);
vdc = U(3);
theta_mec = U(4);
```

2) Removing offSet of current channels evaluated in `DRIVE_INIT` *state*

```
iabc.a = input.ch0 - offset_in.ch0;
iabc.b = input.ch1 - offset_in.ch1;
iabc.c = input.ch2 - offset_in.ch2;
```

3) Input Data Elaboration

3.1) Electric angle

```
SinCos(&theta_mec, &sc_theta_mec);
theta_elt = theta_mec * pp - theta_off;
SinCos(&theta_elt, &sc_theta_elt);
```

3.2) Speed computation

```
pll(sc_theta_mec, &pll_theta_mech);
```

3.2) dc link voltage filter

```
vdc = low_pass_filter(vdc,vdc_filt,double_pi*500.0*Ts);
vdc_filt = vdc;
```

3.3)Phase voltages with Dead Time

```
vabc_DT(&vdc_filt, &duty, &iabc, &vabc_real);
```

4) Rotation to obtain $\alpha\beta$ *variables*

```
DirectClarke (&vabc_real, &valphabeta);
DirectClarke (&iabc, &ialphabeta);
```

This is the minimum number of code lines necessary to prepare the data for their use in the control code. Now the rest of the code can run properly.

4.1.2 Preliminary Flux Build

The flux building procedure is not a peculiarity of FPC control, but it is required for those machine without magnets. In our case the synchronous SYR and the asynchronous IM need it, to be ready for a high-dynamic behavior while FPC is running.

The amplitude of flux must be bring to the minimum desired value λ_{min}, with a controlled ramp of limited dynamic. To do this two regulators like the ones of the PFC are adopted, one controlling the flux amplitude and the other acting on load angle. The current value of flux, obtained from the flux observer structure, is compared with the reference that change at every step, to obtain the $v^*_{d,s}$ voltage. On the other hand, the $v^*_{q,s}$ voltage is set by the load angle regulator that works in order to keep the current load angle value equal to the desired value $\delta^* = 0$. This zero set-point is due to the fact that, the flux build procedure is stationary, thus no torque is required from the machine.

Finally, the reference voltages are rotated, first in $(\alpha\beta)$ and then in (abc) frame, to calculate the duty cycles needed to control the inverter feeding the machine.

The main parts of the code used for SYR machine, that is equal for the IM machine except for the part related to the flux observer, are here showed. In fact, what changes between the machines is the model on which the $I\theta$ estimator is based on, thus requiring is specific structure implementation.

1) The actual value of flux is incremented until the reference one is reached, then a time equal to t_FB is give to flux to settle down:

```
fs_ref += FsMin*0.01;
if (fs_ref>FsMin){
   counter++;
   fs_ref = FsMin;}
if (counter==t_FB){
   StateDrive = GO_MOTOR;
   counter =0;}
```

2) Flux observer structure: $I\theta$ and VI

```
flux_obsv_IT(ialphabeta, theta_elt, &fdq_IT, &falphabeta_IT);
flux_obsv_VI(ialphabeta, valphabeta, falphabeta_IT, &falphabeta_obsVI,
   &falphabeta_obsVI_prd);
falphabeta_obsVI_amp = amplitude (falphabeta_obsVI.alpha,
   falphabeta_obsVI.beta);
```

3) Position evaluation, calculation of current load angle value and frames rotation

```
sc_falphabeta_obsVI.cos = falphabeta_obsVI.alpha / falphabeta_obsVI_amp;
sc_falphabeta_obsVI.sin = falphabeta_obsVI.beta / falphabeta_obsVI_amp;
direct_rot(falphabeta_obsVI, sc_theta_elt, &fdq_obsVI);
```

```
delta_VI = atan2(fdq_obsVI.q,fdq_obsVI.d);
direct_rot(ialphabeta,sc_falphabeta_obsVI, &idqs);
```
4) Regulators
```
   fs_var.ref = fs_ref;
   fs_var.fbk = falphabeta_obsVI_amp;
   fs_par.lim = 1.5 * Rs * I_max;
   fs_var.ffw = Rs * idqs.d;
PIReg(&fs_par, &fs_var);
   la_var.ref = 0.0;
   la_var.fbk = delta_VI;
   la_par.lim = sqrt(vdc*vdc*one_over_three - fs_var.out*fs_var.out);
   la_par.kp = fs_par.kp * falphabeta_obsVI_amp;
   la_par.ki = fs_par.ki * falphabeta_obsVI_amp;
   la_var.ffw = 0.0;
PIReg(&la_par, &la_var);
```
5) Reference voltages evaluation
```
vdqs_ref.d = fs_var.out;
vdqs_ref.q = la_var.out;
inverse_rot(vdqs_ref, sc_falphabeta_obsVI, &valphabeta_ref);
InverseClarke(&valphabeta_ref, &vabc_ref);
PWMCompute(&vabc_ref, &duty, &vdc);
```

A focus on the code of the $VI\theta$ flux observer is performed to highlight its predictive behavior and also the different structure of the $I\theta$ part for SM and AM.

As explained in Section 3.2.4, the flux observer is based on the current and voltage models of the machine. While the voltage model is always the same, being a simple integrator of the back-EMF force, the one based on current is characteristics of the machine. For SM it means that the maps containing the direct flux relations, $\lambda_d(i_d, i_q), \lambda_q(i_d, i_q)$ must be changed, following the machine type. On the other hand, for the AM, all the parameter describing the machine in the $I\theta$ ($L_m, \tau_r, k_r, \sigma, L_s$), must be adapted.

The code adopted to implement the flux observer is showed in the following.

$I\theta$ estimator for SM

1) direct current rotation from (α, β) to (dq)
```
direct_rot(ialphabeta, sc_theta_elt, &idq);
```
2) d-axis Flux Computation
```
fdq->d = read_two_dim_lut(&FdObsMap[0][0], fabs(idq.q), idq.d, DIQ, INV_DIQ,
```

```
DID, INV_DID, IQ_MAX, IQ_MIN, ID_MAX, ID_MIN, NPOINTS_Q);
fdq->q = read_two_dim_lut(&FqObsMap[0][0], fabs(idq.q), idq.d, DIQ, INV_DIQ,
DID, INV_DID, IQ_MAX, IQ_MIN, ID_MAX, ID_MIN, NPOINTS_Q);
```

3) inverse flux rotation from (dq) to (α, β*)*

```
inverse_rot(fdq, sc_theta_elt, &falphabeta_IT);
```

$I\theta$ estimator for AM

In this case a LUT containing the behavior of the machine inductance is required to adapt this parameter to the actual working point. The `Lm_fs` LUT is obtained from experimental test, carry out in the frequency range needed for the application.

1) Flux-dependent parameter:

```
Lm_est = read_one_dim_lut(&Lm_fs[0], fabs(fs_ref), Lm_FMAX, Lm_FMIN, Lm_DF,
Lm_INV_DF);
Lr = Lm_est + Llr;
Ls = Lm_est + Lls;
sigma = 1 - Lm_est*Lm_est/(Lr*Ls);
```

2) Flux estimator implentation:

```
direct_rot(ialphabeta, sc_theta_elt, &ialphabetaR);
Fr_alphabetaR.alpha += Ts * (ialphabetaR.alpha * Lm_est - Fr_alphabetaR.alpha)
* Rr_DC/Lr;
Fr_alphabetaR.beta += Ts * (ialphabetaR.beta * Lm_est - Fr_alphabetaR.beta)
* Rr_DC/Lr;
Fs_alphabetaR_IT.alpha = Fr_alphabetaR.alpha* Lm_est/Lr +
+ sigma*Ls * ialphabetaR.alpha;
Fs_alphabetaR_IT.beta = Fr_alphabetaR.beta * Lm_est/Lr +
+ sigma*Ls * ialphabetaR.beta;
inverse_rot(Fs_alphabetaR_IT, sc_theta_elt, &Fs_alphabeta_IT);
```

$VI\theta$ estimator

Finally, from the composition of this two structure, the $VI\theta$ is obtained.

1) Predictive behavior of VI flux estimator: the actual value (k) is the one evaluated in the previous time step (k-1):

```
falphabeta_VI->alpha = falphabeta_VI_prd->alpha;
falphabeta_VI->beta = falphabeta_VI_prd->beta;
```

2) Flux observer:

```
tmp = falphabeta_IT.alpha - falphabeta_VI->alpha;
err = valphabeta.alpha - Rs * ialphabeta.alpha + g * tmp;
```

```
falphabeta_VI_prd->alpha += err * Ts;
tmp = falphabeta_IT.beta - falphabeta_VI->beta;
err = valphabeta.beta - Rs * ialphabeta.beta + g * tmp;
falphabeta_VI_prd->beta += err * Ts;
```

4.1.3 Flux Polar Control code

Finally, the main part of the control code. After the flux building procedure, for those that need it, the machine is ready to work, in FPC control.

In the first part of the code the flux observer structure is again used to evaluate at every ISR call the amplitude and load angle of the flux, necessary as current values for the two regulators to generate the error from the two references. The code is the same viewed in flux building phase, part *2)* and *3)*. With angle θ_s and speed ω_s obtained by the stator flux vector, the variables are converted in (dq_s) frame. The phase advancing angle is evaluated using ω_s, to partially compensate the delay between the calculation of the command and its application.

Then, the control constrain are adopted to reach, in the end, the two reference value, from whom the (vdq_s^*) voltages are obtained. These are rotated in (abc) to evaluate the duty cycles. The code is showed.

Phase-advancing angle

```
theta_ph_adv = 0.5 * Ts * ws;
SinCos(&theta_ph_adv,&sc_theta_ph_adv);
```

Flux Polar Control

1) Limitation of torque slope

```
mech_ref = two_level_saturation(mech_ref, TORQUE_MAX, (-TORQUE_MAX));
mech_ramp = slew_rate_limit(mech_ref,mech_rate*Ts,mech_ramp);
```

2) MTPA flux

```
fs_mtpa = read_one_dim_lut(&FS_MTPA[0], fabs(mech_ramp), FS_TMAX, FS_TMIN,
FS_DT, FS_INV_DT);
```

2.1) saturation of the flux to the minimum value to be guaranteed in the machine - only for magnet-less machine

```
fs_ref = min_level_saturation(fs_mtpa,FsMin);
```

2.2) Dynamic flux limit - only for machine with magnets

```
FluxLimitDyn = read_one_dim_lut(&F_DYN_Vect[0], fabs(delta_VI), FS_LAMAX,
FS_LAMIN, FS_DLA, FS_INV_DLA);
fs_ref = max_level_saturation(fs_ref, FluxLimitDyn);
```

3) Flux-weakening law

```
f_FW = k * sqrt(vdc*vdc*one_over_three + Rs*is_amp*is_amp - 4/3*Rs*Pe) /
(fabs(pll_VI.omega_filt) + 0.001);
f_FW = k * sqrt(vdc*vdc*one_over_three + Rs*is_amp*is_amp - four_over_three*Rs*Pe)
/ (pp * fabs(pll_theta_mech.omega_filt) + 0.001);
fs_ref = max_level_saturation(fs_ref,f_FW);
```

4) Torque limits

```
Te_lim = read_one_dim_lut(&T_LIM[0], fs_ref, T_FMAX, T_FMIN, T_DF, T_INV_DF);
mech_ramp = two_level_saturation(mech_ramp, Te_lim, (-Te_lim));
```

5) Normalization of torque and flux values

```
te_pu = fabs(mech_ramp/F_TMAX);
fs_min = read_one_dim_lut(&F_MIN[0], fabs(mech_ramp), F_TMAX, F_TMIN, F_DT,
F_INV_DT);
fs_max = read_one_dim_lut(&F_MAX[0], fabs(mech_ramp), F_TMAX, F_TMIN, F_DT,
F_INV_DT);
fs_pu = (fs_ref-fs_min) / (fs_max-fs_min);
```

6) Load angle evaluation

```
la_ref = read_two_dim_lut(&La_TeFsmap_pu[0][0], te_pu, fs_pu, DTE, INV_DTE,
DFS, INV_DFS, TE_MAX, TE_MIN, FS_MAX, FS_MIN, NPOINTS_TE);
```

7) Regulators

```
fs_var.ref = fs_ref;
fs_var.fbk = falphabeta_obsVI_amp;
fs_par.lim = 1.5 * Rs * I_max;
fs_var.ffw = Rs * idqs.d;
PIReg(&fs_par, &fs_var);
la_var.ref = la_ref * sign(mech_ramp);
la_var.fbk = delta_VI;
la_par.lim = sqrt(vdc*vdc*one_over_three - fs_var.out*fs_var.out) ;
la_par.kp = fs_par.kp * falphabeta_obsVI_prd_amp;
la_par.ki = fs_par.ki * falphabeta_obsVI_prd_amp;
la_var.ffw = (Rs * idqs.q) + (pll_theta_mech.omega_filt * pp *
falphabeta_obsVI_prd_amp);
PIReg(&la_par, &la_var);
```

8) Phase Advancing of Voltage References and Inverse Rotation

