

POLITECNICO DI TORINO

Master's Degree in Electrical Engineering



Master's Degree Thesis in

**STUDY AND ADVANCED
MODELLING OF MULTIPHASE
MOTORS WITH RECONFIGURABLE
NUMBER OF PHASES**

Supervisors

Prof. Gianmario PELLEGRINO

Dr. Simone FERRARI

ETEL SA Tutor

Dr. Alessandro FASOLO

Candidate

Lorenzo CIFALINÒ

MARCH 2023

Summary

This thesis focuses on Permanent Magnet Synchronous Motors (PMSMs) with three or more phases, specifically addressing the multi-threephase structures. At the beginning of the last century, multi-phase solutions were adopted in generators with power greater than 1MW to overcome current limitations imposed by the circuit breakers of that period. However, with the advancement of new power electronics devices, multi-phase technology is now used in various applications, such as naval and aircraft motors, due to their improved efficiency and fault-tolerance, respectively.

Multi-phase solutions are increasingly being adopted in industrial applications to achieve better torque performance and improved utilization of individual components, thanks to the increased focus on efficiency and cost optimization. Moreover, the reduction in size of individual power electronics components in the multi-phase structures facilitates the integration of components in a complete structure composed of power converter and electric motor. New power electronics technologies such as SiC and GaN devices enable the use of multi-phase solutions to achieve these objectives, in line with the latest market trends.

The aim of this work is to demonstrate how an existing ETEL motor can be upgraded by modifying only the control strategy from a standard three-phase configuration to enable the control of four three-phase systems in an Internal Permanent Magnets synchronous motor. Additionally, this thesis will highlight the study of the windings configuration to ensure proper motor operation without the need for modifying any active motor parts.

The aforementioned solutions are justified by the significant improvements obtained in terms of torque, efficiency, and speed range.

If a threephase and a four-threephase solution are compared at the same operating point, the torque increases by approximately more than 4%, and the torque ripple is significantly reduced by about 91%. Additionally, the efficiency is increased by 2%, as the power factor, at the rated working condition. Furthermore, the speed range is extended by approximately 14% while maintaining a constant efficiency of 87%. In terms of the single power converter rated current, it has been reduced by 75% compared to the standard three-phase solution.

The just mentioned results evidence how a multi-threephase solution could represent a solution to improve the performance of a motor and allow to integrate all the component in a single structure easier.

Software like Matlab, Simulink, FEMM and Syre have been used to study the control's dynamic and windings configuration of the system. Finite Element Method analysis was used to study improvements in torque, efficiency, speed range, and power factor. Simulations in Simulink were conducted to verify the control strategy and propose a new decoupling method in a safe working conditions.

Finally, the results of the work are shown in order to demonstrate that reconfiguring multi-phase machines is not so complex as it seems and it could brings new product in the future even for an industrial application.

Acknowledgements

These last six months of my Master's degree have represented an important moment of growth both professionally and, above all, personally. An experience abroad, working for a company in Switzerland, was something I never imagined I would do, but I always wished for. And now, here I am, after a challenging academic journey, realizing how much has happened and how quickly it's all gone by. First and foremost, I would like to thank Politecnico di Torino, especially Professor Gianmario Pellegrino and Dr. Simone Ferrari, who gave me the wonderful opportunity to carry out my thesis work in a company like ETEL and who guided me through this adventure. With their patience and understanding, they provided me with valuable suggestions and clarifications without which I wouldn't have been able to complete this work.

Last but not least, I would like to express my gratitude to ETEL SA for providing me with the opportunity to carry out my work in a dynamic and stimulating job environment, where I was able to meet highly skilled and friendly colleagues. I am particularly grateful to my company supervisor, Dr. Alessandro Fasolo, who introduced me to this new world and acted as a valuable reference throughout my time in Switzerland. His patience and availability were particularly appreciated during the most challenging moments of my experience.

Table of Contents

List of Tables	VIII
List of Figures	IX
Acronyms	XV
Introduction and Motivation	1
1 State of the art and machine model	4
1.1 Model of multiphase machines	4
1.1.1 Multiphase machines windings	4
1.1.2 Multistator Vector Space Decomposition technique's funda- mentals	8
1.1.3 Machine model	11
1.2 Field-oriented control of the multiphase machine	17
1.2.1 Simulation ambient and type of the data used	17
1.2.2 Control strategy and logic-block schemes	22
1.3 Simulation toolchain	27
2 Automated FEM simulation of the multi-phase machine	29
2.1 Introduction to the motor used and windings configuration adopted	29
2.2 Syre: standard and multi-threephase motor study	39
2.2.1 Introduction to the software	39
2.2.2 Different models comparison	44
2.3 Results and final comments	54
3 Control Simulation	59
3.1 Control of n 3-phase machine	59
3.1.1 Motor model	59
3.1.2 Mechanical equations	64
3.2 Inverter Model	65

3.3	Digital Control	67
3.4	Simulation of the 3-phase motor drive	78
3.5	Simulation of the 6-phase motor drive	84
3.5.1	Modifications to the original model	84
3.5.2	Results	96
3.6	Simulation of the 12-phase motor drive	101
3.6.1	Modifications to the original mode	101
3.6.2	Results	114
4	Conclusions	119
	Bibliography	121

List of Tables

2.1	Main data of TMK0291-050-3VBS motor, in order to represent its main characteristics.	30
2.2	Iron losses comparison respect to the "Point 1" of the threephase motor. The values considered are related to the nominal performance requirements.	52
2.3	Sixphase and twelvephase motors' quantities comparison respect to the threephase one.	57
3.1	Settings of the control strategy adopted for the threephase motor.	78
3.2	Electrical parameters' values estimated: R_s, L_d and L_q are obtained with Syre study, while the leakage one is the result of one FEMM simulation imposing a constant and common value to a threephase set turned on.	87
3.3	Electrical parameters' values estimated: R_s, L_d and L_q are obtained with Syre study, while the leakage one is the result of one FEMM simulation imposing a constant and common value to a threephase set turned on.	104

List of Figures

1.1	Fractional-slot concentrated windings: a) Single Layer (SL); b) Double Layer (DL).	5
1.2	Drive with multiple single-phase units, [4].	5
1.3	Drive with multiple threephase units,[4].	5
1.4	Three examples of the remaining phase coils after removing one of the two three-phase DL windings. (a) DL-1. (b) DL-2. (c) DL-3, [6].	6
1.5	Symmetric windings star for a sixphase motor.	8
1.6	Asymmetric windings star for a sixphase motor.	8
1.7	(d,q) frame adopted in MS approach.	9
1.8	Example of the matrix T_{12} and T_{12}^{-1} used for the VSD technique in the four-threephase motor drive model.	15
1.9	Example of a 3-phase Simulink model as Syre output.	18
1.10	Components of Simulink's "Motor model" block.	18
1.11	Direct flux-maps $\lambda_{dq}(i_d, i_q)$	19
1.12	Inverse flux-maps $i_{dq}(\lambda_d, \lambda_q)$	19
1.13	MTPA and MTPV curves as function of torque and currents along d and q axis.	20
1.14	Current grid in (d,q) plane used to compute the fluxes' maps. . . .	21
1.15	Flux identification using i_{dq} currents.	21
1.16	General torque-control scheme adopted using a novel decoupling matrix for common and differential mode control.	23
1.17	Flux-observer scheme for a sixphase machine.	25
1.18	Flux-observer scheme for a twelvephase machine.	25
2.1	Example of the initial parametric approach used to model the machine.	31
2.2	Example of a sixphase voltage vector configuration of a machine. . .	34
2.3	Example of a twelvephase voltage vector configuration of a machine.	34
2.4	Threephase machine preliminary simulation. From the top to the bottom: (a) no-load fluxes, (b) fluxes with load, (c) currents injected. The load-angle between (a) and (b) can be noticed.	35

2.5	Threephase machine preliminary simulation. Torque obtained at no-load working condition, with zero average value.	36
2.6	Threephase machine preliminary simulation. Result of the torque got with the rated value of TMK motor. $\bar{T} = 302.5Nm$	36
2.7	Twelvephase machine preliminary simulation. From the top to the bottom: (a) no-load fluxes, (b) fluxes with load, (c) currents injected. The load-angle between (a) and (b) can be noticed.	37
2.8	Twelvephase machine preliminary simulation. Torque obtained at no-load working condition.	38
2.9	Twelvephase machine preliminary simulation. Result of the torque got with the rated value of new version of TMK motor. $\bar{T} = 313.6Nm$ and $T_{pk-pk\%} = -98\%$ respect to the threephase machine.	38
2.10	GUI-Syre.app interface. The main page is shown with the uploaded model.	40
2.11	GUI_Syre.app interface. The geometric section is shown with the uploaded model.	41
2.12	GUI_Syre.app interface. The options section is shown with the uploaded model.	41
2.13	GUI_Syre.app interface. The windings configuration selections are shown with the uploaded model.	42
2.14	GUI_Syre.app interface. The materials' settings available for a model.	42
2.15	GUI_Syre.app interface. The parameters selection used for the simulation.	43
2.16	Results obtained with load about the threephase machine.	44
2.17	Fluxes transformed in (d,q) domain thanks the Clarke and rotational transformation in the threephase system.	44
2.18	Torque value obtained for threephase electric motor with the Syre simulation in the single working point.	45
2.19	Results obtained with load about the sixphase machine.	46
2.20	Fluxes transformed in (d,q) domain in the sixphase system.	46
2.21	Torque value obtained for sixphase electric motor with the Syre simulation in the single working point.	46
2.22	Results obtained with load about the twelvephase machine.	47
2.23	Fluxes transformed in (d,q) domain in the twelvephase system.	47
2.24	Torque value obtained for twelvephase electric motor with the Syre simulation in the single working point.	47
2.25	Torque maps for threephase motor.	49
2.26	Torque maps for sixphase motor.	49
2.27	Torque maps for twelvephase motor.	49
2.28	Threephase machine's permanent magnets losses-map as function of the speed.	51

2.29	Threephase machine rotor's eddy currents losses-map as function of the speed.	51
2.30	Threephase machine rotor's hysteresis losses-map as function of the speed.	51
2.31	Threephase machine stator's eddy currents losses-map as function of the speed.	51
2.32	Threephase machine stator's hysteresis losses-map as function of the speed.	51
2.33	Iron losses-map of threephase motor: point 1 is related to rated current along q -axis, point 2 is for the same torque following the MTPA characteristic.	52
2.34	Values of d and q currents to use the MTPA characteristic.	53
2.35	Efficiency maps for the three machines.	55
2.36	Torque ripple maps for the three machines.	55
2.37	Iron losses maps for threephase machine.	56
3.1	Motor model block, with which representing the whole dynamic of the motor is possible using a physical model combined with 1D-2D LUTs model.	60
3.2	Trigonometric transformation from the absolute angular position to the sinusoidal-cosinusoidal related coordinates.	61
3.3	Fluxes computation, from the back-EMF to the fluxes in (d,q) domain.	61
3.4	"Inverse maps" logic block, in which the currents and the torque are computed with the right convention.	62
3.5	Torque and currents calculation. For the first the cross.multiplication is used, while for the latter some 2D LUTs are used.	62
3.6	Iron losses addition in terms of current to the required one for the energy conversion.	63
3.7	Iron losses-map of threephase motor: point 1 is related to rated current along q -axis, point 2 is for the same torque following the MTPA characteristic.	64
3.8	2D-LUTs used approach to compute the iron losses with the speed adaptation. Then the contribution of the iron to the currents is computed.	64
3.9	Mechanical load's representation depending on the control type adopted: current, flux, torque or speed control.	65
3.10	Inverter model used to represent the real dynamics of the power electronics.	66
3.11	PWM logic used to obtain the gate-signals from the duty-cycles due to the control side.	67

3.12	From the top to the bottom: torque reference and motor torque, dragged load speed and filtered one by the control.	79
3.13	Threephase machine duty-cycles implementing the third harmonic injection. In the bottom graph a zoom at steady state is present. . .	80
3.14	From the top to the bottom: i_{abc} and a zoom to highlight the difference between the physic quantities and the sampled ones. . . .	81
3.15	Currents in (d,q) frame: it is shown the "real" motor current, the one used in the control after the feedback currents transformation, the reference.	82
3.16	Fluxes in (d,q) frame: it is shown the "real" motor fluxes and the observed ones.	83
3.17	Sixphase machine Simulink model overview.	84
3.18	Motor model in the sixphase motor: the analogous blocks of threephase version are present, but with some modification explained in this section.	85
3.19	Motor model in the sixphase motor: the analogous blocks of threephase version are present, but with some modification explained in this section.	86
3.20	Motor model in the sixphase motor: the analogous blocks of threephase version are present, but with some modification explained in this section.	87
3.21	Modifications adopted in the iron-losses calculation in order to be consistent with the VSD theory.	88
3.22	Current calculation block for a sixphase motor, considering the inverse transformation T_6^{-1}	88
3.23	From the top to the bottom: Torque reference and the torque applied by the motor, while in the second graph the load speed and the filtered one with a digital filter are represented.	96
3.24	Threephase machine duty-cycles implementing the third harmonic injection. In the bottom graph a zoom at steady state is present. . .	97
3.25	From the top to the bottom: $i_{abc,1}$ and $i_{abc,2}$ with 30° displacement electrical degrees, and a zoom to highlight the difference between the physic quantities and the sampled ones.	98
3.26	Currents in (d,q) frame: it is shown the "real" motor current, the one used in the control after the feedback currents transformation, the reference.	99
3.27	Currents in (d,q) frame: it is shown the "real" motor current, the one used in the control after the feedback currents transformation, the reference.	100
3.28	Simulink model overview about the twelvephase model.	101

3.29	Motor model of twelvephase machine, with four threephase systems with isolated neutral points.	102
3.30	"Flux Integration" block, where the T_{12} is implemented and the (α, β) coordinates are rotated to the (d,q) ones.	103
3.31	"Inverse Flux Maps" block, in which the currents are computed in the common (d,q) domain for the VSD technique. The torque calculation takes into account the new multiplication factor of 6.	104
3.32	"Currents calculation" block, in which the twelvephase currents are computed.	105
3.33	From the top to the bottom: motor torque and related reference, dragged load speed and the filtered one with the control.	114
3.34	Duty-cycles adopted for the twelvephase motor. The third harmonic injection is recognizable. The second graph shows with the same color the three duty-cycles related to the same threephase system. .	115
3.35	Currents of twelvephase motor with the typical displacement of 15° electrical degrees of an asymmetrical machine. The currents related to the same star are have been represented with the same color. The second graph shows also the sampling phenomenon delay for one threephase system only, for reasons of graphic clarity.	116
3.36	From the top to the bottom: physical current, sampled one and its reference along d-axis and the same quantities for the q-axis.	117
3.37	From the top to the bottom: physical flux and observed one along d-axis and the same quantities for the q-axis.	118

Acronyms

IPM

Internal Permanent Magnets

SPM

Surface Permanent Magnets

FEM

Finite Element Method

DWs

Distributed Windings

FSCWs

Fractional-Slot Concentrated Windings

PM

Permanent Magnet

SR

Switched Reluctance

DL

Double-Layer

SL

Single-Layer

EMF

Electric and Magnetic Field

VSD

Vector Space Decomposition

MS

Multistator

MTPA

Maximum Torque Per Ampere

LUTs

Look Up Tables

Introduction and Motivation

With the rapid development of power electronics even an increase of more complex structures can be seen, in order to achieve better performances and guarantee greater efficiency. It is interesting to observe that the dawn of multiphase machines dates back to 1920s in a field not directly related to the propulsion applications. In particular, to satisfy the demand for higher power generators, a three-phase machine was converted into a six-phase one to overcome the current limit imposed by the circuit breakers of that period. However, the development of multiphase solutions slowed sharply after the increase of the maximum allowable current of the circuit breakers[1].

During the end of the last century new technologies have been developed with the increasing of power electronic components' performances. So, motors with more phases than the standard three began to come up into the market with the aim to reach better behaviour and higher reliability. With studies like [2] was started the analysis of new structures that allowed more torque, less torque-ripple, cheaper power electronics and more degrees of freedom about the control side.

Nowadays, the main field in which this new technology is a standard are ship and aircraft applications, because they clearly need safe and continuous working operations. In this context, they use multiphase motors that are already a robust and consolidated technology which allows them to have better dynamics as regards the fault-tolerance behaviour [3], [4].

However, structures like the latter are going to be used even in industrial applications with the goal to reach better torque performances and better exploitation of the single components. In this way, obtaining more degrees of freedom and different kind of control strategies with which the customers may be facilitated to use new techniques is easier, understanding in addition which kind of motor is better for the application, if an Internal Permanent Magnets (IPM) or a Surface Permanent Magnets (SPM) one, as in [5], which are the trend in industrial applications.

Besides, with multiphase motors optimizing volumes, costs and efficiencies is possible, thanks to the better modularity given by the system, now composed on more smaller components as regards the power electronics side.

Finally, summarizing the positive effect of a multiphase machine, it is possible to

obtain:

- slightly higher torque value
- reduced torque ripple
- better efficiency
- improved power factor for high load
- extended speed range
- more fault tolerance
- modularity of power electronics (cheaper components and optimised volumes)
- more degrees of freedom with the control strategy

However, the complexity of the control and of the hardware system is increased.

Another aspect has to be considered, since not all the windings configurations are possible for each machine, because each kind of multiphase motor needs a proper shift angle between the different windings, in order to obtain the correct spatial positioning, as presented in [6], [7] and [8]. As regards the problem just mentioned, Finite Element Analysis (FEM) were used in order to study the correctness of the windings distribution, following the suggestions of [9], [10] and [11]: FEM studies have been used to analyse the machine and to extract currents, fluxes and torque from the model.

The control of a system with multiple phases has to take into account the mutual coupling between each different system, so even the technique used to control the motor has to be changed, considering a decoupling algorithm and different kind of Clarke transformation [12], [13]. They will be presented and explained in the following paragraphs, as shown in [1], [14], [15], [16]: a common and differential mode control will be presented.

In conclusion. the model does not take into account a lumped approach but a 2D or 3D maps based one, thanks to flux-maps obtained with procedures as described in [17].

The aim of the following work is to study multiphase configurations and how they can be adopted in an industrial application, showing as an already existing motor used in ETEL can be renovated: starting from a standard three-phase configuration the presented thesis will show how the control strategy can be modified and what change there are in it, with the purpose to achieve the control of four threephase

systems of an Internal Permanent Magnets (IPM) motor. Besides, even the supply source has chosen, considering different single threephase inverters independently, in order to achieve a better modularity, as it can be seen in [18].

The approach adopted took into account a simulation ambient, using finite element software to study the proper windings configuration and the own behaviour of the motor, and then combining it with block logic simulation and C-language in order to design the new control chosen.

Software like Matlab, Simulink, FEMM and Syre have been used to simulate the control's dynamic and the windings configuration of the system. Values are obtained via 2D simulations for the sake of comparison of the different configurations presented.

The work is organised as follow:

- Introduction to the theory of multiphase motor (windings configuration, motor model and control) and explanation of the software used, presenting even the approach adopted.
- Presentation of the Finite Element Method simulations and comparing them with the Syre's ones, introducing the new software. The improvements due to the increasing number of phases will be shown.
- Control simulation ambient: description in details of the Simulink system used and C-script adopted to design the control of the motor drive, comparing the results between the standard threephase system, the dual-threephase one and the four-threephase machine.
- Conclusion as overview of obtained results and used to suggest the possible future works related to this thesis.

Finally, the thesis will give to the readers all the elements to analyse the advantages of a multiphase structure and to understand the feasibility of the presented design of motor, showing that industrially speaking it is a solution which should be taken into account, even for future products.

Chapter 1

State of the art and machine model

1.1 Model of multiphase machines

1.1.1 Multiphase machines windings

When a threephase motor has to be redesigned in a multiphase one, several different aspects have to be considered: firstly the kind of windings to choose represents the main problem which comes up at the beginning. Not all the motors can be adapted with any number of phases because each one of the latter need a specific shift angle between two adjacent windings, especially if they are studied with the most used Vector Space Decomposition (VSD) technique, as explained later.

From the stator winding point of view, multiphase machines can be equipped with Distributed Windings (DWs) as well as with Fractional-Slot Concentrated Windings (FSCWs). With respect to the DWs, the second solution is characterized by shorter end-windings length. This results in lower machine axial length and lower Joule losses, and thus in an enhanced power and torque density of the machine. For this reason, the FSCWs are often used for surface mounted PM and SR machines in many industrial sectors. In addition, FSCWs allow to have an increased value of inductance, which is reflected in less short-circuit current within the system. So, in the following paragraphs, only windings typology as FSCWs will be considered.

As shown in Fig.1.1, two main layouts for FSCWs are possible: the Double-Layer (DL) and the Single-Layer (SL). In particular, in the DL configuration each stator tooth carries a coil; therefore, in each slot there are conductors of two different phases or coils. In the SL configuration, the stator teeth are wound alternatively; thus in each slot there are conductors of the same phase only [1]: as regards the flux oscillations, the first option has to be preferred, which guarantees the best

uniformity of the rotating field. In this way, even the reluctance seen from each winding has not huge oscillations and the torque ripple due to rotor's asymmetric components is reduced.

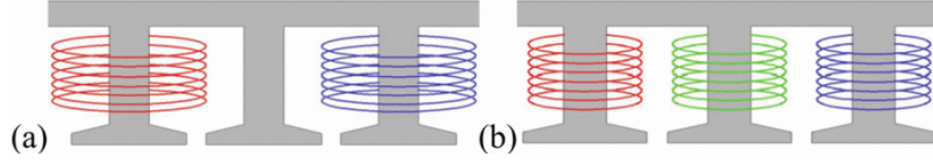


Figure 1.1: Fractional-slot concentrated windings: a) Single Layer (SL); b) Double Layer (DL).

An other aspect that has to be taken into account is the power supply. There is the possibility to consider an independent single coil configuration, in which each coils is supplied by an own inverter, or the multi threephase one, in which the windings are managed in order to create several threephase stars [1], like in Figg.1.2-1.3. At the end, there is also the possibility to obtain the same configuration of Fig.1.2 but with a common point in order to create a multiphase star: this case was not considered in this work since it needs a proper control of the common mode that influences the dynamic of the motor. For the same reasons, also the drive with multiple threephase with the same neutral point was ignored.

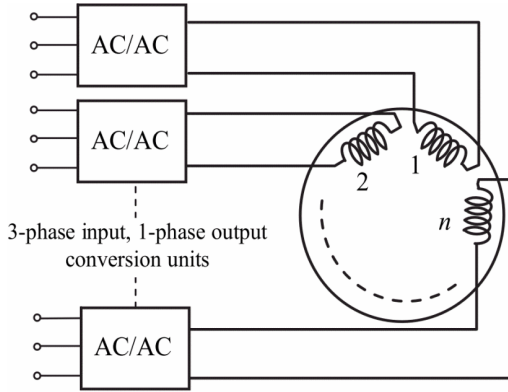


Figure 1.2: Drive with multiple single-phase units, [4].

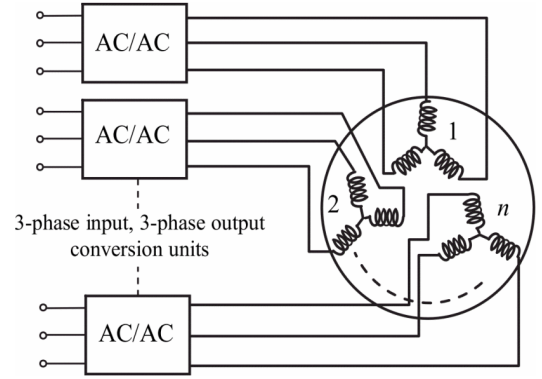


Figure 1.3: Drive with multiple three-phase units,[4].

The main advantage of the first solution is the maximum number of degrees of freedom, because each coil can be controlled properly in order to obtain the best exploitation of the power electronics and the best quality of the motor performances.

However the control of the latter structure is more complex because a synchronization among all the inverters is needed and there are the power oscillation at a doubled frequency of the supply that incomes for each inverter, since each one of them is a singlephase system: from the perspective of a commercial industrial product it is less attractive.

Often is therefore chosen the second kind of structure, in Fig.1.3, that allows to manage more three phase systems, well known in the standard applications and feasible even for customers that want to improve their product easier. Clearly, as regards the fault tolerant condition, multiple threephase solution reaches a lower level of modularity than the singlephase one: if a fault occurs, an entire threephase system has to be switched off, while with the first solution only a phase can be separated from the system and all the other ones can continue to work with a small overload.

At the end, multiple threephase units is a good trade-off between complexity and improvement of performances, so it was the solution adopted in the present thesis.

Then, considering some FSCWs, DL windings in this work, Fig.1.4 shows the different kind of connections possible for a sixphase motor (the simplest version), representing only a three-phase system: depending on the distribution of the windings the torque and its ripple can change, reaching the best values for the case (c), as described in [6].

In addition, the symmetrical arrangement of the coils yields the DL-3 configuration to achieve the higher overload and thermal performance, which results to be useful in order to improve the torque capability under a fault.

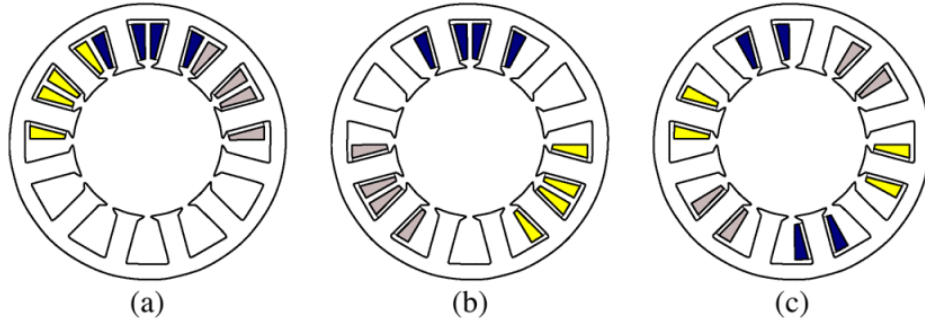


Figure 1.4: Three examples of the remaining phase coils after removing one of the two three-phase DL windings. (a) DL-1. (b) DL-2. (c) DL-3, [6].

The main drawback of the DL-3 configuration is the mutual coupling between

the spatial phases of two windings. It involves a lower torque during a threephase short-circuit fault and the presence of undesired mutual EMF in the windings [3]. In addition, the choice selected has the best performances in fault conditions, in which the distribution of the windings remains more uniform along the circumference of the air-gap, showing dynamics similar to a threephase engine; while in the case (a) of Fig.1.4 the oscillations are too huge to be used, and in the case (b) they are still high enough to be considered not useful as motor focused on a fault tolerance condition.

For the reasons just presented, the third configuration was chosen for the motor analysed in the thesis.

In addition, depending on the own structure and shape of the motor used, an other choice about configuration of the windings can be made by the designer.

In literature it is known that there are two main "families" of multiphase motor: symmetrical and asymmetrical. When a "symmetric" machine is chosen, it means that the first phases of the two stars have to be spaced by $\frac{360^\circ}{N_{ph}}$ electrical degrees, like in Fig.1.5.

However, when the number of phases N_{ph} is multiple of three and an even number (e.g., $N_{ph} = 6, 12$), the machine is usually designed as "asymmetric", with the electrical displacement between the magnetic axes of two adjacent stator phases of $\frac{180^\circ}{N_{ph}}$, as shown in Fig.1.6, [1], [4],[13].

The latter are machines more interesting than the first ones, since they avoid the sixth torque harmonic, as mentioned even in [13].

In this context, considering twelve phases, the displacement angle between two stars needed is of 15° electrical degrees, if an asymmetric machine is considered. Certainly, having a stator that allows any kind of multiphase motor is not always possible: for this reason, different kind of modelling techniques have been created, and in the following chapter the two main ones will be described in detail.

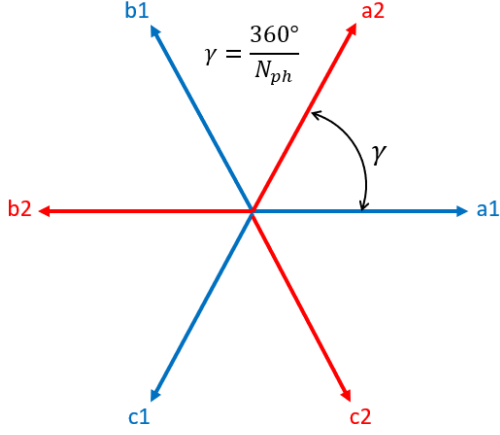


Figure 1.5: Symmetric windings star for a sixphase motor.

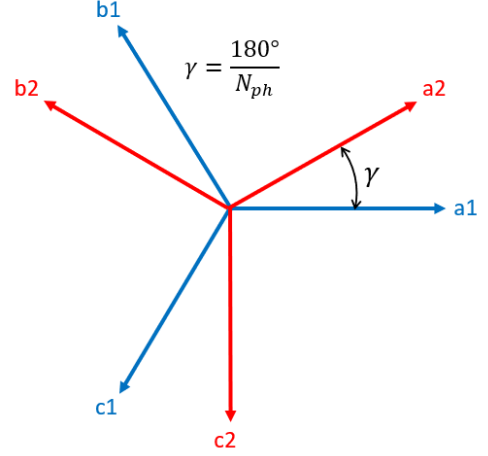


Figure 1.6: Asymmetric windings star for a sixphase motor.

At the end, as regards the winding configuration considered in this work, some FSCWs, DL-3, multiple threephase, asymmetric structures was chosen.

1.1.2 Multistator Vector Space Decomposition technique's fundamentals

One of the most used techniques to represent a multiphase motor is the so called "Multistator" (MS) approach. The name derives from considering the machine as the sum of several independent machines, each one with an own Clarke transformation, like 1.1, that allows to control the drive totally even if the structure has not the possibility to have the standard displacement angle for a specific number of phases. For instance, if a four threephase machine is required but the stator structure does not permit to use the 15° common for an asymmetric machine, with MS approach is possible to describe the motor considering each star as an independent motor with its proper (d,q) domain but coincident with the other ones. Of course the torque will be split depending of the sharing-ratio chosen for each set.

$$[C_k] = \frac{2}{3} \cdot \begin{bmatrix} \cos(\gamma_{s_k}) & \cos(\gamma_{s_k} + 2\pi/3) & \cos(\gamma_{s_k} + 4\pi/3) \\ \sin(\gamma_{s_k}) & \sin(\gamma_{s_k} + 2\pi/3) & \sin(\gamma_{s_k} + 4\pi/3) \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \quad (1.1)$$

In the previous equation γ_{s_k} is the characteristic angle of the considered set ($\pi/6, \pi/12, \dots$).

In fact, the aim of the MS method is to described each star into the same (d,q) frame, in order to make coincident the d-axis with the main flux direction, equal for all the threephase systems, as in Fig.1.7.

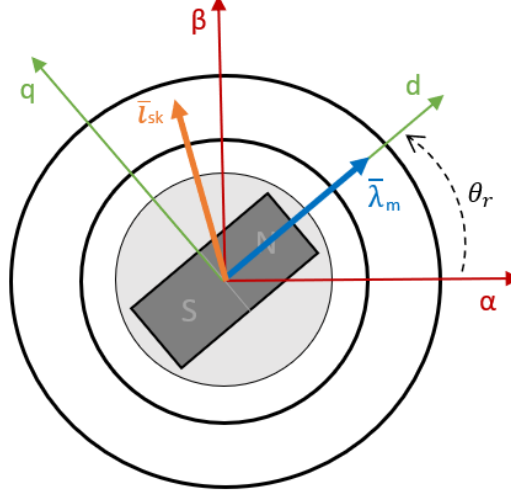


Figure 1.7: (d,q) frame adopted in MS approach.

Finally, the technique just described has to take into account the natural coupling between the different threephase sets, which needs a decoupling algorithm in order to improve the dynamics and make the motor usable.

MS approach, however, was not used in the present work for its complexity in the control strategy but it was mandatory presenting it to show how a multiphase machine can be represented with the greatest number of degrees of freedom available.

In contrast, VSD approach simplifies the analysis of the multiphase motors considering them as a whole structure and neglecting their modularity as multi threephase systems. A sixphase machine will be considered in the following paragraphs in order to make simpler the treatment.

Since a sixphase machine has six independent currents, the structure can be considered as a system with six possible variables, which mathematically speaking is like describing a space with six dimensions. Geometrically speaking is like rotating a vector around the origin at a certain speed, spanning a surface in the six-dimensions space. However, controlling this kind of system is really hard and it can be seen from a different point of view, making the whole problem easier to be studied.

The main idea, in fact, is creating a new vectors' base with which describing the

entire system is more feasible, choosing the new six vectors that, being orthogonal each other, allow to analyse the motor more quickly.

Considering the general vector in the following equation 1.2 it can be seen that each vector obtained with $\gamma = \pi/6$ and $k = 0,1,3,5, \dots$ harmonics spans a two-dimensional surface, which can be considered as a subspace of the original six-dimensional one.

$$S_k(\omega t) = [\cos k(\omega t), \cos k(\omega t - \gamma), \cos k(\omega t - 4\gamma), \cos k(\omega t - 5\gamma), \cos k(\omega t - 8\gamma), \cos k(\omega t - 9\gamma)]^T \quad (1.2)$$

It is known that the electromechanical energy conversion is brought by the fundamental component which spans an own surface: inside the latter two vector will be found in order to describe the system as a standard one, along the (d,q) frame.

Consequently, selecting values of $k = 1$ in 1.2 and the two instants $\omega t = 0$ and $\omega t = \pi/2$ to obtain two orthogonal vectors, the **d** and **q** axis are found, as follow:

$$\begin{aligned} d &= [1, \cos(\gamma), \cos(4\gamma), \cos(5\gamma), \cos(8\gamma), \cos(\gamma)]^T, \quad \omega t = 0 \\ q &= [0, \sin(\gamma), \sin(4\gamma), \sin(5\gamma), \sin(8\gamma), \sin(\gamma)]^T, \quad \omega t = \pi/2 \end{aligned} \quad (1.3)$$

Replacing in the latter relation $\gamma = \pi/6$ consistently with the sixphase motor drive, it can be noticed that the 1.3 is related to the configuration in Fig.1.6 with the variables' order of 1.4.

$$[X] = [x_{a1}, x_{a2}, x_{b1}, x_{b2}, x_{c1}, x_{c2}]^T \quad (1.4)$$

Now the other four vectors have to be orthogonal and other order of harmonics from 1.2 have to be considered in order to complete the six-dimensional space. It is noticed that surfaces with $k = 1,3,5$ are orthogonal each other, as follow:

$$S_1^T \bullet S_3 = S_1^T \bullet S_5 = S_3^T \bullet S_5 = 0 \quad 0 \leq \omega t \leq 2\pi \quad (1.5)$$

The same results can be obtained with higher order of harmonics ($k = 8,9, \dots$) which were not considered since they generate the same surfaces of the previous ones. Then, considering $k = 3$:

$$\begin{aligned} o_1 &= [1, 0, 1, 0, 1, 0]^T \quad \omega t = 0 \\ o_2 &= [0, 1, 0, 1, 0, 1]^T \quad \omega t = \pi/2 \end{aligned} \quad (1.6)$$

While, considering $k = 5$ can be obtained:

$$\begin{aligned} x &= [1, \cos(5\gamma), \cos(8\gamma), \cos(\gamma), \cos(4\gamma), \cos(9\gamma)]^T & \omega t = 0 \\ y &= [0, \sin(5\gamma), \sin(8\gamma), \sin(\gamma), \sin(4\gamma), \sin(9\gamma)]^T & \omega t = \pi/2 \end{aligned} \quad (1.7)$$

Six vectors have been obtained and a new base of six orthogonal vectors can be taken into account. The new reference has different properties:

1. k_{th} harmonics orders represents the energy conversion of the machine, with $k = 12m \pm 1 (m = 1, 2, 3, \dots)$, and they are transformed in the (d,q) plane as a standard threephase motor.
2. all the harmonics with the order $k = 6m \pm 1 (m = 1, 3, 5, \dots)$ are represented in the (x,y) reference, in which the phenomena regard each sub-harmonic and leakage fluxes.
3. Zero sequence harmonics are mapped into (o_1, o_2) frame to form the conventional zero sequence not related with energy conversion of the system.

1.1.3 Machine model

The following hypothesis have to be taken into account to study properly the model of the motor. As regards the latter, an IPM engine has taken into account:

1. The windings of the machine are ideal and sinusoidally distributed.
2. All the parameters of the different threephase systems are the same.
3. Effects in the windings like proximity and skin effect are neglected.
4. The magnetic linearity is considered for the equations.

In the original sixphase space the motor can be described with the conventional equations, as follow.

As regards the stator voltage equation:

$$[V_s] = [R_s] \cdot [i_s] + \frac{d[\lambda_s]}{dt} \quad (1.8)$$

while for the magnetic equation:

$$[\lambda_s] = [L_s] \cdot [i_s] + [\lambda_m] \quad (1.9)$$

where each vector can be represented as the general one, in which the first three terms are for the first star, while the second three ones represent the second threephase system, in a different way of 1.4:

$$[X] = [x_{a1}, x_{b1}, x_{c1}, x_{a2}, x_{b2}, x_{c2}]^T \quad (1.10)$$

Replacing 1.9 in 1.8, a more complete equation can be reached, in order to be transformed in the new six dimensional space:

$$[V_s] = [R_s] \cdot [i_s] + \frac{d([L_s] \cdot [i_s] + [\lambda_m])}{dt} \quad (1.11)$$

where the inductances matrix is:

$$[L_s] = L_{ls} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + L_m \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{\sqrt{3}}{2} & 0 & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \quad (1.12)$$

As for the conventional standard model is made, each equation is multiplied by a transformation matrix which allows to obtain the (d, q, x, y, o_1, o_2) system, as mentioned before. As consequence, the multiplication can be written:

$$[T_6] \cdot [V_s] = [T_6] \cdot [R_s] \cdot [T_6^{-1}] \cdot [T_6] \cdot [i_s] + \frac{d([T_6] \cdot [L_s] \cdot [T_6^{-1}] \cdot [T_6] \cdot [i_s] + [T_6] \cdot [\lambda_m])}{dt} \quad (1.13)$$

Considering the transformation matrix as result of the different transformation described in 1.3, 1.6 and 1.7 and taking into account of the order of the variables in 1.10, writing the extended transformation matrix is possible:

$$[T_6] = \frac{1}{3} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -1 \\ 1 & -\frac{1}{2} & -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (1.14)$$

The just written matrix is fundamental in the future model of the motor in Simulink because it gives the possibility to represents the sixphase machine, changing from the six dimensional representation to the $(\alpha, \beta, x, y, o_1, o_2)$ and treating the (α, β) space as the conventional threephase one. Then, multiplying the

latter for the following well known rotational matrix, obtaining the (d,q) domain is possible in order to control the machine as the standard technique. The zero sequence has been neglected since the neutral point of each threphase system has been considered isolated.

$$[R]_{\alpha,\beta \rightarrow d,q} = \begin{bmatrix} \cos(\theta_r) & \sin(\theta_r) \\ -\sin(\theta_r) & \cos(\theta_r) \end{bmatrix} \quad (1.15)$$

Now the stator machine model in (d,q) subspace is available, if 1.15 is inserted in 1.13, and the new model is:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} + \mathbf{J} \cdot \omega_r \cdot \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} \quad (1.16)$$

where \mathbf{J} is for the mutual coupling in (d,q) frame:

$$[J] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (1.17)$$

while $\omega_r = p \cdot \omega_m$ and $[\lambda_d, \lambda_q]^T$ is:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = L_{ls} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} n \cdot M_d & 0 \\ 0 & n \cdot M_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \quad (1.18)$$

with n that is the number of threephase sets.

The main difference between a model of sixphase motor and a standard one is the presence of a new subspace, the $x-y$, which represents the unbalancing phenomenon of the stars. The equation 1.19 gives the information about the subspace just mentioned and represents the dynamic of the leakage fluxes.

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \cdot \begin{bmatrix} i_x \\ i_y \end{bmatrix} + \frac{d}{dt} \left\{ \begin{bmatrix} L_{ls} & 0 \\ 0 & L_{ls} \end{bmatrix} \cdot \begin{bmatrix} i_x \\ i_y \end{bmatrix} \right\} \quad (1.19)$$

And finally even the zero-sequence subspace can be written, with the same meaning of the standard threephase motor:

$$\begin{bmatrix} v_{o1} \\ v_{o2} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \cdot \begin{bmatrix} i_{o1} \\ i_{o2} \end{bmatrix} + \frac{d}{dt} \left\{ \begin{bmatrix} L_{ls} & 0 \\ 0 & L_{ls} \end{bmatrix} \cdot \begin{bmatrix} i_{o1} \\ i_{o2} \end{bmatrix} \right\} \quad (1.20)$$

As can be noticed, 1.19 and 1.20 are quite similar in the form. The reasons is that they represent two different subspace but always relative to fluxes that do not contribute to energy conversion and torque generation. In fact, as can be seen in 1.21 after a power balancing of the system, the torque expressed with (d,q) variables of the multiphase machine is in general:

$$T = \frac{n}{2} \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \quad (1.21)$$

where n represents the number of phases of the system. The final result is that each star can share a certain amount of torque with the other ones: in this way, the total torque available is the sum of the single contribute of every stars, which sees the own torque reduced in a inversely proportional respect to the number of the phases.

The main difference that comes up is the multiplication factor that in the sixphase model is 3 and not more 3/2 as in the traditional threephase motor, as below.

$$T = 3 \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \quad (1.22)$$

The sixphase case has been showed, and now a brief presentation of the theory regard the twelvephase will be presented.

The model of the motor with twelve phases can be considered analogous to the one presented in equations 1.8 and 1.9, but instead of the general vector of 1.10, it has to be used the following one:

$$[X] = [x_{a1}, x_{b1}, x_{c1}, x_{a2}, x_{b2}, x_{c2}, x_{a3}, x_{b3}, x_{c3}, x_{a4}, x_{b4}, x_{c4}]^T \quad (1.23)$$

The rotational matrix of 1.15 used for the previous version is still correct, since the VSD approach has the aim to describe the motor in the (α, β) domain, the same for all the stars: the transformation from (α, β) to (d, q) frame is always the same, independently if the original model has three, six or twelve phases.

The transformation matrix is the factor that changes and allows to describe the system from the original twelve phases to the six bidimensional sub-spaces, according to the new variables' description similar to the sixphase already seen in the previous paragraphs:

$$[X_{dq}] = [x_d, x_q, x_{x1}, x_{y1}, x_{x2}, x_{y2}, x_{x3}, x_{y3}, x_{o1}, x_{o2}, x_{o3}, x_{o4}]^T \quad (1.24)$$

The matrix that allows the transformation just mentioned is the following one, with which is possible to describe the system using a (d, q) domain for the energy conversion phenomena, three bidimensional sub-planes linked to the leakage fluxes and not related to the torque production, and finally the four zero-sequence sub-planes that represent the common voltage mode (one for each threephase system), which are negligible since the four threephase system have not a common neutral point:

$$\begin{aligned}
 [T_{12}] &= \frac{1}{6} \cdot \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} \\ C_{5,1} & C_{5,2} & C_{5,3} & C_{5,4} \\ C_{7,1} & C_{7,2} & C_{7,3} & C_{7,4} \\ C_{11,1} & C_{11,2} & C_{11,3} & C_{11,4} \\ Z \end{bmatrix} \\
 C_{l,g} &= \begin{bmatrix} \cos(\frac{l(g-l)\pi}{12}) & \sin(\frac{l(g-l)\pi}{12}) \\ \cos(\frac{l(g-l)\pi}{12} + \frac{2l\pi}{3}) & \sin(\frac{l(g-l)\pi}{12} + \frac{2l\pi}{3}) \\ \cos(\frac{l(g-l)\pi}{12} - \frac{2l\pi}{3}) & \sin(\frac{l(g-l)\pi}{12} - \frac{2l\pi}{3}) \end{bmatrix}^T \\
 [Z] &= \begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \end{bmatrix}
 \end{aligned} \tag{1.25}$$

where $l = 1, 5, 7, 11$ and $g = 1, 2, 3, 4$. Implementing the algorithm of 1.25 with a MATLAB function, an example of result can be gave even with inverse matrix:

```

T_12 =

    0.1667    -0.0833    -0.0833     0.1610    -0.1179    -0.0431     0.1443    -0.1443     0.0000     0.1179    -0.1610     0.0431
         0     0.1443    -0.1443     0.0431     0.1179    -0.1610     0.0833     0.0833    -0.1667     0.1179     0.0431    -0.1610
    0.1667    -0.0833    -0.0833     0.0431     0.1179    -0.1610    -0.1443     0.1443     0.0000    -0.1179    -0.0431     0.1610
         0    -0.1443     0.1443     0.1610    -0.1179    -0.0431     0.0833     0.0833    -0.1667    -0.1179     0.1610    -0.0431
    0.1667    -0.0833    -0.0833    -0.0431    -0.1179     0.1610    -0.1443     0.1443    -0.0000     0.1179     0.0431    -0.1610
         0     0.1443    -0.1443     0.1610    -0.1179    -0.0431    -0.0833    -0.0833     0.1667    -0.1179     0.1610    -0.0431
    0.1667    -0.0833    -0.0833    -0.1610     0.1179     0.0431     0.1443    -0.1443    -0.0000    -0.1179     0.1610    -0.0431
         0    -0.1443     0.1443     0.0431     0.1179    -0.1610    -0.0833    -0.0833     0.1667     0.1179     0.0431    -0.1610
    0.3333     0.3333     0.3333         0         0         0         0         0         0         0         0         0
         0         0         0     0.3333     0.3333     0.3333         0         0         0         0         0         0
         0         0         0         0         0         0     0.3333     0.3333     0.3333         0         0         0
         0         0         0         0         0         0         0         0         0     0.3333     0.3333     0.3333

T_12_inv =

    1.0000    -0.0000     1.0000    -0.0000     1.0000     0.0000     1.0000     0.0000     1.0000    -0.0000     0.0000    -0.0000
   -0.5000     0.8660    -0.5000    -0.8660    -0.5000     0.8660    -0.5000    -0.8660     1.0000    -0.0000    -0.0000     0.0000
   -0.5000    -0.8660    -0.5000     0.8660    -0.5000    -0.8660    -0.5000     0.8660     1.0000     0.0000    -0.0000     0.0000
    0.9659     0.2588     0.2588     0.9659    -0.2588     0.9659    -0.9659     0.2588    -0.0000     1.0000    -0.0000     0.0000
   -0.7071     0.7071     0.7071    -0.7071    -0.7071    -0.7071     0.7071     0.7071     0.0000     1.0000     0.0000    -0.0000
   -0.2588    -0.9659    -0.9659    -0.2588     0.9659    -0.2588     0.2588    -0.9659         0     1.0000     0.0000    -0.0000
    0.8660     0.5000    -0.8660     0.5000    -0.8660    -0.5000     0.8660    -0.5000     0.0000    -0.0000     1.0000    -0.0000
   -0.8660     0.5000     0.8660     0.5000     0.8660    -0.5000    -0.8660    -0.5000    -0.0000     0.0000     1.0000     0.0000
   -0.0000    -1.0000     0.0000    -1.0000    -0.0000     1.0000     0.0000     1.0000    -0.0000     0.0000     1.0000         0
    0.7071     0.7071    -0.7071    -0.7071     0.7071    -0.7071    -0.7071     0.7071    -0.0000     0.0000    -0.0000     1.0000
   -0.9659     0.2588    -0.2588     0.9659     0.2588     0.9659     0.9659     0.2588     0.0000    -0.0000     0.0000     1.0000
    0.2588    -0.9659     0.9659    -0.2588    -0.9659    -0.2588    -0.2588    -0.9659         0    -0.0000     0.0000     1.0000
    
```

Figure 1.8: Example of the matrix T_{12} and T_{12}^{-1} used for the VSD technique in the four-threephase motor drive model.

Besides, considering variables like in 1.24, writing the following relation in (d,q) subspace and the other ones is possible. The new representation of the system is

easier to be studied, in fact the different subspaces are completely decoupled and independently treated. The great advantage of the VSD approach will be seen in the following section, in which the used control strategy is explained and the fact that the sub-planes are decoupled will have a crucial importance.

$$\begin{aligned} [v_{dq}] &= [R_s] \cdot [i_{dq}] + [L_{dq}] \cdot \frac{d[i_{dq}]}{dt} + \mathbf{J} \cdot \omega_e \cdot [L_{dq}] \cdot [i_{dq}] + \omega_e \cdot [\lambda_m] \\ [v_{x_i y_i}] &= [R_s] \cdot [i_{x_i y_i}] + [L_{x_i y_i}] \cdot \frac{d[i_{x_i y_i}]}{dt} + \mathbf{J} \cdot \omega_e \cdot [L_{x_i y_i}] \end{aligned} \quad (1.26)$$

In the previous equations the several matrices can be considered as below:

$$\begin{aligned} [v_{dq}] &= [v_d, v_q]^T \\ [v_{x_i y_i}] &= [v_{x_i}, v_{y_i}]^T \\ [i_{dq}] &= [i_d, i_q]^T \\ [i_{x_i y_i}] &= [i_{x_i}, i_{y_i}]^T \\ [R_s] &= \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \\ [L_{dq}] &= \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \\ [L_{x_i y_i}] &= \begin{bmatrix} L_{ls} & 0 \\ 0 & L_{ls} \end{bmatrix} \\ [\lambda_m] &= [0, \lambda_{PM}]^T \end{aligned} \quad (1.27)$$

where $i = 1, 2, 3, 4$, L_{ls} is for the leakage inductance and $L_{dq} = L_{ls} + n \cdot M_{dq}$. The matrix \mathbf{J} is the same of the previously seen in 1.17.

However, since for the control dynamics flux-maps will be used instead of the lumped model with the inductance, the matrix of all the self and mutual coupling inductances in the original twelve-dimensional plane is not shown and not more considered.

Finally, the torque equation can be written even for a twelve phases motor, considering 1.21:

$$T = 6 \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \quad (1.28)$$

A final comment has to be done for the equations 1.22 and 1.28 because the fluxes and currents used to compute the torque are related to the single threephase system. In fact, while the VSD theory described the whole motor in a common (d,q) domain, the flux-maps are related to a single system, because the simulations

used to compute them had as output that kind of matrices. For the just mentioned reason, the torque has to be computed as seen, multiplying for a constant factor which takes into account the stars' number.

In the following paragraphs the control logic will be explained.

1.2 Field-oriented control of the multiphase machine

1.2.1 Simulation ambient and type of the data used

Even if the software Syre will be described in the following chapters, in the present section it will be briefly introduced as regards its output results, in order to understand what kind of data can be used for the control strategy and which is the simulation ambient for the control.

Firstly, the general simulation ambient has to be shown, in order to understand how the model is organised making the control theory also clearer.

Syre has the possibility to select from a folder the file which has inside the Simulink model of the whole motor drive and use it with the correct parameters of the studied machine. The model represents the dynamic of each motor drive's component, from the microprocessor to the motor, considering even the inverter and all the sampling phenomena that characterize the digital control method.

What is shown in Fig.1.9 is the default Simulink model about a threephase system: the model will be different if compared with a multi-phase motor, but describing how to simulate the whole motor drive is not the aim of this section, which has as purpose to explain the logic flow of the data and describing the control scheme.

Below the main features of the system used in Simulink, representing the general model in Fig.1.9 and the mechanical coupling between the motor and the load in Fig.1.10.