```
vdqs_ref.d = fs_var.out * sc_theta_ph_adv.cos - la_var.out * sc_theta_ph_adv.sin;
vdqs_ref.q = fs_var.out * sc_theta_ph_adv.sin + la_var.out * sc_theta_ph_adv.cos;
inverse_rot(vdqs_ref, sc_falphabeta_obsVI_prd, &valphabeta_ref);
```

4.2 Matlab-Simulink simulation

All the simulation have been carry out in Simulink: the environment that compose the experimental test bench has been accurately modeled in the Simulink model, in order to perform simulations that provide results as close as possible to the ones that will be obtained in real experimental procedure.

The self-build Simulink model is composed of a series of element that are in common between all the models, as the sensors, the MCU, the inverter, the battery and the drive machine. The only characteristic block is the one containing the machine models that has been developed to analyse the machine behavior when controlled with FPC. It contains all the electromagnetic-mechanic equations that describe the machine, like the ones presented in Chapter 2. A general overview of the Simulink environment is provided in Figure 4.2, where, starting from the left, we can find:

- Signal measurement with current, voltage and position sensors, in blue and orange, with the same characteristic of the real component. They are the same for all the machines. The only parameter that can vary is the number of division of the encoder that is changed following the number of division of the real component available for real experiment.

- The MCU, in light blue, that receives in input the measurements, the reference torque and the buttons command for the state machine. In output it produces the duty cycles signals for the inverter.

  This block contains the user-made control code, in the C file, for the torque controller. In Simulink the management of a system controlled by a discrete-time triggered block, is performed using the so-called *S-Function*. It permits to collect, at every trigger signals that corresponds to the desired switching period, all the input signals and to provide them to the code, at time step (k). In the code they are elaborated and used to define the output values at time step ($k+1$): the principle ones are the duty cycles for the inverter ant the PWM enabling signal, while a group of control variables can also be retrieved in order to monitor the behavior of the system while the simulation is running.

- The inverter, in red, receives the duty cycles and the PWM enabling signals from the MCU, the current of the motor, the v dc voltage and the carrier signal. With all these data it is able to produce the phase voltages used to feed the machine. The inverter model can work with instantaneous or average values. It highly depends from the computational power of the adopted workstation. Moreover it implement the dead time voltage drop, emulating the behavior of a real component.

- The battery, in brown, emulates the AC/DC converter adopted in laboratory to produce the desired v dc voltage for the inverter.
- The model of the MUT, in light yellow: in input are provided the voltages from the inverter and the speed of the driving machine, while in output all the electrical-magnetic and mechanical quantities are obtained for the post-simulation analysis.

  Independently on the type of machine, in this block are implemented the voltage, magnetic and mechanic models of the MUT:

  - starting from the mechanical speed of the shaft, imposed by the driving machine, the electrical angle is obtained. It is used for the coordinates rotation of the phase voltages that from (*abc*) have been transformed in (α, β), and finally, using this angle, in (*dq*).
  - At this point the iron loss must be taken into consideration. It is a mandatory procedure for the AM where the value of the equivalent iron resistance is known exactly from no-load test, while for the SM, its value is defined arbitrarily, being this parameter hard to retrieve from an experimental test. Anyway, as explained in Section 2.1.5 and 2.2.4, the equivalent voltage and resistance of the machine are obtained and used in the following.
  - The electromagnetic model of the machine, containing the voltage equation and the magnetic model, is used to obtain the flux and the current component, in (*dq*) coordinates.
  - With these information we can obtain:
    * the current and flux component in ($\alpha\beta$) and in (*abc*);
    * the torque value, using the cross product between flux and current vectors;
    * the load angle value using the (*dq*) component of flux.

    All these variables are recorder to study the behavior of the machine.
- Finally, the driving machine, in purple, that has the role of running the MUT at a certain speed imposed by the user, up to the maximum tolerated by the machine.

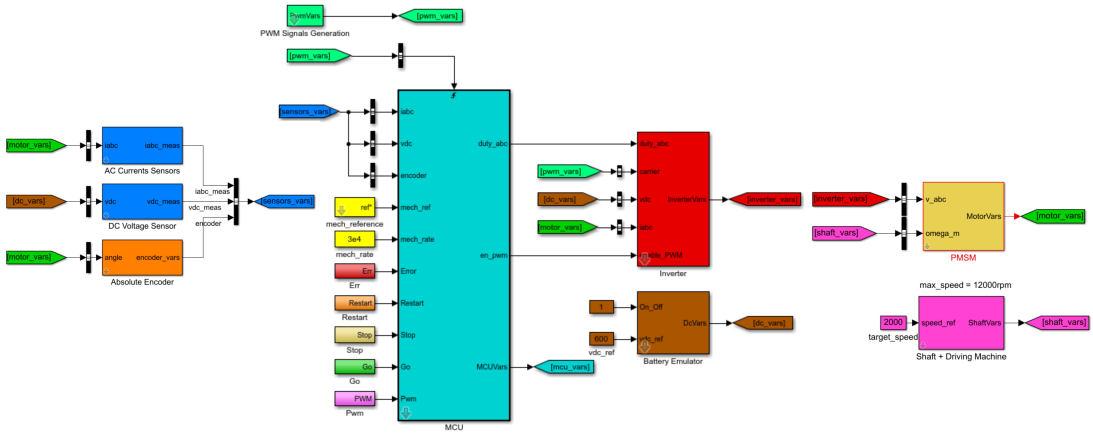


Figure 4.2: Simulink block scheme

With the described architecture, the simulations with the different kind of machines can be performed.

The C file containing the control code is compiled in Matlab, using the mex function that is able to collect all the information in the ancillary H files. In our case the header files contain:

- motor data in the maps and LUTs for the control;
- inverter data, included the dead-time LUT;
- the control functions used in the code;
- constant and general variables declaration.

Once the compiling phase end, the control starts and it is able to imposed the voltage needed to satisfy the torque requested at the desired speed, following the machine type and the feeding setting. All the control variable are recorded during the simulation in order to produce an output file, useful for post-processing analysis on the collected data. In the following, the simulation results obtained on the four type of machines are showed.

4.2.1 Induction Motor

The main machine parameter and control settings are in Table 4.1.

Motor			
Pole Pairs	pp	2	–
Stator Resistance	R_s	0.598	Ω
Rotor Resistance	R_r	0.401	Ω
Stator Leakage Inductance	$L_{\sigma s}$	3.82	mH
Rotor Leakage Inductance	$L_{\sigma r}$	3.82	mH
Unsaturated Magnetizing Inductance	$L_{m,unsat}$	68.4	mH
Nominal Torque	T_{nom}	16	Nm
Inverter			
Switching frequency	f_{sw}	4	kHz
Dead Time	t_{DT}	1	μs
dc link voltage	v_{dc}	600	V
Sensors			
Encoder	n_{div}	1024	–
Controller			
Sampling frequency	f_s	4	kHz
Proportional gain PI_λ (150 Hz)	$k_{p\lambda}$	942.5	$1/s$
Integral gain PI_λ ($k_{i\lambda}/k_{p\lambda}$ = 1/15 ·ω_b)	$k_{i\lambda}$	59,218	$1/s$
Proportional gain PI_δ	$k_{p\delta}$	942.5·λ	V
Integral gain PI_δ	$k_{i\delta}$	59,218·λ	V
Frequency threshold for Flux Observer	g	10	Hz
Flux Weakening coefficient	k_{FW}	0.9	–
PLL bandwidth	$\omega_{b,PLL}$	188.5	rad/s
PLL phase-margin	$\phi_{M,PLL}$	60	°
Flux Building value: max(λ_{MTPA})	λ_{FB}	595	$mVs°$
Over current protection	I_{max}	25	A
Over speed protection	ω_{max}	6050	rpm
Over voltage protection	V_{max}	700	V
Maximum torque (2·T_{nom})	T_{max}	32.5	Nm

Table 4.1: IM - Simulation parameters

The proportional and integral gains values have been chosen accordingly to the adopted switching frequency of the controller. The general rule imposes that the switching frequency and the highest bandwidth of the controller must be, at least, a decade one away

from the other. The adopted switching frequency has been set equal to 4 KHz, while the bandwidth of the controller is 150 Hz, lower then the threshold of 400 Hz.

Motor and Generator test

The first tests carry out on the machine are the Maximum Torque per Speed tests, both in motor and in generator mode. The maximum torque level is requested to the machine from 0 rpm up to the highest reachable speed. The machine, following the several constrains applied, is able to produce the torque profile that corresponds to the maximum exploitation of the working area defined by the *MTPA, CL* and *MTPV* locus.

The evolution of the main variables in the motor test is represented in Figure 4.3.

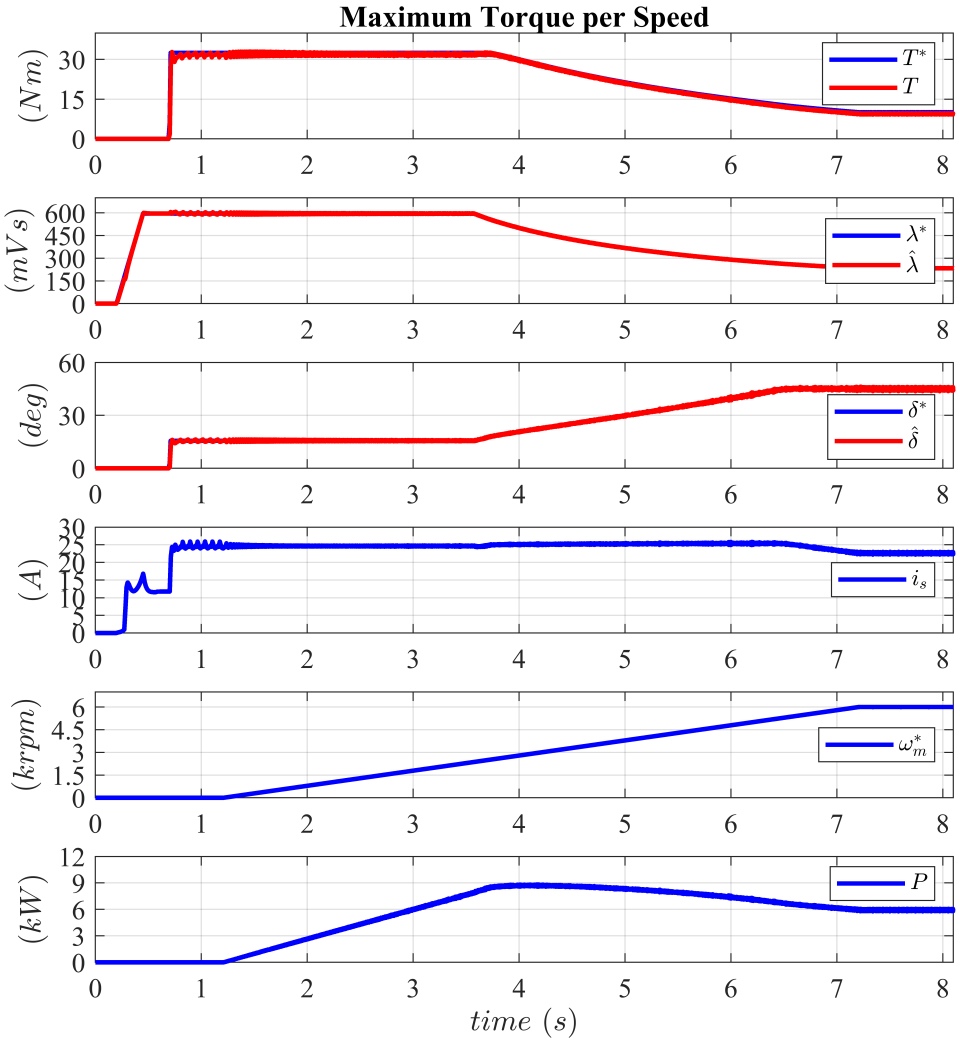


Figure 4.3: IM Maximum Torque per Speed - Motor @ 600 V

Let's analyze how the different variables evolve during the test. In the first half second the flux building phase is performed: the machine is not yet running the FPC torque controller because, before, the flux must be bring to desired value, to guarantee high dynamic performance during control. The desired torque is null, while the flux increase with a ramp until it reaches the final value, that is, in this case, the flux in *MTPA* corresponding to the highest torque level. To obtain this flux ramp the current must be non null. Once the target value has been reached, the machine is ready to perform FPC.

The reference torque step change from zero to the maximum value T_{max}, when the speed is still null: the actual torque value follows the desired perfectly. The flux, being previously set to the one in the extreme point in *MTPA*, doesn't change. The current move to the no-torque value to the value corresponding to the maximum torque, that is, in the end, in the extreme point along the *MTPA*, corresponding to Current Limit of 25A. As well as does the torque, the load angle value has a step change from zero, at which it is maintained during the flux-building phase, to its characteristic value in *MTPA* condition.

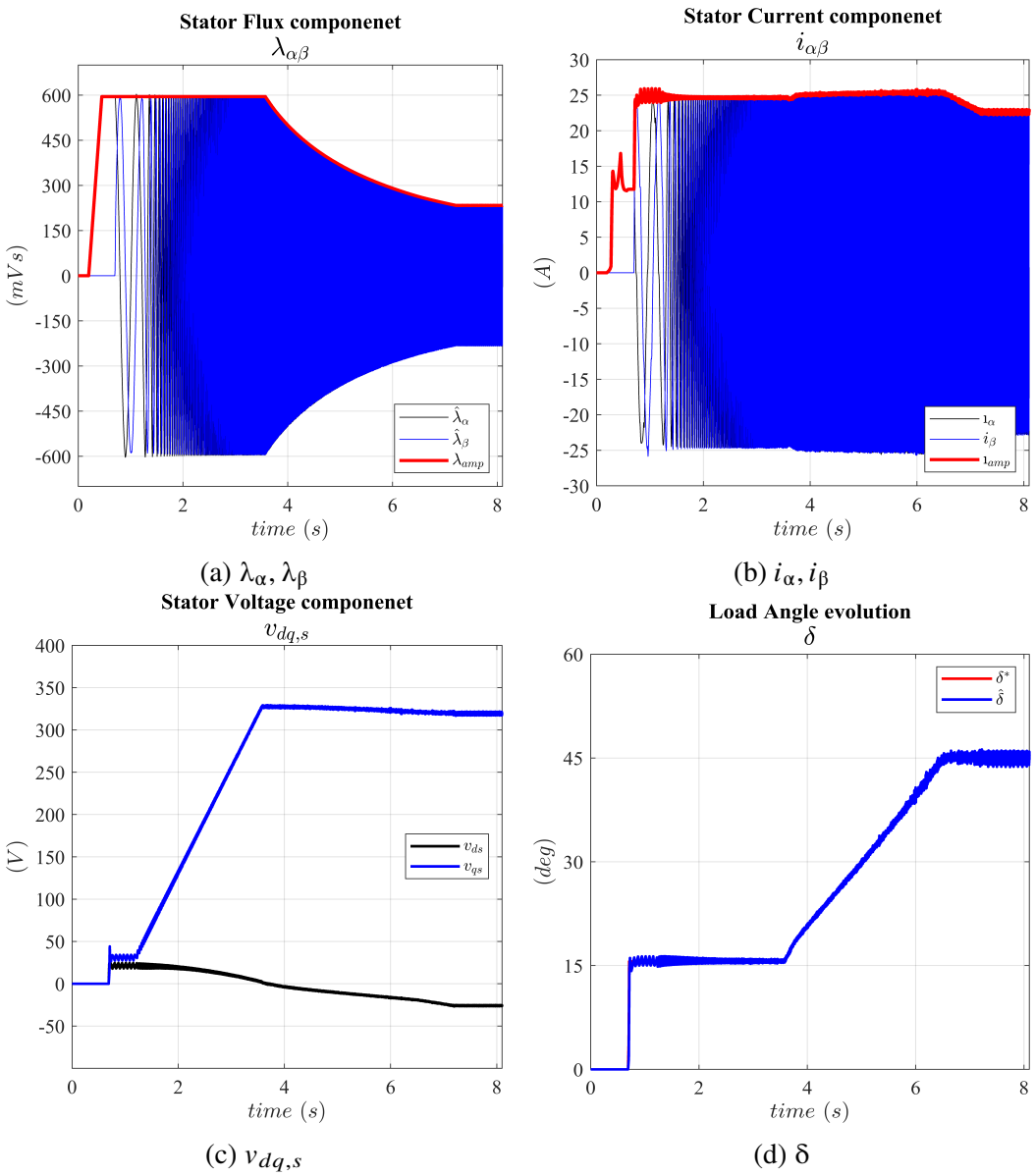
Nothing change in the variables, even when the machine start to move. Only the power rises, given the torque value and the machine speed increasing.

When the speed approaches the base value $\omega_b = V_{max}/\lambda$, the change in the variables is evident. The torque maximum value is no more the previous one, but it is reduced following the limitation imposed to the machine by the *CL*, that set a boundary. The same behavior is performed by the flux amplitude: having reached the base speed means that, if we want the speed overcome this threshold, given a fixed maximum voltage, the flux must be reduced. On the other hand the flux vector, following the *CL* trajectory, is moving from the *MTPA* locus to the *MTPV* one. Thus the load angle is increasing its value from the angle of the *MTPA* (15°) to the one of the *MTPV* (45°). The power assumes the peak value at the end of the MTPA locus, at base speed. When speed goes over this value the power slowly decrease its value.

In the end the *MTPV* locus is reached: this is evident from the load angle plot, that in this condition has a value of 45°. The torque and flux reach their minimum value, while current slowly decrease and power asset to its final permanent value at maximum speed.

It must be noticed how the variable are stable and controlled even if the machine is in this extreme working condition.

A detail of the behavior of fluxes, currents, voltages and load angle is provided in Figure 4.4.



(a) $\lambda_\alpha, \lambda_\beta$ (b) i_α, i_β

(c) $v_{dq,s}$ (d) δ

Figure 4.4: IM Evolution of control variables - Motor @ 600 V

As said previously the fluxes behavior resemble the one of the torque, that is progressively reduced after the end of the *MTPA*.

The current, even if uncontrolled variables in this type of control strategy, is always maintained constant. In this case the peak value is the one imposed by the current limit, equal to 25A because the torque request is always equal to the maximum value. When the machine arrive in *MTPV* condition the peak value start to decrease because the working point is moving back to the origin of axis. The falling is stopped by the speed that reaches its target value, thus is no more increasing.

The reference voltages produced by the two regulators are showed in Figure 4.4c. As anticipated in the introduction of the torque controller, the two axis are decoupled and the voltage level is also highly different between the two. While in d-axis is performed the regulation of flux amplitude, limited only by the voltage drop on the phase resistance, the q-axis take care of all the back-EMF of the machine. This is the reason why the voltage are so different, as can be seen in the picture.

Finally, a zoom in the load angle amplitude evolution from *MTPA* to *MTPV* with an angle that is exactly equal to the expected value of 45°. The first plateau at 15° is in *MTPA*. Slowly, moving along the CL the vector increase its shifting, reaching the *MTPV* plateau in which stay permanently.

The same test is performed also in generator configuration. In this case the speed increase from 0 rpm up to maximum one, but the torque request is negative, thus the power assume a negative value. The behavior of the variables, showed in Figure 4.5 is the same as in the previous case but reversed.

After the preliminary flux-build phase, always needed by the machine, the FPC starts to work, with a reference torque that step change from zero to the minimum value. The main control variables adapt their values in order to satisfy the request: the flux is already the one in *MTPA*, while the load angle step change its value to reach to the target imposed by the 2D LUT, given the current torque and flux references. This configuration is kept until the working point is in *MTPA* condition. As soon as the speed goes over the base one, the variables change following the constrains imposed, first by the current limit and after by the *MTPV* locus. The torque and flux are progressively reduced, while the load angle is moving toward the 45° in *MYPV*. As before, the current is perfectly controlled to its maximum value. In this case the machine reach just the start of the *MTPV* operation: this can be deduced by the constant value of the current vector that is always equal to the maximum value and doesn't reduce is value and by the load angle, equal to 45° only when speed approach the maximum value of 6000 rpm.

The peak power in generation mode is higher then the one in motor mode. This is due to the fact that, in this case, the power losses in the circuit are also present into the total amount of power evaluated with the electromagnetic torque. Their value is subtracted only when considering the power produced in output by the machine.

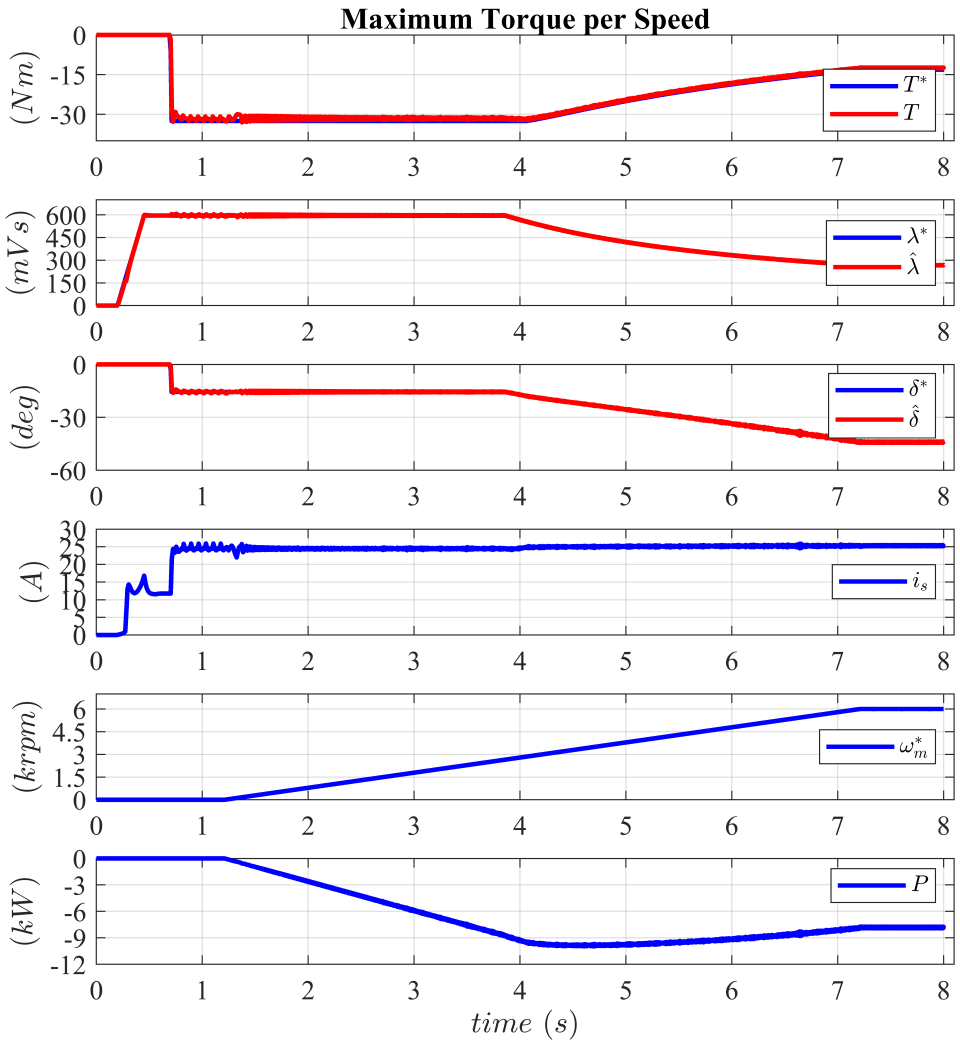
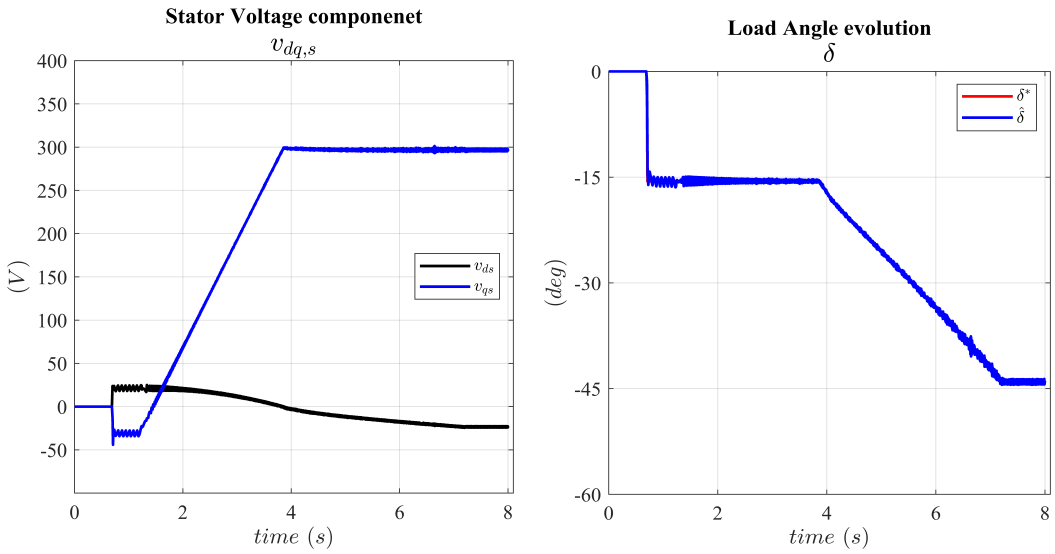


Figure 4.5: IM Maximum Torque per Speed - Generator @ 600 V

A more detailed behavior of voltages and load angle is showed in Figure 4.6. As in the motor case, the (*dq,s*) voltages are highly different between them, facing different phenomena on the two axis. The load angle, as can be seen, present the same evolution, but reversed, that it has in motor mode. From the -15° in *MTPA*, reaches the -45° in *MTPV*,and keep maintaining this value, being at this point the speed constant.



(a) $v_{dq,s}$ (b) δ

Figure 4.6: IM Evolution of control variables - Generator @ 600 V

High dynamic test

To prove the dynamic capability of the FPC torque controller, a square torque reference is imposed to the machine with a frequency of 4 Hz and a variation slope of 3000 Nm/s, acting from zero up to the maximum speed. In this way both the dynamic performance and the ability to control the machine at high speed in *MTPV* condition are proved.

The torque change its value from positive to negative, and the variables follow:

- the flux amplitude is always positive being its target value equal both in motor and in generator mode.
- instead the load angle, being the variable that actually control the sign of the torque, varies in the same way as the torque does. When the torque is positive, also the load angle is, and the same for the negative value. It progressively increase to 45° degrees at high speed, putting the controller in the harder control condition: in fact, here, the flux vector must rotate of 90° between the two working condition.
- the current is always positive and its value change instantaneously only when the reference torque revers, but, except for this moments, it is always quite constant and positive.
- the power, being the result of the product between torque and speed, present the same shape as torque. After the peak value at base speed, it value progressively decrease, but not dramatically, with speed.

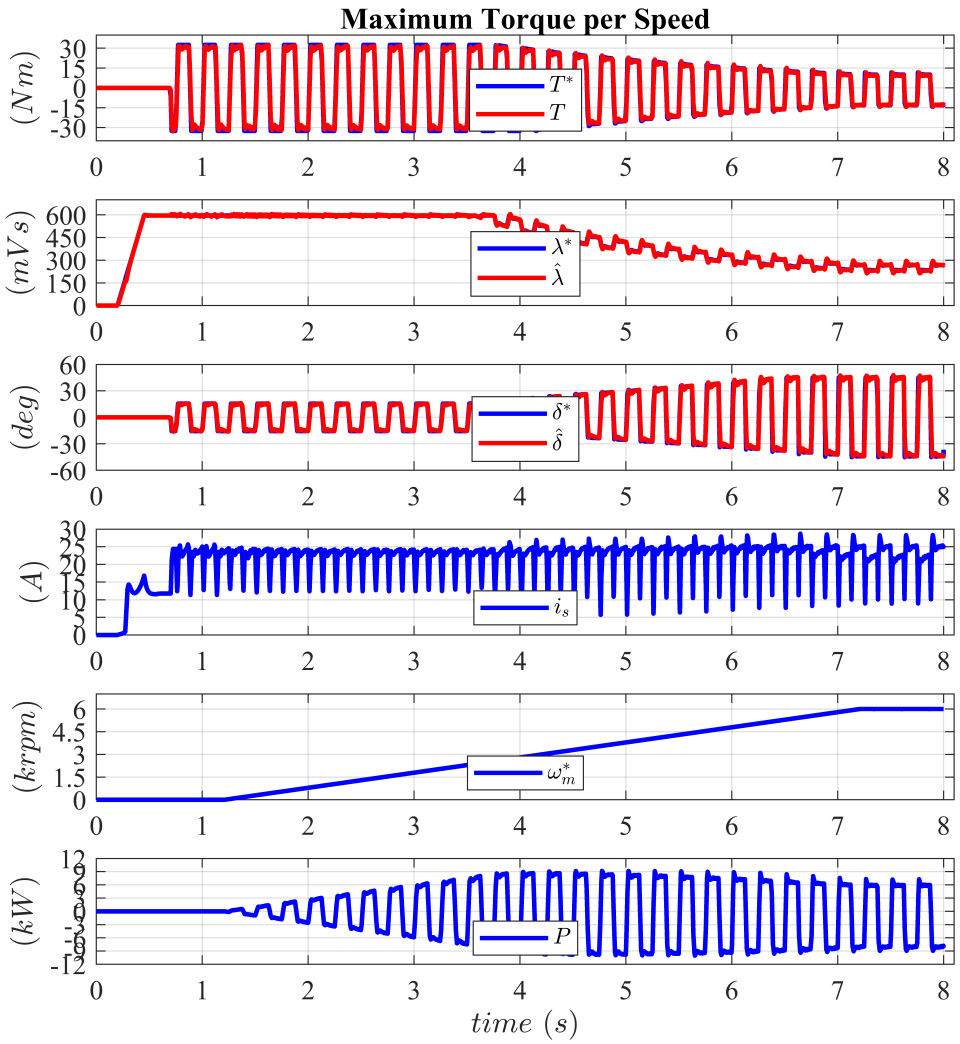
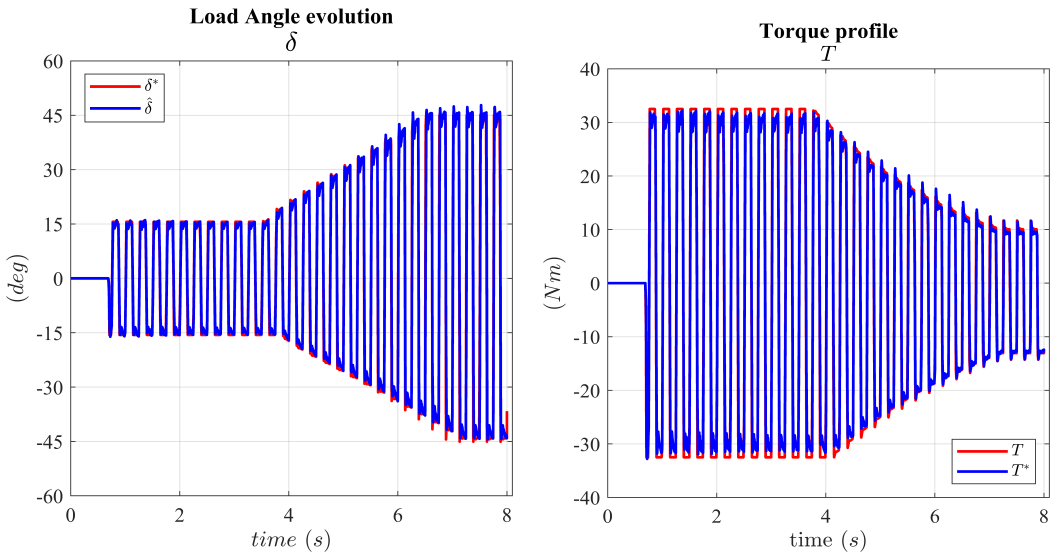


Figure 4.7: IM Fast Torque reversal at 4 Hz - @ 600 V

The evolution of torque, together with load angle, for this type of test is showed in Figure 4.8. Their shape are complementary with the speed that increase: when load angle that progressively rise its value, the torque amplitude is reducing. It must be noticed that the the reference load angle value and obtained one are perfectly overlapped, meaning that the control works properly and it is able to imposed the desired variable value.



(a) δ (b) T

Figure 4.8: IM Fast Torque reversal at 4 Hz - @ 600 V

Accuracy test

Finally, an accuracy test is performed: imposing a speed in which the extreme value of torque can be reached, a stair of torque values is provided as reference for the controller. The torque, starting from the lower value *-Tmax* increase its value until the maximum, *+Tmax* is reached. In Figure 4.9 the reference stair and the obtained one are showed. The level of correspondence between the two is quite satisfying, proving the efficiency of the proposed torque controller.

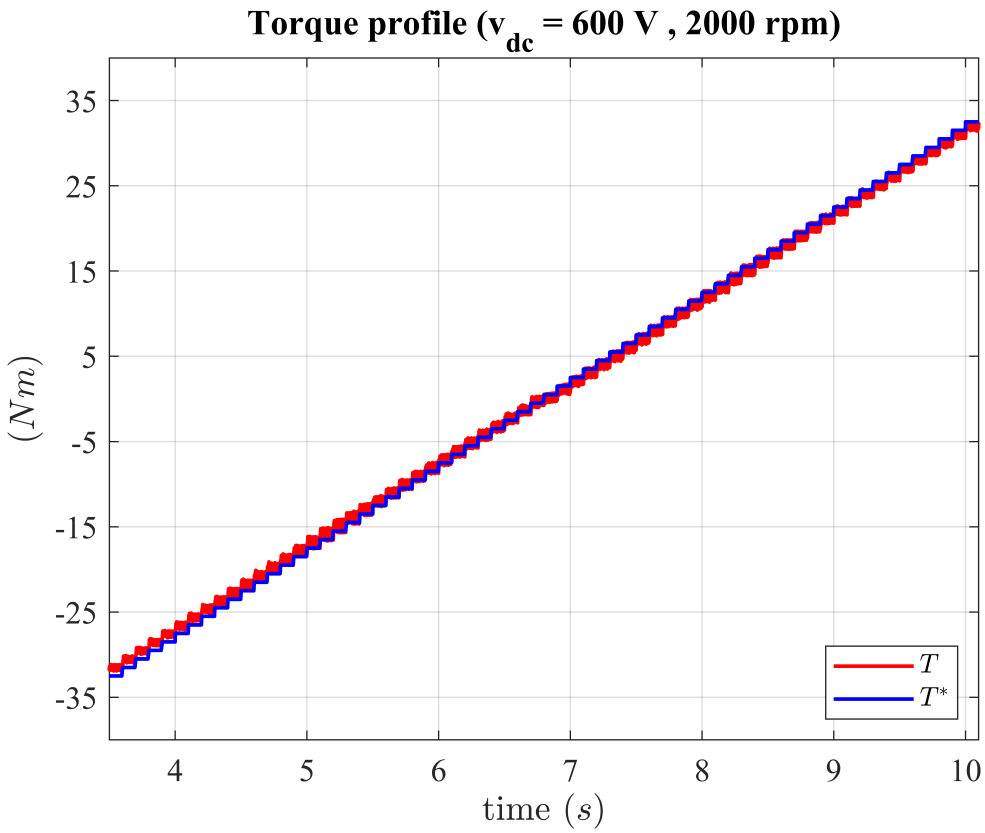


Figure 4.9: IM Torque accuracy test @ (600 Vdc, 2000 rpm)

4.2.2 Internal Permanent Magnets Motor

The main machine parameter and control settings are in Table 4.2.

Motor			
Pole Pairs	pp	2	–
Stator Resistance	R_s	0.42	Ω
d-axis equivalent Inductance	L_d	3.56	mH
q-axis equivalent Inductance	L_q	28.6	mH
Nominal Torque	T_{nom}	20	Nm
Inverter			
Switching frequency	f_{sw}	4	kHz
Dead Time	t_{DT}	1	μs
dc link voltage	v_{dc}	360	V
Sensors			
Encoder	n_{div}	512	–
Controller			
Sampling frequency	f_s	4	kHz
Proportional gain PI_λ (150 Hz)	$k_{p\lambda}$	942.5	$1/s$
Integral gain PI_λ ($k_{i\lambda}/k_{p\lambda}$ = 1/15 ·ω_b)	$k_{i\lambda}$	59,218	$1/s$
Proportional gain PI_δ	$k_{p\delta}$	942.5·λ	V
Integral gain PI_δ	$k_{i\delta}$	59,218·λ	V
Frequency threshold for Flux Observer	g	20	Hz
Flux Weakening coefficient	k_{FW}	0.9	–
PLL bandwidth	$\omega_{b,PLL}$	188.5	rad/s
PLL phase-margin	$\phi_{M,PLL}$	60	°
Over current protection	I_{max}	45	A
Over speed protection	ω_{max}	6050	rpm
Over voltage protection	V_{max}	450	V
Maximum torque	T_{max}	35	Nm

Table 4.2: IPM - Simulation parameters

It must be noticed that the proportional and integral gain of both the flux and load angle regulator are the same as the ones adopted for the Induction Motor. Since the switching frequency is always equal to 4 KHz no change is required for these variables. Also the PLL bandwidth and phase margin are the same.

On the contrary the frequency threshold of the flux observer is moved to 20 Hz, since

the flux model of the machine based on accurate flux map and can be used also in a higher speed range.

Motor and Generator test

As for the Induction Motor, also for the IPM the test in motor and generator condition are performed. The general behavior is the same of the AM, but the specific values assumed by the variables are, of course, different.

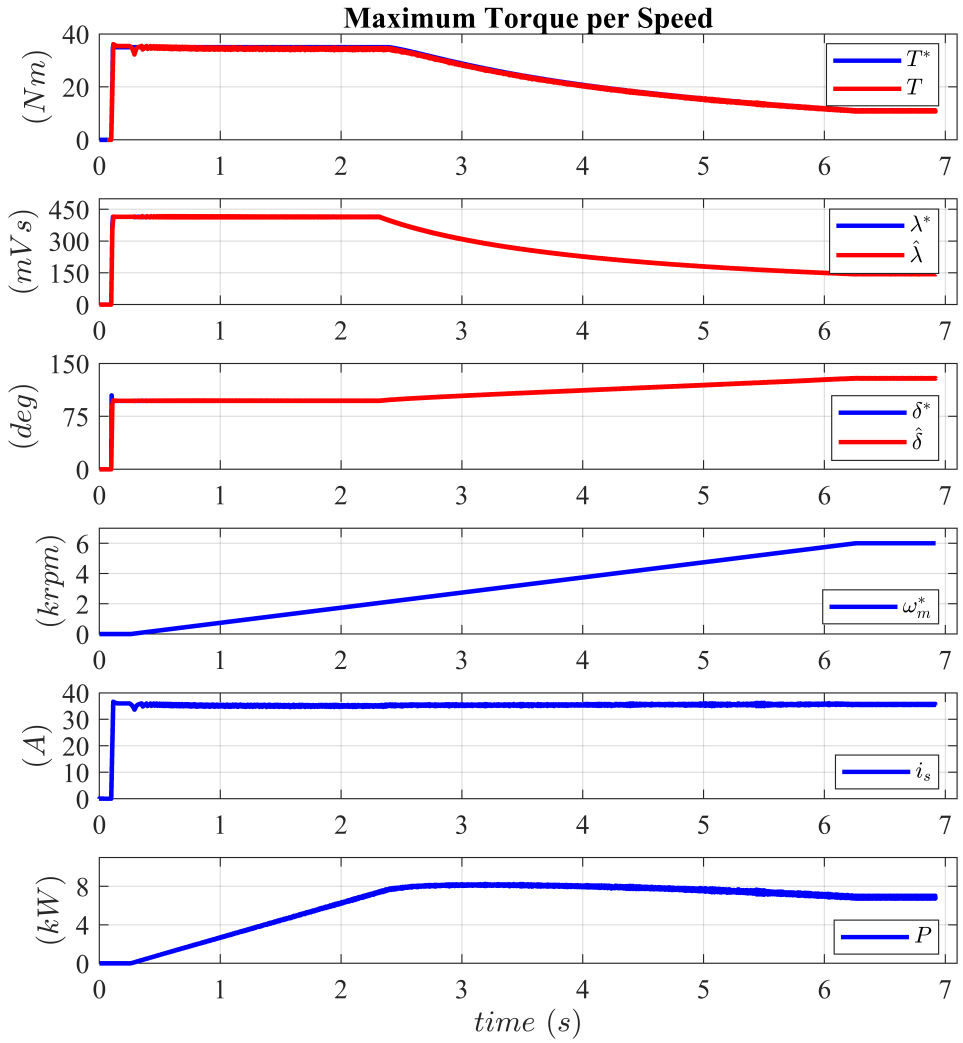


Figure 4.10: IPM Maximum Torque per Speed - Motor @ 360 V

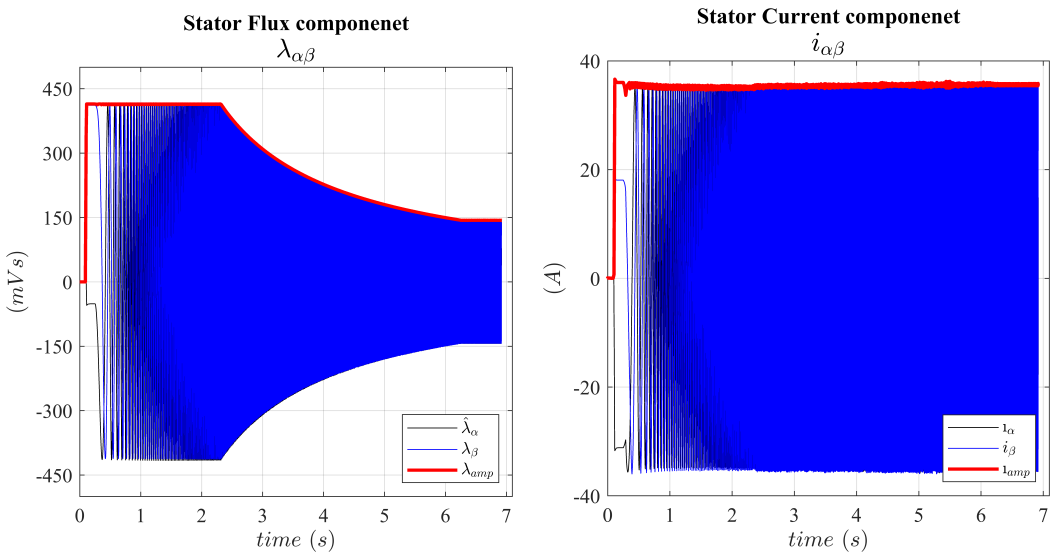
For this type of machine the flux building is not required, since the presence of magnets establish a permanent flux into the machine. Thus the machine is ready to perform FPC as soon as the digital control start working.

The change in the torque impose a variation in the load angle value that, when the reference of flux and torque are established, is imposed equal to its correspondent value, equal to ca. 97°. It overcome this value when speed goes over the base one and move to the final value of 129° when the target speed is reached. In this case this is not yet the *MTPV* load angle: it can be seen by the current amplitude that is always constant and equal to the Current Limit imposed by the control, meaning that the *MTPV* is not used.

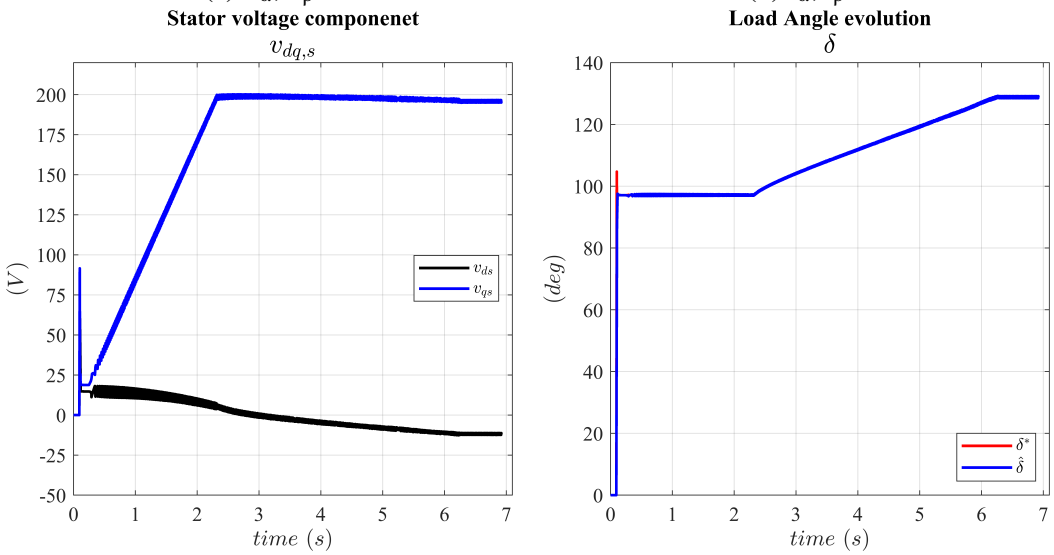
In this type of machine the power tend to remain quite constant when the torque decrease after base speed. This is due to the presence of magnets that impose to the descending torque a behave as *"1/*ω*"*. This, combined with the increasing speed, result into a constant power value.

A more detailed behavior of fluxes, current, reference voltages and load angle is provided in Figure 4.11. The fluxes have the typical behavior that resemble the one of the torque, while the currents are always well controlled and maintained equal to the maximum value.

The reference and obtained load angle are perfectly coincident, meaning an effective control strategy.



(a) $\lambda_\alpha, \lambda_\beta$ (b) i_α, i_β



(c) $v_{dq,s}$ (d) δ

Figure 4.11: IPM Evolution of control variables - Motor @ 360 V

The behave in generator mode is showed in Figure 4.12. The control is fully able to manage the machine also in this condition, without any particular critical issues then the one faced in motor mode.

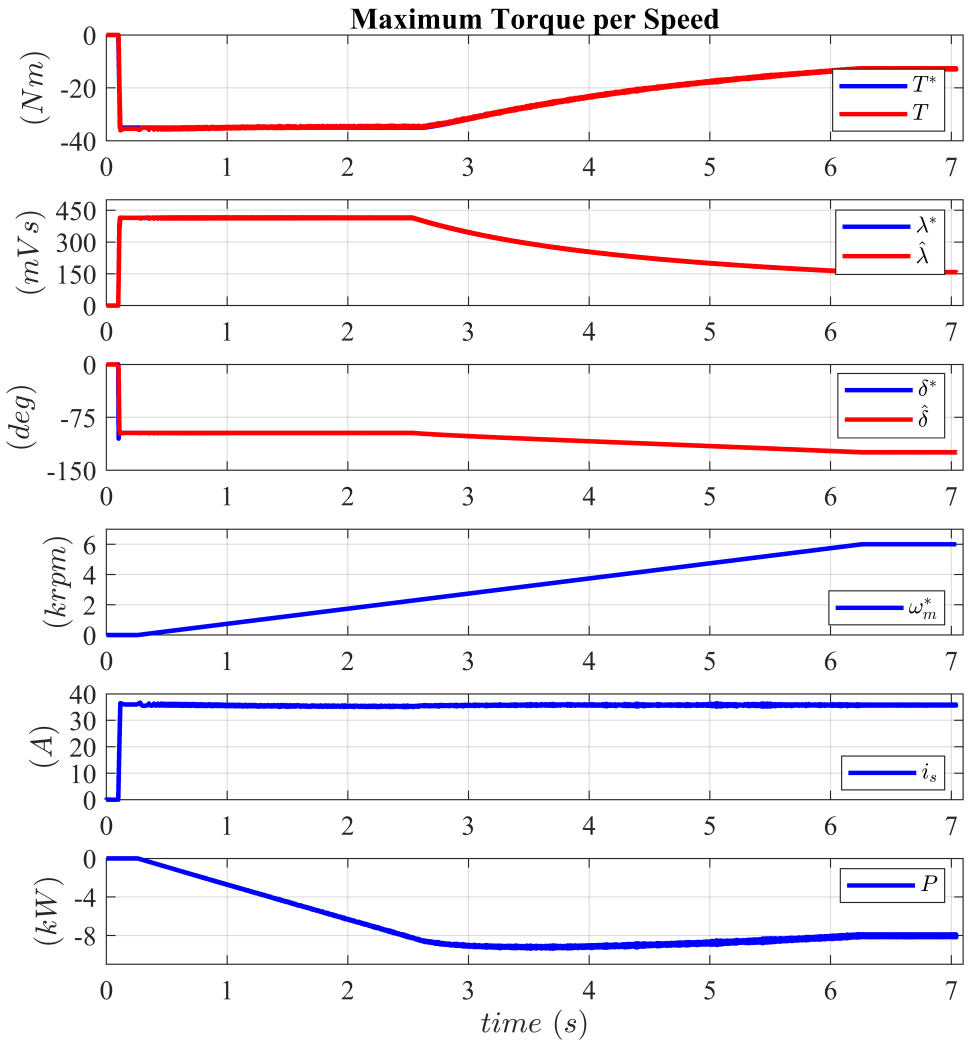
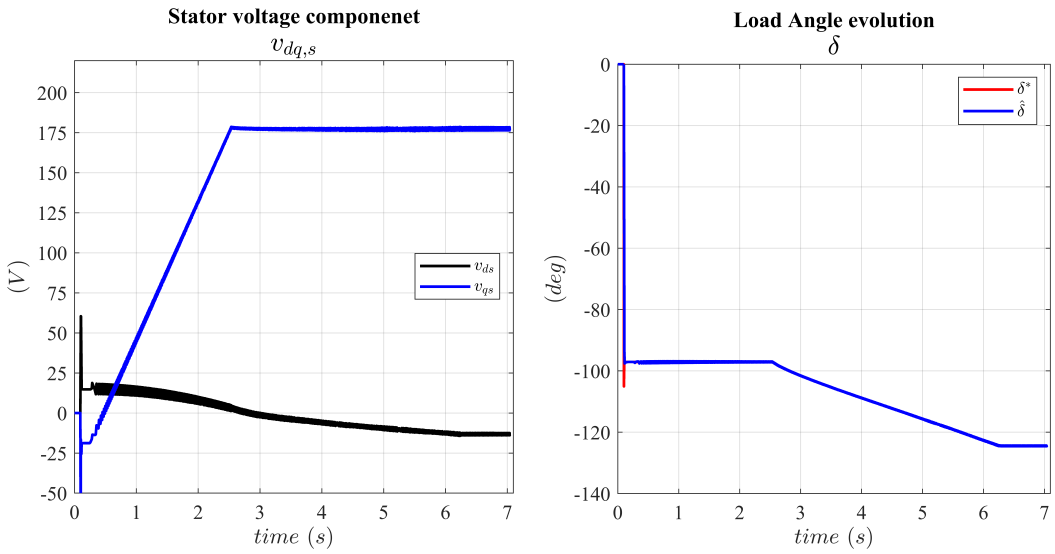


Figure 4.12: IPM Maximum Torque per Speed - Generator @ 360 V

The reference voltages and the load angle are again showed in their evolution in this type of operation.



(a) $v_{dq,s}$ (b) δ

Figure 4.13: IPM Evolution of control variables - Generator @ 360 V

High dynamic test

The high dynamic test is performed in this type of motor: the imposed square reference torque leads to the same behave in all the other control variable that follows the sign of the torque: the load angle and the power. But the former is a cause, while the latter a consequence of the torque variation. In particular the load angle must face a change in its value of more that 200° passing from motor to generator mode, and back. Performing this type of behave at high speed, close the *MTPV* control locus, is a not trivial action. All the others variables perform as yet described for the AM.

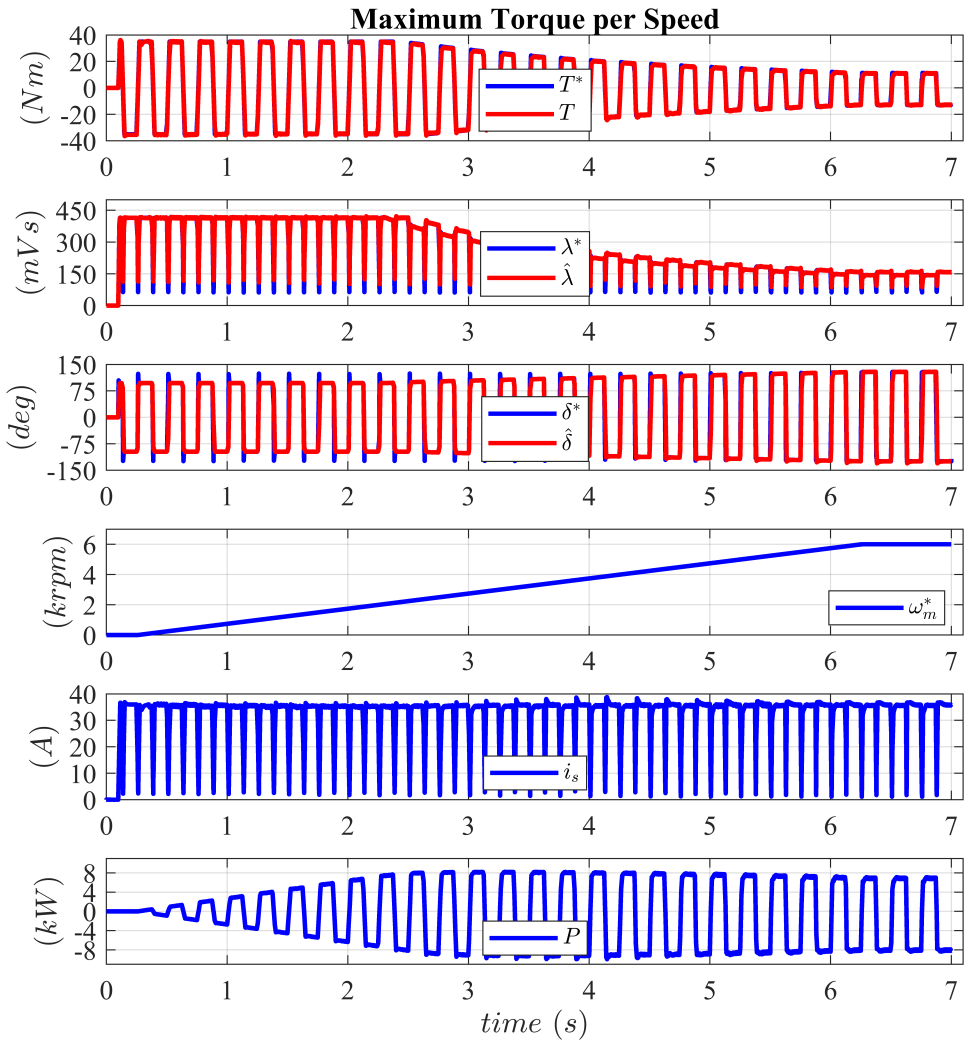
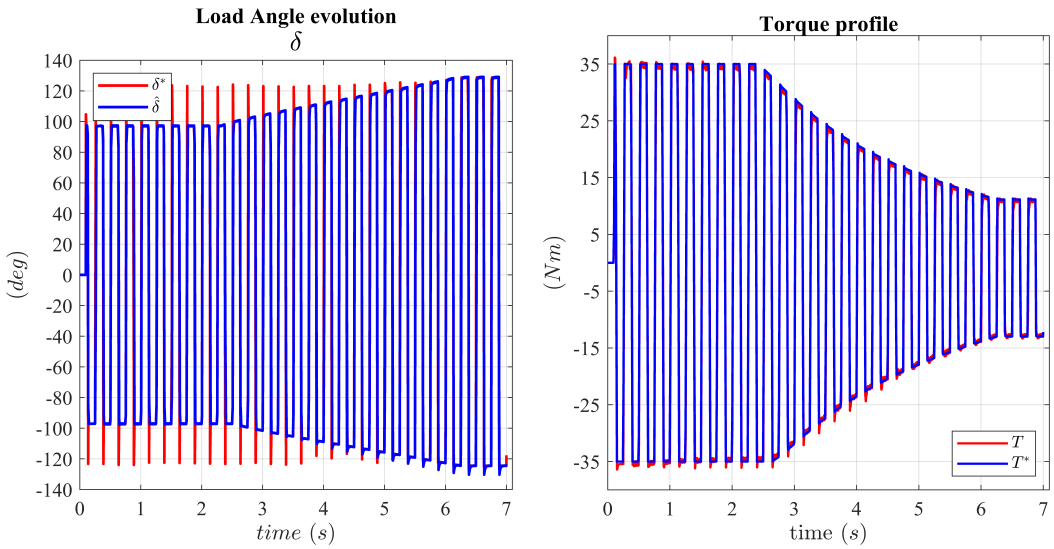


Figure 4.14: IPM Fast Torque reversal at 4 Hz - @ 360 V

The detail of the torque and load angle of this motor are showed in Figure 4.15



(a) δ (b) T

Figure 4.15: IPM Fast Torque reversal at 4 Hz - @ 360 V

Accuracy test

Finally the accuracy test is performed, requiring to the machine at 1500 rpm all the possible torque values, from the highest negative to the maximum positive. Also for this machine, as for the AM, the result obtained is satisfying with a good correspondence between the desired value and the result get.

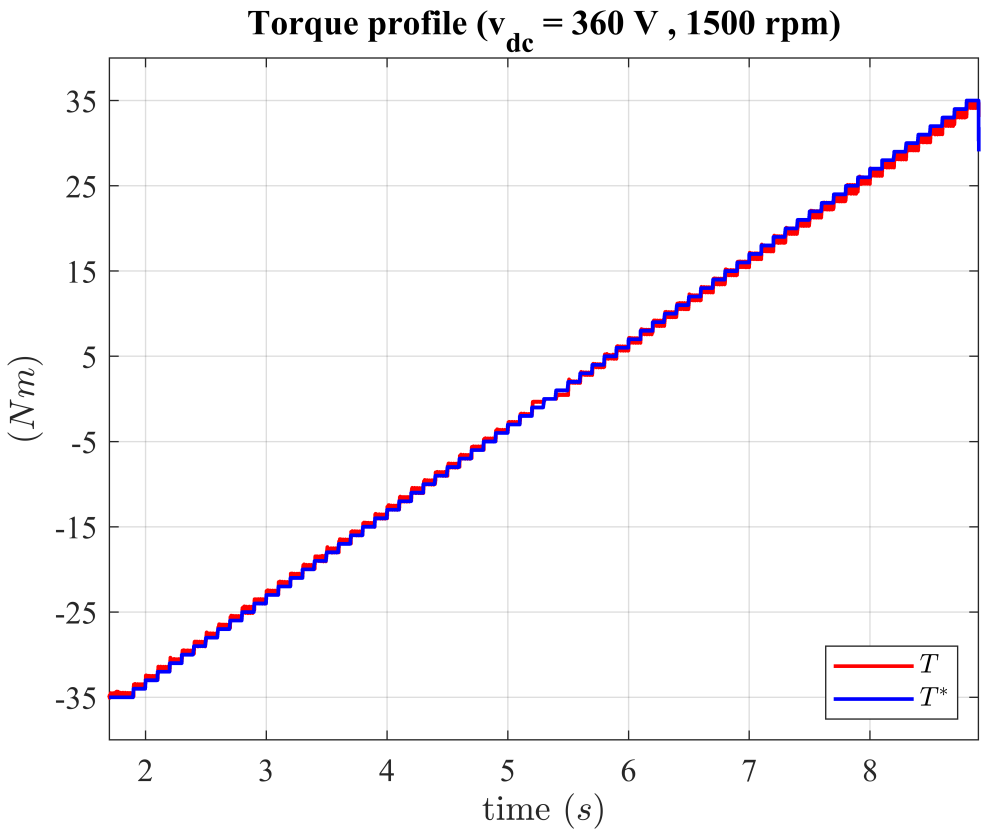


Figure 4.16: IPM Torque accuracy test @ (360 Vdc, 1500 rpm)

4.2.3 Surface Permanent Magnets Motor

The third machine on which the FPC has been tested is a SM with surface mounted magnets. The main machine parameter and control settings are in Table 4.3.

Motor			
Pole Pairs	pp	18	–
Stator Resistance	R_s	7.4	Ω
d-axis equivalent Inductance	L_d	70	mH
q-axis equivalent Inductance	L_q	70	mH
Nominal Torque	T_{nom}	16	Nm
Inverter			
Switching frequency	f_{sw}	5	kHz
Dead Time	t_{DT}	1	μs
dc link voltage	v_{dc}	360	V
Sensors			
Encoder	n_{div}	1024	–
Controller			
Sampling frequency	f_s	5	kHz
Proportional gain PI_λ (150 Hz)	$k_{p\lambda}$	942.5	$1/s$
Integral gain PI_λ ($k_{i\lambda}/k_{p\lambda}$ = 1/15 ·ω_b)	$k_{i\lambda}$	59,218	$1/s$
Proportional gain PI_δ	$k_{p\delta}$	942.5·λ	V
Integral gain PI_δ	$k_{i\delta}$	59,218·λ	V
Frequency threshold for Flux Observer	g	20	Hz