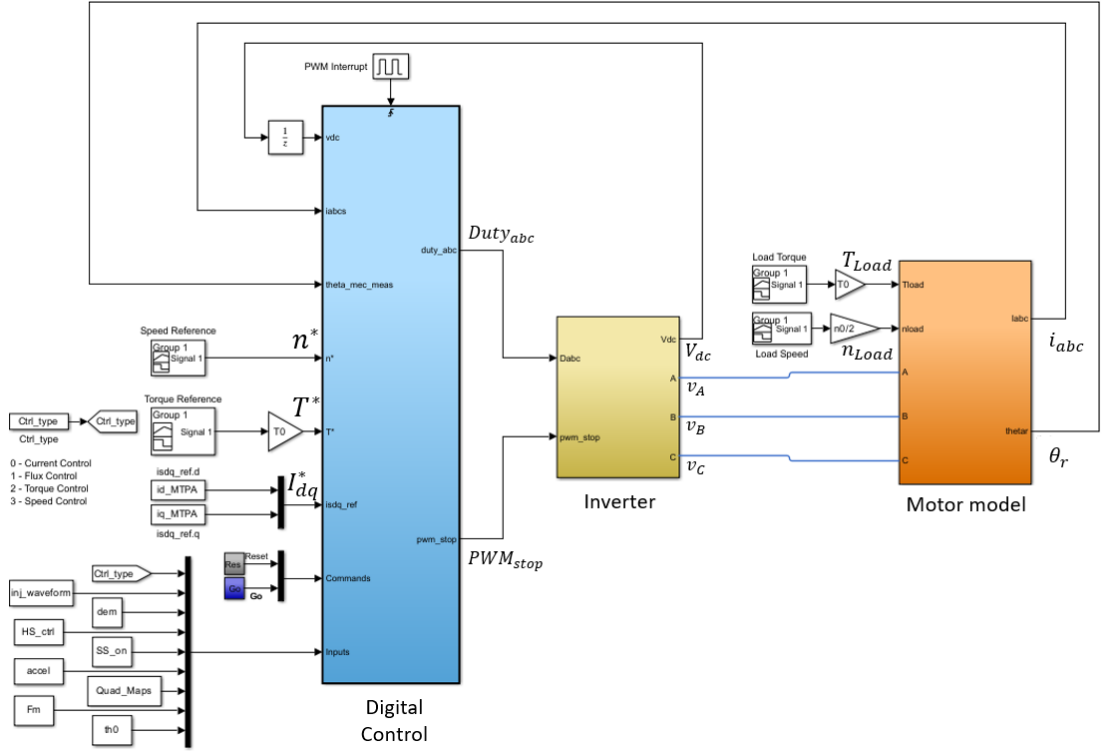


Figure 1.9: Example of a 3-phase Simulink model as Syre output.

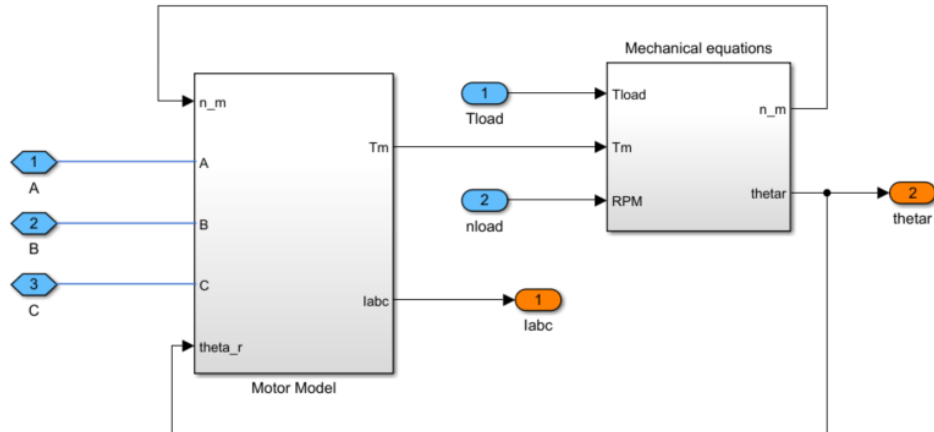


Figure 1.10: Components of Simulink's "Motor model" block.

As discussed in the previous section, it has not been used a lumped approach for the (d,q) plane because the model of the electric motor was mapped along the

whole working area and the relationship between currents and fluxes is described thanks to 1D or 2D LUTs (Look Up Tables). The lumped approach was used only for the (x_i, y_i) plane, for which considering LUTs for the leakage inductances is hard or almost negligible phenomenon.

Using Syre, obtaining the relation between fluxes and currents (direct and inverse maps) is possible like in Figg.1.11-1.12, as well as for the iron losses, which are related with the currents always thanks to some maps and with each different contribution independently: harmonics, permanent magnets and eddy currents contribution are considered separately and then summed together.

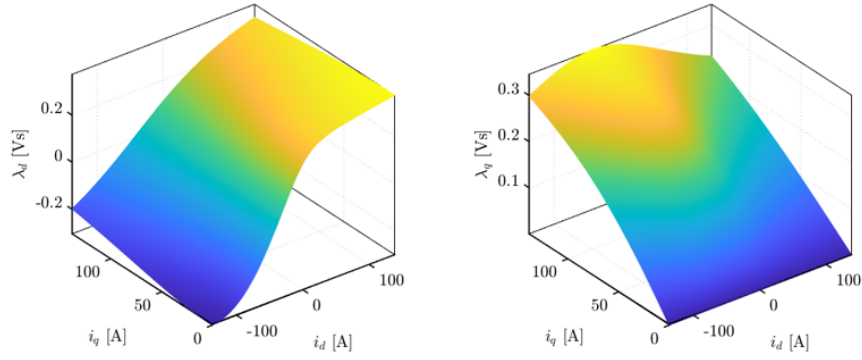


Figure 1.11: Direct flux-maps $\lambda_{dq}(i_d, i_q)$.

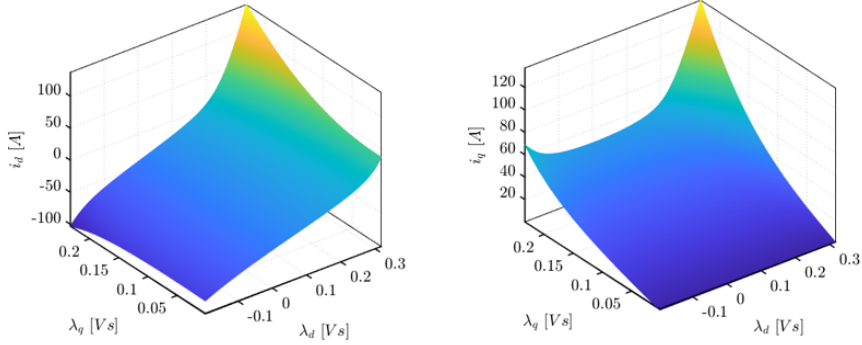


Figure 1.12: Inverse flux-maps $i_{dq}(\lambda_d, \lambda_q)$.

The different machine models are represented as several threephase system and then the different variables are transformed into the (d,q) domain as explained in section 1.1.3 and shown in chapter 3.1. Then, LUTs related to inverted matrix and inverse maps like in Fig.1.12 are computed and the physical 3/6/12-phases

quantities are extracted and used as feedback in the control logic using the just shown inverse transformations. More details there are in the following paragraphs. Obtaining the currents in the motor model from the fluxes (computed by integration of the back electromotive force) and as consequence emulating the machine behaviour is at the end possible; besides, with 1.11 used in the digital control, implementing the flux-observer algorithm is a possibility.

Finally, with all the information gave by Syre about the motor, even the Maximum Torque Per Ampere (MTPA) characteristic can be obtained and used to control the motor drive: the MTPA represents the minimum current's value to be used in order to give to the load the required torque, so in this way minimizing the current and reducing the Joule losses is possible thanks to a 1D LUTs. Therefore, the MTPA takes into account the anisotropy of the machine using the flux-maps. It has to be highlighted that the flux-weakening behaviour has not taken into account in this work, so the Maximum Torque Per Voltage (MTPV) characteristic has been computed to obtain the mechanical behaviour, but it has not been considered in the control side.

An example of the both characteristics are shown in Fig.1.13.

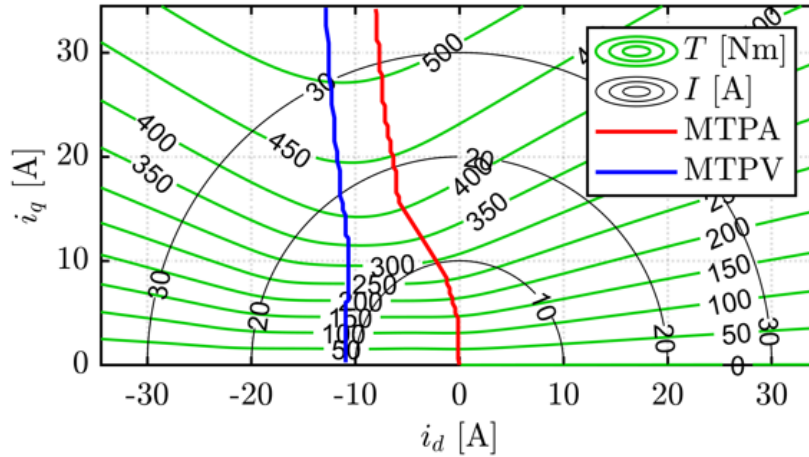


Figure 1.13: MTPA and MTPV curves as function of torque and currents along **d** and **q** axis.

The several variables needed for the control's study and just introduced before are computed thanks to several FEM simulations using FEMM and Syre.

As can be seen in Fig.1.14, the idea is to create a currents' grid, in which currents in the available (d,q) range are considered. Selecting each point in the (d,q) plane, and using the inverse Clarke and rotational transformation, the proper current can

be applied to each threephase set of the motor and then, using the direct ones, results like L_{dq} and λ_{dq} can be computed and saved in LUTs and used in a second moment in order to be interpolated during the simulation time. The explanation of the method to obtain the LUTs is not the aim of the present thesis, but it can be found a deep study in papers like [17]. An example of the results is shown in Fig.1.15, in which the flux along d and q axis is identified thanks the method explained before.

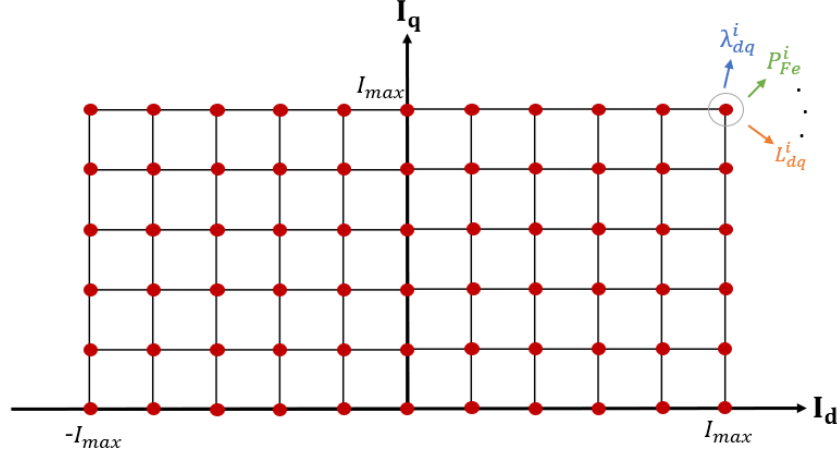


Figure 1.14: Current grid in (d,q) plane used to compute the fluxes' maps.

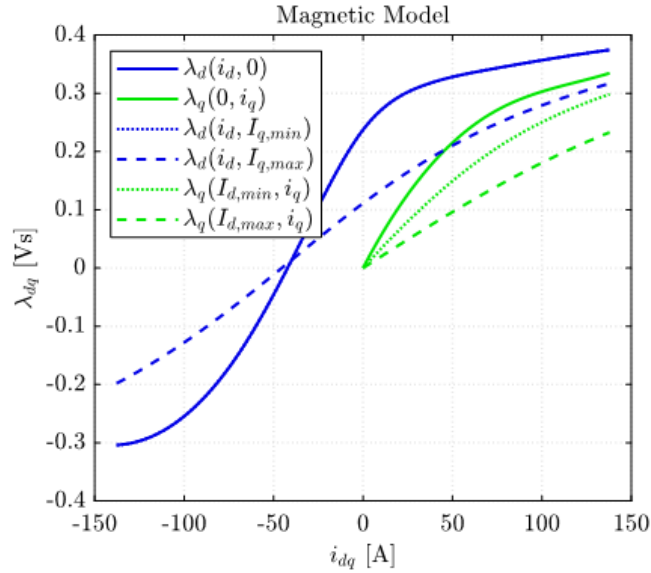


Figure 1.15: Flux identification using i_{dq} currents.

As regards the method to obtain the voltages to apply to each phase, the traditional PWM technique has been considered, obtaining the voltages thanks to the comparison between an high-frequency triangular waveform and a carrier signal that is compared with the latter. A more detailed analysis there is in section 3.1, in which each Simulink's simulation component is described. After a summary about the kind of data that will be treated in the study of the control, it is the moment to describe the control strategy and how it was implemented.

1.2.2 Control strategy and logic-block schemes

With the present work an alternative hybrid method to control a multiphase engine is proposed, considering the work of [12] but taking into account a different decoupling algorithm. The aim of the study is merging the two different techniques presented in section 1.1.2 and used to model the multi-threephase electric motor: the advantages are the modularity of MS and the easier way to represent the motor behaviour with the VSD, avoiding the more complex coupling algorithm to be considered with the first one. The goal to be reached is decomposing the (d,q) model of 1.16 and obtaining a multiple subspaces sets which represent the common and differential mode of the machine. The first one is directly related to the energy conversion with 1.16 and 1.21, with the meaning of the subspace that represent the real "power transmission". The differential mode is related to the unbalances between the different phases along the (d,q) domain, and its purpose is to be set to zero by the control side in order to have the most balanced and symmetrical system possible. Then, there is a decoupling matrix proposed in this thesis, that allows to change each threephase system (d,q) quantities into the decoupled sub-planes in common and differential mode, as below:

$$\begin{bmatrix} \bar{x}_{cm,dq} \\ \bar{x}_{dm,dq-1} \\ \dots \\ \bar{x}_{dm,dq-(k-1)} \end{bmatrix} = [D] \cdot \begin{bmatrix} \bar{x}_{dq-1} \\ \bar{x}_{dq-2} \\ \dots \\ \bar{x}_{dq-k} \end{bmatrix} \quad (1.29)$$

As already explained, the common mode has the meaning of the torque production while the differential one represents the differences between the several stars, the both in (d,q) coordinates. An inverse matrix has to be considered too, since some quantities have to be returned in the original (d,q) sub-planes for the PWM comparison.

For this reason, $[D]$ depends of how many sets there are into the system.

$$\begin{bmatrix} \bar{x}_{dq-1} \\ \bar{x}_{dq-2} \\ \vdots \\ \bar{x}_{dq-k} \end{bmatrix} = [D]^{-1} \cdot \begin{bmatrix} \bar{x}_{cm,dq} \\ \bar{x}_{dm,dq-1} \\ \vdots \\ \bar{x}_{dm,dq-(k-1)} \end{bmatrix} \quad (1.30)$$

Once the that the decoupling matrices have been introduced, the general control scheme can be presented, valid for every number of asymmetrical multi-trheephase machine and related to a direct torque-control, so it misses the speed loop, that is the standard one and depends from the mechanical physical world.

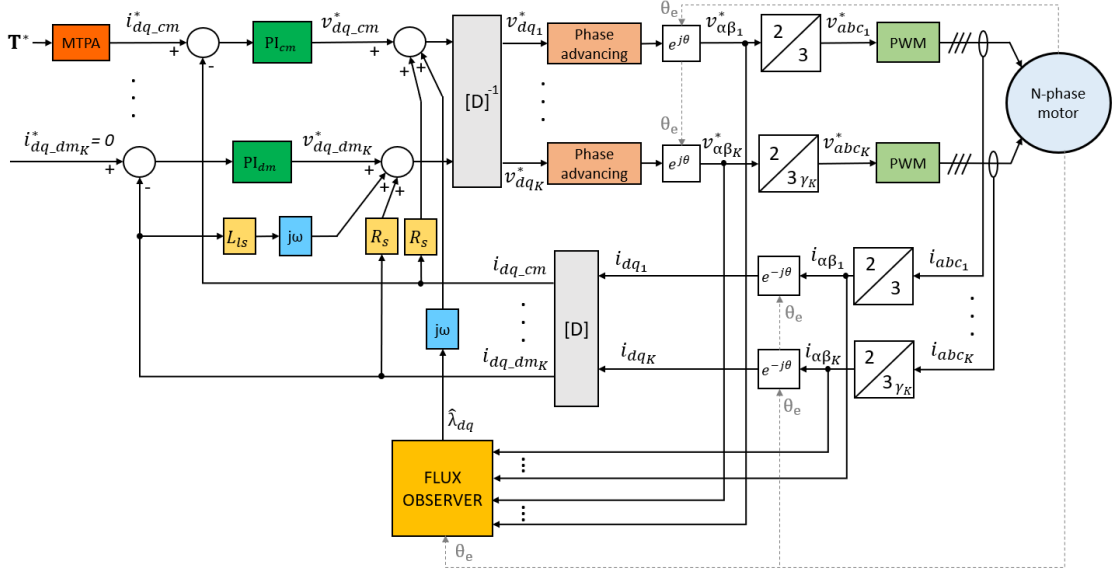


Figure 1.16: General torque-control scheme adopted using a novel decoupling matrix for common and differential mode control.

The torque-control starts imposing obviously a torque reference and thanks to two 1D-LUTs (related to Fig.1.13) choosing properly the common mode currents along d and q axis as reference is possible. In addition the reference of differential mode is set to zero as explained before, trying to reduce the unbalancing dynamics between the phases.

Then the standard PI-controller is applied in which the parameters are set as in 1.31, depending on the switching frequency of the inverters and adjusting the band frequency at least one decade lower.

$$kp_{cm,d} = \omega_b \cdot L_d = 2 \cdot \pi \cdot f_b \cdot L_d$$

$$kp_{cm,q} = \omega_b \cdot L_q = 2 \cdot \pi \cdot f_b \cdot L_q$$

$$ki_{cm,d} = \omega_b \cdot R_s = 2 \cdot \pi \cdot f_b \cdot R_s$$

$$ki_{cm,q} = \omega_b \cdot R_s = 2 \cdot \pi \cdot f_b \cdot R_s$$

$$kp_{dm,d} = \omega_b \cdot L_{ls} = 2 \cdot \pi \cdot f_b \cdot L_{ls}$$

$$kp_{dm,q} = \omega_b \cdot L_{ls} = 2 \cdot \pi \cdot f_b \cdot L_{ls}$$

$$ki_{dm,d} = \omega_b \cdot R_s = 2 \cdot \pi \cdot f_b \cdot R_s$$

$$ki_{dm,q} = \omega_b \cdot R_s = 2 \cdot \pi \cdot f_b \cdot R_s$$

Once that the reference voltage in common and differential planes are computed, they are summed with some feedforward components in order to make faster the dynamics. The feedforward is taken from equation 1.16 and 1.19. Now the inverse-decoupling matrix is considered, in order to allow to the control to take into account the real voltage in (d-q) domain, for each threephase system. Besides, a phase-advancing algorithm is applied to the latter to compensate the delay due to the inverter's dynamics.

with a well known rotational matrix all the voltages in the shared (α, β) plane are transferred and so for each star the own correct Clarke transformation is used, reaching the original plane (a,b,c) quantities to use for each single threephase inverter through the PWM modulation. Then "PWM" and "N-phase motor" will model the inverters and machine's dynamics, as will be explained later.

In this kind of control an encoder is considered, in fact the mechanical angle θ_r is used directly from the motor system.

Finally, the only ones feedback used in the machine's control are the (a,b,c) currents, measured, which are traced back in the common-differential mode system thanks to inverse Clarke transformations, inverse rotational ones and the direct coupling matrix.

The flux observer is needed to predict the flux's variations for the feedforward, so its logic block uses as input the reference voltages and the measured currents, all in the (α, β) domain.

From the scheme point of view it depends from the number of phases considered for the system: they are similar to a classical threephase one but the fact that there is a single general flux for the several threephase systems has to be taken into account. A general block logic scheme was hard to be found, so in Figg.1.17-1.18 the schemes for the used sixphase and twelvephase motor are shown.

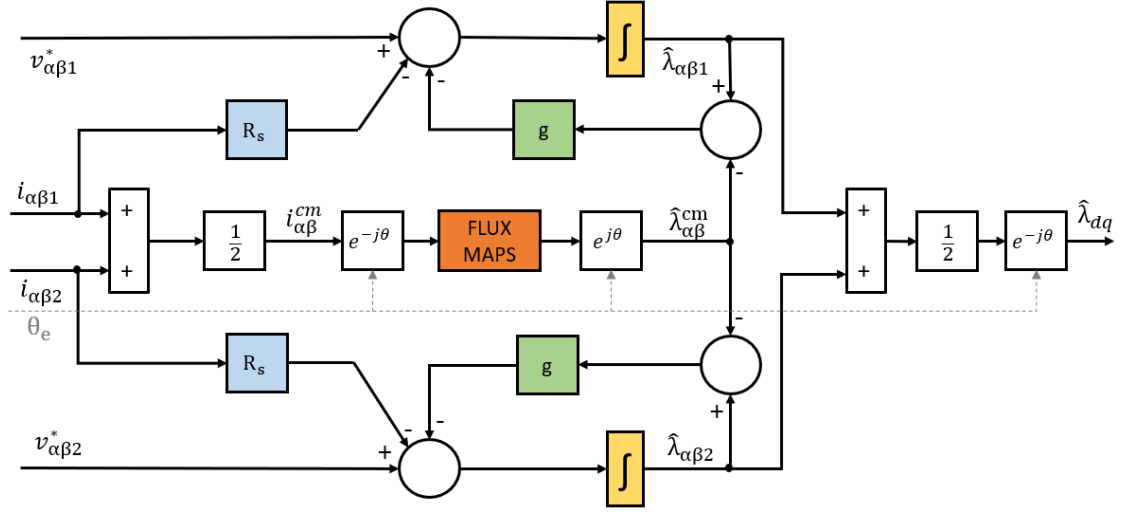


Figure 1.17: Flux-observer scheme for a sixphase machine.

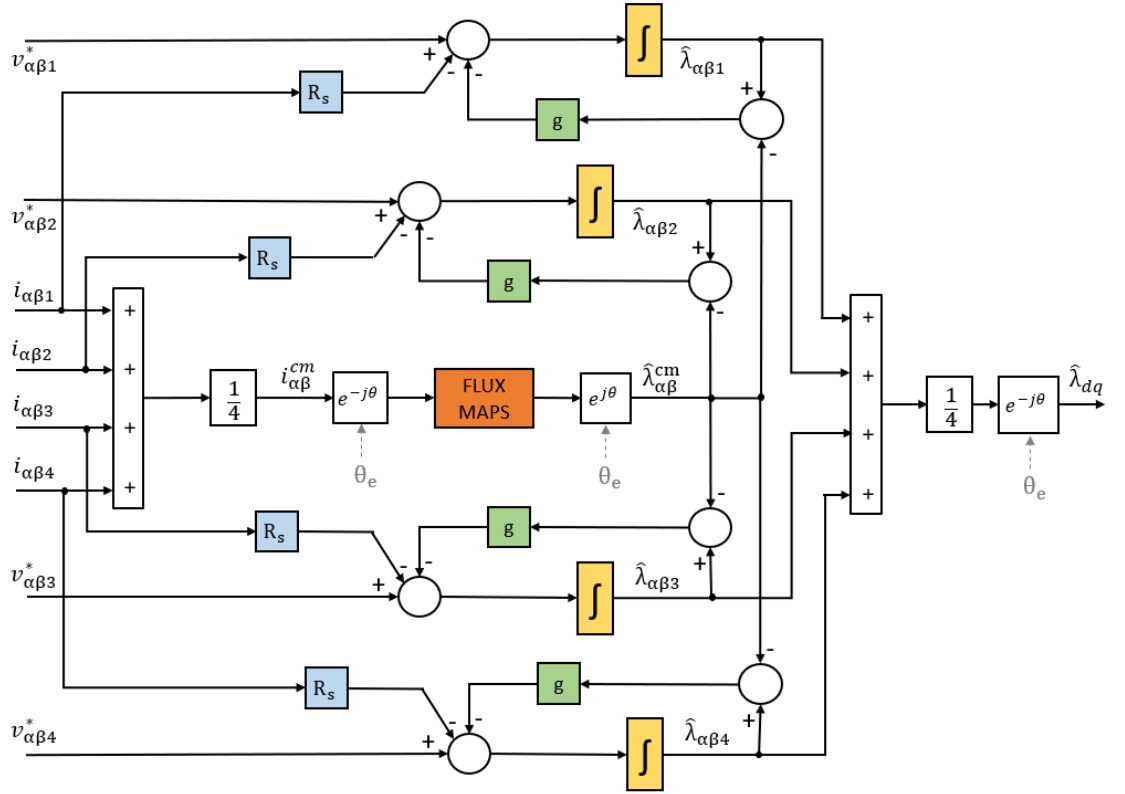


Figure 1.18: Flux-observer scheme for a twelvephase machine.

Neglecting the fact that there are several duplicated standard threephase schemes in the previous observer pictures, a single final value of the flux in (d,q) reference is reached. There is a magnetic model of the system represented by the flux-map related to a phase, which is the direct relationship between the flux and the current in (d,q)-axis. For computational reasons, a flux-map for each threephase system was not used, and to reduce the computational time and the complexity of the control, only a single map has been taken into account considering the common mode, the latter useful for the torque generation. The current to take into account is that computed in $(\alpha\beta)$ domain, considering the average between the different sets. Besides, the electric model is implemented with the electric equation that links the flux with voltage, and the voltage drop across the resistance, for each star, like in 1.31.

$$[V_{\alpha\beta}]_k = [R_s] \cdot [i_{\alpha\beta}]_k + p \cdot [\lambda_{\alpha\beta}]_k \quad (1.31)$$

Then, as can be seen even in Figg.1.17-1.18, the two model are merged using a feedback quantity multiplied for g , the switch-frequency between the two models. At the end, a unique value of the flux is obtained computing the average among the several threephase systems and rotating it in the (d,q)-plane.

After the different schemes which explain the different blocks, only the coupling-decoupling one has remained.

The simplest case is the two-threephase one, in which the common mode is represented by the average of the machine (d,q) quantities, while the differential mode is the easy subtraction of the both. in 1.32 there is an example.

$$\begin{bmatrix} \bar{x}_{cm,dq} \\ \bar{x}_{dm,dq} \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \bar{x}_{dq-1} \\ \bar{x}_{dq-2} \end{bmatrix} \quad (1.32)$$

In order to come back to the original variables an inverse matrix $[D]^{-1}$ is needed, as below:

$$\begin{bmatrix} \bar{x}_{dq-1} \\ \bar{x}_{dq-2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \bar{x}_{cm,dq} \\ \bar{x}_{dm,dq} \end{bmatrix} \quad (1.33)$$

The meaning of the previous equation is to add to the common mode the differential one for a star, and remove it for the other, in order to compensate the unbalancing.

The problem is different for a multi-threephase motor drive, for which a lot of algorithms have been written in literature.

In the present work a simple and intuitive algorithm is proposed for multiphase machines with a number of threephase systems greater than two (in this case they will be twelve).

The main principle is to compute the common mode with the same logic of the sixphase motor, obtaining an arithmetic average through the different sets. The crucial difference from the method shown in [12] is how to compute the differential mode. Instead to link all the phases each together, since the leakage phenomena are not so decisive for the torque production, a new method has be chosen to obtain them, as follows:

$$\begin{bmatrix} \bar{x}_{cm,dq} \\ \bar{x}_{dm-1,dq} \\ \bar{x}_{dm-2,dq} \\ \bar{x}_{dm-3,dq} \end{bmatrix} = \frac{1}{4} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & -2 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 0 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} \bar{x}_{dq-1} \\ \bar{x}_{dq-2} \\ \bar{x}_{dq-3} \\ \bar{x}_{dq-4} \end{bmatrix} \quad (1.34)$$

Remembering that the differential mode embodies the differences between two or more phases, the purpose of 1.34 is linking the second star to the first one, the third to the second, the fourth to the third and so on, in order to reduce their unbalances as soon as one of them is different to the other with which is correlated. with the presented algorithm a sparse matrix can be reached and the computational complexity is low.

Since the inverted relationship is also needed in the control, the inverse matrix of 1.34 is reported in 1.35

$$\begin{bmatrix} \bar{x}_{dq-1} \\ \bar{x}_{dq-2} \\ \bar{x}_{dq-3} \\ \bar{x}_{dq-4} \end{bmatrix} = \begin{bmatrix} 1 & \frac{3}{2} & 1 & \frac{1}{2} \\ 1 & -\frac{1}{2} & 1 & \frac{1}{2} \\ 1 & -\frac{1}{2} & -1 & \frac{1}{2} \\ 1 & -\frac{1}{2} & -1 & -\frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} \bar{x}_{cm,dq} \\ \bar{x}_{dm-1,dq} \\ \bar{x}_{dm-2,dq} \\ \bar{x}_{dm-3,dq} \end{bmatrix} \quad (1.35)$$

Once the that the decoupling matrices are shown, their blocks can be integrated in Fig.1.16 depending of the number of threephase sets taken into account.

1.3 Simulation toolchain

Different kind of software have been used in this work. Presenting them in chronological order in which they have been used, the open source FEMM is briefly introduced.

The latter was used to draw the motor for an electromagnetic finite elements analysis (FEM), in order to understand the magnetic behaviour of the motor and choose the proper windings configuration. So, thanks to the technical drawing of the motor, all the shapes have been reported in FEMM with a MATLAB parametric code which was written in order to be able to represent the motor easily, simply changing some parameters for each desired change, using the "octave-FEMM" interface. Then, a magnetic analysis has been done for the different cases of configuration, three and twelve phases. At the end, the drawing was saved in *dxg* format in order to be used

in Syre, the other software used for the thesis.

In fact, Syre has the possibility to modify and consider several already uploaded kind of rotor, from a so called V-type to a SPM one and other ones. But in the case of the TMK motor used in this work, the shape of the rotor and its magnets is particular and unique, so a personal draw was mandatory. Finally, the model built on FEMM was used even to compare the results with Syre, to be sure that the simulations were correct.

Besides, Syre was used to simulate the dynamic behaviour of the motor using the FEM analysis and a MATLAB's app designed by the Politecnico di Torino. Syre allows to simulate the machine using a 2D FEM simulation interface with FEMM, and also obtaining flux maps, iron losses maps, inductance maps and other results is possible.

The fundamental output of Syre is the Simulink model already seen in Fig.1.9, designed with the proper parameters of the considered motor and the possibility, as introduced before, to simulate the digital control dynamics and the behaviour of the coupled motor-load system and the inverters' PWM phenomena.

In this context studying a whole machine is easier because one can simulate the magnetic behaviour and optimize it changing the parameters, and at the same time even the control strategy of the whole drive can be considered and analysing it for a commercial product.

Finally, starting from a threephase Simulink model, several change were introduced in order to adapt the simulation for a multi-threephase, since Syre can manage a multi threephase motor only magnetically speaking (flux-maps, torque maps, iron losses,...), but not for the control side yet.

Chapter 2

Automated FEM simulation of the multi-phase machine

2.1 Introduction to the motor used and windings configuration adopted

ETEL is a company that produces even electric motors, and the model called "TMK" is one of their products. The used model is the torque-motor "TMK0291-050", a kind of IPM motor.

Since the TMK is a torque-motor, its main characteristic is an high torque value, obtained without any gear-box, to the disadvantages of the maximum speed range available. In contrast with the automotive world, in these kinds of motor, the main problem which brings then a flux-weakening is not the maximum voltage reached but the heating phenomena of the magnets, due to the high values of magnetic induction.

Some representative data are shown in table 2.1 and they have been taken from the public datasheet of the motor considered. If the official datasheet is considered, two version of the same motor can be noticed, depending of the connection configuration selected: the 3VBS and the 3VDS. It depends on the number of series/parallel circuits chosen and model considered in the present work is the first one.

Besides has to be considered that in this thesis some magnets slightly more performing than the real motor have been used: the results about the torque will be a bit greater but for the reason just mentioned they are consistent with the datasheet's one. Finally, some data could be different for the reason that Syre uses a 2D software (FEMM) that needs an additional analytical computations.

Motor Performance	Unit	Value
Peak torque T_{pk}	Nm	440
Continuous torque T_c	Nm	271
Continuous current I_c	A	32.5
Electrical resistance R_s @20°C	Ω	0.66
Electrical inductances L_d/L_q	mH	6.5/5.05
Base speed ω_b	rpm	436
Max speed with flux-weakening ω_{max}	rpm	2410
Rated speed ω_0	rpm	387
Number of poles	-	44
Nominal DC bus voltage V_{DC}	V	600
External diameter	cm	310
Length	cm	50

Table 2.1: Main data of **TMK0291-050-3VBS** motor, in order to represent its main characteristics.

For company secret's reasons it was not possible to show the real and detailed shape of the motor in order to show it better with air-gaps close to the magnets and other parameters used, but it will be not a crucial problem to the the explanation of the whole work.

In Fig.2.1 a general model of an IPM motor is shown, where observing the approach used to study is possible. Only a quarter of the whole machine has been considered for periodicity reasons, and in Fig.2.1 the green arrows indicate some example of the quantities that are introduced in the parametric model, made in FEMM and an interface with MATLAB, and which could be changed using parameters; the zoom represents the same aspects for the magnets. The latter have different materials speaking about the values of the magnetic density and the sizes. The kind of magnets used for each pole serve to concentrate the flux to the air-gap and obtaining greater values of torque, as can be noticed in the figure below.

For confidentiality reasons, the drawing of the motor can not be shown in detail.

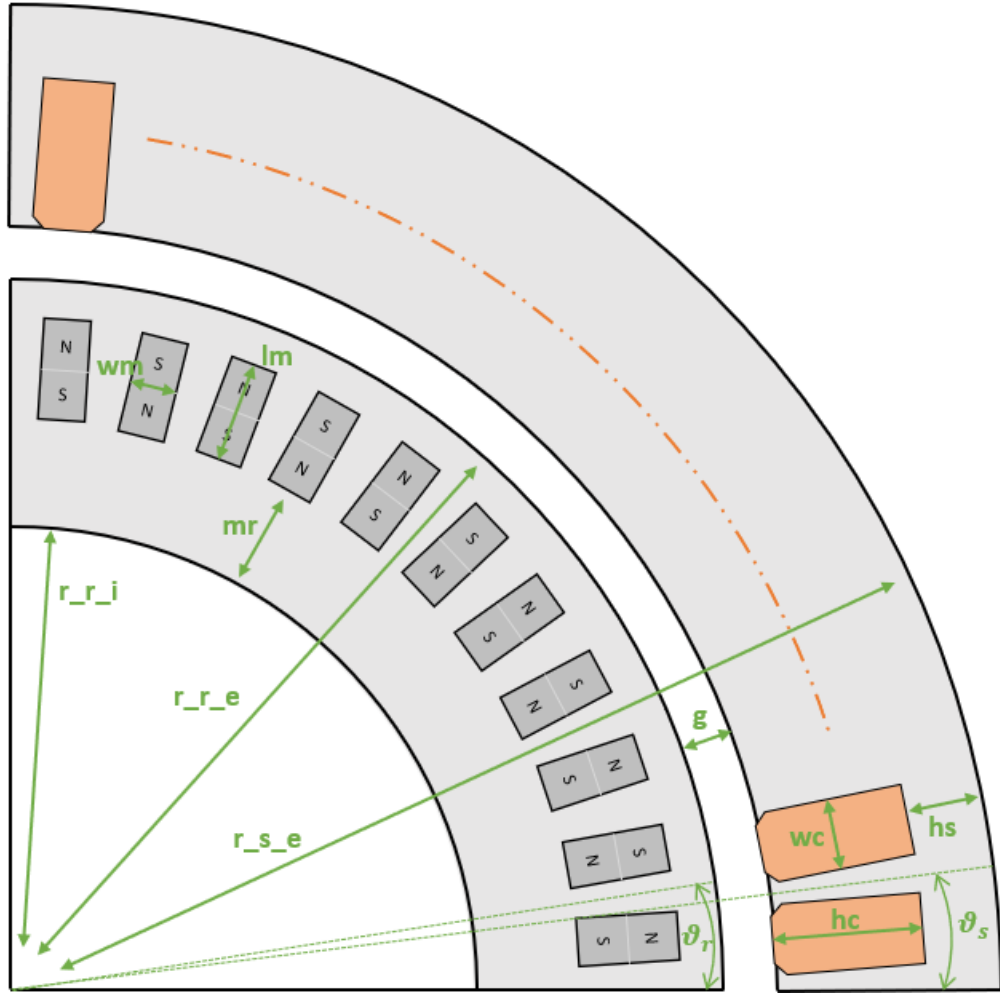


Figure 2.1: Example of the initial parametric approach used to model the machine.

Once that the general model has been introduced, some example of the code used to generate the previous figure is reported in order to make it clearer.

Listing 2.1: Stator coordinates' definition.

```

1 %STATOR
2
3 AAx=r_s_i; AAy=0;
4 BBx=r_s_i*cosd(AngTooth/2); BBy=0;%r_s_i*sind(AngTooth/2);
5 CCx=AAx+hc; CCy=0;%r_s_i*sind(AngTooth/2);
6 DDx=r_s_i*cosd(AngTooth/2+AngCave)+hc*cosd(0.5*AngSecStat); DDy=r_s_i
    *sind(AngTooth/2+AngCave)+hc*sind(0.5*AngSecStat);
7 EEx=r_s_i*cosd(AngTooth/2+AngCave); EEy=r_s_i*sind(AngTooth/2+
    AngCave);
8 FFx=r_s_i*cosd(AngTooth/2+AngCave+AngTooth); FFy=r_s_i*sind(AngTooth
    /2+AngCave+AngTooth);
9 GGx=r_s_i*cosd(AngTooth/2+AngCave+AngTooth)+hc*cosd(0.5*AngSecStat);
    GGy=r_s_i*sind(AngTooth/2+AngCave+AngTooth)+hc*sind(0.5*AngSecStat
    );
10 HHx=r_s_i*cosd(AngSecStat)+hc*cosd(AngSecStat); HHy=r_s_i*sind(
    AngSecStat)+hc*sind(AngSecStat);
11 LLx=(r_s_i)*cosd(AngSecStat); LLy=(r_s_i)*sind(AngSecStat);
12 IIX=LLx; ILY=LLy;
13 MMx=(r_s_e)*cosd(AngSecStat); MMy=(r_s_e)*sind(AngSecStat);
14 NNx=r_s_e; NNy=0;
15 PP1x=EEx+cs*cosd(0.5*AngSecStat); PP1y=EEy+cs*sind(0.5*AngSecStat);
16 OO1x=AAx+cs; OO1y=BBy;
17 QQ1x=0.5*(OO1x+PP1x); QQ1y=0.5*(OO1y+PP1y);
18 RR1x=0.5001*(CCx+DDx); RR1y=0.5001*(CCy+DDy);
19 OO2x=FFx+cs*cosd(0.5*AngSecStat); OO2y=FFy+cs*sind(0.5*AngSecStat);
20 PP2x=IIX+cs*cosd(AngSecStat); PP2y=ILY+cs*sind(AngSecStat);
21 QQ2x=0.5*(OO2x+PP2x); QQ2y=0.5*(OO2y+PP2y);
22 RR2x=0.5001*(GGx+HHx); RR2y=0.5001*(GGy+HHy);

```

As can be seen, all the points of the model have been written by code in MATLAB using the OctaveFEMM ambient and then, from the boundary conditions and material's definition, up to the windings configuration, the drawing has been completed used the same method.

As described in 1.1.1, not all the windings configuration are possible for a given machine, but fortunately the used motor allows to configure the both sixphase and twelvephase motor. However, the windings can not be connected with the same order as they are spatially distributed since the electrical angle has to be taken into account instead the mechanical one. So, after a reorganization of the windings, a correct configuration of them has reached and some general example are given below with the voltage vectors diagrams with the aim to understand the logic followed.

In Figg. 2.2-2.3 the angle γ represents the electrical angle between two adjacent coils (as can be seen with the numbers next to the vectors that indicate the mechanical order of the coils in the counter-clockwise), while θ is the angle needed in order to obtain an asymmetrical multi-threephase machine. The latter, for this particular

motor, could be obtained so it was possible to obtain a sixphase or a twelvephase machine as multi-threephase one.

Then, each color represents a different threephase system and for this reason only two different ones (blue and red) can be noticed in the first figure, while four different colors are used in the scheme related to the twelvephase system, which has four threephase sets.

The graphs below were obtained with all the coils equally wound, so the dotted vectors mean that, starting from a conventional positive verse of winding, an opposite one has to be considered in order to obtain the proper polarity of the electromotive force, while the voltages with continuous lines are already in the correct direction.

Besides, it is clear in Figg.2.2-2.3 that, in order to obtain the proper verse for the second coil (electrically speaking), inverting the direction of the latter is needed rewinding it in the opposite way. Taking into account Fig.2.2, since the minimum angle is $\theta = 15^\circ$ and the needed one to have a sixphase machine is $\delta = 30^\circ$, two voltage vectors for the same phase have to be connected in the example given below (the first two vectors of each threephase system are colored with the same color). As regards the preliminary FEMM model, only two version of the motor have been taken into account, in order to have an initial idea about the expected results at the end of the work, and to be able to compare them with the Syre's ones and make a check of them. So only the threephase machine and the twelve one were considered, as follows.

Firstly, the threephase windings configuration is considered and the first study is about the no-load fluxes in order to understand the initial electric phase of the motor and to check the correctness of the same configuration. As can be seen on the top of Fig.2.4 the three fluxes were obtained and in order to verify the value of the torque, it was decided to inject a current along the q -axis in order to get about the rated torque, using the convention of Fig.1.7: the considered machine has not an huge anisotropy, as can be noticed in table 2.1 and at the beginning the motor can be controlled without following the MTPA-trajectory, not available yet at this point of the study, as in graphs (b) and (c).

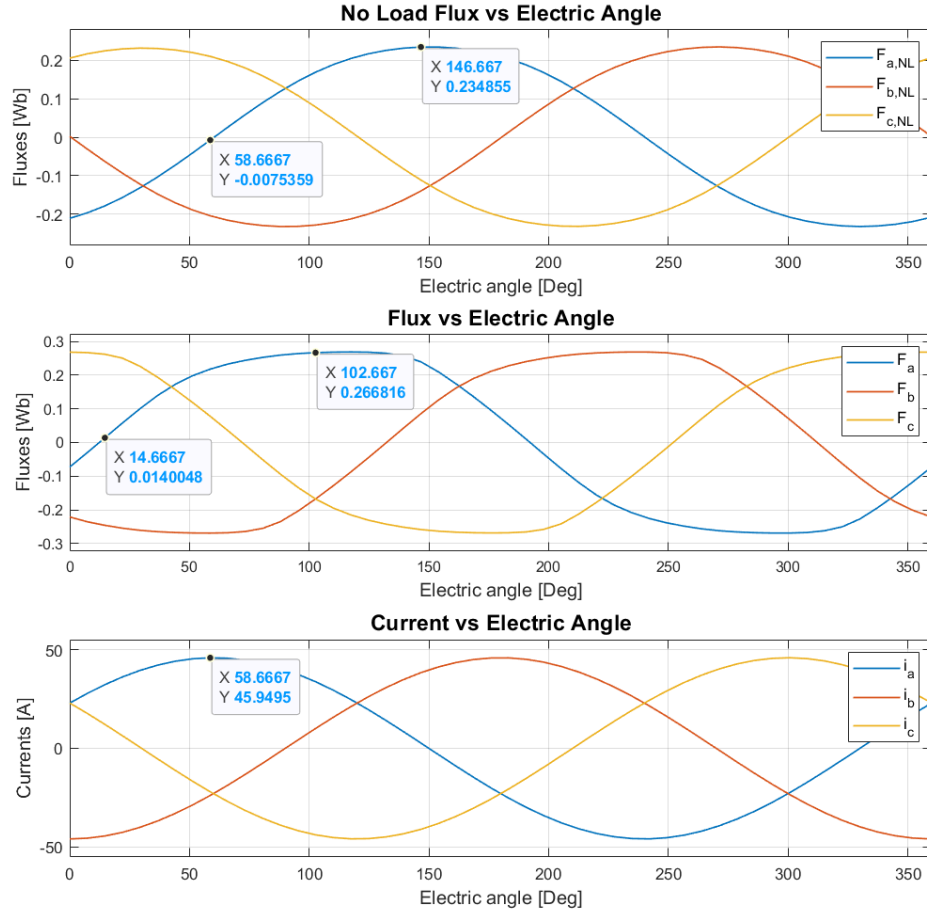


Figure 2.4: Threephase machine preliminary simulation. From the top to the bottom: (a) no-load fluxes, (b) fluxes with load, (c) currents injected. The load-angle between (a) and (b) can be noticed.

In the first study done, at no-load, in fact the torque has a value of zero as regards the average value, like in Fig.2.5, while when the rated current is injected the torque value reaches an average value of about 302.5 Nm with an evident peak to peak torque ripple. An other fact that has to be seen, is the sixth harmonic dynamic that characterizes the behaviour of a threephase machine: Fig.2.6 shows what just said.

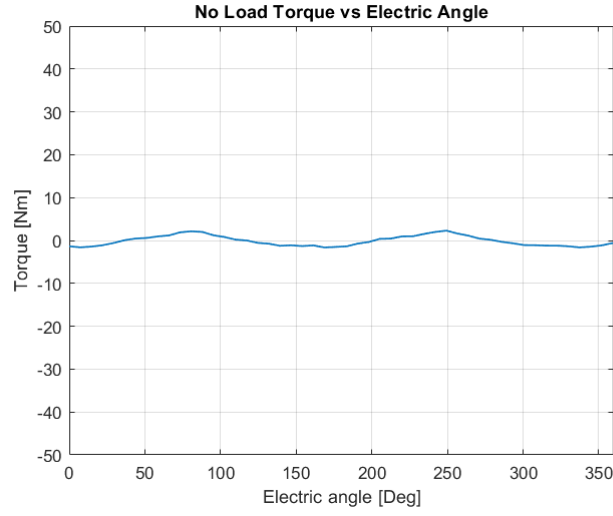


Figure 2.5: Threephase machine preliminary simulation. Torque obtained at no-load working condition, with zero average value.

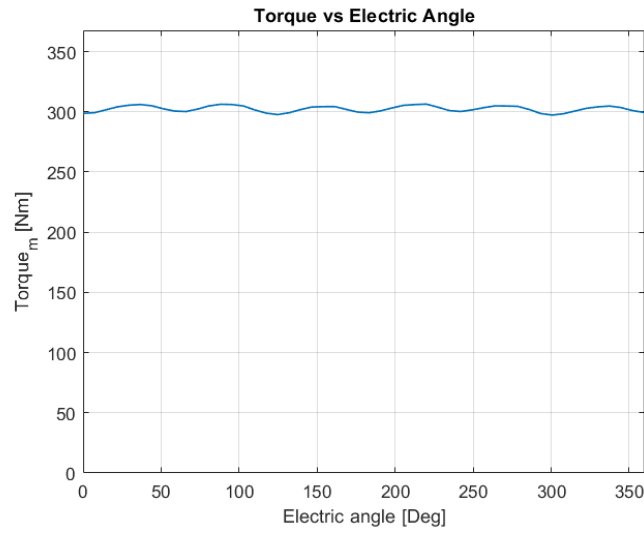


Figure 2.6: Threephase machine preliminary simulation. Result of the torque got with the rated value of TMK motor. $\bar{T} = 302.5 Nm$.

The torque in Fig.2.6, taking into account the rated value of table 2.1 and the fact the more performing magnets are considered, seems being in accordance with the real motor and the phase between fluxes at no-load and with the latter represents the same value of the real motor, with about forty four electrical degrees in the same working condition.

The next step was to make the same analysis for the twelve version of the same motor, in order to understand what kind of results should be obtained at the end of the study. Even in this case, a first analysis was about the no-load condition, to check if the coils are connected correctly or not, following the suggestions of Fig.2.3. There is what has been just said into the first diagram in Fig.2.7. Then, as already done for the threephase motor, a current in quadrature with the no-load flux has been used, with the same value of *ampere-turns*, in order to make a consistent comparison with the previous study. Only in post-processing analysis, the current was scaled to represent the correct value because, as it will be seen after, the bus DC-voltage is the same of the standard machine, so four time the number of turns are needed in order to use a quarter of the rated current and being able to use smaller power electronics in terms of power. Finally, the currents properly controlled and shifted for each phase are shown in the rest of the graphs below, with the related fluxes in a working condition with load.

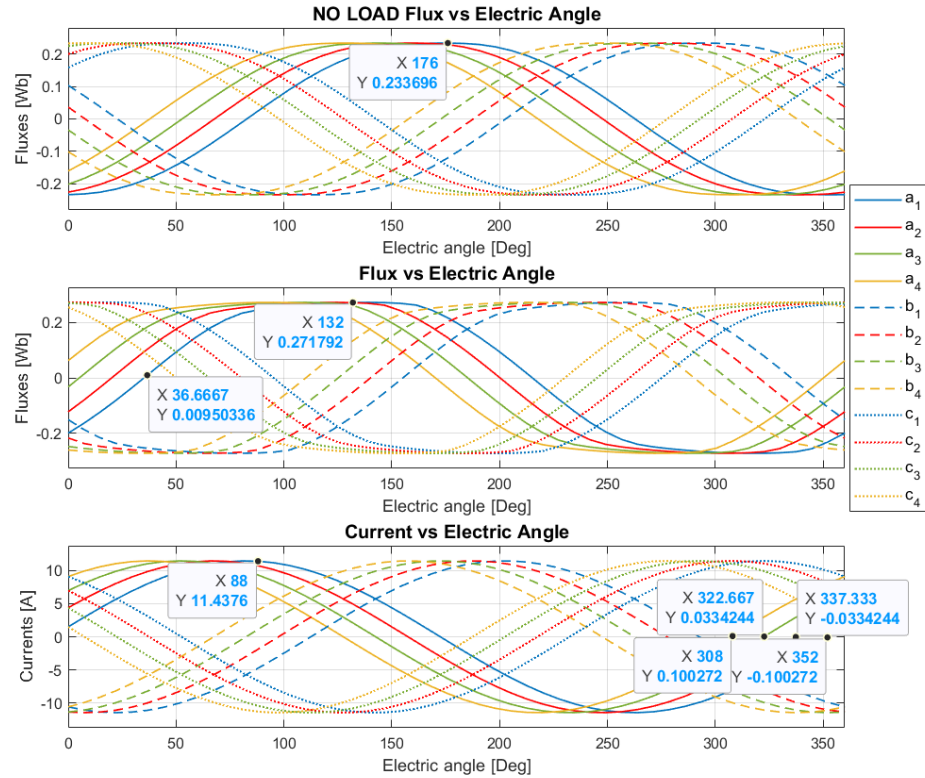


Figure 2.7: Twelvephase machine preliminary simulation. From the top to the bottom: (a) no-load fluxes, (b) fluxes with load, (c) currents injected. The load-angle between (a) and (b) can be noticed.

In the previous picture there is the same load-angle seen for the threephase case between the fluxes in different working conditions and even the fifteen electric degrees between the different phases can be noticed in the second plot. Finally, the peak current of the current is reduced four times if compared with the first model. At the end, the torques are reported in order to see how the behaviour changes.