Flux Weakening coefficient	k_{FW}	0.9	–
PLL bandwidth	$\omega_{b,PLL}$	188.5	rad/s
PLL phase-margin	$\phi_{M,PLL}$	60	°
Over current protection	I_{max}	10	A
Over speed protection	ω_{max}	1800	rpm
Over voltage protection	V_{max}	450	V
Maximum torque	T_{max}	29	Nm

Table 4.3: SPM - Simulation parameters

Also in this case the proportional and integral gain of both the flux and load angle regulator are the same as the ones adopted for the Induction Motor and the IPM machine. Also the PLL bandwidth and phase margin are the same.

Motor and Generator test

The FPC control is tested on the machine performing a motor and generator test. The result in motor mode are showed in Figure 4.17.

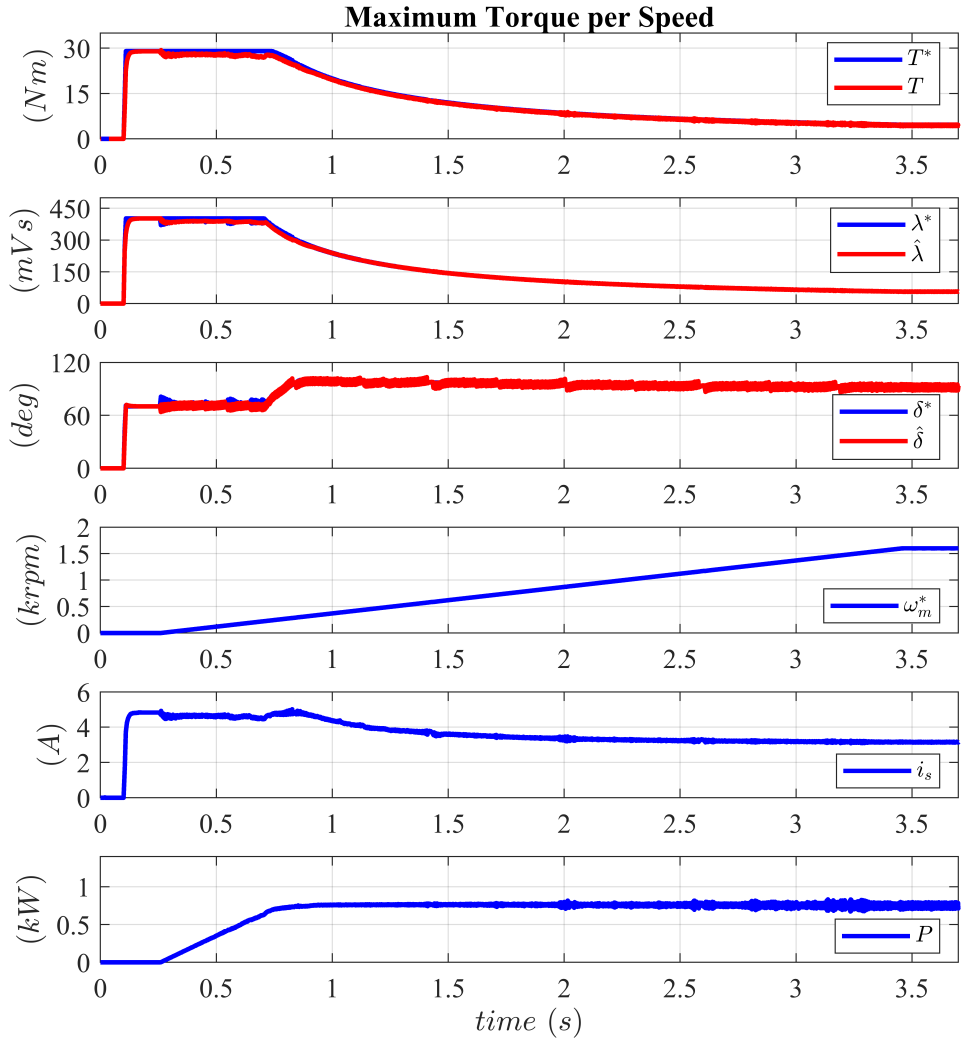


Figure 4.17: SPM Maximum Torque per Speed - Motor @ 360 V

For this type of machine the flux building phase is unnecessary, being equipped with magnets. As soon as the torque reference change to the maximum reachable value, the machine is able to follow instantaneously the target with the one produced. All the variable adapt their values to the ones in *MTPA* condition: the load angle, the flux and the current are the ones in the extreme point of *MTPA*, where the torque is at its maximum level.

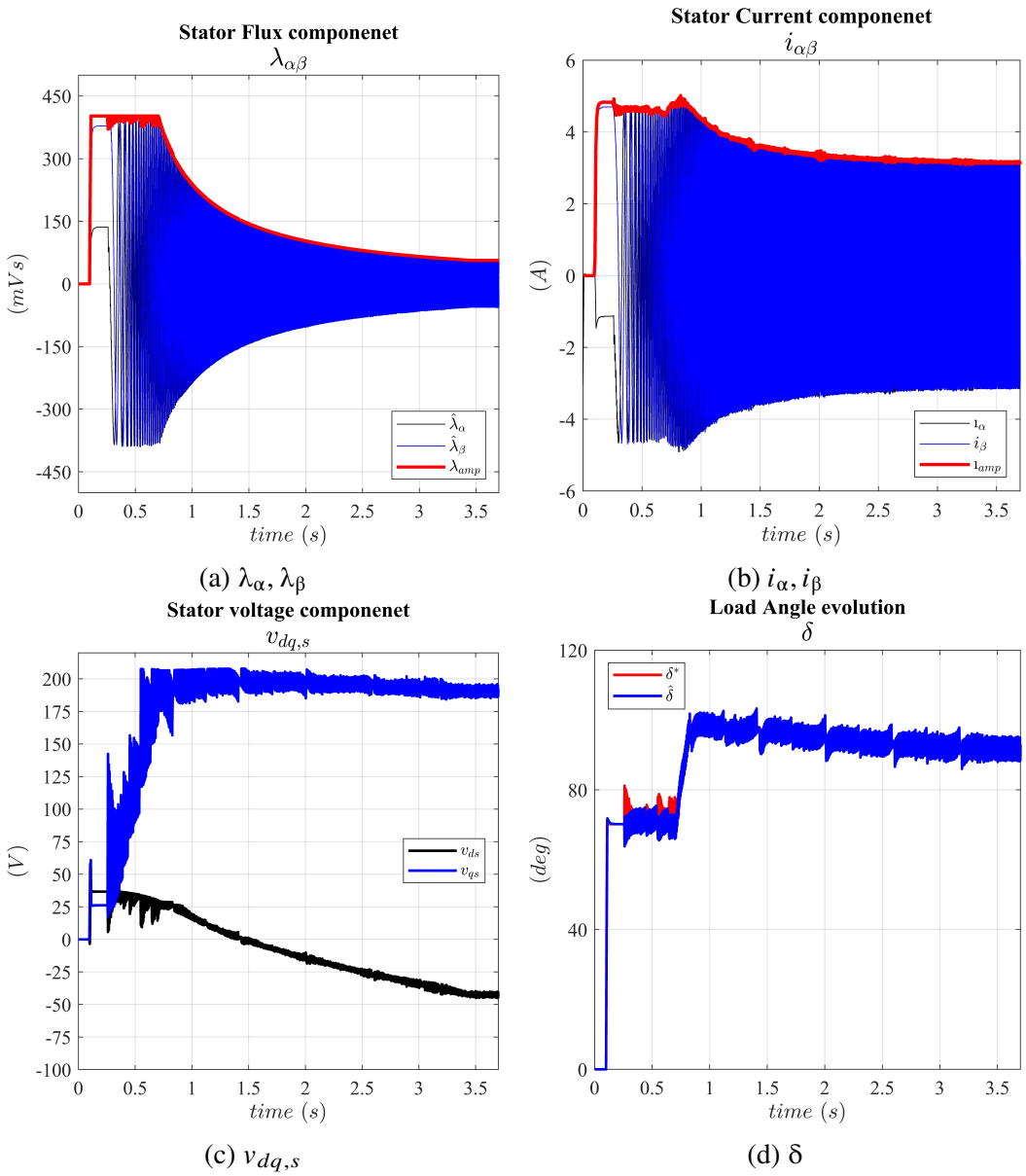
When the speed rises over the base one, the Current Limit is walked quickly, reaching

the *MTPV* with a small speed increase. This is evident both from the current that start to reduce its value from this point, and from the load angle value that stabilize to the *MTPV* value of 90°.

The power, after the base speed is maintained always constant, thanks to the magnets. This is a typical behave of machine with magnets.

The detail of the evolution of the main variables during the test is showed in Figure 4.18. The flux weakening performed is able to reduce the amplitude of flux from the starting value of 400 mVs to the final equal to 50 mVs. This means a reduction of 8 time, that is quite impressive. The current, as anticipated, reduce its value after the CL, when the *MTPV* is explored with the speed rising. The voltages present the typical decoupled behavior already experienced for the other machine.

Finally the desired and the actual load angle are plotted: they are perfectly overlapped, proving the control accuracy.



(a) $\lambda_\alpha, \lambda_\beta$ (b) i_α, i_β

(c) $v_{dq,s}$ (d) δ

Figure 4.18: SPM Evolution of control variables - Motor @ 360 V

In Figure 4.19 the generator test are showed. The results are the same as in motor-mode but with the torque, load angle and power values with the opposite sign.

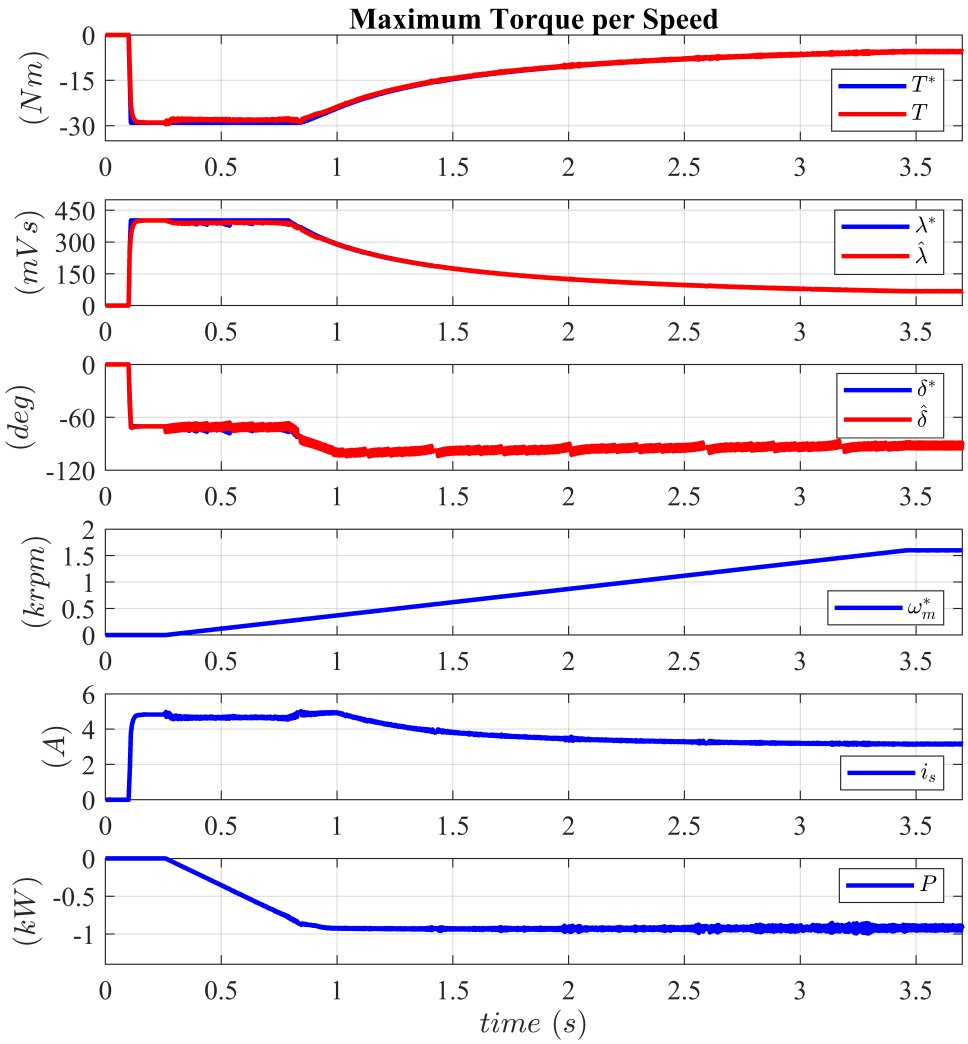
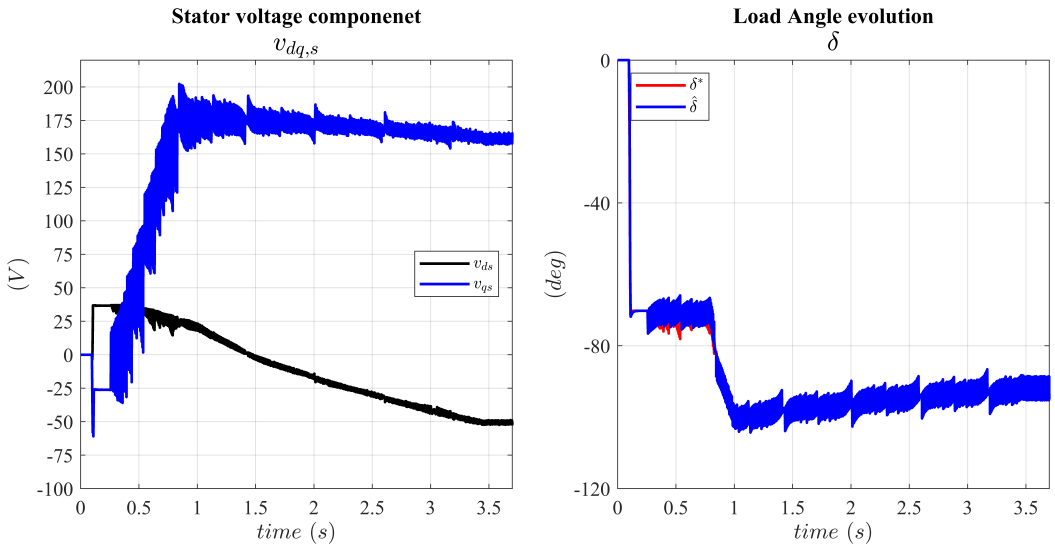


Figure 4.19: SPM Maximum Torque per Speed - Generator @ 360 V

The detail of voltages and load angle are showed in Figure 4.20.



(a) $v_{dq,s}$ (b) δ

Figure 4.20: SPM Evolution of control variables - Generator @ 360 V

High dynamic test

The dynamic test is performed with a reference square torque switching at 2 Hz from motor to generator. The results are the ones of the two previous tests combined to produce the desired level of torque in both the configuration.

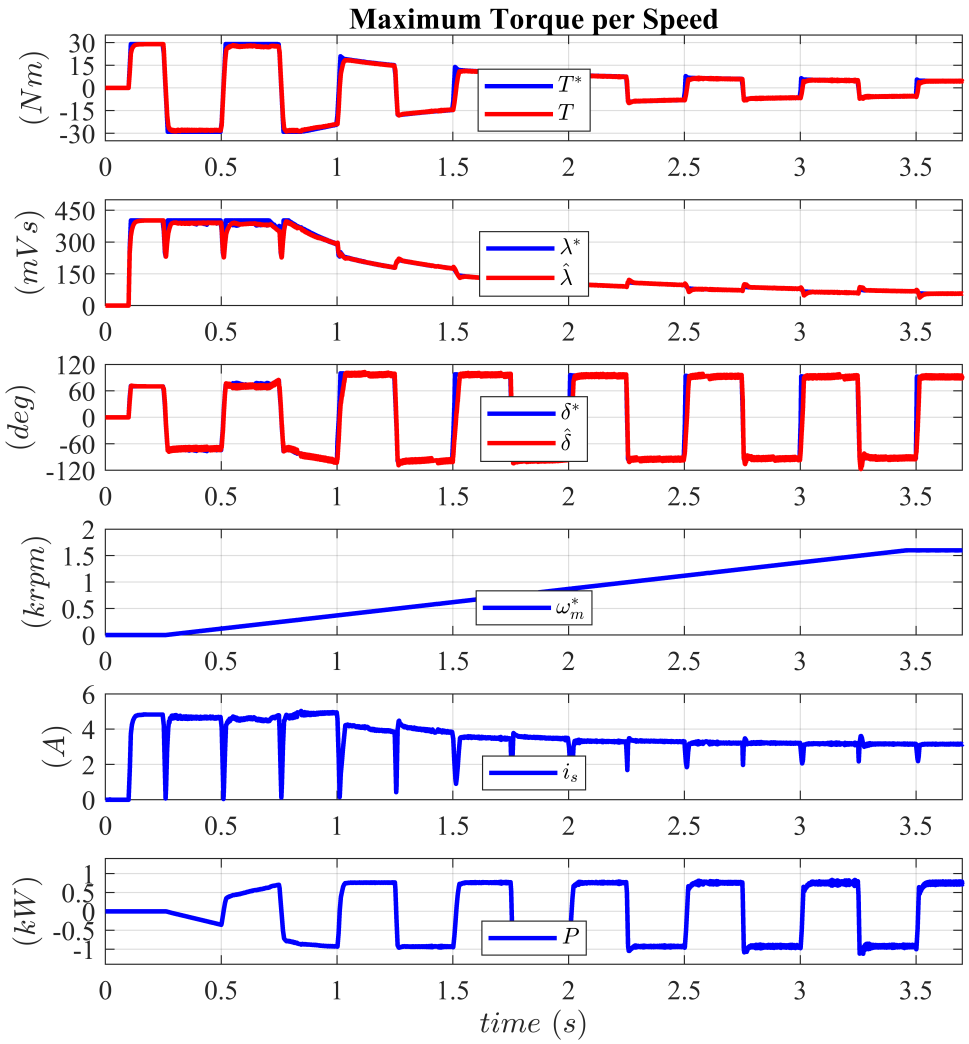
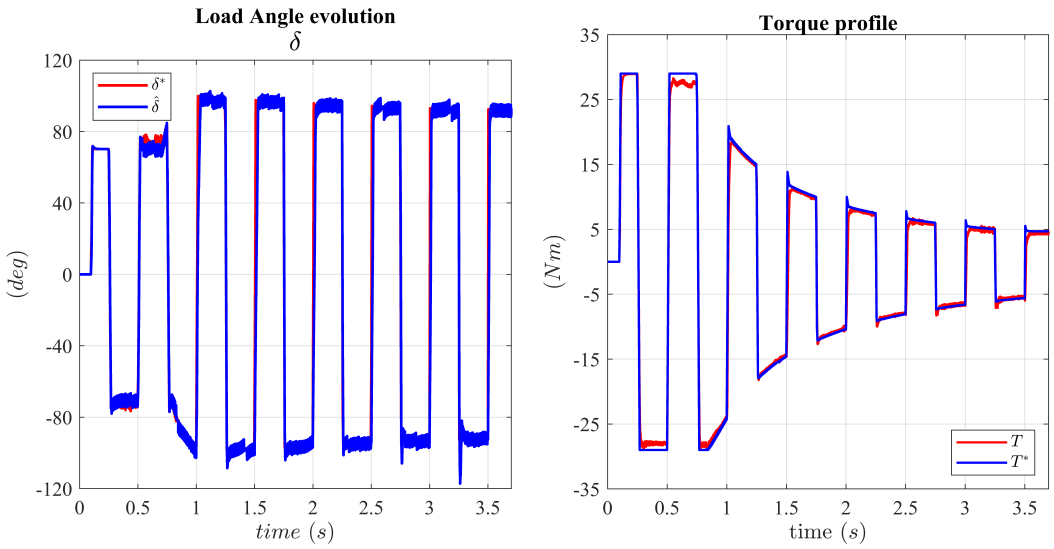


Figure 4.21: SPM Fast Torque reversal at 2 Hz - @ 360 V

The load angle and torque perfectly reproduce the reference values. In particular at high speed in *MTPV* condition the load angle jump from +90° to -90°, imposing an overturning to the flux vector every time the torque reverse its value.



(a) δ (b) T

Figure 4.22: SPM Fast Torque reversal at 2 Hz - @ 360 V

Accuracy test

The last test is the accuracy one: it is performed at 200 rpm where the machine can reach the extreme torque values. The accuracy between the desired and the actual torque is quite satisfying, unless for high torque value where they are not coincident. This can be due to the reduction of accuracy of magnetic maps for high current levels, like the ones necessary to obtain this levels of torque.

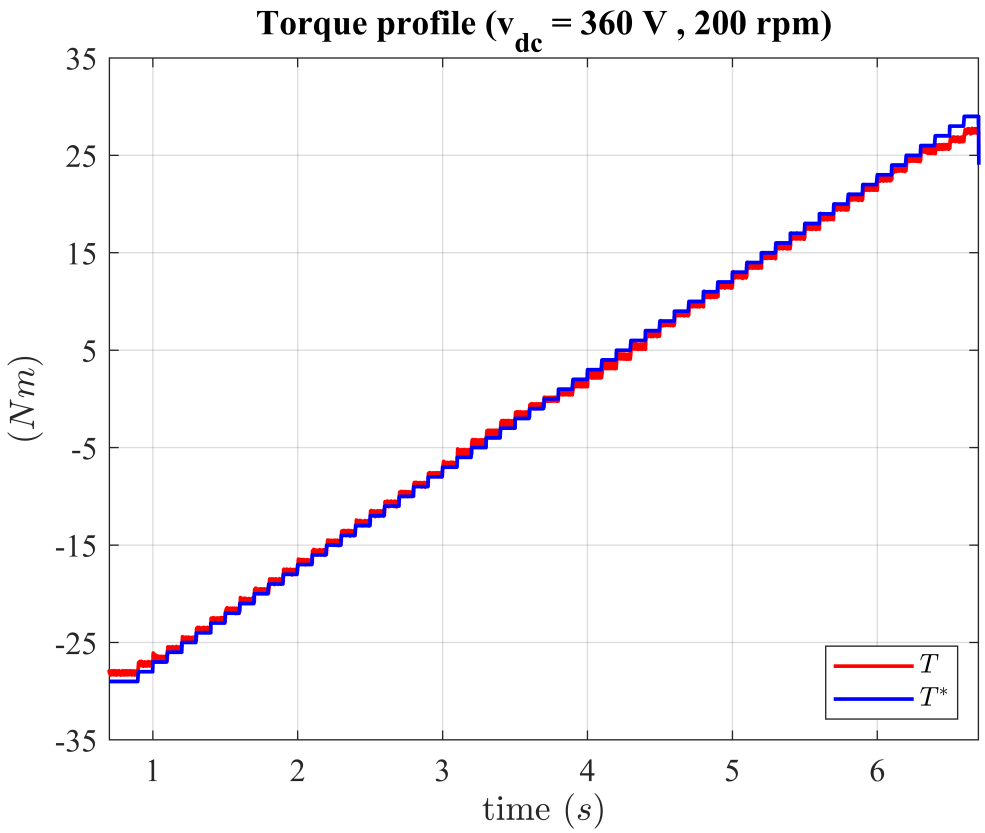


Figure 4.23: SPM Torque accuracy test @ (360 Vdc, 200 rpm)

4.2.4 Synchronous Reluctance Motor

The last motor used to evaluate the performance of the torque controller is the magnet-less reluctance SM. The main machine parameter and control settings are in Table 4.4.

Motor			
Pole Pairs	pp	2	–
Stator Resistance	R_s	3.6	Ω
d-axis equivalent Inductance	L_d	51.5	mH
q-axis equivalent Inductance	L_q	147.5	mH
Nominal Torque	T_{nom}	36	Nm
Inverter			
Switching frequency	f_{sw}	4	kHz
Dead Time	t_{DT}	1	μs
dc link voltage	v_{dc}	360	V
Sensors			
Encoder	n_{div}	1024	–
Controller			
Sampling frequency	f_s	4	kHz
Proportional gain PI_λ (150 Hz)	$k_{p\lambda}$	942.5	$1/s$
Integral gain PI_λ ($k_{i\lambda}/k_{p\lambda}$ = 1/15 ·ω_b)	$k_{i\lambda}$	59,218	$1/s$
Proportional gain PI_δ	$k_{p\delta}$	942.5·λ	V
Integral gain PI_δ	$k_{i\delta}$	59,218·λ	V
Frequency threshold for Flux Observer	g	20	Hz
Flux Weakening coefficient	k_{FW}	0.9	–
PLL bandwidth	$\omega_{b,PLL}$	188.5	rad/s
PLL phase-margin	$\phi_{M,PLL}$	60	°
Flux Building value: $\lambda(T_{nom})$	λ_{FB}	900	$mVs°$
Over current protection	I_{max}	50	A
Over speed protection	ω_{max}	12000	rpm
Over voltage protection	V_{max}	450	V
Maximum torque	T_{max}	52.5	Nm

Table 4.4: SYR - Simulation parameters

Also in this case the proportional and integral gain of both the flux and load angle regulator are the same as the ones adopted for the Induction Motor and the IPM machine. Also the PLL bandwidth and phase margin are the same.

Motor and Generator test

The first test to be performed is the one in motor-mode before, the flux building is required to bring the flux to a minimum value guarantee. The value of λ_{FB} has been set equal to the flux correspondent to the nominal torque in *MTPA* condition. A small amount of current is present in the machine to produce this minimum flux. When the amplitude reaches the target value, the FPC can start to work.

It is evident that the actual value of flux is not enough to satisfy the torque request of the machine, that is close to the double of the nominal torque. For this reason the flux increase again when the torque step change is applied. The behave of all the variables is showed in Figure 4.24.

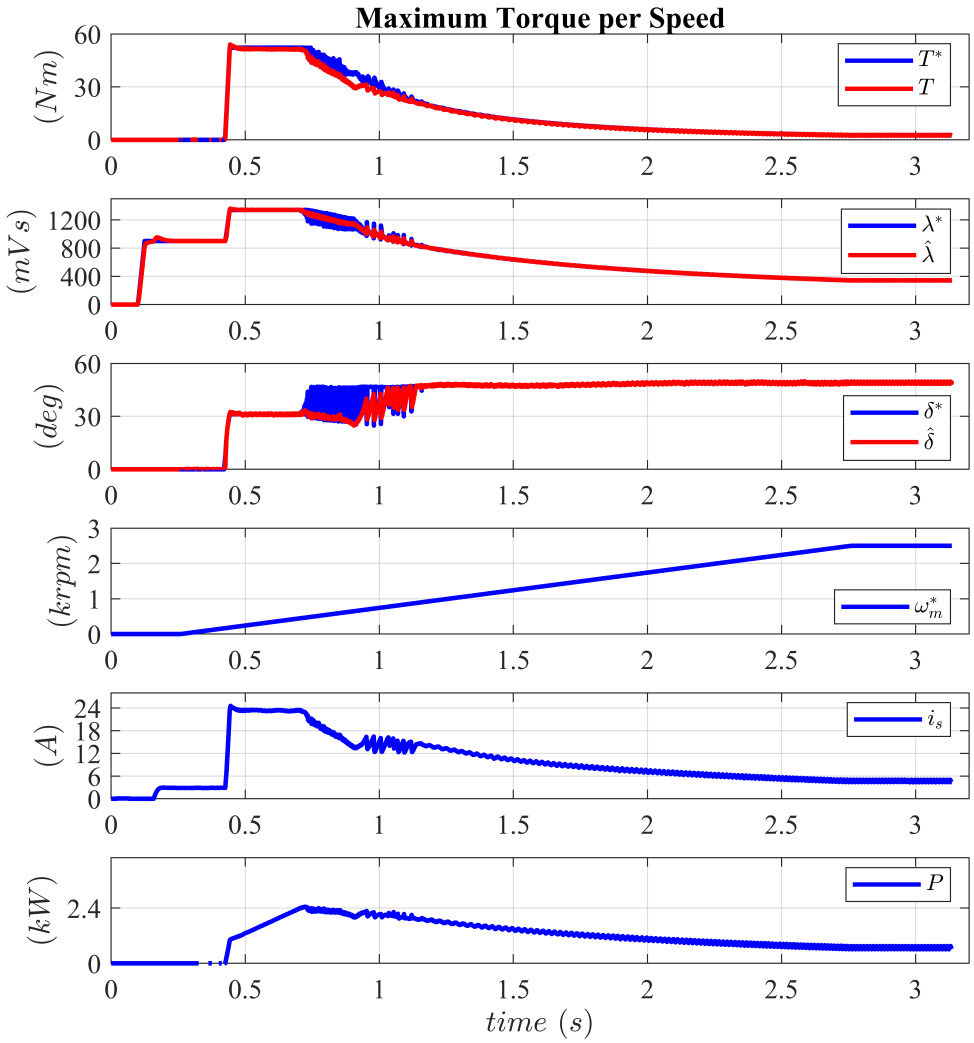
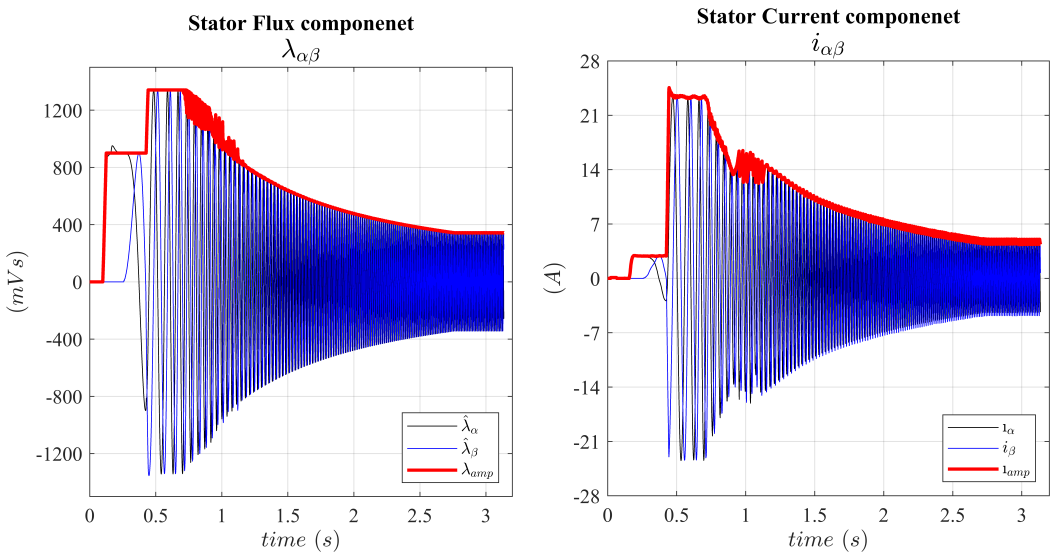


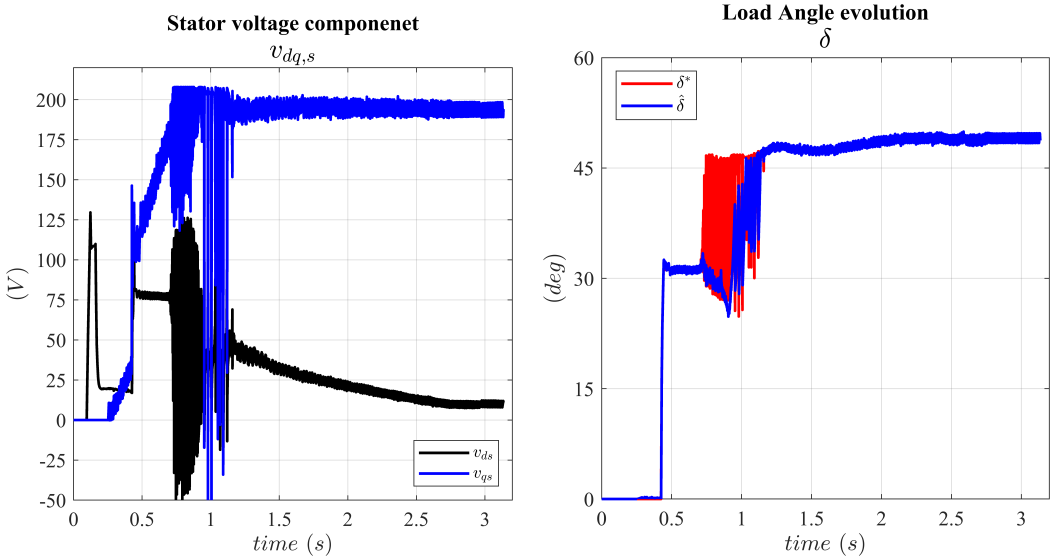
Figure 4.24: SYR Maximum Torque per Speed - Motor @ 360 V

The value in the extreme *MTPA* point is maintained by all the variable until base speed is reached. At this point the working point start to move along the Current Limit and the flux and torque value decrease, while the load angle is moving to its characteristic value in *MTPV* condition. The *MTPV* is rapidly reached after a small increase in speed and the load angle value is established permanently. All the variable evolve in *MTPV* with a perfect correspondence between the reference and the obtained values. It can be noticed how the current reduce its value after the *MTPA* and particularly in the *MTPV* condition. This behave is highlighted in Figure 4.25.

The transition between *MTPA* and *MTPV* in this case is quite confusing, but, in the end, the machine is perfectly controlled once the *MTPV* is reached.



(a) $\lambda_\alpha, \lambda_\beta$ (b) i_α, i_β



(c) $v_{dq,s}$ (d) δ

Figure 4.25: SYR Evolution of control variables - Motor @ 360 V

The generator-mode tests in Figure 4.26 reproduce exactly the motor one with the same evolution of variable. Also in this case the peak value of power in generator-mode is higher then the one reached in motor-mode: this is due to the account of power losses in the machine. For this type of machine the power highly change after the base speed: being a magnet-less machine the torque fall with a slope higher then $1/\omega$ when the flux weakening start to work causing a high reduction in the power level.

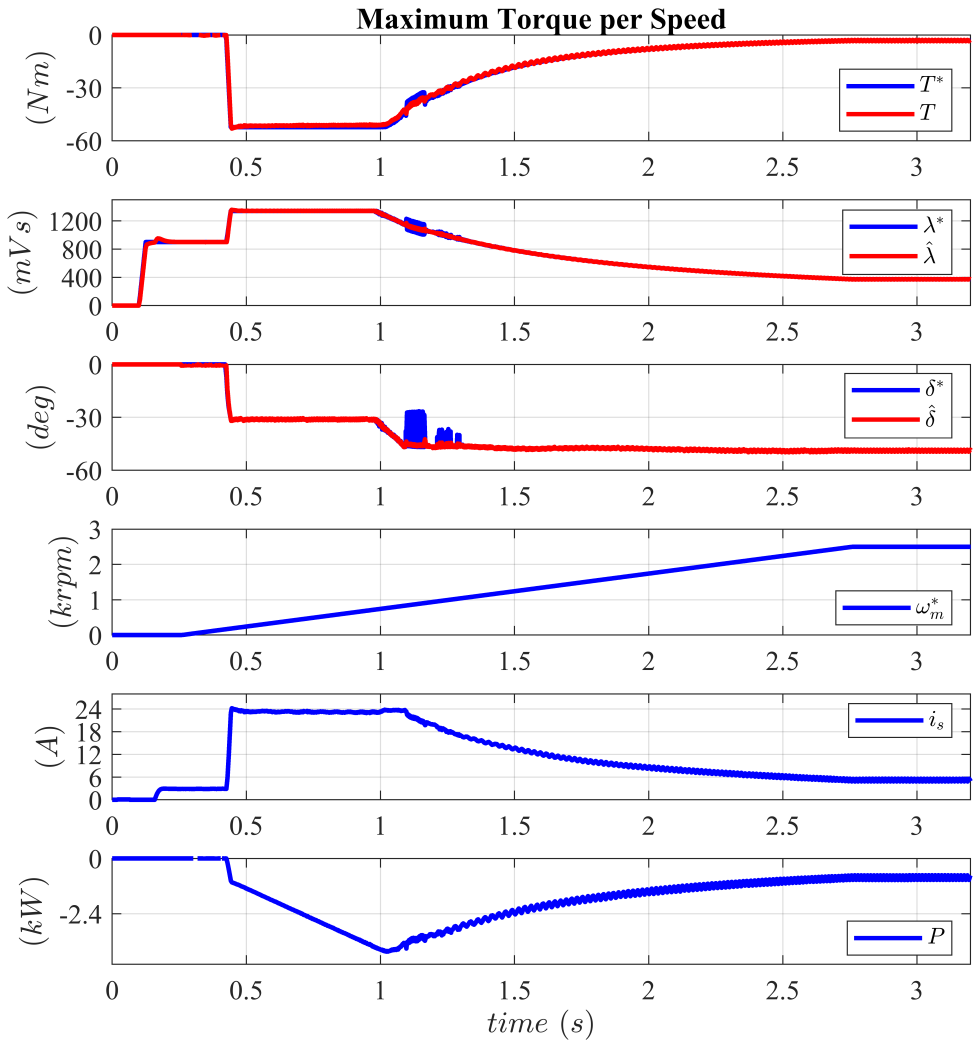
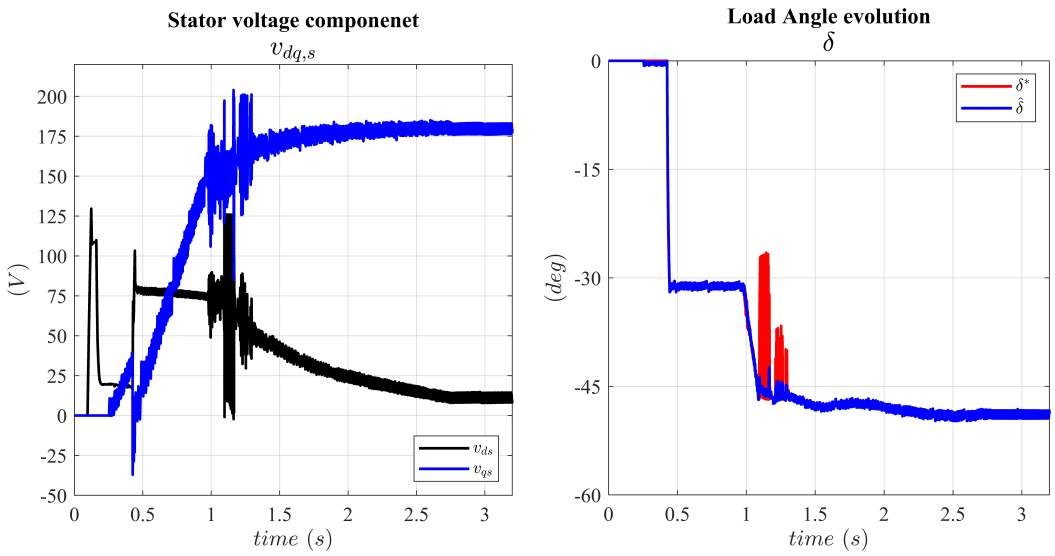


Figure 4.26: SYR Maximum Torque per Speed - Generator @ 600 V

The evolution of voltages and load angle is in Figure 4.27.



(a) $v_{dq,s}$ (b) δ

Figure 4.27: SYR Evolution of control variables - Generator @ 360 V

High dynamic test

In Figure 4.28 the high dynamic test on the synchronous reluctance machine is showed. As for the other machines the motor and generator behavior are exploited in this type of high-stress test.

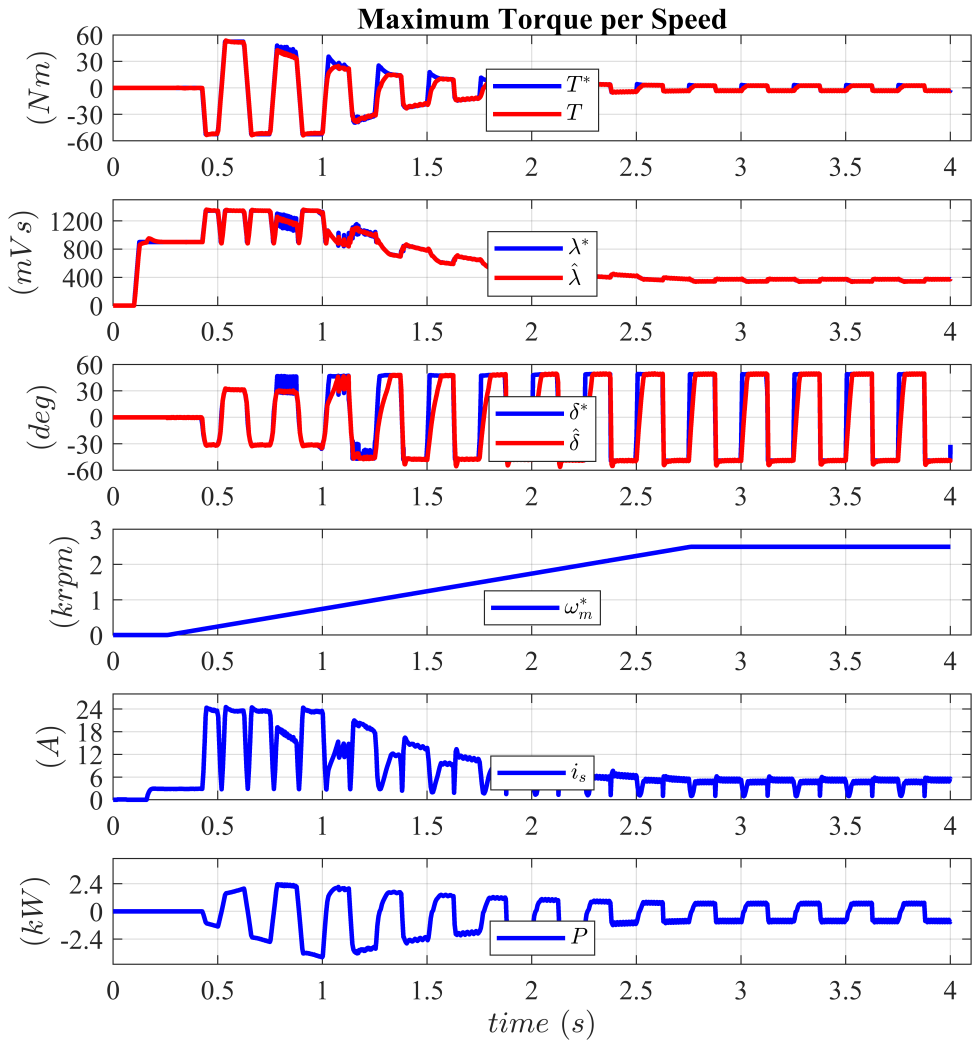
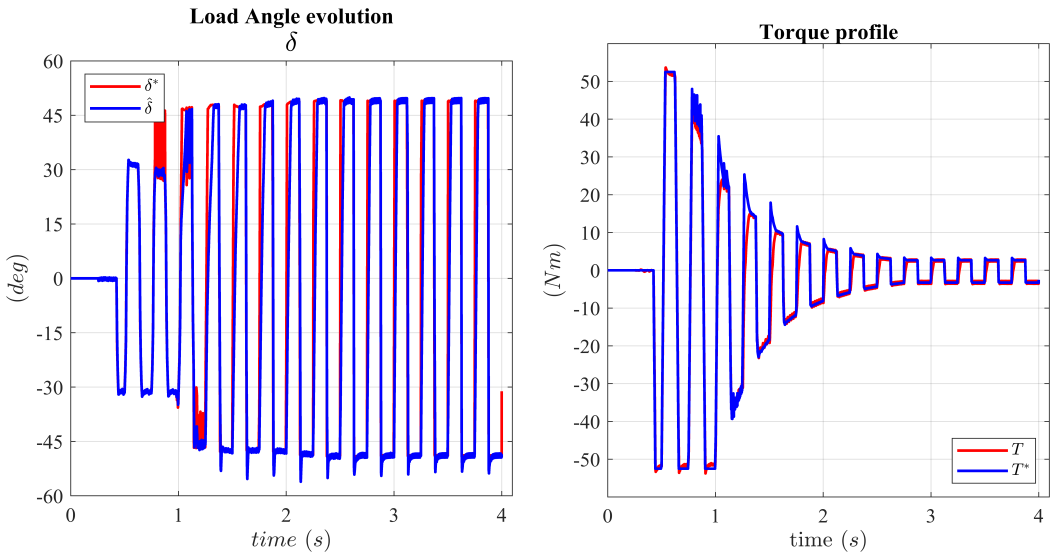


Figure 4.28: SYR Fast Torque reversal at 2 Hz - @ 360 V

The load angle and the torque values during the test are showed in Figure 4.29. The flux vector in *MTPV* must face a rotation of 90° when the reference torque reverse its value.



(a) δ (b) T

Figure 4.29: SYR Fast Torque reversal at 2 Hz - @ 360 V

Accuracy test

The accuracy test, at 400 rpm, is showed in Figure 4.30. The reference torque change with a stair from the lower minimum to the upper maximum values in order to analyze the behave in all the possible torque levels. The results are satisfying, being the two plot overlapped in a quite good way.

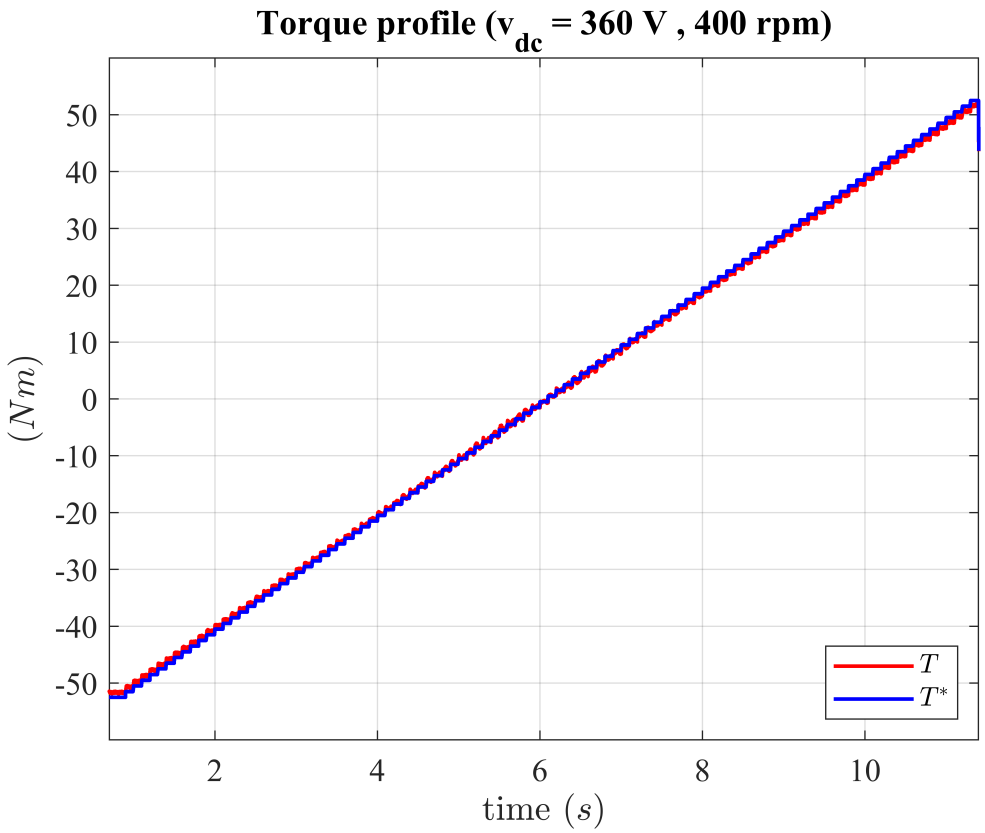


Figure 4.30: SYR Torque accuracy test @ (360 Vdc, 400 rpm)

4.3 Experimental validation

Despite the simulation results can be considered a valid starting point on which improve the control performance and accuracy, the proving results of the controller capability are the one obtained from the experimental simulations on the real machine. In this case, the simulation test have been carry out only on the IM and IPM machine.

Test bench realization

The experimental test have been realized in the PEIC (Power Electronic Innovation Center) laboratory of Politecnico di Torino. The test bench, showed in Figure 4.31, is composed by a number of element that are the same modeled in the simulation environment in Simulink.

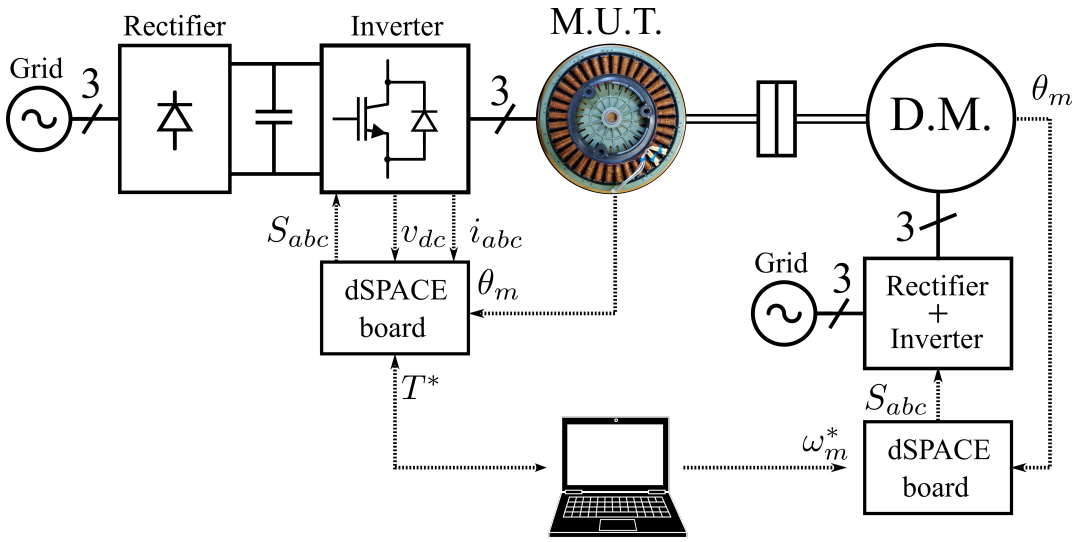


Figure 4.31: Test bench for the experimental validation of the FPC

The main element are:

- an ac/dc rectifier: it is a battery emulator, that is able to produce the desired dc voltage v_{dc} in output.
- a dc/ac inverter: given the input dc link voltage v_{dc} and the switching signals S_{abc} obtained by the PWM modulator in the dSPACE board, produces the desired voltages v_{abc} that are used to feed the machine. From it, the actual current value and the dc link voltages are measured and their value is used to run the control code, i.e. real voltages evaluation with dead time.

- the MUT (Machine Under Test) is the machine on which the FPC is applied to impose the desired level of torque. It is controlled via the v_{abc} voltages in input that are the one needed to obtain the reference torque value. The angular position of the machine is measured with an encoder and provided to the control.
- the dSPACE board, together with the working station, is the core of the machine control. It collects all the measurement (i_{abc} current, v_{dc} dc link voltage and mechanical speed ω_m) and receive from the control the user-defined reference torque T^* to be obtained from the machine. With this information, running the control code, the duty cycles are evaluated and finally the command for the six inverter switches produced.
- On the other hand the driving machine must be controlled to impose the desired speed to the MUT. This is done with a different dSPACE board that perform the speed control of the DM defining the command for the inverter adopted, together with the rectifier, to control this machine.
- Finally the driving machine is mechanically joined with the MUT to execute the speed management in the system.

4.3.1 Induction Motor

The experimental results on the induction motor are here showed. Several test have been carry out in order to evaluate the performance of the machine with the FPC torque controller.

- motor and generator test to obtain the maximum torque per speed behavior of the machine;
- high dynamic test with reference square torque reversing at 4 Hz with a slope of 5 kNm/s;
- accuracy test to verify the accuracy of the torque controller.

The machine parameters are the same used for simulation reported in Table 4.1.

Motor and Generator test

The maximum torque per speed profile test in motor configuration in Figure 4.32 have been carry out on the machine with a dc voltage of 600 V. The behave is the one obtained in simulation, proving the accuracy of the model realized.

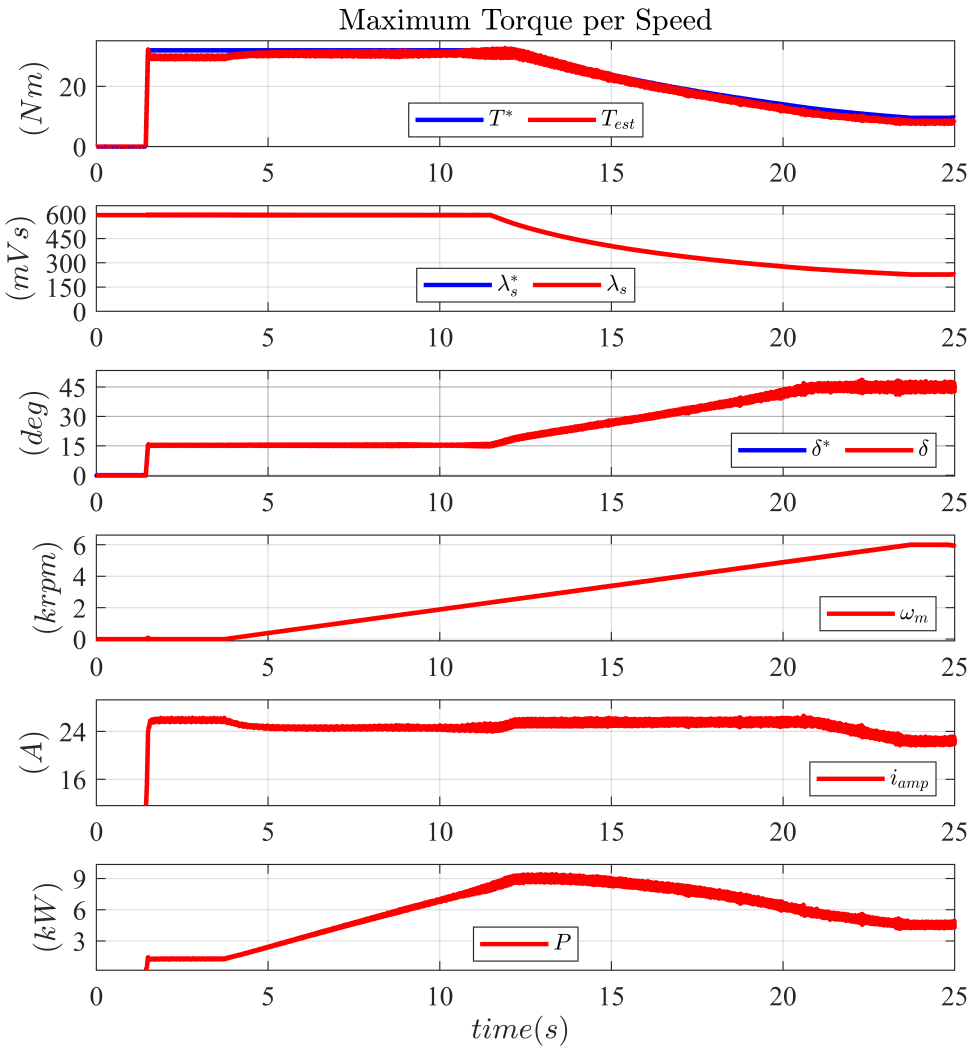
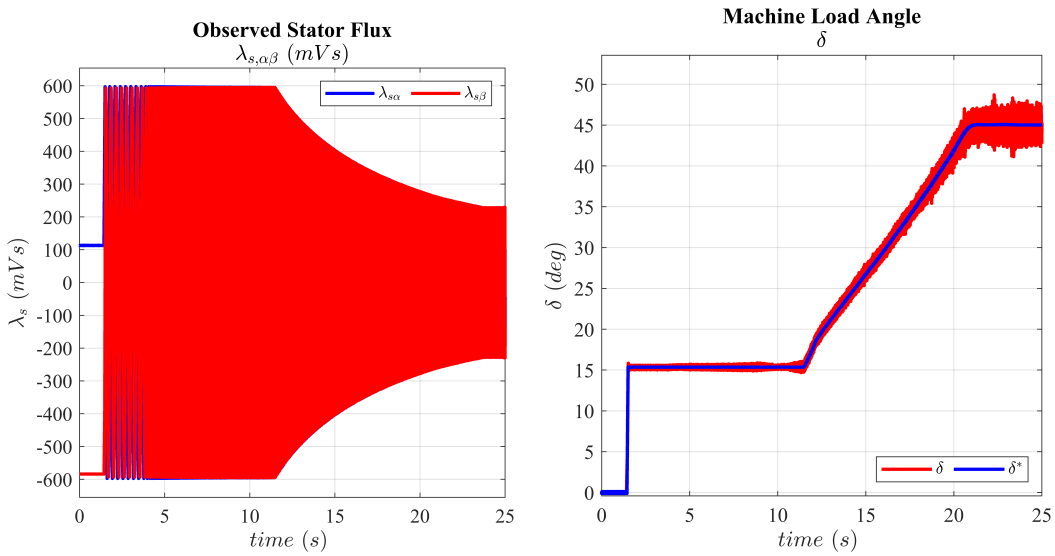
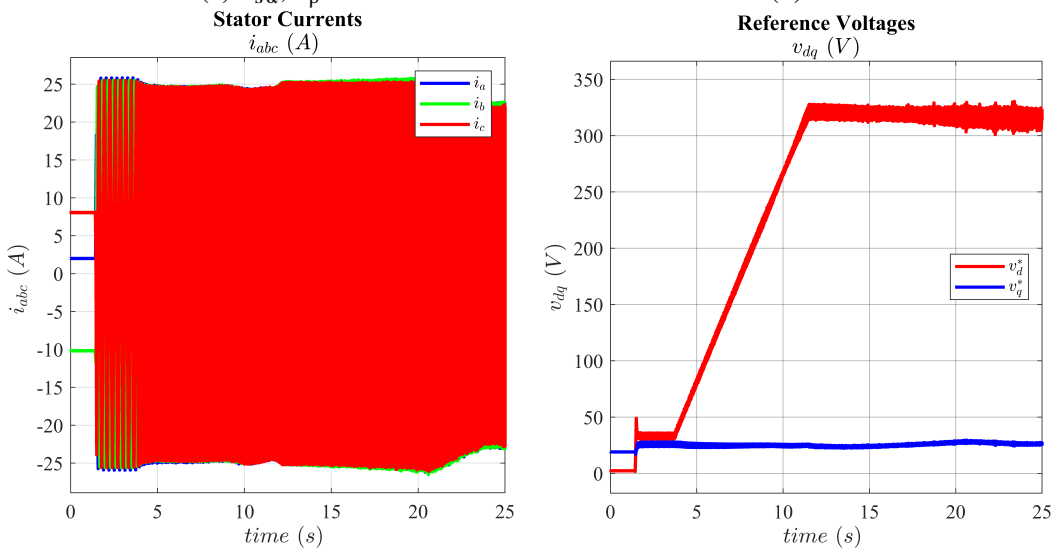


Figure 4.32: IM - Overview of main variables behave in motor experimental test @ 600V

The detailed behave of the fluxes, the load angle, the stator current and the references voltages is showed in Figure 4.33. The results are as expected from simulation.



(a) $\lambda_{s\alpha}, \lambda_{\beta}$ (b) δ



(c) i_{α}, i_{β} (d) v_d, v_q

Figure 4.33: IM - Fluxes, load angle, stator current and reference voltage in motor experimental test @ 600V

The generator-mode test is performed to guarantee the capability of control of FPC on both machine configuration. The results are showed in Figure 4.34.

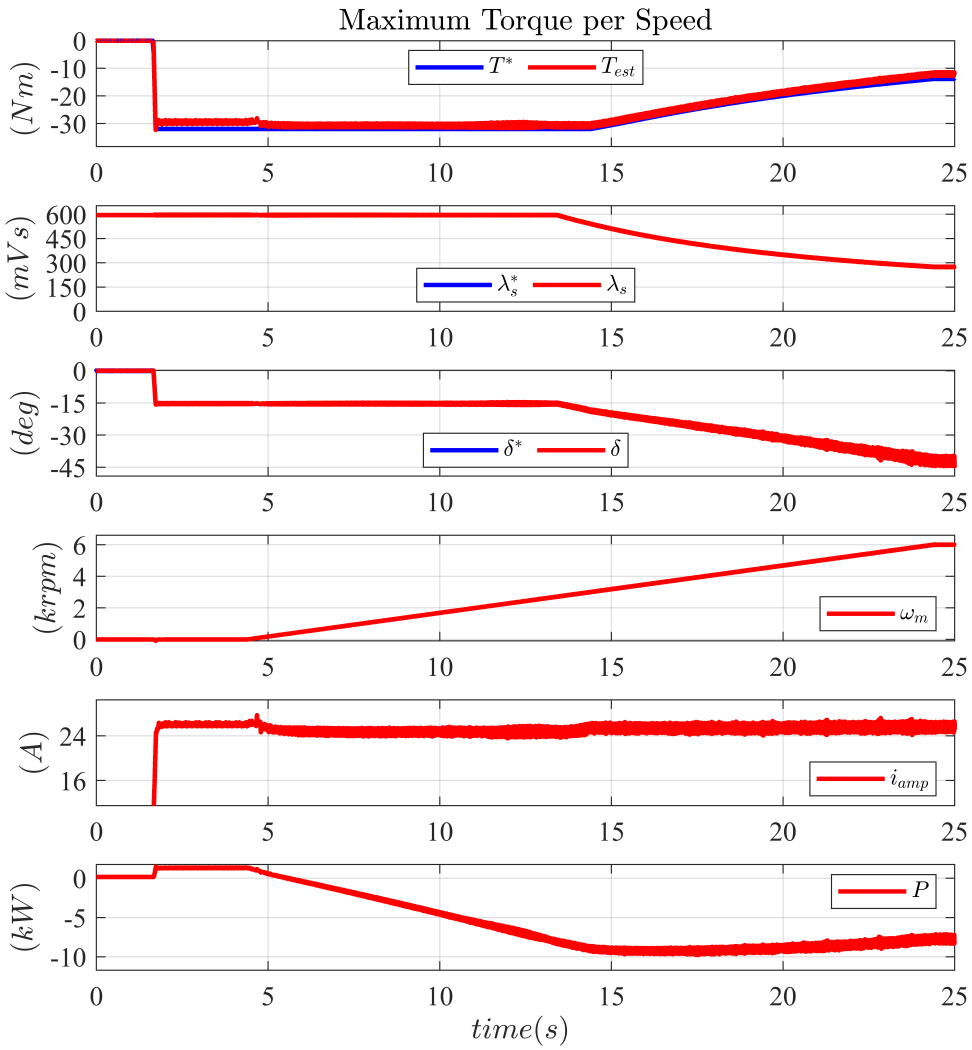
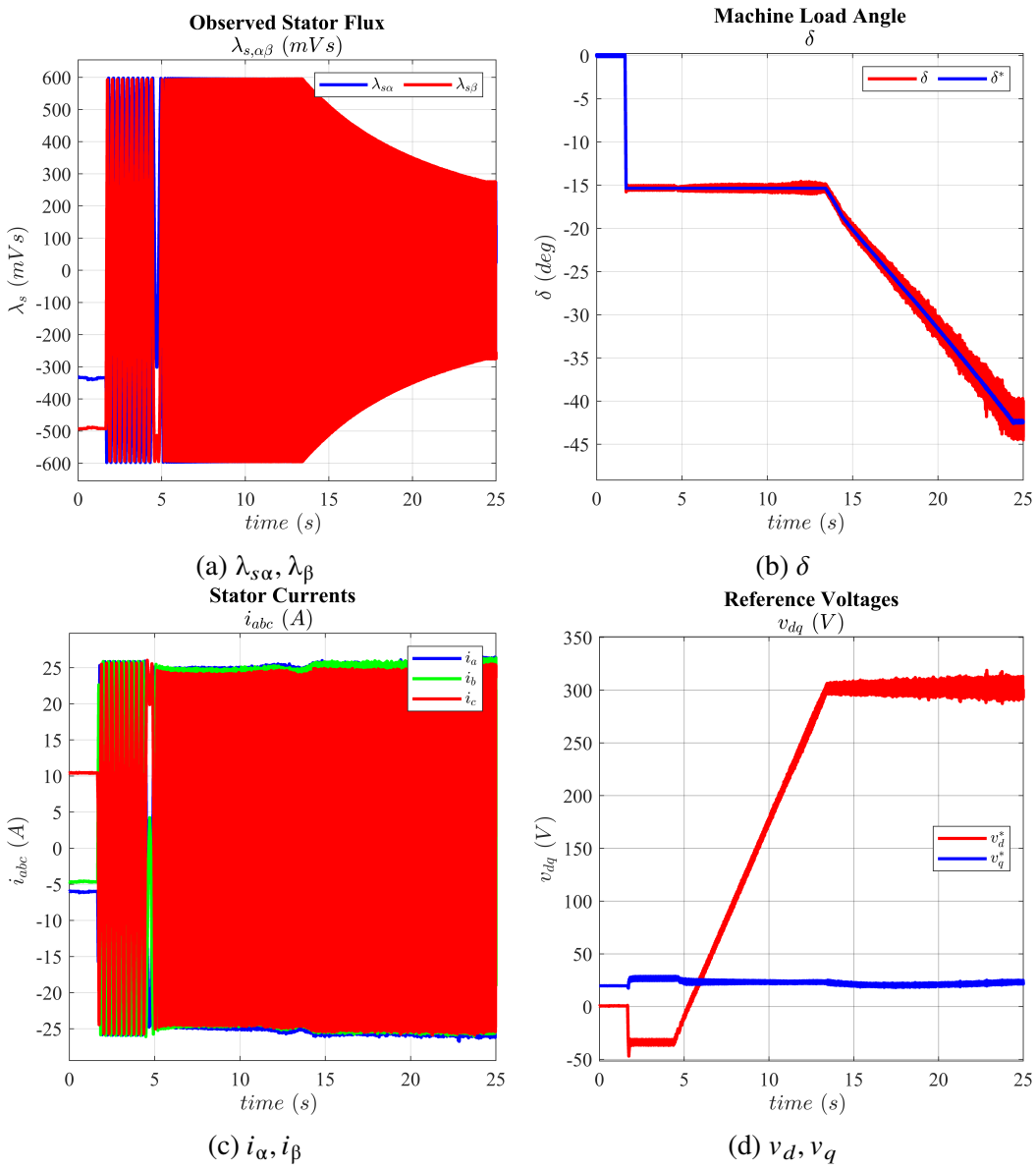


Figure 4.34: IM - Overview of main variables behave in generator experimental test @ 600V

The detail of the evolution of the main variables is in Figure 4.35.



(a) $\lambda_{s\alpha}, \lambda_{\beta}$ (b) δ

(c) i_{α}, i_{β} (d) v_d, v_q

Figure 4.35: IM - Fluxes, load angle, stator current and reference voltage in generator experimental test @ 600V

High Dynamic test

The dynamic behavior of the machine when high performance is required, is presented in Figure 4.36, and in the following zoom in Figure 4.37 obtained from the high speed working region. The stress of the torque inversion is mainly due to the rotation of the flux vector that must change its position, rotating of two times the value of the correspondent load angle, to be in the right place to satisfy the reference torque.

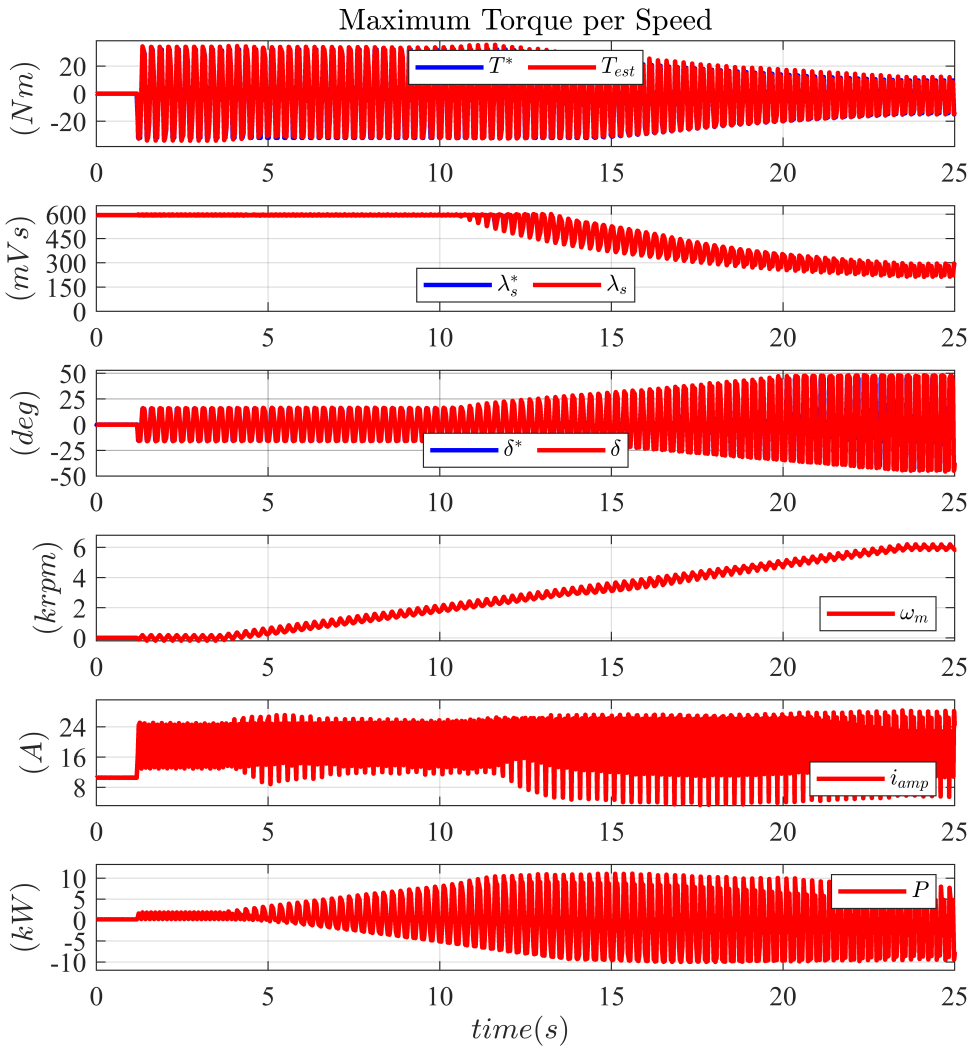


Figure 4.36: IM - Overview of main variables behave in high dynamic experimental test @ 600V, switching at 4 Hz with a slope of 5kNm/s

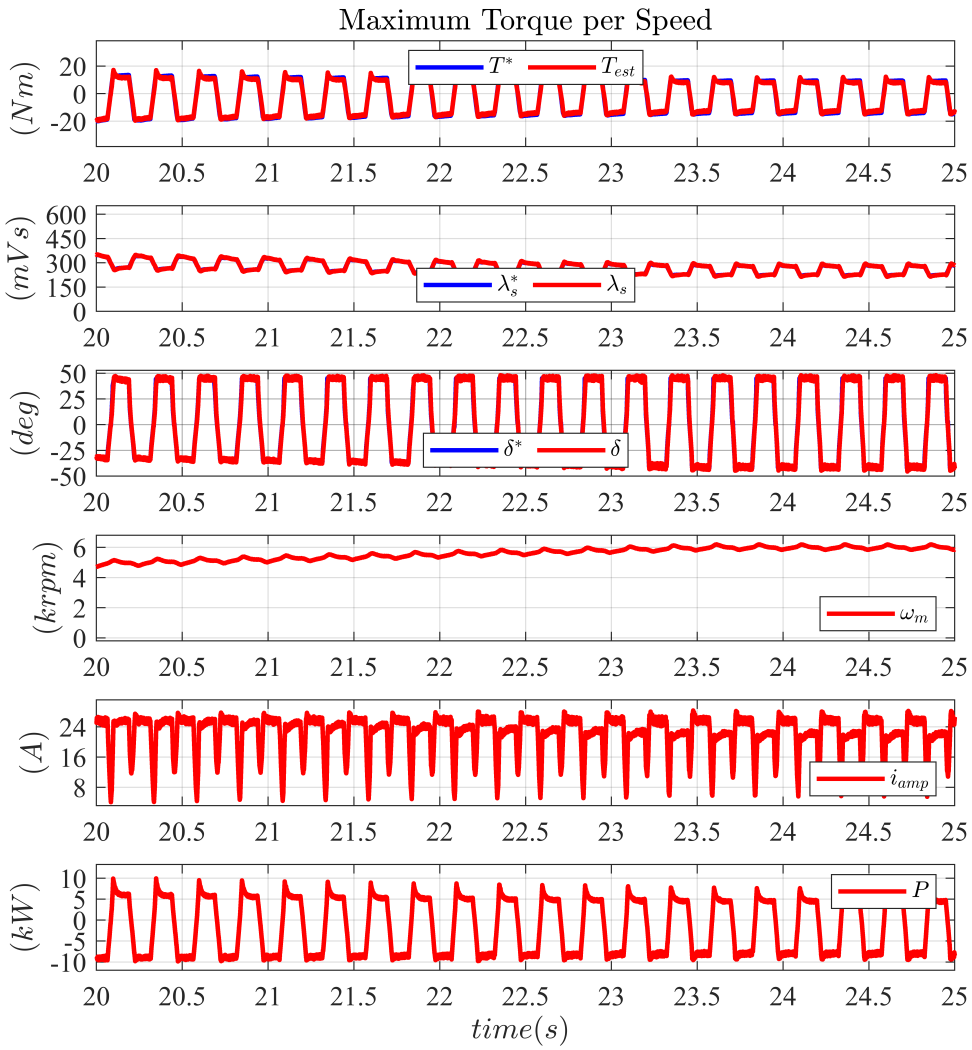
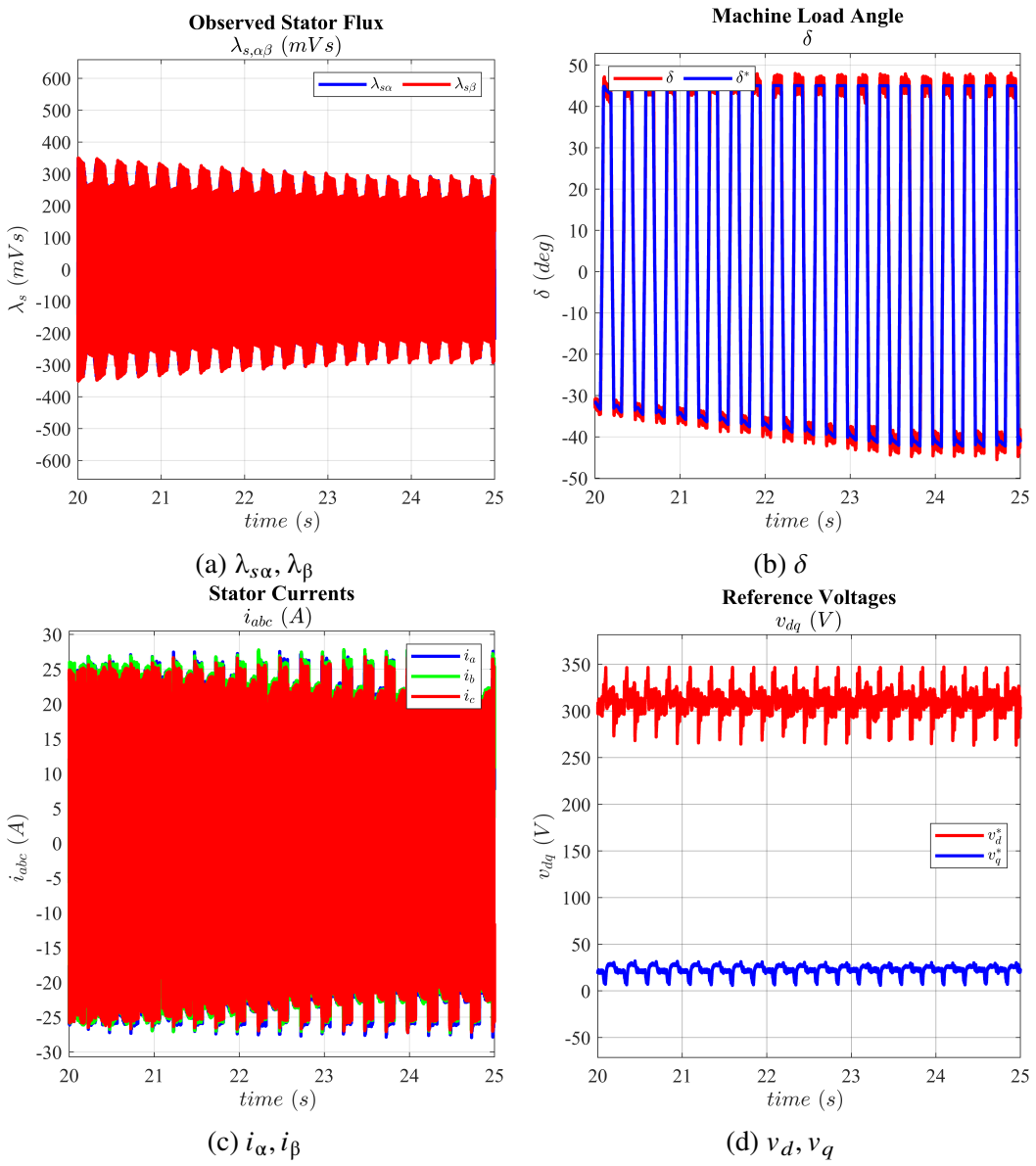


Figure 4.37: IM - Overview of main variables behave in high dynamic experimental test @ 600V, switching at 4 Hz with a slope of 5kNm/s: zoom in high speed working region

The behave of the main variables in the high speed working region is provided in Figure 4.38. Despite the high stress suffered by the machine in this condition, all the variables are perfectly controlled to the desired values.



(a) $\lambda_{s\alpha}, \lambda_{\beta}$ (b) δ

(c) i_{α}, i_{β} (d) v_d, v_q

Figure 4.38: IM - Fluxes, load angle, stator current and reference voltage in high dynamic experimental test @ 600V, switching at 4 Hz with a slope of 5kNm/s: zoom in high speed working region

Accuracy test

Finally, the accuracy test is performed. To obtain the results plotted in Figure 4.39 the speed is set into the machine at the desired value. All the speed in the range from zero speed up to the maximum one are explored at fixed step. Given the speed value, all the torque levels, from the minimum to the maximum, are imposed into the machine.

Following the actual speed the extreme values of torque are automatically defined by the control code that limits the maximum exploitable torque.

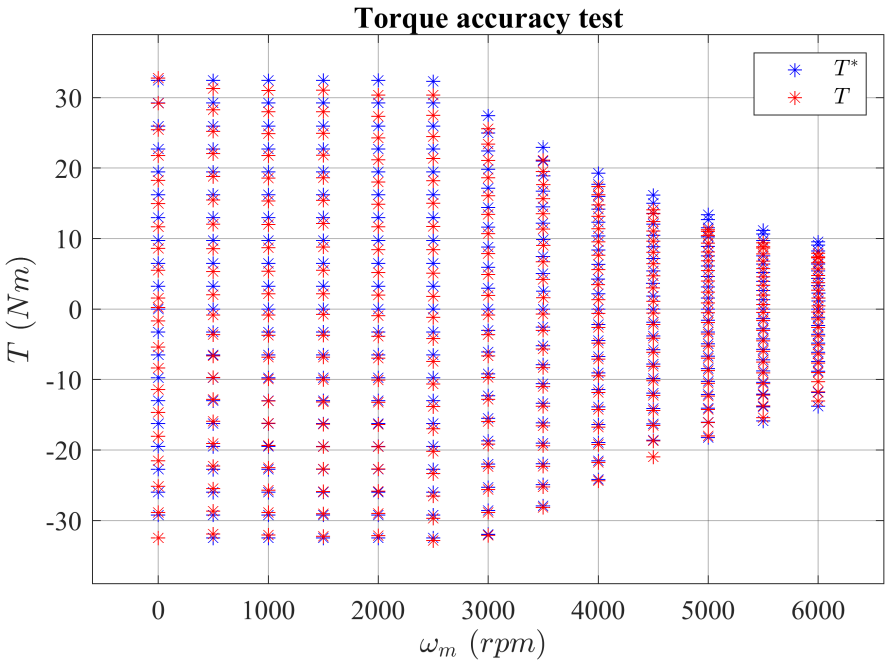


Figure 4.39: IM - Torque accuracy test @ 600V

4.3.2 Internal Permanent Magnets Motor

The experimental results on the internal permanent magnet motor are here showed. Several test have been carry out at different dc link voltage levels in order to evaluate the performance of the machine with the FPC torque controller.

Motor and Generator test

The maximum torque per speed profile test in motor configuration in Figure 4.40 has been carry out on the machine with a dc voltage of 360 V. The behave is like the one obtained in simulation, proving the accuracy of the model realized. The machine parameters are the same used for simulation reported in Table 4.2.

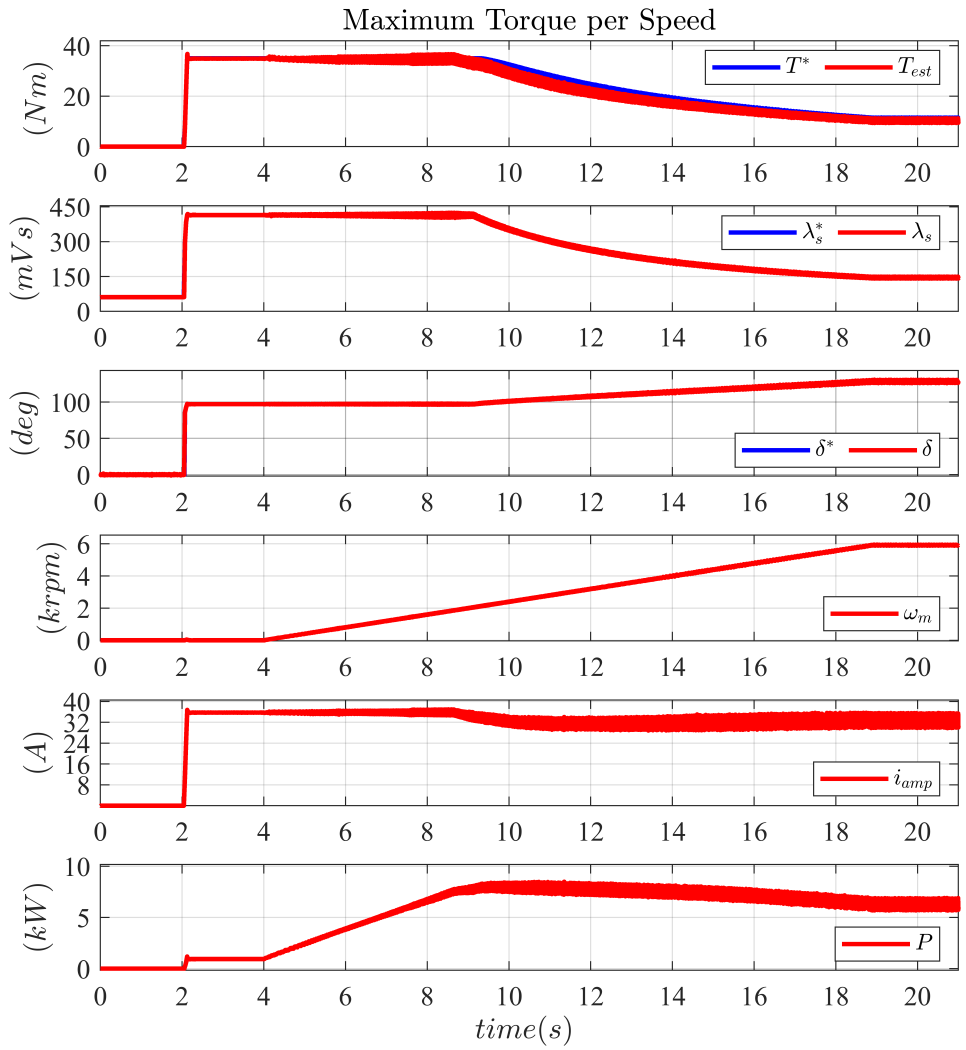
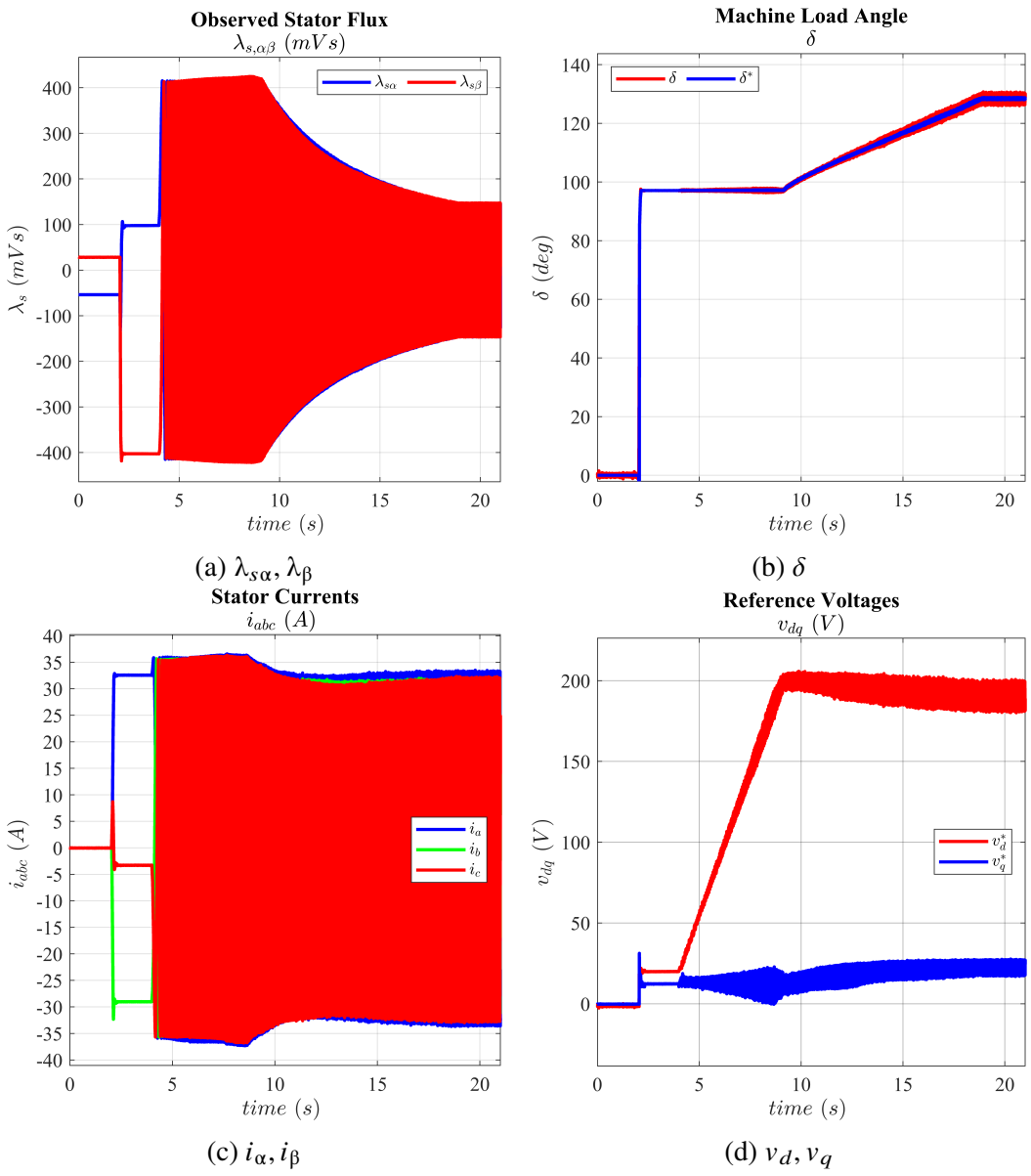


Figure 4.40: PM - Overview of main variables behave in motor experimental test @ 360V

The detailed behave of the fluxes ,the load angle ,the stator current and the references voltages is showed in Figure 4.41. The results are as expected from simulation.



(a) $\lambda_{s\alpha}, \lambda_{\beta}$ (b) δ

(c) i_{α}, i_{β} (d) v_d, v_q

Figure 4.41: PM - Fluxes, load angle, stator current and reference voltage in motor experimental test @ 360V

The generator-mode test is performed to guarantee the capability of control of FPC on both machine configuration. The results are showed in Figure 4.42.

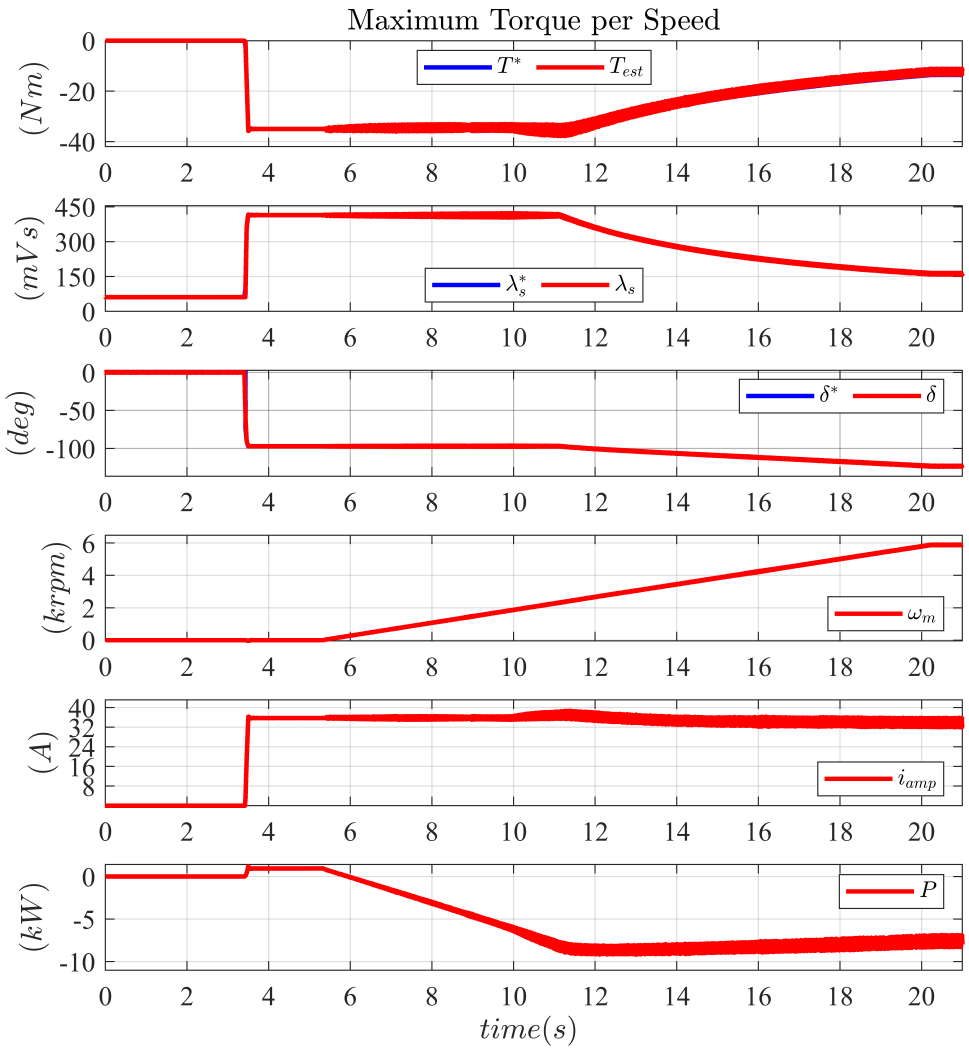
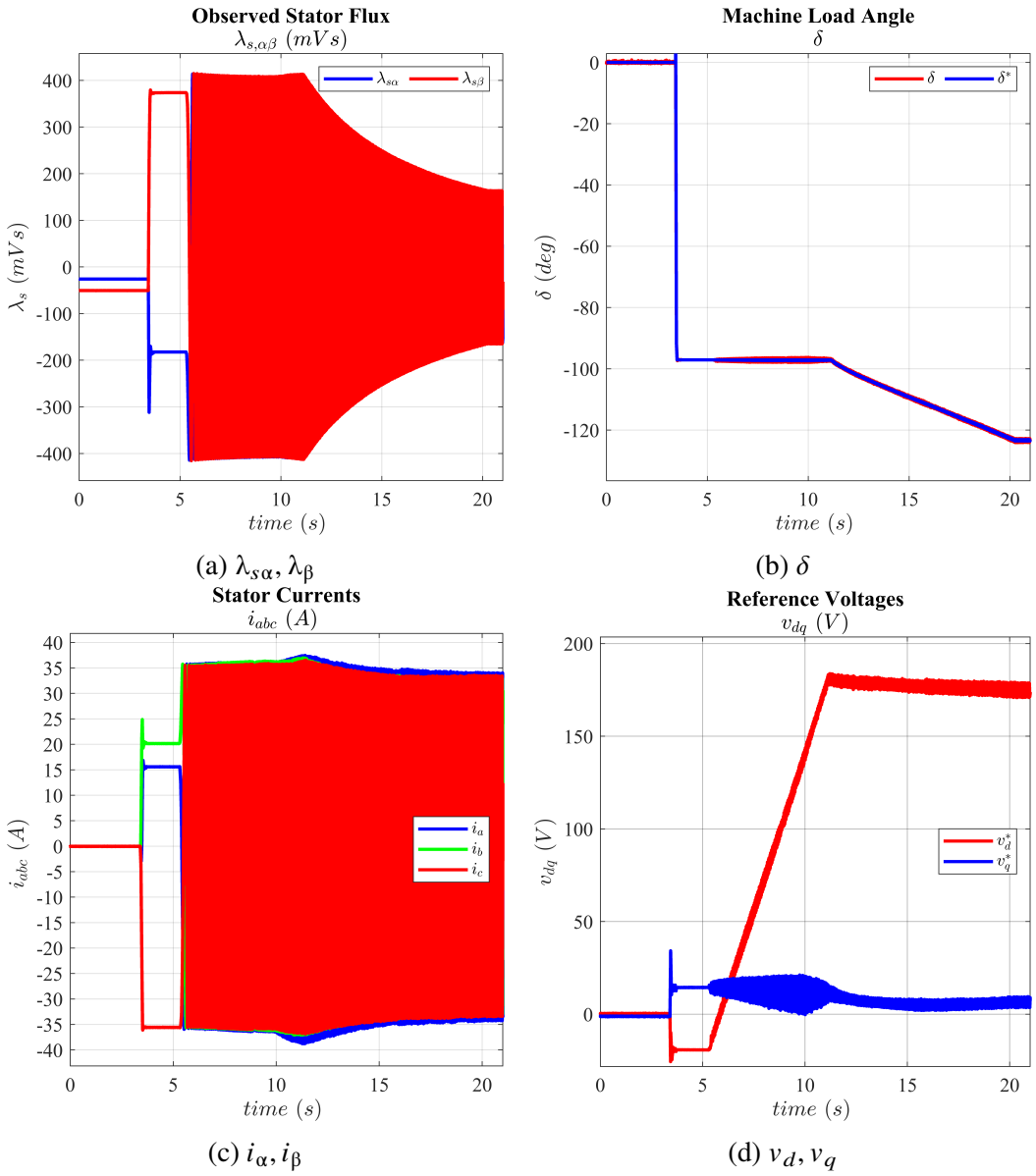


Figure 4.42: PM - Overview of main variables behave in generator experimental test @ 360V

The detail of the evolution of the main variables is in Figure 4.43.



(a) $\lambda_{s\alpha}, \lambda_{\beta}$ (b) δ

(c) i_{α}, i_{β} (d) v_d, v_q

Figure 4.43: PM - Fluxes, load angle, stator current and reference voltage in generator experimental test @ 360V

High Dynamic test

The dynamic behavior of the machine when high performance is required, is presented in Figure 4.44, and in the following zoom in Figure 4.45 obtained from the high speed working region.

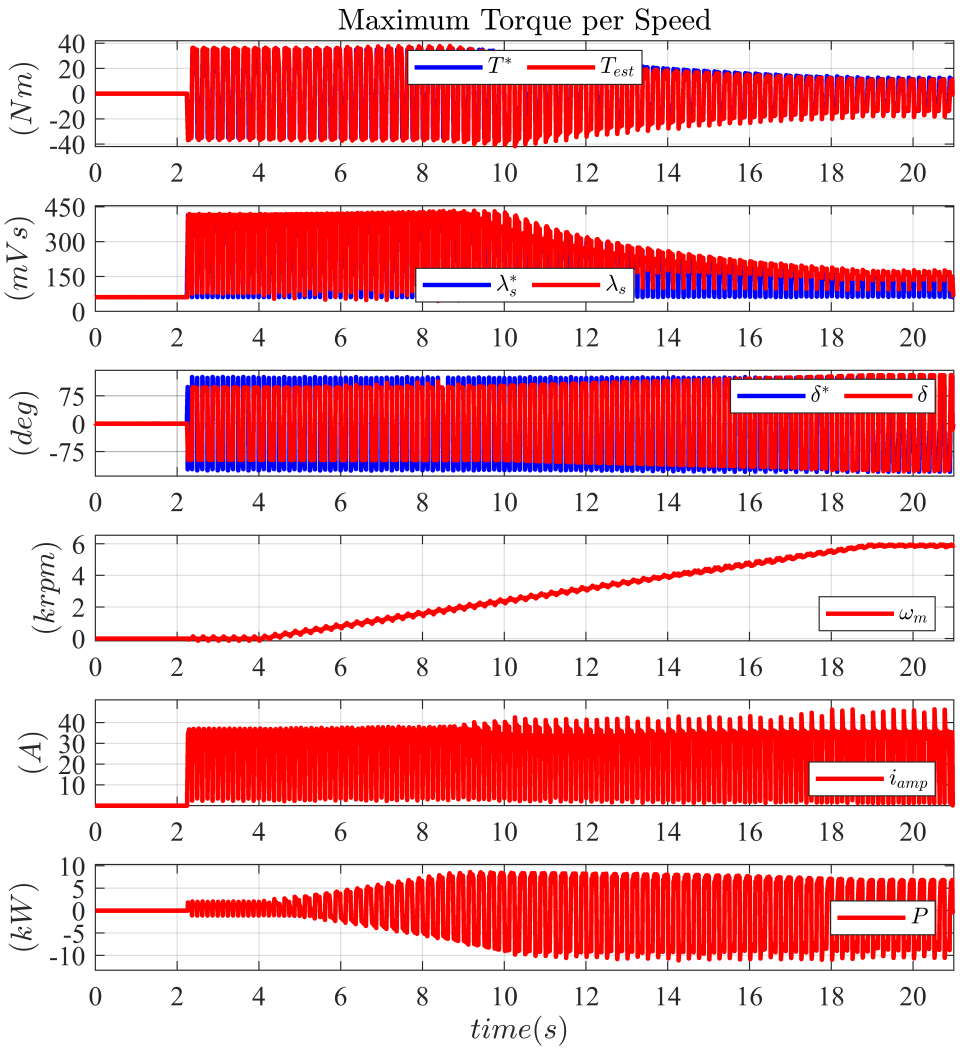


Figure 4.44: PM - Overview of main variables behave in high dynamic experimental test @ 360V

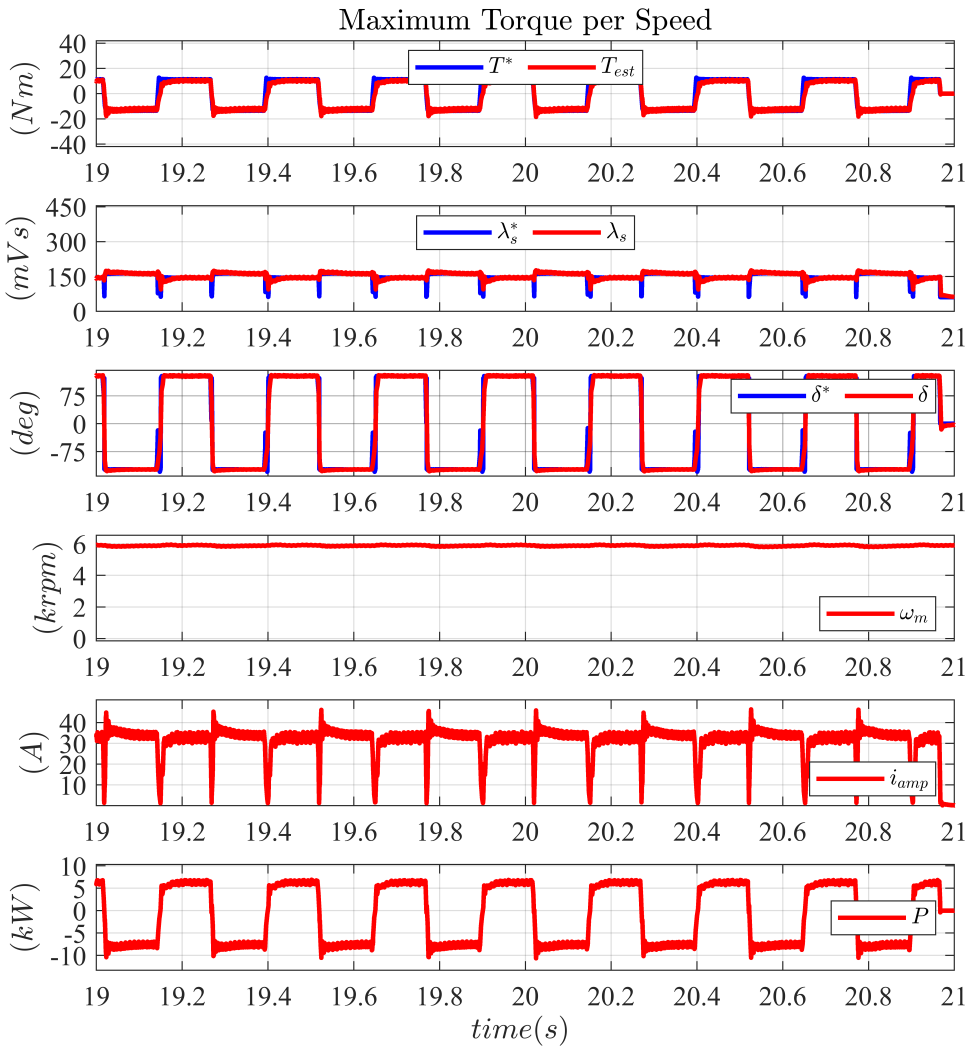
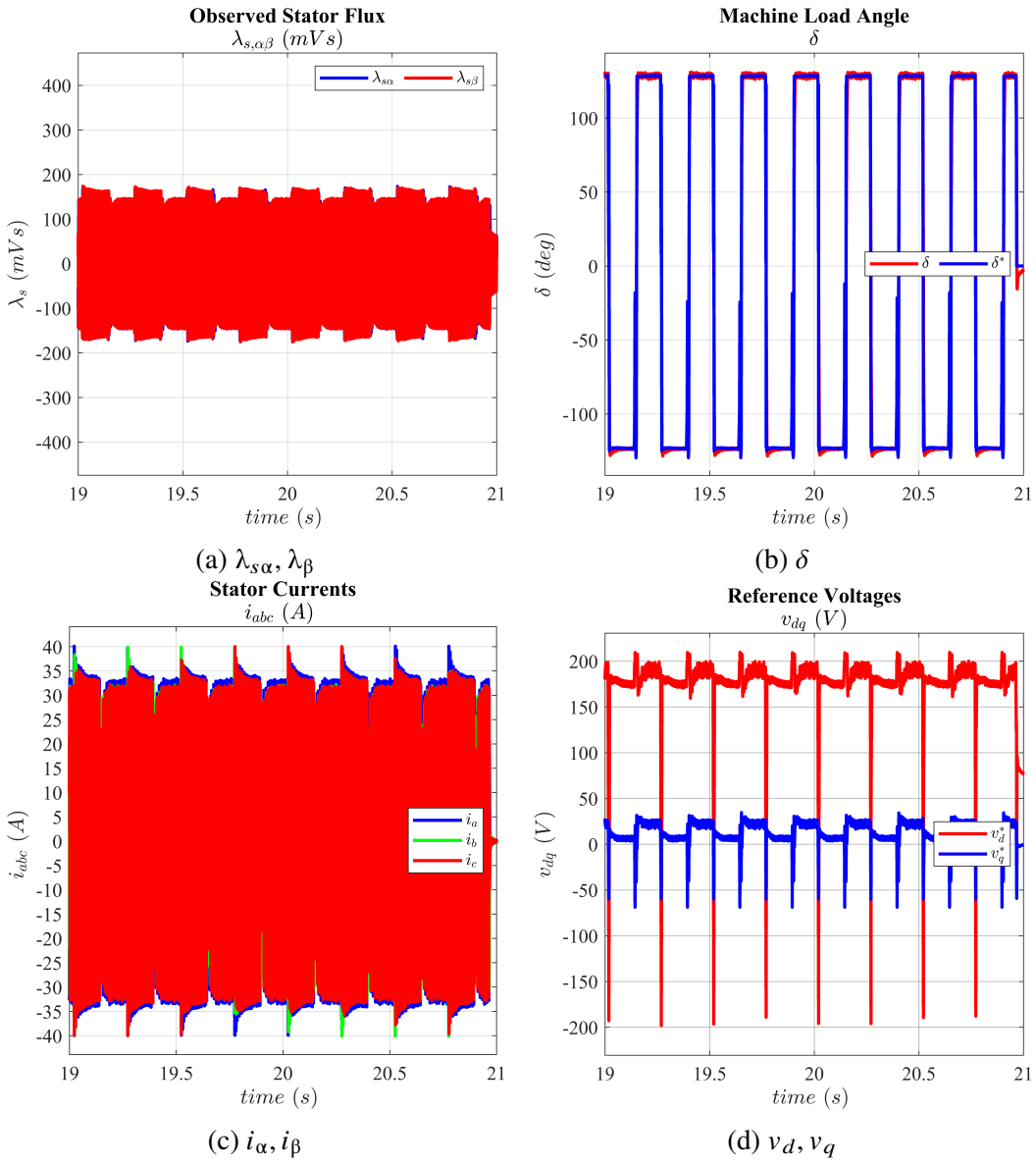


Figure 4.45: PM - Overview of main variables behave in high dynamic experimental test @ 360V: zoom in high speed working region

The behave of the main variables in the high speed working region is provided in Figure 4.46. All the variables, even the uncontrolled current, maintain the proper value.



(a) $\lambda_{s\alpha}, \lambda_{\beta}$ (b) δ

(c) i_{α}, i_{β} (d) v_d, v_q

Figure 4.46: PM - Fluxes, load angle, stator current and reference voltage in high dynamic experimental test @ 360V: zoom in high speed working region

Accuracy test

The results of the accuracy test are here showed in Figure 4.47.

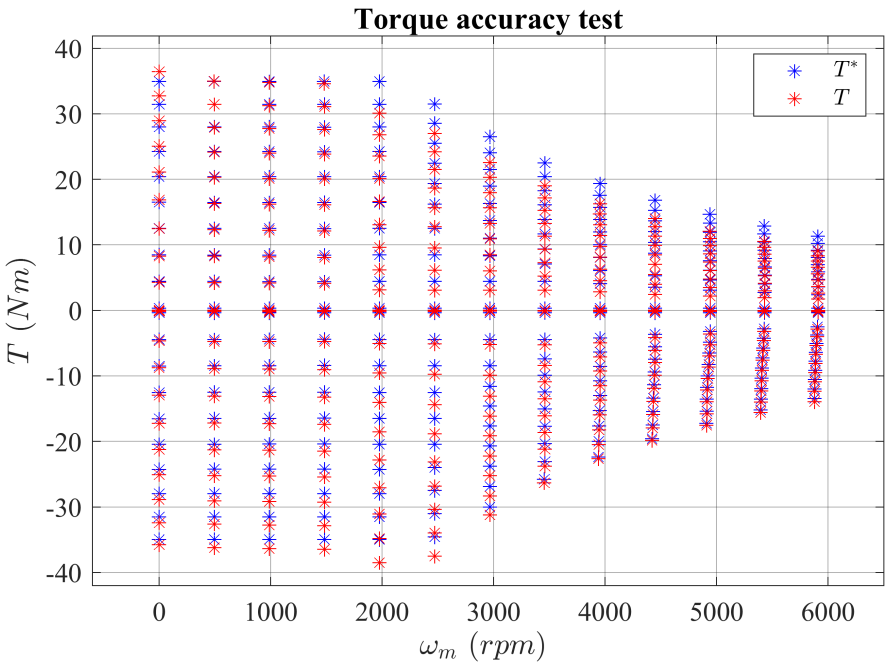


Figure 4.47: PM - Torque accuracy test @ 360V

Chapter 5

Conclusions

Within the report, a polar control technique has been proposed and analyzed for drives of any type of ac motor, one different from each other for controlled status variables and reference employed. Although the new proposed algorithm can be used in any field, the study of performance was carried out with particular interest for its use in the field of electric vehicle traction, where the exploitation of all the available working range, even over the base speed, is of primary importance for the application.
The strengths of the proposed technique are summarized:

- the control parameters, i.e. the value used for the flux and load angle proportional and integral regulators, are the same for all the machine tested. No care must be give to this parameters that, thanks to the principle on which the control is based, can be maintained always the same for all the machines. This is for sure the main strength of this type of control. In fact, all the simulation and experimental test have been carry out maintaining always the same values for any machine. For any other tye of controller at least one of the two inner regulators require a tuning procedure, being dependent on the machine working point.

- The only experimental test that must be carry out on the machine under test is the well known procedure to obtain direct flux map. No additional experimental procedure is required. This is an appreciable point because the direct flux maps is the same starting point required for other conventional control strategy.

- The high speed operation, in *MTPV* condition, is exploited by the machine using the same control approach used in the more common *MTPA* locus. This is possible thanks to the strategy adopted to build the control maps and control limitation stored in LUTs, that take into account also the *MTPV* limitation to define the working area. No additional regulator are require to work in MTPV region, as in other conventional control strategy.

- The voltage limit of the two inner regulators are unbalanced, exploiting the machine equation in (dq_s) coordinates. This permits to privilege the variation of the load angle value, rather then the flux amplitude, in order to regulate the torque value product in output. Moreover, the dynamic performance are improved by this asymmetry because, having the major part of voltage available, the load angle can vary rapidly.

The main drawbacks are:

- as for other conventional control strategy several LUTs are required to control the machine. In particular the load-angle control map must be created following a customized procedure that permit to obtain the proper value of the target load angle, given the torque and flux amplitude references.
- A flux observer structure is required to obtain the current amplitude of flux and its position, useful to retrieve the load angle value, that, in this case, represents the two strategic information for the control.
- With this strategy the torque's regulation linearity is losses because, in this case, the value of the reference load angle cannot be retrieved with an analytical relation as can be done for the reference value of current in the DFCV approach, but only using a LUT.

Despite the few drawbacks, the advantages that characterize the Flux Polar Control make it a valid alternative to the conventional solutions already present for the drives applied in the same field of application as this controller.

Chapter 6

Appendix

Here are showed and explained the code for the reading of one and two dimension LUT adopted in the control code.

Read of 1D LUT

The function to read one dimension LUT is showed. In the following is explained the basic principle on which it works:

- Following the order of the function argument, the inputs are: the index of memory of the first element of the vector storing the LUT (`pointer`), the actual value of the variable from which the output depends (`X`), the extremes of admissible input for the function (`Xmin,Xmax`), the regular step between two element (`DX`) and it inverse (`INV_DX`).

- The input value is saturated between the minimum and maximum admissible values. In this way the function is always able to produce an output, whichever value of input it receives.

- The position (`index`) of the lower element (`Xlow`) next to the input one is evaluated and used as current pointer value, summing its value to the one of the original pointer.

- The output is evaluated as the value corresponding to the lower element (`Ylow`) plus the difference due to the current input. The adopted formula is the one of the line passing through two points: the lower and upper ones (`Xlow,Xhigh=Xlow+DX`), with their correspondent output, (`Ylow,Yhigh`), with respect to the current value

(X).

$$Yout = Ylow + \frac{Yhigh - Ylow}{Xhigh - Xlow} \cdot [(X - Xlow) - DX \cdot index]$$
$$Yout = Ylow + \frac{Yhigh - Ylow}{DX} \cdot [(X - Xlow) - DX \cdot index]$$
$$Yout = Ylow + (Yhigh - Ylow) \cdot INV\_DX \cdot [(X - Xlow) - DX \cdot index]$$

The code is showed:

```
float read_one_dim_lut (float *pointer, float X, float Xmax, float Xmin,
float DX, float INV_DX) {
float Y_interp;
int index;
if (X > Xmax) X = Xmax - 0.001 * DX;
if (X < Xmin) X = Xmin + 0.001 * DX;
index = floor((X - Xmin) * INV_DX);
pointer += index;
Y_interp = *(pointer) + (*(pointer + 1) - *(pointer)) * INV_DX * ((X - Xmin)
- DX * index);
return Y_interp;
}
```

Read of 2D LUT

The function works as the one for the one dimension LUT but it must repeat the interpolation procedure three times to obtain the output.

- The inputs are: the index of memory of the first element of the matrix storing the LUT (`P`), the actual value of the variables from which the output depends (`x,y`), the steps and the extremes of the inputs (`Dx,invDx,Dy,invDy,Xmin,Xmax,Ymin,Ymax`), and the number points along a line of the matrix (`Npointx`).
- The input value is saturated between the minimum and maximum admissible values in both directions.
- The exact position (`rx,ry`) of the input value (`x,y`) is evaluated as distance from the origin of the matrix (`Xmin,Ymin`), that corresponds to the position pointed initially by the pointer.

Also the integer part of the distance (`ix,iy`), that are the closer lower coordinates of the point, are obtained from the exact position.

The difference between the exact and integer position is the deficit of the exact input values from the closer lower coordinate that exist in the matrix.

- The value of the pointer is upgraded to the lower coordinate of the closer existing point in the matrix. The square interpolation is performed here, between this point and the following, along x direction, obtaining `V1`.
- The value of the pointer is upgraded to the upper coordinate, incrementing the index of the number of element in each line (`Npointx`). The square interpolation is performed here, between this point and the following, again along x direction, obtaining `V2`.
- Finally, the output of the function, is the interpolation between the two interpolated values, that produce `V`.