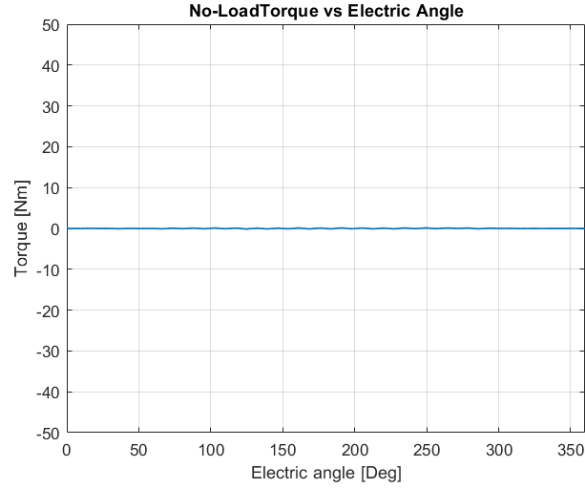


Figure 2.8: Twelvphase machine preliminary simulation. Torque obtained at no-load working condition.

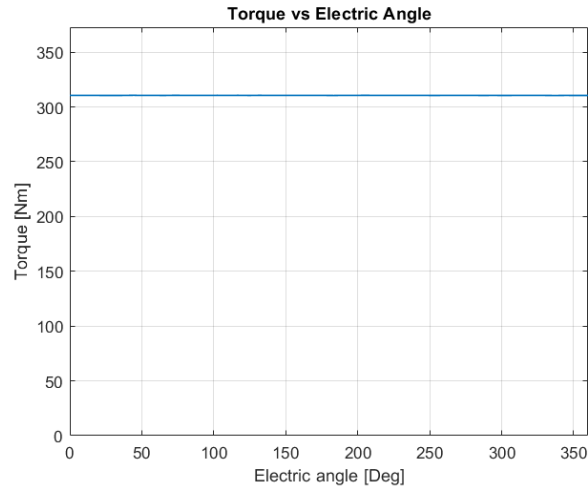


Figure 2.9: Twelvphase machine preliminary simulation. Result of the torque got with the rated value of new version of TMK motor. $\bar{T} = 313.6 Nm$ and $T_{pk-pk\%} = -98\%$ respect to the threephase machine.

It is clear thanks to the Figg.2.8-2.9 that as already mentioned in the chapter 1, the torque is increased of about 3.7% while the ripple is drastically reduced of about 98% if compared with the original machine, avoiding the sixth harmonic distortion. Besides, the power electronics should use a quart of the current respect to the threephase machine, allowing to use more smaller components, easier to be managed in terms of space and volume and more helpful thermally speaking. The same study was considered with Syre and some other results were reached, considering even the sixphase motor.

2.2 Syre: standard and multi-threephase motor study

2.2.1 Introduction to the software

Syre was born in a MATLAB ambient and it allows to study a machine with an interface with FEMM. In fact it uses FEMM for each simulation point and in this way currents, fluxes and torque can be obtained. However, since FEMM is a static FEM software and it does not have any time-harmonic analysis, some result like iron losses and permanent magnets' ones are obtained thanks to an analytical approach using equations.

Since showing the real shape of the used motor is not possible, an example is presented to understand how the software works. To do it, an already existing model is used from the folder "motorExamples", so called "THOR". As in Fig 2.10, with the GUI_Syre "Main Data" section is possible to select the main characteristics of the motor. They are needed to choose the desire kind of machine and the main sizes of the motor, like the inner and outer diameters and so on, or the number of pole pairs and the number of slots per pole per phase.

The second section used with GUI_Syre is the "Geometry" one, with which is possible to choose the detailed geometry of the stator slot and the rotor barriers, if we are speaking about a SR or PM motor. As is shown in Fig.2.11, slot shape can be selected, between trapezoidal and rectangular one, or it is possible to change the sizes of it or the amplitude of the opened side. Besides, as can be seen on the right of the figure below, the number of the barriers is an option to choose, like for their angular direction, sizes and radial position along the rotor. Finally one can select even the angle with which terminate the barriers.

Then, with "Options" is possible to insert the material's exploitation, inserting the value of current density and the admitted copper losses and so one, choosing only some of them since they are linked to each other and it is not possible to obtain everything. Even the optimisation of the mesh has to be chosen, as for the limits of the speed range. At the end, optimising the geometry of the ribs is possible

changing the details about their angle and inclination. An example is given in Fig.2.12.

After have optimised the whole geometric side, the windings have to be configured in order to use the correct connections. Some settings like the number of turns in series per phase or the number of conductors needed in the slot there are, and at the same time one can choose how many threephase sets have to be considered. An other aspect to be considered is the kind of the slot layer position: stacked (like in Fig.2.13) or side-by-side one. At the end, the software considers a distributed windings scheme by default, but even a concentrated one can be design manually and saving it as new configuration.

Once that the main geometric parameters and the windings type are chosen, it is the moment for the material to be used inside each region, in fact there is a proper section called "Materials" with which is possible to use a different for each region: the iron for the stator and the rotor, the copper and its insulation, and even if one would use magnets or not, selecting the material for the barriers and designing the geometry of the same magnets. Besides, thank to FEMM computation and analytical adjustments, the mass and inertia of the motor can estimated and customized materials can be introduced in the software, like the iron's characteristic or the behaviour of some new magnets with the changing of the temperature. The related picture in Fig.2.14.

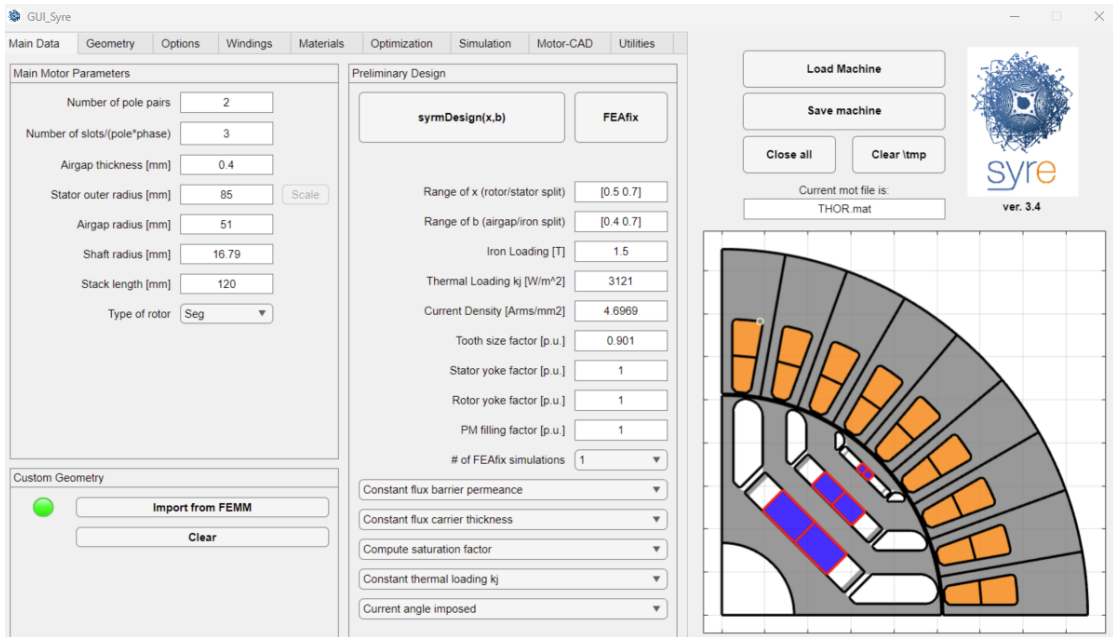


Figure 2.10: GUI-Syre.app interface. The main page is shown with the uploaded model.

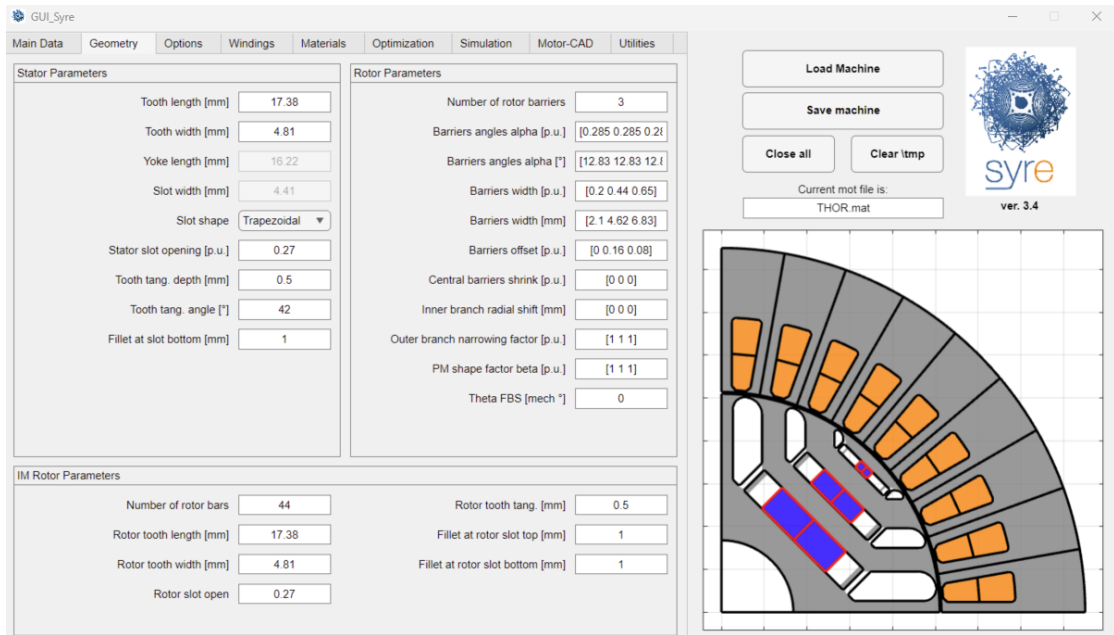


Figure 2.11: GUI_Syre.app interface. The geometric section is shown with the uploaded model.

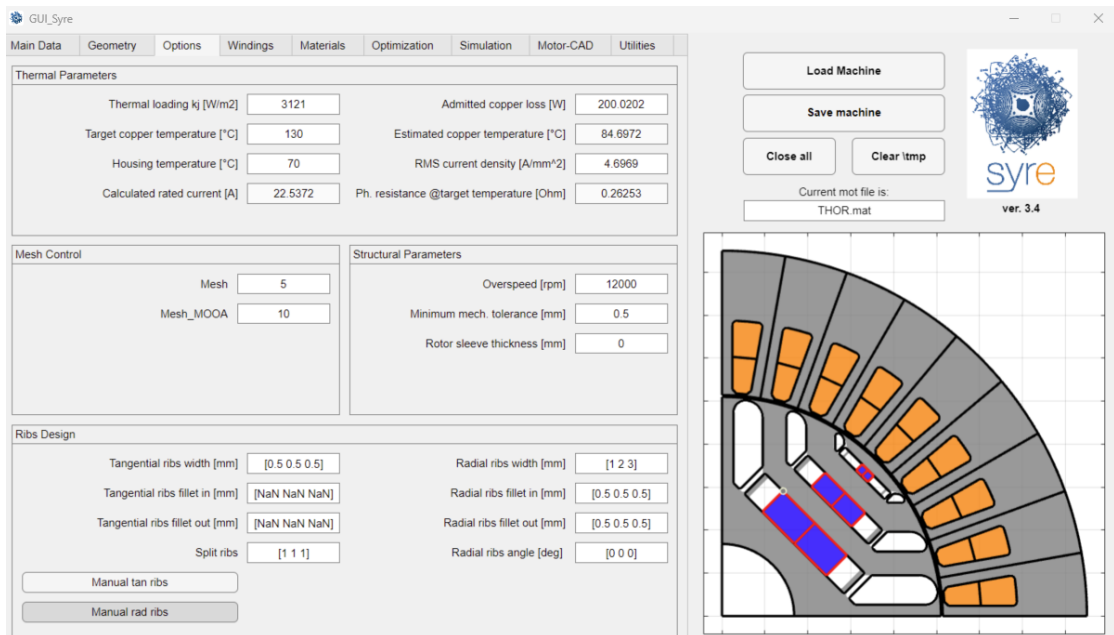


Figure 2.12: GUI_Syre.app interface. The options section is shown with the uploaded model.

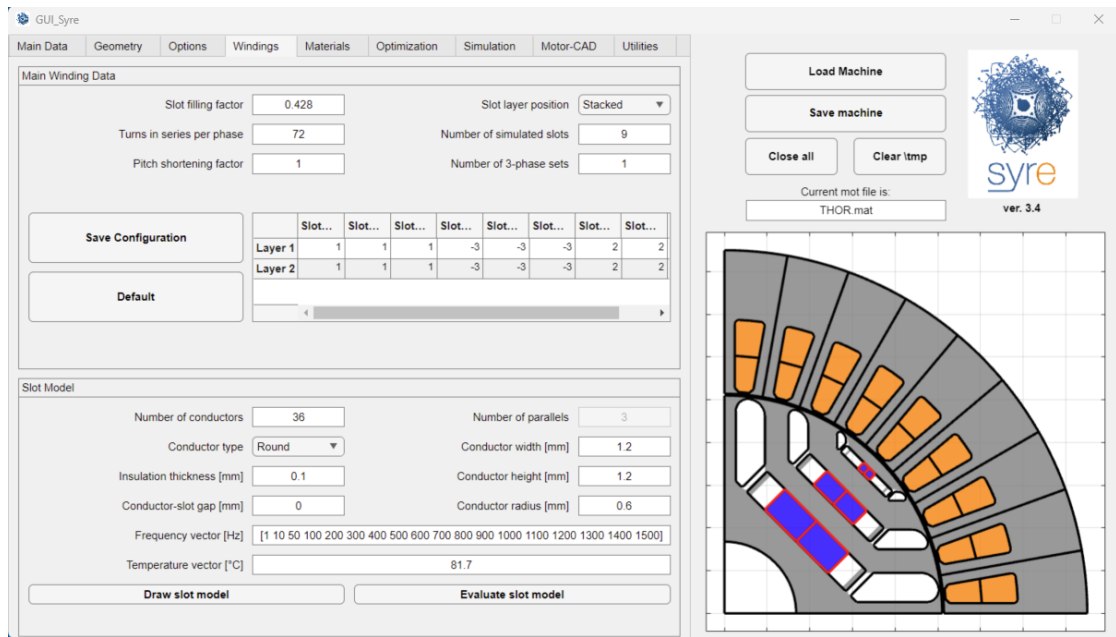


Figure 2.13: GUI_Syre.app interface. The windings configuration selections are shown with the uploaded model.

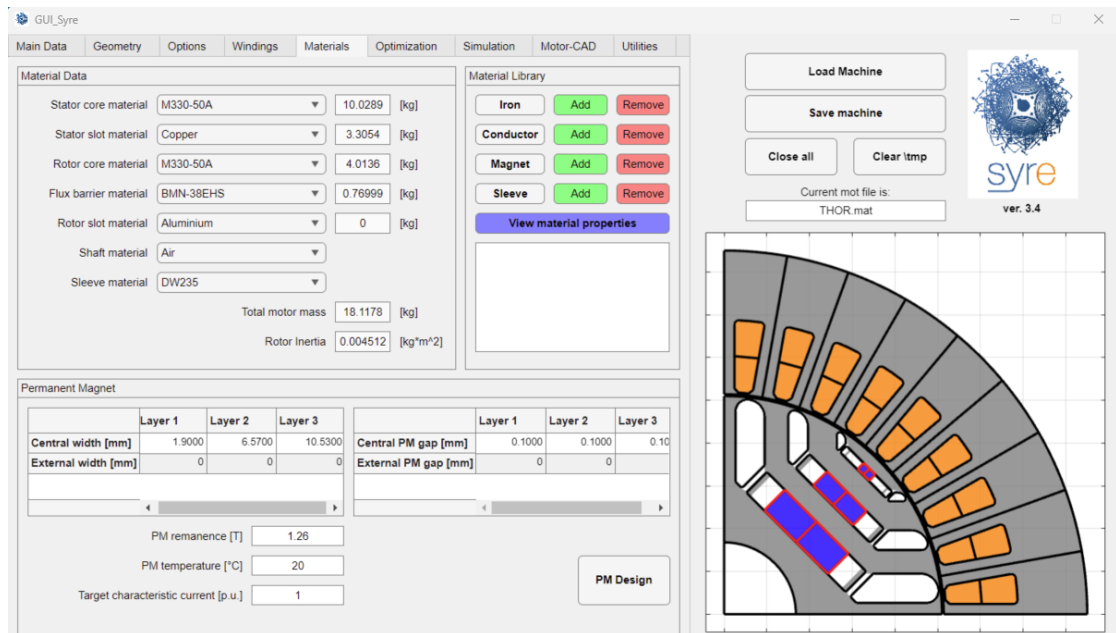


Figure 2.14: GUI_Syre.app interface. The materials' settings available for a model.

Neglecting the sections like "Optimization", "Motor-CAD" and "Utilities" which were not used in the present thesis, the last one to be explained is "Simulation", in which the user can select the working point and the number of points simulated or the electric angular excursion spanned. The latter can be equal to 60° electric degrees if a single working point or a flux map has to be computed, but if one needs an iron losses map, he needs to change the value to 180° in order to take into account the whole harmonic dynamic. The same is for the number of positions simulated: it can be chosen any number but if the iron losses are the aim of the study, they have to be 90 or 180, coherent with the electric period considered. As is shown in Fig.2.15, the convention about the direction of magnetic flux has to be chosen: "SR" if it has to coincide with the q -axis, "PM" if it will be on the d -axis, as in the example used. Then, even the temperature of the magnet has been taken into account in order to change automatically their behaviour, inserted before in the previous section. Other aspect that can be considered are the value/range of the current used for the simulation, the speed and the number of active sets (For more sets than one, for example four, they have to be activated using a vector like $[1\ 1\ 1\ 1]$ or $[1\ 0\ 1\ 1]$ if one wants to turn off the second star).

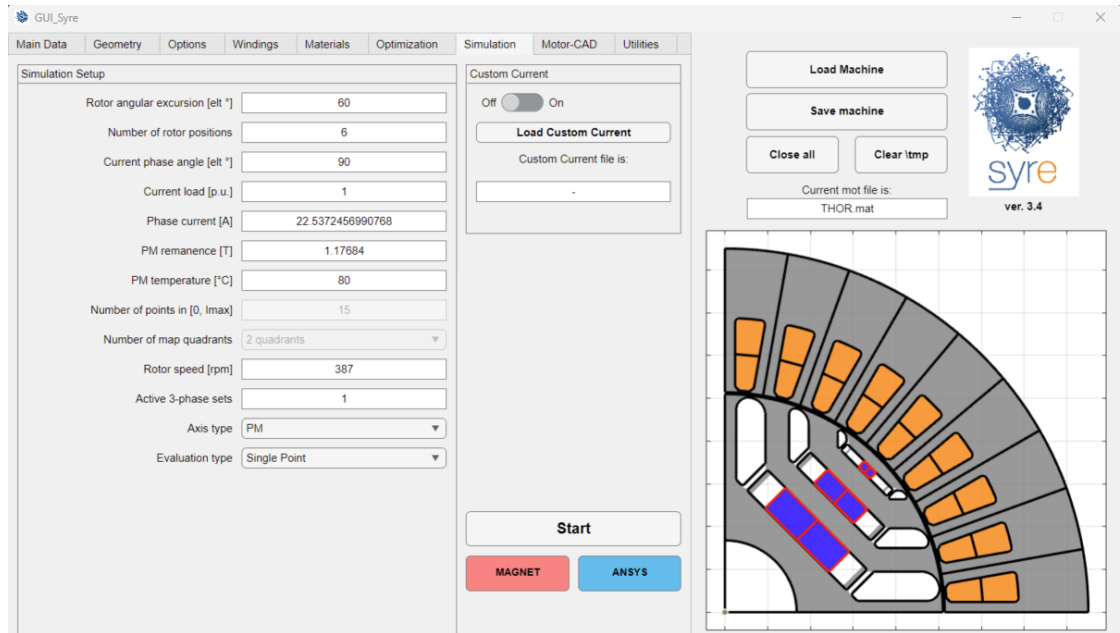


Figure 2.15: GUI_Syre.app interface. The parameters selection used for the simulation.

As introduced before, the TMK motor shape was not already designed in Syre, so firstly the closer structure was drawn in order to assign properly the materials in the right points of the space, and then the rotor structure was imported in the

FEMM linked file in order to study it. The stator shape used was the Syre's one so some different results between Syre and the preliminary study could be due to the some details about the slot geometry and the different mesh used. However the values obtained are consistent and they can be considered good.

2.2.2 Different models comparison

The first results shown are regard the threephase system, the original system considered that can be compared with the first values had in the preliminary analysis. Considering a working point related to the rated current in quadrature with no-load flux, in Fig.2.16 can be observed almost the same values showed in Fig.2.4, with same peak current and peak flux values and with the same phase-shift between the both.

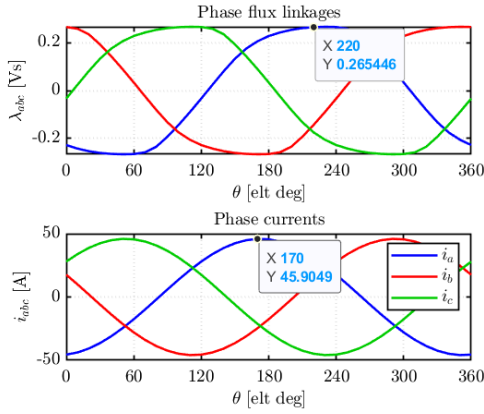


Figure 2.16: Results obtained with load about the threephase machine.

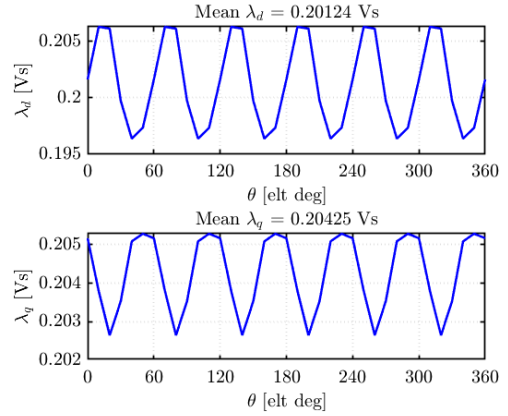


Figure 2.17: Fluxes transformed in (d,q) domain thanks the Clarke and rotational transformation in the three-phase system.

In fact, even the torque can be considered similar to the previous results got, since it is different only for about 1% from the initial study, as can be noticed in Fig.2.18. The same is for the torque ripple, not so far from the FEMM simulations seen before. The main cause of the differences are surely the different mesh used and even the rotor positions spanned: while in the preliminary study each step was one mechanical angle until the end of the related electrical period, in Syre only the first sixty electrical angle are simulated and the rest of the quantities are interpolated as post-processing. An other factor could be the slightly different

stator geometry, since in Syre the slot has a less angular geometry respect to the real one.

Finally, the IPF's value coincides with the shift angle noticed between the fluxes in Fig.2.4, in load and no-load conditions.

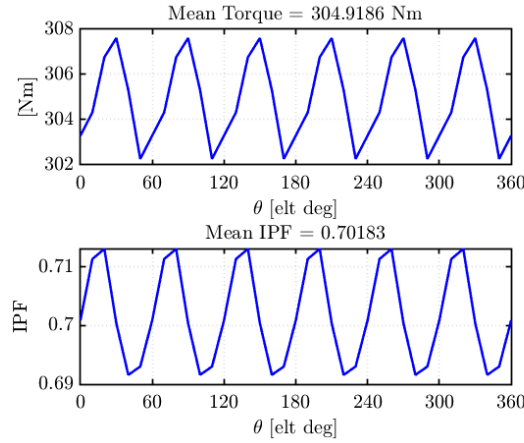


Figure 2.18: Torque value obtained for threephase electric motor with the Syre simulation in the single working point.

The second step to verify the correctness of the Syre model is to check if it works in the same way with the multi-threephase system, since it was not the main purpose of the software. But firstly, since not only the twelve phase will be considered in the work, an intermediate step has been considered: the sixphase version.

For the sixphase motor the number of turns in series per phase has not been modified while the number of conductor into the slot model has been doubled, in order to obtain the same V_{DC} on the DC-link.

Then, the configuration of the windings has been modified following the theory described before. Values in the middle, between threephase and twelvephase, are expected and at the beginning it seems to be in accordance with this statement.

In fact, as can be seen in the initial results in Figg.2.19-2.20-2.21, the fluxes have quite the same peak value, but the current is reduced of an half and th torque is increased.

Besides, the torque ripple range is reduced: the sixphase has only the 14% of peak to peak ripple respect to the original model. In addition, the average torque value has been improved of about 2.7%. The inductive behaviour seems to be similar, since the IPF value can be considered practically the same.

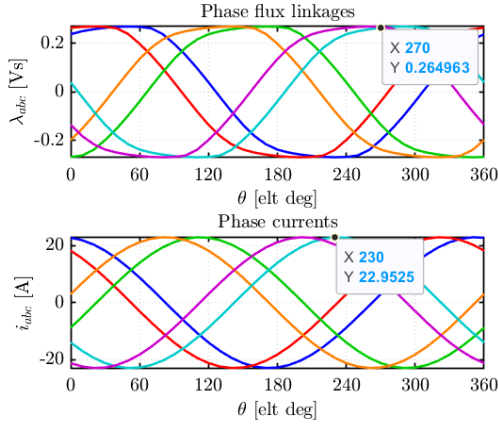


Figure 2.19: Results obtained with load about the sixphase machine.

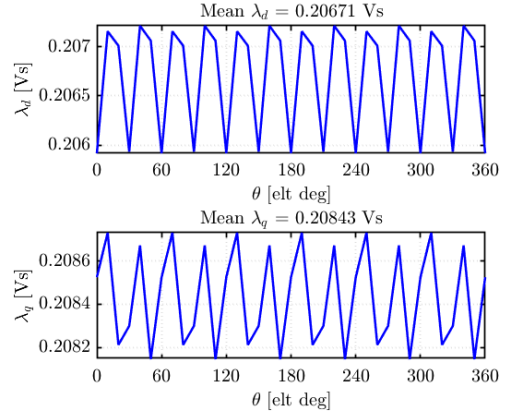


Figure 2.20: Fluxes transformed in (d,q) domain in the sixphase system.