```
float read_two_dim_lut(float *P, float x, float y, float Dx, float invDx,
float Dy, float invDy, float Xmax, float Xmin, float Ymax , float Ymin,
int Npointx) {
float V; % output
// temp variables
float ix, iy, rx, ry, V1, V2;
int delta;
// inputs limitations
if (x > Xmax) x = Xmax - 0.001 * Dx;
if (x < Xmin) x = Xmin + 0.001 * Dx;
if (y > Ymax) y = Ymax - 0.001 * Dy;
if (y < Ymin) y = Ymin + 0.001 * Dy;
// initial position
rx = (x - Xmin) * invDx;
ry = ((y - Ymin) * invDy) + 1;
// int indexes
ix = floor(rx);
iy = floor(ry);
// interpolation deficits
rx = rx - ix;
ry = ry - iy;
// first point
delta = ix + ((iy - 1) * Npointx);
```

```
P = P + delta;
// square interpolation
V1 = *P + ((*(P + 1) - *P) * rx);
P = P + Npointx;
V2 = *P + ((*(P + 1) - *P) * rx);
// inperpolated value
V = V1 + ((V2 - V1) * ry);
return V;
}
```

Phase Locked Loop

Another function, self-made, is the one that implement the Phase Locked Loop structure that, given an input angle, produce in output the filtered version of it, beside the speed of rotation of the vector whose angle is used.

The inputs are the sine and cosine of the angle of interest. The function calculate the sine and cosine of the PLL angle evaluated at the previous step (`k-1`), that are compared with the ones in input . From the vectorial product the error is evaluated and gave to a PI regulator that produces in output the speed. From the integral of the speed the filtered angle is obtained and normalize between [`-180°:180°`].

```
void pll(Xsc sc_theta, Xpll *pll_var){
SinCos(&(pll_var->theta), &sc_theta_pll);
pll_var->err = sc_theta.sin * sc_theta_pll.cos - sc_theta.cos * sc_theta_pll.sin;
pll_var->prop = pll_var->kp * pll_var->err;
pll_var->acc = pll_var->ki * pll_var->err;
pll_var->omega_filt += pll_var->acc * Ts;
pll_var->omega_unfilt = pll_var->prop + pll_var->omega_filt;
pll_var->theta += pll_var->omega_unfilt * Ts;
pll_var->theta = angle_normalization (pll_var->theta);
}
```

Dead Time Evaluation

The Dead Time function, self-made, is used to implement the process explained in Section 3.2.1 that treat the phase voltages to take into account the voltage drop on the inverter component.

The inputs of the function are (`vdc_filt,duty,iabc`), while the output are the real motor phase voltages (`vabc_real`), from whose the real inverter voltage drop has been

removed.

The ideal voltages (vabc_ideal) have been evaluated with the current duty and dc voltage. The value of drop on the inverter (vDT_pu) is obtained from the LUT that has the voltage drop in pu function of the current. The pu value multiplied with the dc voltage amplitude and the sign of the current, is the final value (vDT) to be subtracted from the ideal voltages to obtain the real ones.