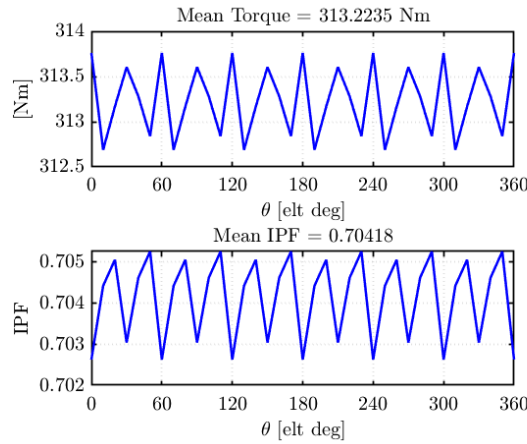


Figure 2.21: Torque value obtained for sixphase electric motor with the Syre simulation in the single working point.

Of course, the total *ampere-turns* into the motor are the same because even if the current is halved, there are two threephase system to be supplied and the doubled turns (with halved diameter) in order to reach the same voltage on the bus with the same stator slot: thermally speaking the stator has the same behaviour, what changes is the possibility to have a modularity with smaller components as regards the power electronics and they can be organised better in the space and in the volume of the system.

Speaking about the twelvephase, the same approach has to be used and in fact the

same trend can be observed. Taking into account the Figg.2.22-2.23-2.24, again, the fluxes' maximum value and the displacement between the several phases is coherent with what has been mentioned in the previous chapters and obtained in Fig.2.7. Besides, the average torque value is increase respect to the threephase machine and the sixphase one, and even in this case, the ripple peak peak about the torque is the smallest one.

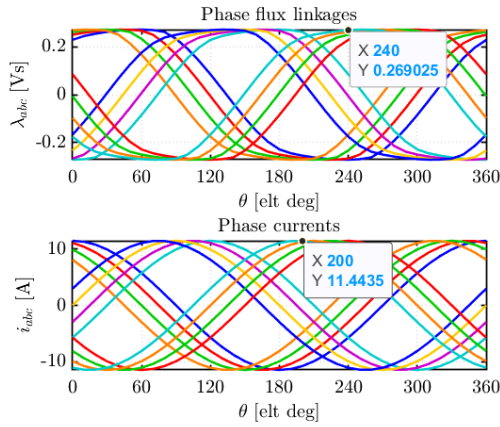


Figure 2.22: Results obtained with load about the twelvephase machine.

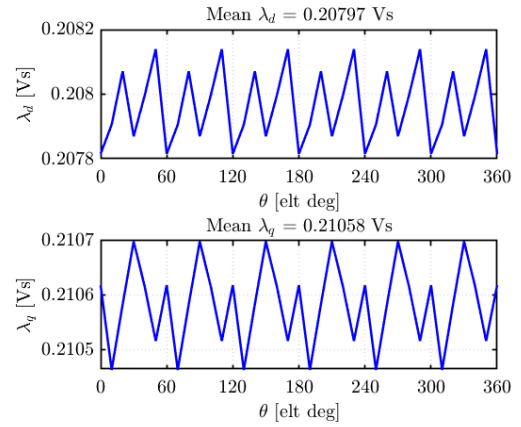


Figure 2.23: Fluxes transformed in (d,q) domain in the twelvephase system.

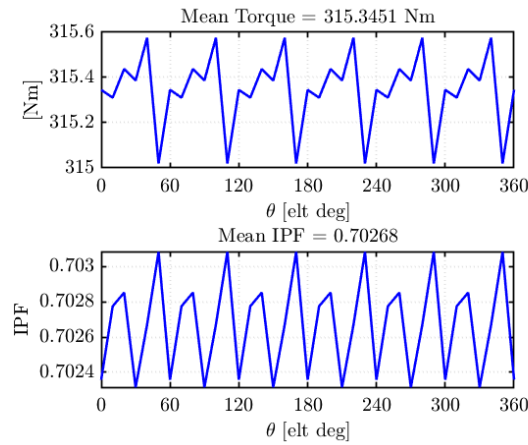


Figure 2.24: Torque value obtained for twelvephase electric motor with the Syre simulation in the single working point.

In terms of percentage, comparing the twelvephase with the standard threephase one, there is an increment of about 4.2% regard the average torque value, while the peak to peak torque ripple is only the 8.6% of the first case: so the latter has been reduced of about the 91.4%. Then, the computation of the torque and fluxes with Syre, if compared with the results seen in the preliminary study, are almost the same and the differences are due to the same reasons described before.

Some other interesting results could be the torque-maps and the iron losses-maps which represent, at a certain speed, the behaviour of the machine for each couple of points (i_d, i_q) , as explained before in section 1.2.

A first relevant comparison between the several models is the torque, the main objective of TMK motor. Firstly, as shown in Figg.2.25-2.26-2.27, the average torque values can be compared for the same relative rated working point. The current injected into the model is shifted of 90° electrical degrees respect to the d -axis, in which there is the whole magnetic flux of the motor. It can be considered as the rated working point as regards the torque reached since the TMK motor, even if it an IPM machine, has not an huge reluctance and the inductances in the sixphase and twelvephase models decrease linearly with the increasing of the number of the threephase system: the ratio between L_d and L_q will be the same. So, with maps/LUTs related to Figg.2.25-2.26-2.27 is possible to know each value of the torque for each couple (i_d, i_q) in a plane that considers a current range up to three times the rated one.

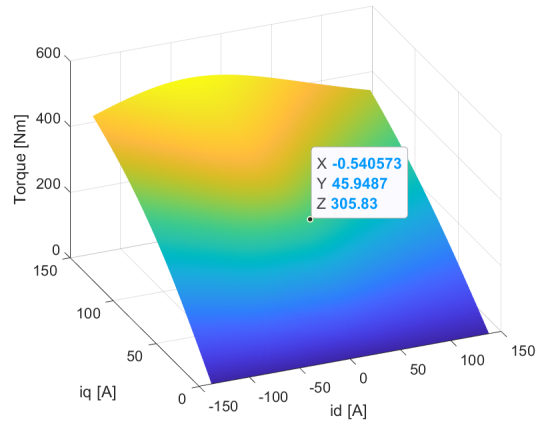


Figure 2.25: Torque maps for three-phase motor.

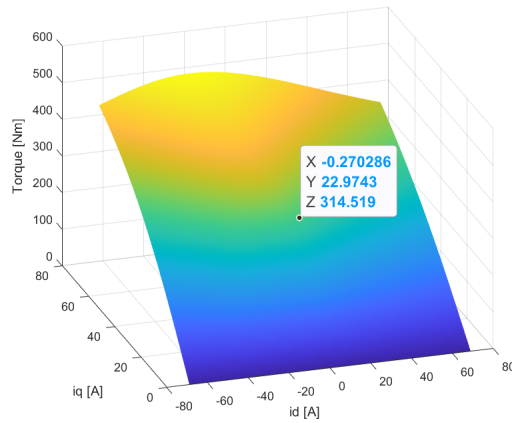


Figure 2.26: Torque maps for sixphase motor.

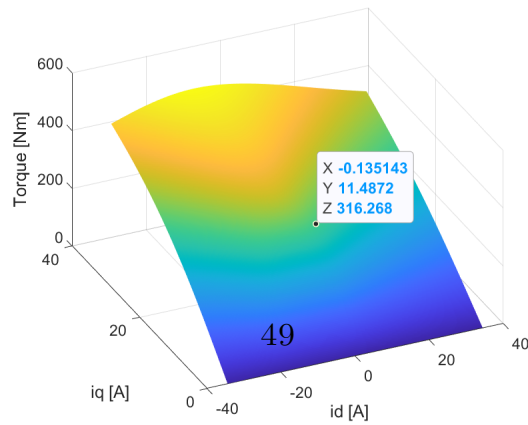


Figure 2.27: Torque maps for twelvephase motor.

The second main results are the iron losses maps, computed for a fixed speed value. In order to extend the results of iron losses even to a greater speed range, the equation from [19] taken into account are:

$$P_{Fe} = P_{h,0} \cdot \left(\frac{f}{f_0}\right)^\alpha + P_{c,0} \cdot \left(\frac{f}{f_0}\right)^2 \quad (2.1)$$

where $P_{h,0}$ and $P_{c,0}$ are respectively the hysteresis and eddy-currents losses at the rated frequency f_0 . The same thing can be written for the magnets loss:

$$P_{PM} = P_{PM,0} \cdot \left(\frac{f}{f_0}\right)^2 \quad (2.2)$$

The factor P_{Fe} and P_{pm} are obtained with Steinmetz equations analytically using the results of the FEM simulations.

An example of maps is given in Figg.2.28-2.29-2.30-2.31-2.32, in which iron losses maps are computed for several speed values: half rated speed, rated speed and doubled one. As can be noticed in the pictures, the losses increase almost quadratically with the speed and they are function of the currents.

However, once that an example is given, it is enough to compare the different machine configurations with a fixed speed value, the same for of the three different models.

Considering the losses about the iron only, their 3D map can be computed representing, at the fixed rated speed, the iron losses as function of current in the (d,q) plane, as in Fig.2.33.

For confidentiality reasons, the values of the graphs are shown between zero and the maximum available.

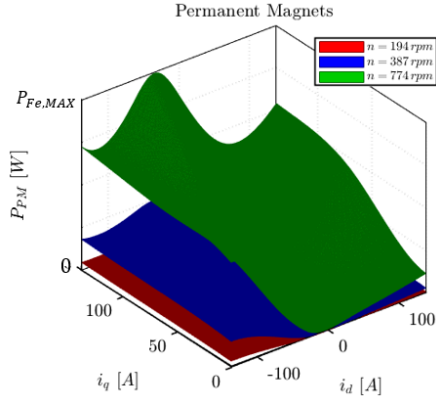


Figure 2.28: Threephase machine's permanent magnets losses-map as function of the speed.

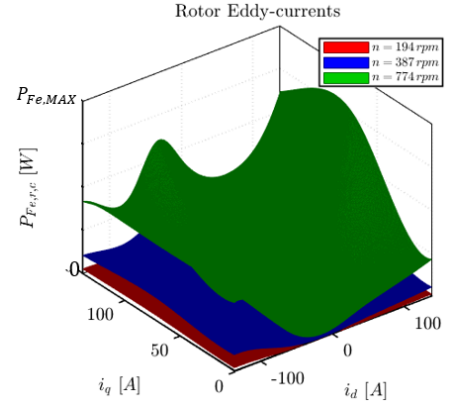


Figure 2.29: Threephase machine rotor's eddy currents losses-map as function of the speed.

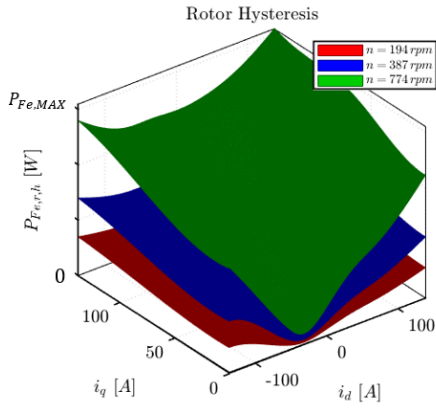


Figure 2.30: Threephase machine rotor's hysteresis losses-map as function of the speed.

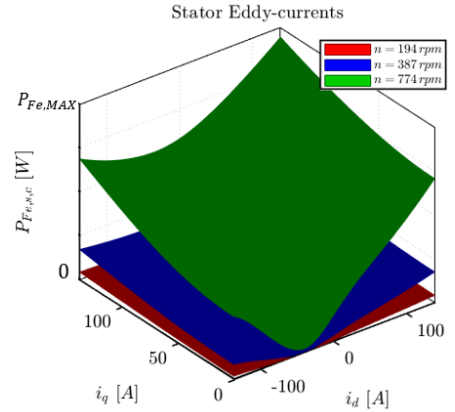


Figure 2.31: Threephase machine stator's eddy currents losses-map as function of the speed.

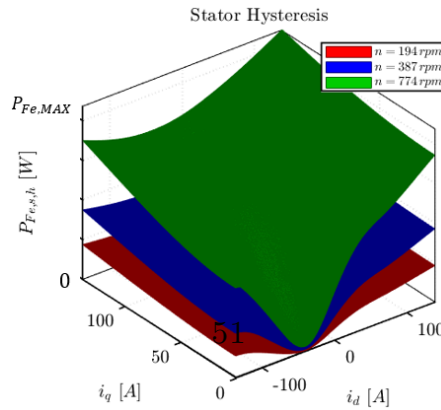


Figure 2.32: Threephase machine stator's hysteresis losses-map as function of the speed.

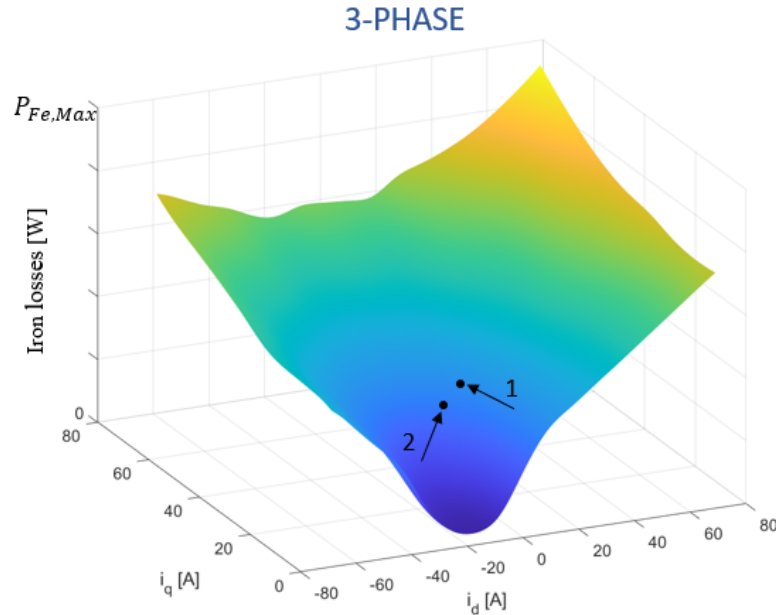


Figure 2.33: Iron losses-map of threephase motor: point 1 is related to rated current along q -axis, point 2 is for the same torque following the MTPA characteristic.

Considering the point "1" of the threephase machine as reference, it can be built a comparative matrix respect the mentioned point, in order to understand the multi-phase advantages in terms of iron losses, as in Table 2.2.

Number of phases	Point 1	Point 2
3	-	-10%
6	-2%	-11%
12	-5%	-16%

Table 2.2: Iron losses comparison respect to the "Point 1" of the threephase motor. The values considered are related to the nominal performance requirements.

In the previous graph the point 1 is related to the rated current injected entirely along the q -axis, while the second one represents the iron losses when the control follows the MTPA trajectory for the same torque. It was possible to estimate the latter thanks to the FEM analysis that allow to study the machine in each physical aspect. In fact, diagrams like that in Fig.2.34 are possible to be obtained, in which the values to have the smallest current amplitude possible to reach the required torque are available.

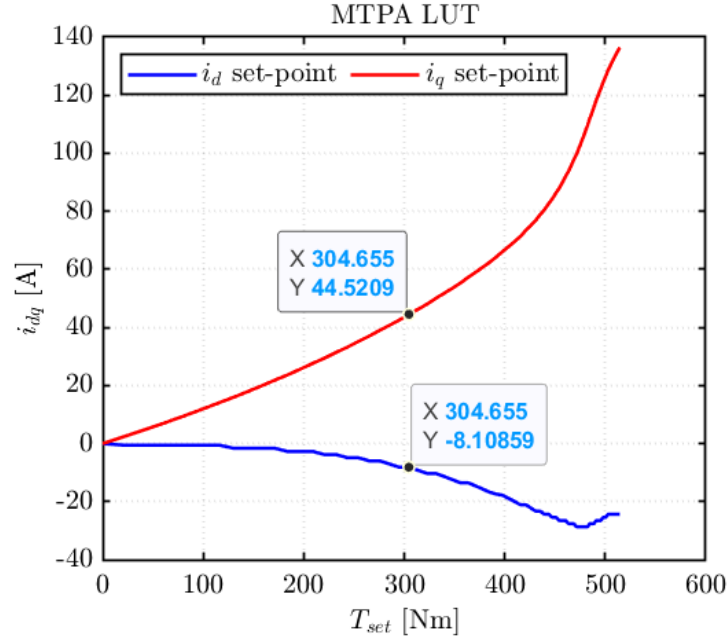


Figure 2.34: Values of d and q currents to use the MTPA characteristic.

Coming back to Fig.2.33, it can be observed that using MTPA trajectory to control the machine can make save some dissipated power because of the smaller current used, exploiting the materials better. The same graphs are available even for sixphase and twelve versions with the difference that the current's values are scaled respectively to one half or a quarter respect to Fig.2.34.

For all the motors considered in the present work, the decreasing of the iron losses is about of 10% for each one if the MTPA trajectory is used, allowing a thermal benefit at the rated working point.

The same maps related to the other two versions have been shown before, noticing that with the increasing of the number of the threephase systems, due to a lower harmonic distortion, the iron losses decrease and thermally speaking it can represent ad advantage because the iron and the magnets will be colder or at least managing the thermal behaviour will be easier.

2.3 Results and final comments

After an analysis about the accuracy of the FEM results obtained with Syre, some final conclusion can come up using the post-processing tool whose name is "Syre_MMM" (Syre Magnetic Model Manipulation), with which analysing the data already computed and reaching interesting analysis is possible, in order to take some decision about the design and the convenience of the choices. The main results available with Syre_MMM, which will be used in the present chapter, are surely the several quantities (like power factor, efficiency, DC-bus voltage, iron losses, ...) represented under the mechanical characteristic, helpful to understand and to predict the behaviour of the motor in every working point, at a certain torque and speed. Below there will be some comparison between the three motors. As first decision to take to build the mechanical characteristic, the maximum limits regard the torque and the speed are needed. The both are remained the same of the threephase model, since at a certain point of working area, some mechanical high stress can arise independently if the motor has one, two or four sets. So, the peak torque value and the maximum speed have been considered from table 2.1. Finally, to compute each working point, fifty five points have been used along the whole rectangle.

Considering the results in order of relevance, the efficiency motor maps are taken into account. In Fig.2.35 the efficiency in the working area of the three considered models are present. As is clear with the pictures below, the maximum efficiency value is increased at low load, and the same trend can be noticed even at about the rated working condition, where the efficiency is increased of almost the 2% from the threephase machine to the twelvephase one. Even if the increment is not so relevant, however is highlighted the constant efficiency curve at the value of 87% in red, and it is shown how its speed range at low load is extended through the different models, allowing better performances for faster applications. For the efficiency value of 87%, for example, the speed range is increased of about the 7% between sixphase system and threephase one, while the ratio between twelvephase and threephase solution is of 14%. Besides, the lower is the efficiency value, the greater is the improvement regard the speed range spanned.

For confidentiality reasons, the values of the graphs are shown between zero and the maximum available.

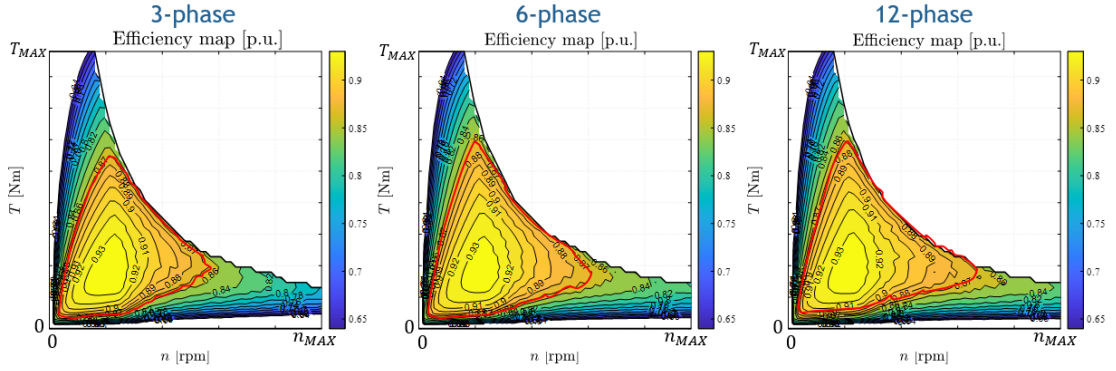


Figure 2.35: Efficiency maps for the three machines.

Secondly, since the TMK is a torque motor, also analysing the torque ripple behaviour could be interesting. For company reason's secret the graphs can not be shown in detail, however they evidence that using more number of threephase systems allows to reduce it drastically. Especially at high speed-low load and at rated working point it seems completely reduced, improving the performances in a such a way to justify the choice to change the number of phases.

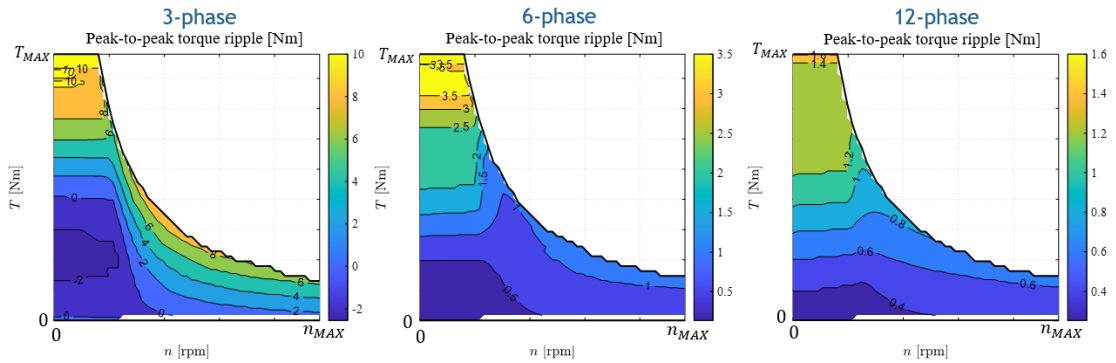


Figure 2.36: Torque ripple maps for the three machines.

Thanks to equations like 2.1-2.2 computing the iron losses of each speed value is possible and drawing the mechanical characteristic with the iron losses values is a *Syre_MMM* possibility.

In Fig.2.37 the value of 200W has been evidenced in all graph (as said, numerical values comes from 2D computations and are used for the sake of comparison). It is evident that for the rated working points of the three types of machine the improvement is not huge, but if it is considered a low-load working condition, the main advantage thanks to the increasing of the number of phases is surely the extended speed range available for the same value of losses. In fact, as regards

the value of 200W, the speed-range is increased of about 80% between sixphase and threephase motor, while if the four-threephase machine is compared with the threephase one, the range is increased of about 100%. Taking into account more threephase systems could represent a good solution to use the motor for higher speeds, allowing a better thermal behaviour of the system.

Finally, one last important aspect has been remained and it concerns the inductive behaviour: the power factor.

No great differences have been found comparing the threephase, sixphase and twelve systems, highlighting only a performances' improvement only with high-load working conditions. In fact, as regards about the rated working point a better power factor of 2.2% has been found in the sixphase model respect to the threephase machine. The same comparison done between twelvephase motor and the threephase one brings an improvement of 2.3%, for the same working condition. So, if the aim of the re-design of the machine is to reach better power factor values, the increasing of the machine's number of phases does not arise evident advantages, except if the machine has to work in an intermittent working cycle, in which an high load for a limited time period has to be considered and power factor can be improved more.

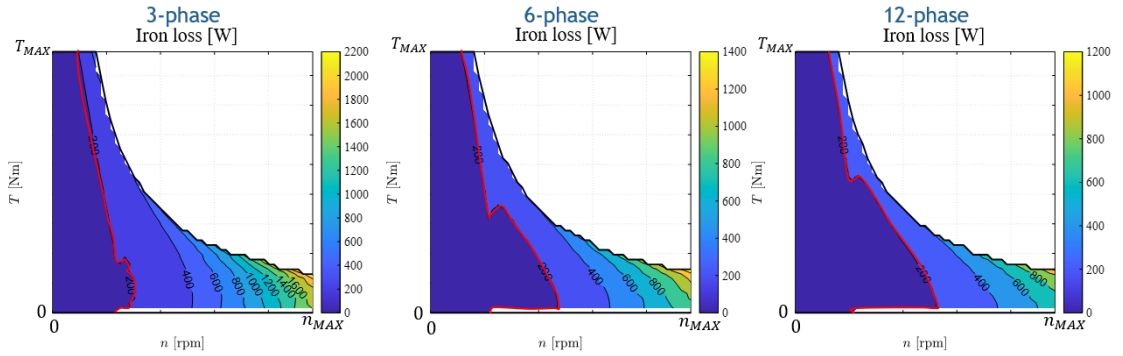


Figure 2.37: Iron losses maps for threephase machine.

After that all the main results have been shown, a final comparison table can be used in order to have an overview about the new technology proposed in this work. The table 2.3 does it and compare the sixphase and twelvephase machines respect to the threephase one.

Quantities	6-phase	12-phase
Torque ripple peak to peak T_{pk-pk}	-86%	-91.4%
Average torque \bar{T}	+2.7%	+4.2%
Iron losses @Rated working-point	-2%	-6%
Efficiency speed range @87%	+7%	+14%
Efficiency @Rated working-point	+1.7%	+2%
Iron losses speed range @200W-Low load	+180%	+200%
Power factor @Rated working-point	+2.2%	+2.3%
DC Bus-voltage V_{DC}	=	=
Power electronics rated current	-50%	-75%

Table 2.3: Sixphase and twelvephase motors' quantities comparison respect to the threephase one.

As it is clear in the previous table, changing from a threephase system to multi-threephase one brings several advantages, and how much they are convenient depends from the number of threephase systems are considered. The DC-link is remained the same of the threephase version because of the re-windings technique adopted for the new models: in this way the coils have been doubled or considered four times in terms of number of turns, but in contrast the current has been halved or divided four times in order to maintain the same total power of the motor drive. So, a reduced torque ripple could be noticed in the both versions sixphase and twelvephase, while the average value of the torque is increased a bit respect to the original machine. Due to the less harmonic distortions in the multi-threephase solutions, a less amount of losses in the iron are present and it means a better value of efficiency; an extended speed range at low load can be reached, since at high-load the behaviour is quite the same and changes only about the power factor. The interesting aspect of the new design is that it does not need a new design of the structural side, but with only re-designing the coils and the windings connection all the benefits of table 2.3 are accessible by the customer without too many complications as regards the hardware aspects.