```
void vabc_DT(float *vdc_filt, Xabc *duty, Xabc *iabc, Xabc *vabc_real){
float vDT_pu;
vabc_ideal.a = (*vdc_filt) * one_over_three * (2.0 * duty->a - duty->b -
duty->c);
vabc_ideal.b = (*vdc_filt) * one_over_three * (2.0 * duty->b - duty->c -
duty->a);
vabc_ideal.c = (*vdc_filt) * one_over_three * (2.0 * duty->c - duty->a -
duty->b);
vDT_pu = read_one_dim_lut(&DT_Vdt[0], abs(iabc->a), DT_IMAX, DT_IMIN, DT_DI,
DT_INV_DI);
vDT.a = sign(iabc->a) * vDT_pu * (*vdc_filt);
vDT_pu = read_one_dim_lut(&DT_Vdt[0], abs(iabc->b), DT_IMAX, DT_IMIN, DT_DI,
DT_INV_DI);
vDT.b = sign(iabc->b) * vDT_pu * (*vdc_filt);
vDT_pu = read_one_dim_lut(&DT_Vdt[0], abs(iabc->c), DT_IMAX, DT_IMIN, DT_DI,
DT_INV_DI);
vDT.c = sign(iabc->c) * vDT_pu * (*vdc_filt);
vabc_real->a = vabc_ideal.a - vDT.a;
vabc_real->b = vabc_ideal.b - vDT.b;
vabc_real->c = vabc_ideal.c - vDT.c;
}
```

MinMax Modulation

The code performing "MinMax" Modulation is showed. After the normalization of reference voltages on the value of the dc link, the current medium value between the three voltages is found and used as zero-sequence voltage. Its value is summed to the normalized voltages to calculate the duty cycles, that, in the end, are eventually saturated between [0,1].

```
void PWMCompute (Xabc *ABC, Xabc *Duty, float *Vdc) {
```

```
// tmp variables
float a, b, c, x1, y1, pwm_zero_seq;
// Vdc Reciprocal
x1 = 1.0 / (*Vdc);
// Normalization of Reference Voltages
a = ABC->a * x1;
b = ABC->b * x1;
c = ABC->c * x1;
// Computation of Common Mode
if (a > b) {
x1 = a;
y1 = b;
}
else {
x1 = b;
y1 = a;
}
if (x1 < c) pwm_zero_seq = x1;
else {
if (y1 > c) pwm_zero_seq = y1;
else pwm_zero_seq = c;
}
// Add Common Mode
Duty->a = 0.5 + a + 0.5 * pwm_zero_seq;
Duty->b = 0.5 + b + 0.5 * pwm_zero_seq;
Duty->c = 0.5 + c + 0.5 * pwm_zero_seq;
// Saturation of Duty-Cycle
Duty->a = two_level_saturation(Duty->a, DUTY_MAX, DUTY_MIN);
Duty->b = two_level_saturation(Duty->b, DUTY_MAX, DUTY_MIN);
Duty->c = two_level_saturation(Duty->c, DUTY_MAX, DUTY_MIN);
}
```

Bibliography

[1] E. Trancho et al. “IPMSM torque control strategies based on LUTs and VCT feedback for robust control under machine parameter variations”. In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Oct. 2016, pp. 2833–2838. DOI: 10.1109/IECON.2016.7793556.

[2] Bon-Ho Bae et al. “New field weakening technique for high saliency interior permanent magnet motor”. In: *38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003.* Vol. 2. Oct. 2003, 898–905 vol.2. DOI: 10.1109/IAS.2003.1257641.

[3] Gianmario Pellegrino, Radu Iustin Bojoi, and Paolo Guglielmi. “Unified Direct-Flux Vector Control for AC Motor Drives”. In: *IEEE Transactions on Industry Applications* 47.5 (Sept. 2011). Conference Name: IEEE Transactions on Industry Applications, pp. 2093–2102. ISSN: 1939-9367. DOI: 10.1109/TIA.2011.2161532.

[4] Sandro Rubino et al. “Direct Flux and Load Angle Vector Control of Permanent Magnet Synchronous Motors”. In: (2021).

[5] Eric Armando et al. “Accurate Magnetic Modelling and Performance Analysis of IPM-PMASR Motors”. In: *2007 IEEE Industry Applications Annual Meeting*. ISSN: 0197-2618. Sept. 2007, pp. 133–140. DOI: 10.1109/07IAS.2007.29.

[6] Paul Krause et al. In: *Analysis of Electric Machinery and Drive Systems*. 2013. DOI: 10.1002/9781118524336.

[7] Eric Armando et al. “Flux linkage maps identification of synchronous AC motors under controlled thermal conditions”. In: *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*. May 2017, pp. 1–8. DOI: 10.1109/IEMDC.2017.8002334.

[8] Eric Armando et al. “Experimental Identification of the Magnetic Model of Synchronous Machines”. In: *IEEE Transactions on Industry Applications* 49.5 (Sept. 2013). Conference Name: IEEE Transactions on Industry Applications, pp. 2116–2125. ISSN: 1939-9367. DOI: 10.1109/TIA.2013.2258876.

[9] I.R. Bojoi et al. “Self-commissioning of inverter nonlinear effects in AC drives”. In: *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*. Sept. 2012, pp. 213–218. DOI: 10.1109/EnergyCon.2012.6347755.

[10] Tobias Huber, Wilhelm Peters, and Joachim Bocker. “Voltage controller for flux weakening operation of interior permanent magnet synchronous motor in automotive traction applications”. In: *2015 IEEE International Electric Machines Drives Conference (IEMDC)*. May 2015, pp. 1078–1083. DOI: 10.1109/IEMDC.2015.7409195.

[11] Oleg Buchholz and Joachim Böcker. “Gopinath-observer for flux estimation of an induction machine drive system”. In: *2017 IEEE Southern Power Electronics Conference (SPEC)*. Dec. 2017, pp. 1–7. DOI: 10.1109/SPEC.2017.8333614.

[12] D. Grahame Holmes and Thomas A. Lipo. In: *Pulse Width Modulation for Power Converters: Principles and Practice*. 2003. DOI: 10.1109/9780470546284.