Another design that can be available is that obtained considering the same windings of the threephase machine. In this way, the current is the quantity that remains with the same values, while the DC-link voltage is decreased since less flux is seen from each windings: it will be an half for the sixphase, a quarter for the twelve phase. It could be a good solutions for the customers which ask to Etel, for example, applications with the same rated current but they have some problems with the power electronics' too high voltage.

If the flux is reduced but the DC-link voltage is considered the same of the

threephase solution, the speed reached is inversely proportional to it, and the base speed is increased twice for the sixphase machine, four times for the twelvephase one. The last solution needs a more detailed thermal analysis about the motor side, because even if the thermal behaviour of the coils should be almost the same, increasing the speed there are more harmonics and the iron and the magnets might suffer.

Chapter 3

Control Simulation

3.1 Control of n 3-phase machine

Coming back to Fig.1.9, the simulation ambient has been shown in order to understand where and how the control strategy is modeled and studied. Besides, as first example the description of each simulation's components in the threephase machine will be given. Then, as regards the sixphase and twelvephase motors, there will be the explanations of the changes added to each version, in order to understand the modifications considered respect to the original simulation.

The whole motor drive is represented by three main blocks: the control side, the power electronics inverter and the motor model. The latter is zoomed in Fig.1.10, in which two subsystems there are, one is for the real motor model while the other one is for mechanical load. Firstly, the motor model will be explained in details in the following sub-section. After, the power electronics side will be also described and at the end the control strategy written in C-language will be shown in details, in order to allow the readers to understand how the simulation works. In the last two sections, however, the modifications adopted to obtain a multi-threephase engine will be presented and at the end of each one even the final results about the torque-control are presented.

3.1.1 Motor model

If the block "Motor model" in Fig.1.10 is opened, what is shown in Fig.3.1 can be observed. The model representation starts with the three voltages given by the inverter, which supply the "Simscape" threephase electrical model: the three phases have a common neutral point and each one is composed of a series resistance and a parallel between a resistance and a driven current generator, which represent the iron resistance behaviour and the current injected into the motor respectively. In this way extracting the voltages used for the energy conversion of the motor is

possible, and they go inside the "Flux integration" block, the green one, in which the fluxes are computed by integration. Then, thanks to the inverted LUTs, the currents are calculated together with the torque, in the red block. After, using into the light blue block the already computed fluxes and currents, the calculation of the iron losses is done with several LUTs and the total currents are obtained. Finally, the currents in the (a,b,c) domain are obtained in the magenta block and they represent even the output, with the torque, of the "Motor model" block.

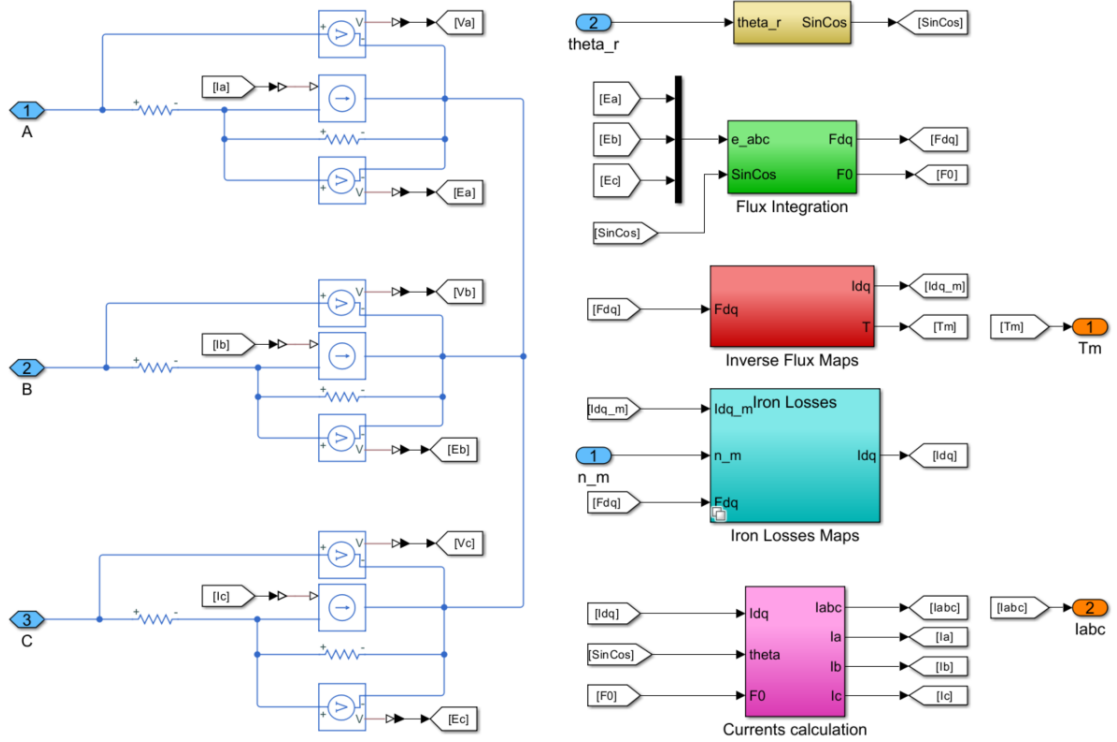


Figure 3.1: Motor model block, with which representing the whole dynamic of the motor is possible using a physical model combined with 1D-2D LUTs model.

Before to describe in detail each block used in "Motor model", the last block has to be introduced, which allows to compute the angular position in the trigonometric coordinates. In Fig.3.2 is shown how it is made, changing the angular absolute position value with the related sinusoidal and cosinusoidal value. The latter technique is used in order to have a continuous value of the angular position instead of the classic discontinuous ramp waveform which characterize the encoder measurement. Then, with a mathematical transformation, is possible to obtain the angular position for each instant, without any discontinuity.

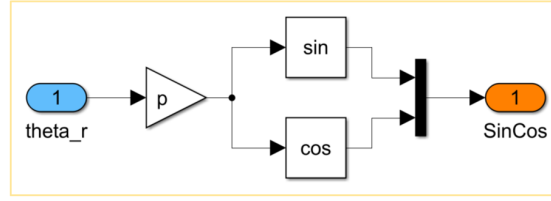


Figure 3.2: Trigonometric transformation from the absolute angular position to the sinusoidal-cosinusoidal related coordinates.

Flux integration

The "Flux integration" block, in green, has as inputs the back electromotive force of the motor, measured on the resistance that represents the iron dynamic. As can be seen in Fig.3.3, after a Clarke transformation, the back-EMF are integrated in the $(\alpha, \beta, 0)$ plane: in this way the cross coupling due to the (d, q) representation is avoided. Then, the fluxes along (α, β) plane are rotated with the well known rotational transformation implemented in block logic, obtaining the fluxes in (d, q) domain; while the zero-sequence flux is left as output.

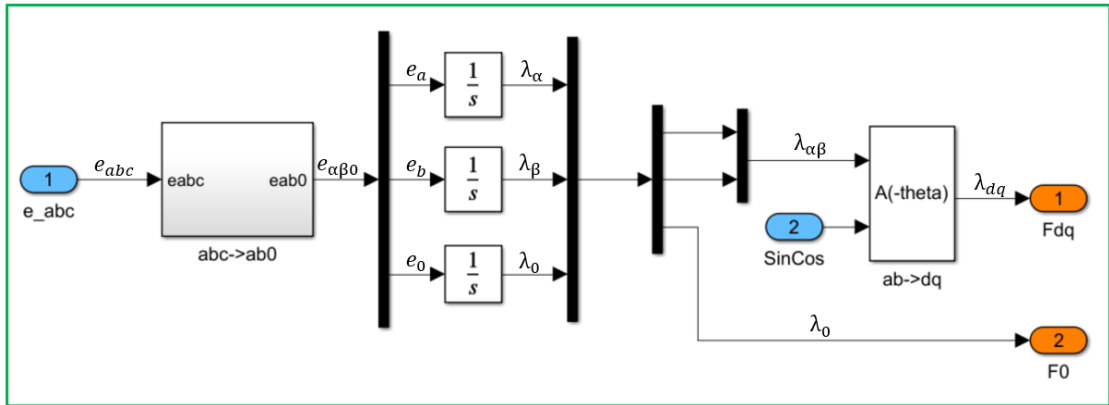


Figure 3.3: Fluxes computation, from the back-EMF to the fluxes in (d, q) domain.

Inverse maps

The Syre's Simulink model is a general one, so it has to be adaptable for each kind of situation. In fact, in Fig.3.4 the block "Quad_Maps" is a constant value with which is possible to suggest the correct convention with which the motor has to be treated: SR, IPM or an induction motor. Depending of the convention chosen, the different quantities will be taken with the positive value only or not. The same block will be used even for the other blocks.

In "Inverse maps" logic block the inputs are the fluxes in the (d, q) frame and in

Iron losses calculation

The iron represents an amount of losses that can be taken into account as an additional current to the torque generation's required one, as explained in Fig.3.6. In the dark green block, zoomed in Fig.3.7, the amount of additional current due to the iron losses is calculated and summed to the current required for the torque generation. In this way the whole current to inject in the physical model representation is obtained.

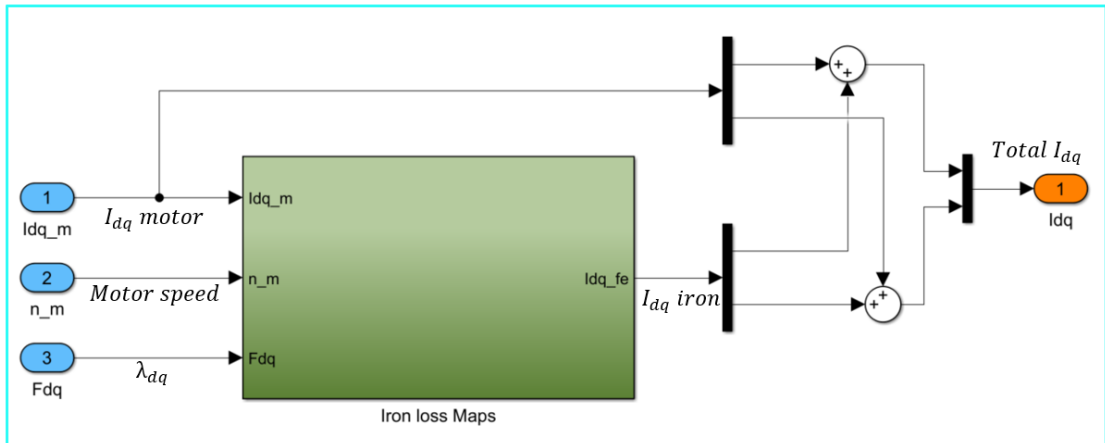


Figure 3.6: Iron losses addition in terms of current to the required one for the energy conversion.

In Fig.3.7 is explained how the "Iron Loss Maps" works. After that the currents are managed depending of the type of the motor's convention is used, three 2D LUTs are interpolated to reach each loss contributions: the losses due to the iron, the eddy currents and the permanent magnets are used, the same kind seen in Fig.2.33 in the previous chapter. As explained in section 2.2.2 with the equations 2.1.2.2, the speed adaptation is taken into account and it is multiplied with the losses considered at the rated speed, in order to obtain the proper losses for each speed.

Since the VSD technique is considered to model the machine, the power to be considered in order to have the current has to be re-scaled, because the VSD approach consider the whole machine but the quantities represented are related only to a system: the scaling factor of $2/3$ has to be taken into account for the threephase motor.

Finally, in the MATLAB function the method introduced in [19] has been adopted to compute the currents contributions.

load is represented in different waves. Several type of control are available in the model, managed by the parameter "Ctrl_type": current, flux, torque and speed control. As regards the first three ones, the load is considered as a dragged mechanical load, which impose its speed as dynamic behaviour, thanks to a external signal. The case is in rectangle numbered with "1". But, if the considered control is a speed one, the load is represented as the red rectangle "2", in which the load has an own dynamic as function of the inertia and the three setttable parameters "Tf", "Tv" and "Bm". The latter are respectively the friction loss, the ventilation loss component and the damping constant.

Besides, the "Tm" is the motor's torque that arrives from the "Motor model", while the "Tload" is from an external signal and represent the main torque load. In addition, if properly set, the load can be modeled as in the rectangle "2" even if the control is not a speed one, just changing the terminal links.

"Tf", "Tv" and "Bm" parameters are set as zero for the final results.

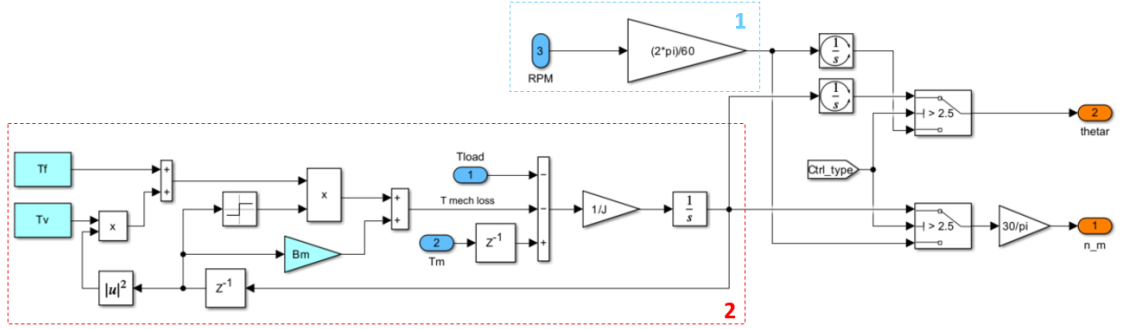


Figure 3.9: Mechanical load's representation depending on the control type adopted: current, flux, torque or speed control.

Finally, the rectangle "2" in Fig.3.9, analytically speaking, represents the equations system of the 3.1, allowing to obtain the mechanical speed and the angular mechanical position of the motor.

$$\begin{aligned}
 J \cdot \frac{d\omega}{dt} &= T_m - T_L - [\omega \cdot B_m + \text{sgn}(\omega) \cdot (T_F + |\omega|^2 \cdot T_V)] \\
 \theta_f &= \int_0^\infty \omega(t) dt + \theta_0
 \end{aligned} \tag{3.1}$$

3.2 Inverter Model

The second "main" block analysed is the inverter's one.

The inverter model adopted for the simulation represents the dynamic behaviour of

the PWM technique, using the comparison between a carrier signal and a triangular one, in order to switch from the duty-cycles to the real binary commands that the switches need to impose the voltages.

As it is reported in Fig.3.10, the duty-cycles which come from the microprocessor are used for the computation of the inverter's commands. The outputs of "PWM" block are the binary signals that need to be converted in a "physical" quantities because of the realistic inverter model of Simulink. The model for the power electronics side is a standard block already present in Simulink and allows to consider different kinds of switches: IGBT, MOSFET, thyristor and ideal one. The latter has been considered in the thesis. So, even physical quantities like the on-resistance or the voltage drop could be taken in consideration. Finally, neglecting the fast-switching phenomena, the voltage to apply to the motor's phases are given as outputs.

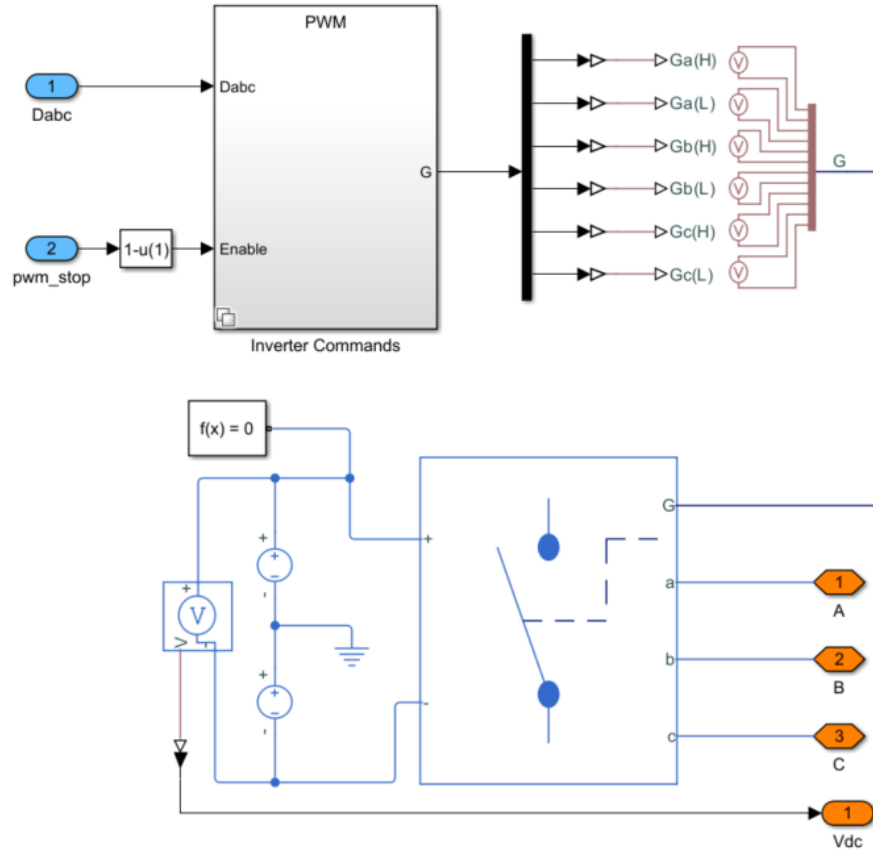


Figure 3.10: Inverter model used to represent the real dynamics of the power electronics.

As a better explanation, in the following picture is shown the logic used to obtain

the PWM binary signals, using the comparison between the triangular waveform and the carrier one, enabled by an external signal which comes from the control from safety reasons.

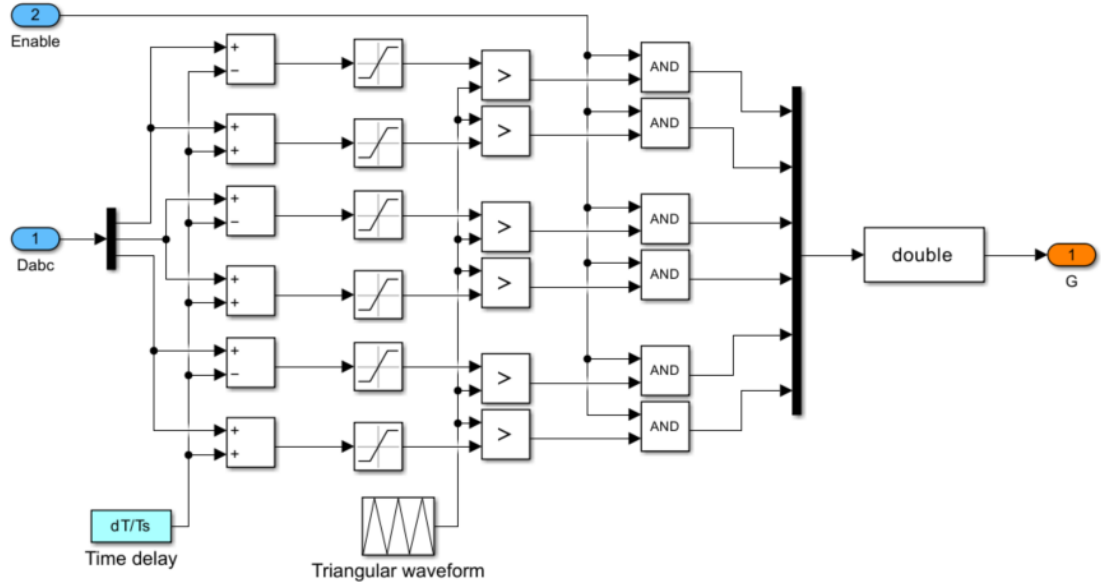


Figure 3.11: PWM logic used to obtain the gate-signals from the duty-cycles due to the control side.

In Fig.3.11 all the signals are summed/subtracted with a delay to avoid the short-circuit during the real utilization of the inverter. After that they are saturated to ± 1 , they are compared with a triangular waveform with a unit amplitude. The results are used in a logic "and" with the "Enable" signals, obtaining the final gate-signals.

3.3 Digital Control

In the threephase model the control logic is already implemented by default, using more or less the classical technique available for a standard threephase motor, similar to the one in Fig.1.16. Even if its study is not the aim of this work, only an introduction about it will be given to the reader in order to understand what has been changed to achieve the control of a multi-threephase.

After some initial commissioning instructions that the motor has to do before the beginning (set the offsets, charge the bootstraps if needed, set the initial λ_{dq} values,...), the control starts with the case so called "START".

In the listing 3.1 can be observed how the control begins, with the angular position

θ recognition and the related speed calculation. After the angular position is saturated to $\pm 2\pi$, the related sinusoidal and cosinusoidal values are computed, as described in the previous section. Then with the function "speed_compute_sc" explained in the listing 3.2 the speed is computed and then filtered thanks to the macro "_Filter", represented in the listing 3.3. Then, the speed value is saved for the next iteration as "old" and easier name to be used are saved as "SinCos_elt" for the electrical angular position and the "omega_elt" for the filtered electrical speed value.

Listing 3.1: Position detection and speed computation.

```

1 case START:
2
3     //-----Speed Compute-----//
4
5     theta_elt_meas      = PP * theta_mec_meas;
6     while(theta_elt_meas > PI)
7         theta_elt_meas -= TWOPI;
8     while(theta_elt_meas < -PI)
9         theta_elt_meas += TWOPI;
10    SinCos_elt_meas.sin = sin(theta_elt_meas);
11    SinCos_elt_meas.cos = cos(theta_elt_meas);
12
13    speed_compute_sc(SinCos_elt_meas, &SinCos_elt_meas_old, &
omega_elt_meas);
14    _Filter(omega_elt_meas, omega_elt_meas_f, Ts*TWOPI*10);
15    omega_mec_meas      = omega_elt_meas/PP;
16    omega_mec_meas_f    = omega_elt_meas_f/PP;
17    omega_mec_meas_rpm  = omega_mec_meas_f*30/PI;
18
19    SinCos_elt_meas_old.sin = SinCos_elt_meas.sin;
20    SinCos_elt_meas_old.cos = SinCos_elt_meas.cos;
21
22    SinCos_elt.sin = sin(theta_elt_meas);
23    SinCos_elt.cos = cos(theta_elt_meas);
24    omega_elt = omega_elt_meas_f;

```

Listing 3.2: Function "speed_compute_sc" for the speed computation.

```

1 void speed_compute_sc(Xsc sincos, Xsc *sincos_old, float* omega) {
2
3     float tmp1, tmp2;
4
5     tmp1 = sincos.sin * sincos_old -> cos;
6     tmp2 = sincos.cos * sincos_old -> sin;
7
8     tmp1 = tmp1-tmp2;
9     *omega = fs * tmp1;
10

```

```

11     sincos_old -> sin = sincos.sin;
12     sincos_old -> cos = sincos.cos;
13
14 }

```

Listing 3.3: Filter macros.

```

1 #define _Filter( x, x_filt, k_filt) x_filt= x_filt + k_filt*(x-
   x_filt) ;

```

After the angular position and speed calculation, depending on the type of the control adopted, the reference for the currents has obtained to be used in the current-loop control, as in listing 3.4. Since it is a general control code, with "Quad_Maps" can selected the convention used for the studied motor, as mentioned in the Simulink model explanation before.

Listing 3.4: Currents' reference computation.

```

1  switch (Ctrl_type){
2
3      case 0: //CurrentControl
4          isdq_ref.d = isdq_ext.d;
5          isdq_ref.q = isdq_ext.q;
6          break;
7
8      case 2: //TorqueControl
9          ReadLut(&ID_REF[0], fabs(T_ext), TMAX, TMIN, DT,
10 INV_DT, &isdq_ref.d);
11          ReadLut(&IQ_REF[0], fabs(T_ext), TMAX, TMIN, DT,
12 INV_DT, &isdq_ref.q);
13          switch (Quad_Maps){
14              case 0:
15                  isdq_ref.d = sgn(T_ext)*isdq_ref.d;
16                  break;
17
18              case 1:
19                  isdq_ref.d = sgn(T_ext)*isdq_ref.d;
20                  break;
21
22              case 2:
23                  isdq_ref.q = sgn(T_ext)*isdq_ref.q;
24                  break;
25          }
26
27          break;
28
29      case 3: //SpeedControl
30          omega_ref_in = n_ref_in * RPM2RAD;
31          ramp(omega_ref_in, accel * RPM2RAD*Ts, &
32 omega_ref_ramp);

```

```

29         sp_var.ref = omega_ref_ramp;
30         sp_var.fbk = omega_elt/PP;
31         sp_par.lim = T_rated;
32         kp_w      = 2*OMEGA_BW*J;
33         ki_w      = pow(OMEGA_BW,2)*J;
34         sp_par.ki  = ki_w*Ts;
35         sp_par.kp  = kp_w;
36         PIReg(&sp_par, &sp_var);
37         T_ext     = sp_var.out;
38
39         ReadLut(&ID_REF[0], fabs(T_ext), TMAX, TMIN, DT,
40 INV_DT, &isdq_ref.d);
41         ReadLut(&IQ_REF[0], fabs(T_ext), TMAX, TMIN, DT,
42 INV_DT, &isdq_ref.q);
43         switch (Quad_Maps){
44             case 0:
45                 isdq_ref.d = sgn(T_ext)*isdq_ref.d;
46                 break;
47             case 1:
48                 isdq_ref.d = sgn(T_ext)*isdq_ref.d;
49                 break;
50             case 2:
51                 isdq_ref.q = sgn(T_ext)*isdq_ref.q;
52                 break;
53             }
54         break;
55     }

```

For the current control the reference are imposed automatically with external values. For the torque control, however, once that the torque reference is given by an external signal, the currents are extracted from the 1D LUTs which have the related current on the MTPA trajectory: it could be modified with a simpler characteristic vertical direction along the q -axis. Finally, if a speed control is adopted, since the speed control has less dynamic than the torque one, the reference is saturated regard the acceleration with a ramp. The latter is describe in the following listing 3.5. At the end, in order to have the torque reference to insert in the LUTs, the PI controller is set with the related values k_p, k_i and the speed-loop is solved thank to the PIReg function, as in listing 3.6.

Listing 3.5: Ramp function used to saturated any kind of acceleration about a certain quantity.

```

1 void ramp(float target, float delta, float *output) {
2
3     if (*output <= target) {
4         *output = (*output + delta);

```

```

5         if (*output>target)
6             *output=target;
7     }
8     else {
9         *output = (*output - delta);
10        if (*output<target)
11            *output=target;
12    }
13 }

```

Listing 3.6: PIReg function adopted to compute a loop in the control strategy.

```

1 void PIReg(XPIRegPars *par ,XPIRegVars *var)
2 {
3     float int_lim;
4
5     //Error computation
6     var->err = var->ref-var->fbk;    //Proportional regulator
7     var->prop = par->kp*var->err;
8     //Saturation of the proportional part to lim
9     if (var->prop>par->lim)
10        var->prop=par->lim;
11     if (var->prop<(-par->lim))
12        var->prop=-par->lim;
13
14     //Limit of the intg part
15     int_lim = par->lim-fabs(var->prop);
16
17     //Integral part
18     var->intg+=par->ki*var->err;
19
20     //Saturation of the intg part
21     if (var->intg > int_lim)
22        var->intg = int_lim;
23     if (var->intg < (-int_lim))
24        var->intg = -int_lim;
25
26     //Output computation
27     var->out=var->prop+var->intg;
28 }

```

The last listing about the regulator is characterised by the error computation and then the "anti-windup" technique is implemented in order to obtain better performances and do not allow the integral part to accumulate too much error in case of high dynamic. In this way, the integrator will reach only the error needed at steady-state to avoid any error in a constant dynamic. After the anti-windup, the integral error is calculated and then the both proportional and integral parts are limited to the limits in order to obtain the output values: the torque and

the voltages will be the results of PIReg function for the speed and current loops respectively.

After the currents references have been computed, it is the time to start the real control.

Firstly, the Clarke transformation is applied to the feedback currents, in order to describe them in (α, β) domain and after even the rotational transformation is considered to bring them in the (d,q) plane using the already computation of sine and cosine of the electrical angle.

The, $v_{\alpha\beta}(k-1)$ is saved, taking the value of $v_{\alpha\beta}^*$ from the previous step and it will be used in the "FluxObserver" function, shown below in listing 3.8 below. It uses "KOBS" parameters instead of the "g" used in the control scheme seen in the theory chapter, but the meaning is the same. In fact, the "FluxObserver" function represents in C-language the flux observer scheme for a threephase system. Firstly, with a LUTs, the flux is estimated using the current's values. Then, the latter is compared with the previous obtained result in order to create the error quantity to be added as feedback. As last step, the flux is observed as integration of an error due to the (α, β) voltages, the voltage drop on the phase resistance and the feedback just calculated and weighted with the parameter "KOBS".

The following step is the setting of the current loop parameters and the "Current_loop" function is applied. The latter is in listing 3.9. The current loop has as output the voltage references to which the feedforward is summed in order to improve the dynamic behaviour of the system.

After it is the turn of the phase-advance algorithm, with which the execution's delay is taken into account. Finally, with the inverse transformations is possible to come back to the (a,b,c) plane and use the voltages obtained to compute the duty using the "PWMduty" function, the last one analysed here, in listing 3.10. The latter use the introduction of the third harmonic inside the duties and the compensation of the dead-time is taken into account.

The last action done by the control is the saturation of the duty-cycles, in this case 0.99, and the check in case of fault using the variable "pwm_stop" thanks to the "CurrentProtection" function: the latter checks if the feedback-currents exceed the maximum value possible and if they do it, the function changes the "State" in "ERROR" and switches the "pwm_stop" value in 1.

Listing 3.7: Duty-cycles computation.

```

1 _clarke(isabc , isab);
2 _rot(isab , SinCos_elt , isdq);
3
4 //vsab_kml is voltage reference at the previous step
5 vsab_kml = vsab_ref;
6 FluxObserver();
7
8 kp_id = OMEGA_BI*Ld_inic;
```

```

9 ki_id = kp_id*OMEGA_BI/10;
10 kp_iq = OMEGA_BI*Lq_inic;
11 ki_iq = kp_iq*OMEGA_BI/10;
12
13 id_par.kp = kp_id;
14 id_par.ki = ki_id*Ts;
15 iq_par.kp = kp_iq;
16 iq_par.ki = ki_iq*Ts;
17
18 //Current loop
19 Current_loop(vdc, Imax_mot, isdq_ref, isdq, &id_par, &id_var, &iq_par
    , &iq_var, &vsdq_ref);
20
21 //Feedforward
22 vsdq_ref.d += RS*isdq.d - omega_elt*lambda_dq.q;
23 vsdq_ref.q += RS*isdq.q + omega_elt*lambda_dq.d;
24
25 //Phase advance
26 dTheta = 1.5*f*omega_elt*Ts;
27 SinCos_elt_dTheta.sin = SinCos_elt.sin*cosf(dTheta) +SinCos_elt.cos*
    sinf(dTheta);
28 SinCos_elt_dTheta.cos = SinCos_elt.cos*cosf(dTheta) -SinCos_elt.sin*
    sinf(dTheta);
29
30 //Compensation of delay execution with phase advance
31 _invrot(vsdq_ref, SinCos_elt_dTheta, vsab_ref);
32 _invclarke(vsab_ref, vsabc_ref);
33 PWMduty(vsabc_ref, vdc,&duty_abc, deadtime, isabc);
34
35 //Duty cycles saturation
36 if (duty_abc.a > 0.99f) duty_abc.a=0.99f;
37 if (duty_abc.b > 0.99f) duty_abc.b=0.99f;
38 if (duty_abc.c > 0.99f) duty_abc.c=0.99f;
39
40 if (duty_abc.a < 0.01f) duty_abc.a=0.01f;
41 if (duty_abc.b < 0.01f) duty_abc.b=0.01f;
42 if (duty_abc.c < 0.01f) duty_abc.c=0.01f;
43
44 //Safe control
45 CurrentProtection(isabc, &State, &pwm_stop);
46 if(pwm_stop){
47     duty_abc.a=0.0f;
48     duty_abc.b=0.0f;
49     duty_abc.c=0.0f;
50 }

```

Listing 3.8: "FluxObserver" function.

```

1 void FluxObserver(void) {

```

```

2
3   lambda_CM_ab_km1 = lambda_CM_ab;
4
5
6   switch (Quad_Maps){
7   case 0:
8       ReadLut2d(&FD_LUT[0][0], fabs(isdq.d), fabs(isdq.q), DIDD
9       , INV_DIDD, DIQD, INV_DIQD , ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX ,
10      IQ_TAB_MIN, n_size , &lambda_CM_dq.d);
11       ReadLut2d(&FQ_LUT[0][0], fabs(isdq.q), fabs(isdq.d), DIQQ
12       , INV_DIQQ, DIQD, INV_DIQD , IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX ,
13      ID_TAB_MIN, n_size , &lambda_CM_dq.q);
14       if (isdq.d < 0)
15           lambda_CM_dq.d = -lambda_CM_dq.d;
16       if (isdq.q < 0)
17           lambda_CM_dq.q = -lambda_CM_dq.q;
18       break;
19
20   case 1:
21       ReadLut2d(&FD_LUT[0][0], fabs(isdq.d), isdq.q, DIDD,
22       INV_DIDD, DIQD, INV_DIQD , ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX ,
23      IQ_TAB_MIN, n_size , &lambda_CM_dq.d);
24       ReadLut2d(&FQ_LUT[0][0], isdq.q, fabs(isdq.d), DIQQ,
25       INV_DIQQ, DIDQ, INV_DIDQ , IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX ,
26      ID_TAB_MIN, n_size , &lambda_CM_dq.q);
27       if (isdq.d < 0)
28           lambda_CM_dq.d = -lambda_CM_dq.d;
29       break;
30
31   case 2:
32       ReadLut2d(&FD_LUT[0][0], isdq.d, fabs(isdq.q), DIDD,
33       INV_DIDD, DIQD, INV_DIQD , ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX ,
34      IQ_TAB_MIN, n_size , &lambda_CM_dq.d);
35       ReadLut2d(&FQ_LUT[0][0], fabs(isdq.q), isdq.d, DIQQ,
36       INV_DIQQ, DIDQ, INV_DIDQ , IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX ,
37      ID_TAB_MIN, n_size , &lambda_CM_dq.q);
38       if (isdq.q < 0)
39           lambda_CM_dq.q = -lambda_CM_dq.q;
40       break;
41   }
42
43   _invrot(lambda_CM_dq, SinCos_elt_dTheta, lambda_CM_ab);
44
45   feedback_OBS.alpha = lambda_CM_ab_km1.alpha - lambda_OBS.alpha
46   ;
47   feedback_OBS.beta   = lambda_CM_ab_km1.beta - lambda_OBS.
48   beta;
49
50   // Integration

```

```

37     lambda_OBS.alpha    += Ts*(vsab_kml.alpha - RS*isab.alpha + KOBS*
feedback_OBS.alpha);
38     lambda_OBS.beta     += Ts*(vsab_kml.beta  - RS*isab.beta  + KOBS*
feedback_OBS.beta);
39     lambda_OBS.amp      = sqrt(pow(lambda_OBS.alpha,2) + pow(
lambda_OBS.beta,2));
40
41     _rot(lambda_OBS, SinCos_elt_dTheta, lambda_dq);
42     T_elt = 1.5*f*PP*(lambda_OBS.alpha*isab.beta - lambda_OBS.beta*
isab.alpha);
43     delta = atan(lambda_dq.q/lambda_dq.d);
44
45 }

```

Listing 3.9: "Current_loop" function.

```

1 void Current_loop(float vdc, float Imax, Xdq isdq_ref, Xdq isdq,
XPIRegPars* id_par, XPIRegVars* id_var, XPIRegPars* iq_par,
XPIRegVars* iq_var, Xdq* vsdq_ref) {
2
3     float vs_max;
4     float tmp1;
5
6
7     vs_max = vdc * SQRT1OVER3;
8
9     //d-axis control loop
10    id_par->lim = vs_max;
11    id_var->ref = isdq_ref.d;
12    id_var->fbk = isdq.d;
13    PIReg(id_par, id_var);
14    vsdq_ref->d=id_var->out;
15
16    // saturation to Imax
17    tmp1 = sqrtf(Imax*Imax-isdq_ref.d*isdq_ref.d);
18    if (isdq_ref.q>tmp1)    isdq_ref.q = tmp1;
19    if (isdq_ref.q<-tmp1)   isdq_ref.q =-tmp1;
20
21    //q-axis control loop
22    iq_par->lim    =vs_max;
23    iq_var->ref    =isdq_ref.q;
24    iq_var->fbk    =isdq.q;
25    PIReg(iq_par, iq_var);
26    vsdq_ref->q=iq_var->out;
27
28 }

```

Listing 3.10: "PWMduty" function.


```

1 void PWMduty(Xabc vsabc_ref, float vdc, Xabc* duty_abc, float
  deadtime, Xabc isabc) {
2
3     float tmp1, tmp2, tmp3, vdc_inv, vzs;
4
5     if (vdc>0) vdc_inv = 1.0f/vdc; // avoid division by zero
6
7     // zero-sequence detection:
8     // tmp1 = max, tmp2 = min
9     // tmp3 is the phase in between: tmp3 = -(max + min)
10
11    tmp1 = vsabc_ref.a; // max
12    tmp2 = vsabc_ref.b; // min
13
14    if (vsabc_ref.a < vsabc_ref.b) {
15        tmp1 = vsabc_ref.b; // max
16        tmp2 = vsabc_ref.a; // min
17    }
18
19    if (tmp1<vsabc_ref.c) tmp3 = tmp1;
20    else {
21        if (tmp2>vsabc_ref.c) tmp3 = tmp2;
22        else tmp3 = vsabc_ref.c;
23    }
24
25    vzs=tmp3*0.5f; // zero sequence voltage
26
27    // Duty-cycles, with zero-sequence
28    duty_abc->a = 0.5f + (vsabc_ref.a + vzs)*vdc_inv;;
29    duty_abc->b = 0.5f + (vsabc_ref.b + vzs)*vdc_inv;;
30    duty_abc->c = 0.5f + (vsabc_ref.c + vzs)*vdc_inv;;
31
32    duty_abc->a += deadtime/Ts*sgn(isabc.a);
33    duty_abc->b += deadtime/Ts*sgn(isabc.b);
34    duty_abc->c += deadtime/Ts*sgn(isabc.c);
35 }

```

Only two functions are remained to be shown and they are not related to the physic of the machine used. They are regard the interpolation of 1D and 2D LUTs and they have been used to obtain the values of reference-current and the fluxes for the flux observer. Their explanation is not interesting for a better understanding of the theory, so they will be only reported below.

Listing 3.11: "ReadLut" and "ReadLut2d" functions.

```

1 void ReadLut(float *tab0, float Xin, float Xmax, float Xmin, float DX
2     , float inv_DX, float* Yout)
3 {

```

```

4      int X0;
5
6      //Limit input range
7      if (Xin > Xmax) Xin = Xmax;
8      if (Xin < Xmin) Xin = Xmin;
9
10     //Select element X0
11     X0=floor (( Xin-Xmin)*inv_DX);
12     //address of X0
13     tab0 = tab0 + X0;
14     //Interpolate between Y(tab0) and Y(tab0+1)
15     *Yout=*tab0 +(*(tab0+1)-*tab0)*inv_DX*(Xin-Xmin-DX*X0);
16
17 }
18
19 // Read a 2 dimensional look-up-table (matrix)
20 void ReadLut2d(float *tab0, float x, float y, float Dx, float invDx, float
    Dy, float invDy, float Xmax, float Xmin, float Ymax, float Ymin, int
    Npointx, float* V)
21 {
22     float ix, iy;
23     float rx, ry;
24     float V1, V2;
25     int delta;
26
27     //Limit input range
28     if (x > Xmax) {x=Xmax-0.1f*Dx;}
29     if (x < Xmin) {x=Xmin+0.1f*Dx;}
30     if (y > Ymax) {y=Ymax-0.1f*Dy;}
31     if (y < Ymin) {y=Ymin+0.1f*Dy;}
32     //Select element 0 for x and y
33     rx=(x-Xmin)*invDx;
34     ry=(y-Ymin)*invDy;
35
36     // indexes of element 0
37     ix=floor(rx);
38     iy=floor(ry);
39     // distance from element 0
40     rx=rx-ix;
41     ry=ry-iy;
42     //address of element 0
43     delta=(ix)+(iy)*Npointx;
44
45     tab0=tab0+delta;
46     //interpolate on x
47     //V1 is exact on x and FLOOR along y
48     V1=*tab0+(*(tab0+1)-*tab0)*rx;
49     //V2 is exact on x and CEIL along y
50     tab0=tab0+Npointx;

```

```

51 | V2=*tab0+(*(tab0+1)-*tab0)*rx;
52 | // interpolate on y
53 | *V=V1+(V2-V1)*ry;
54 | }

```

In the following section, since the threephase simulation has been already treated, only some considerations and the results will be explained, in order to check the correctness of the control strategy adopted. In order to simplify the argumentation, the simulation results will be described after the commissioning states of the control which finish after 0.1s.

3.4 Simulation of the 3-phase motor drive

The threephase model has been already used as example to explain the simulation's structure in Simulink. So, the control has been already shown and now it is the time to show the results obtained, in order to check if the control works or not. Several quantities will be presented to have a complete overview about the system studied and dynamics considered.

In table 3.1 there are the frequencies set to obtain the stable control in the threephase model.

Quantities	value
Switching frequency f_s [kHz]	10
Current loop band-pulsation $\omega_{b,I}$ [rad/s]	$2\pi \cdot 700$
KOBS band-pulsation [rad/s]	$2\pi \cdot 800$
Speed low-pass digital filter [rad/s]	$2\pi \cdot 50$

Table 3.1: Settings of the control strategy adopted for the threephase motor.

Firstly, the crucial quantities to be shown are the mechanical ones, in order to check if the torque control loop works well with a dragged mechanical load.

As on the top of Fig.3.12, for the first 0.17 seconds, the torque-reference is changed from the rated value to the negative half one, in order to test the dynamic system response. Since it is a simulation ambient and not a real system, it has been done instantaneously, with step signals.

Moreover, imposing a ramp load-speed starting from 0.18s until 0.19s, in Fig.3.12 can be seen the torque response to a dynamic load, with a natural increasing of the torque-ripple.

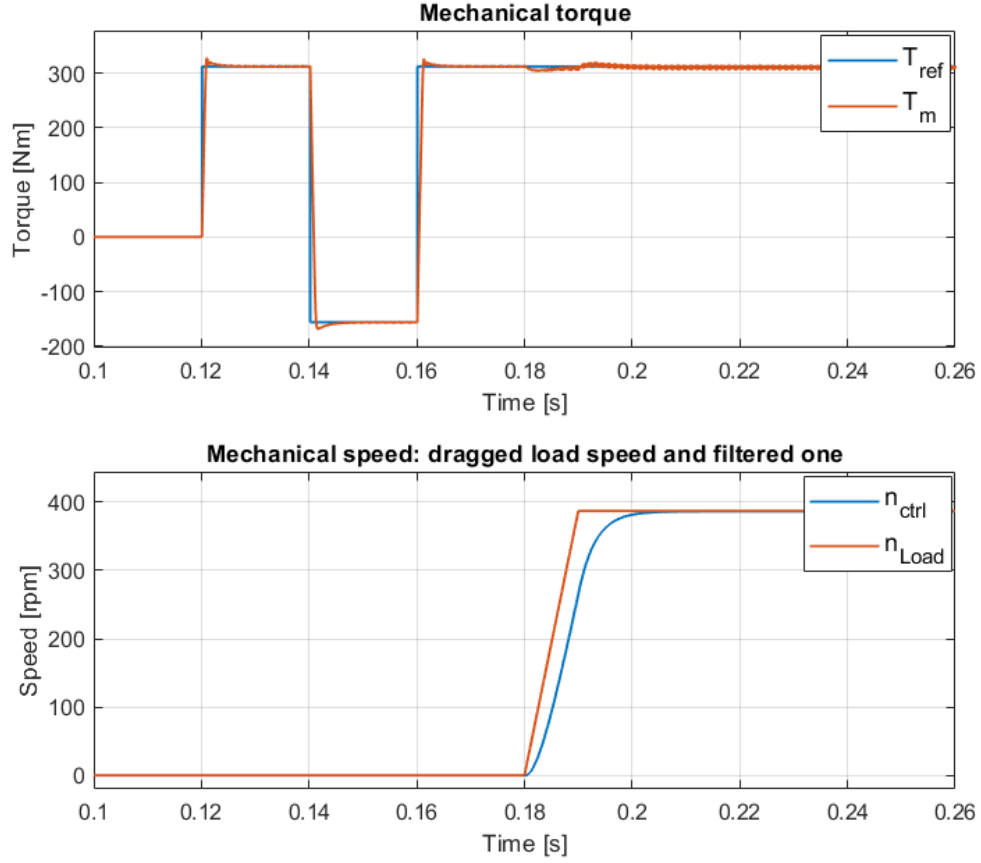


Figure 3.12: From the top to the bottom: torque reference and motor torque, dragged load speed and filtered one by the control.

In the previous figure, it is evident that the speed used in the control to compute the flux observation is digitally filtered to prevent excessive dynamics during the control process.

Secondly, it could be interesting analysing the duty-cycles used in the control, which represents the final control output and could be helpful to check it. In Fig.3.13 there are the duty-cycles, saturated up to the value of 0.99 and that implement the third harmonic injection, as it is evident below.

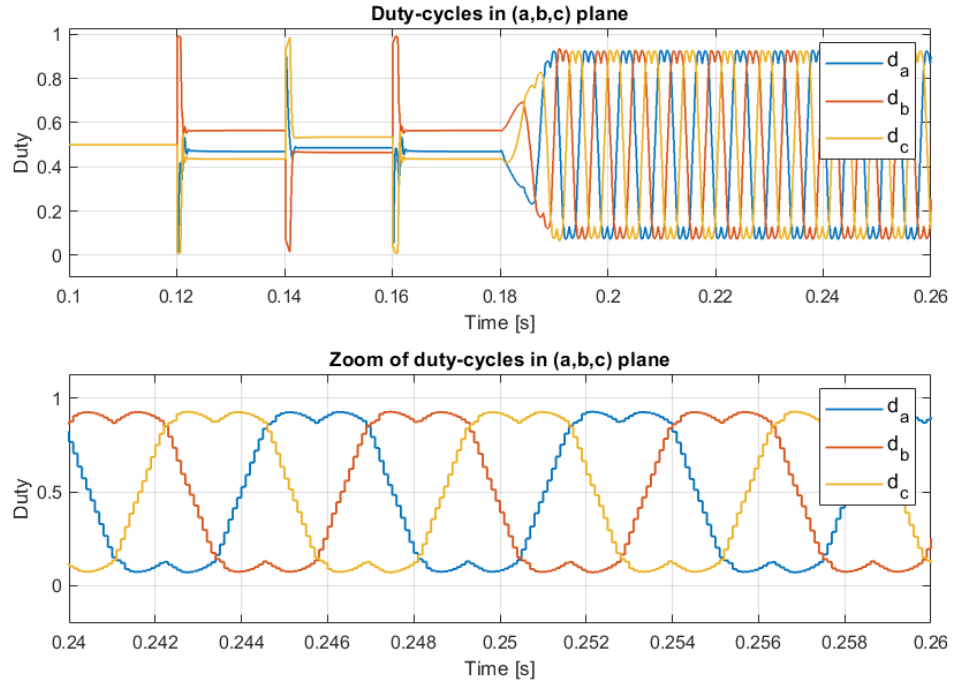


Figure 3.13: Threephase machine duty-cycles implementing the third harmonic injection. In the bottom graph a zoom at steady state is present.

It is clear that the dynamic behaviour starts as soon as the speed increases, and the third harmonic is the main characteristic of the quantities just shown. Other quantities that can be analyzed include the currents in the (a,b,c) domain and their corresponding I_{dq} components after coordinate transformations. In the first graph of Fig.3.14, the three-phase currents are displayed, showing a balanced three-phase system. The waveforms of the currents change when the speed becomes non-zero, as the magnetic field rotates along the rotor structure. In the second graph of Fig.3.14, a zoomed-in view highlights the difference between the current directly measured from the motor model and the ones sampled by the control, with the digital quantization that characterizes them. After a Clarke and rotational transformation the currents along d and q axis are available, and in Fig.3.15 they are presented. The both are displayed in three different representations: the current extracted from the motor model, possible only in a simulation, the used in the control one, quantized, and the related references which come up from the LUTs used to link the torque reference with the MTPA trajectory.

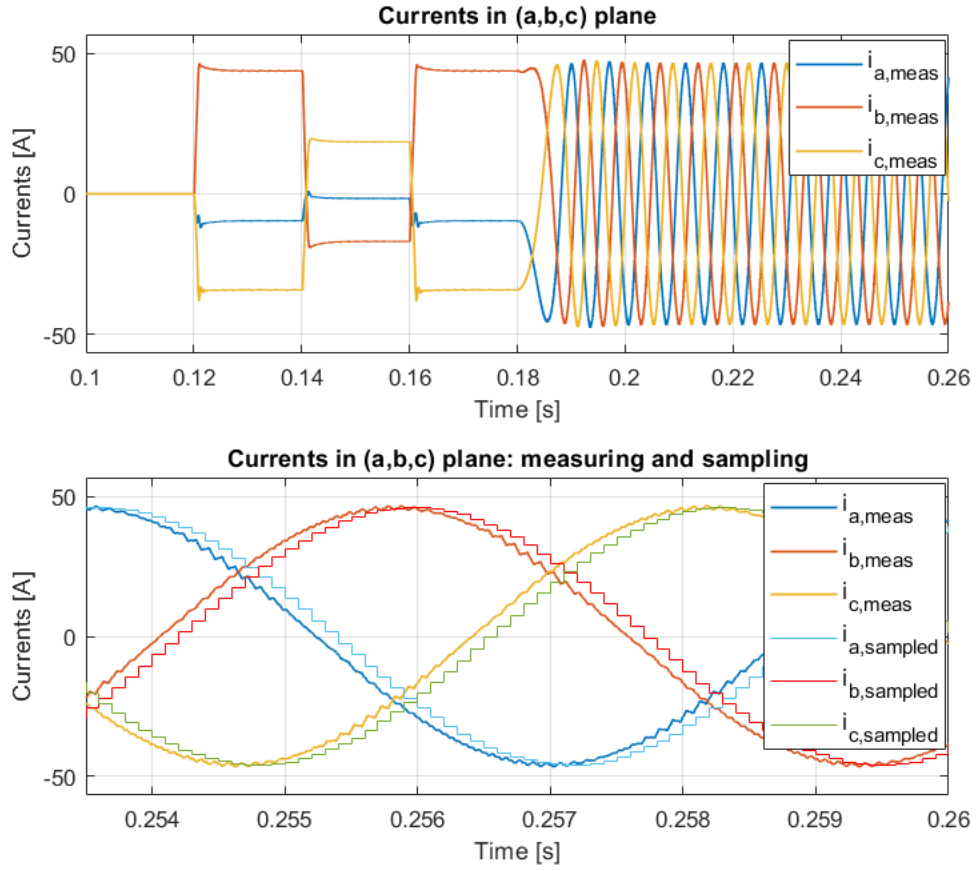


Figure 3.14: From the top to the bottom: i_{abc} and a zoom to highlight the difference between the physic quantities and the sampled ones.

The control works in the good way, in fact, after a brief transient in which the control has to stabilize the quantities, they rich perfectly the references given by the LUTs used.

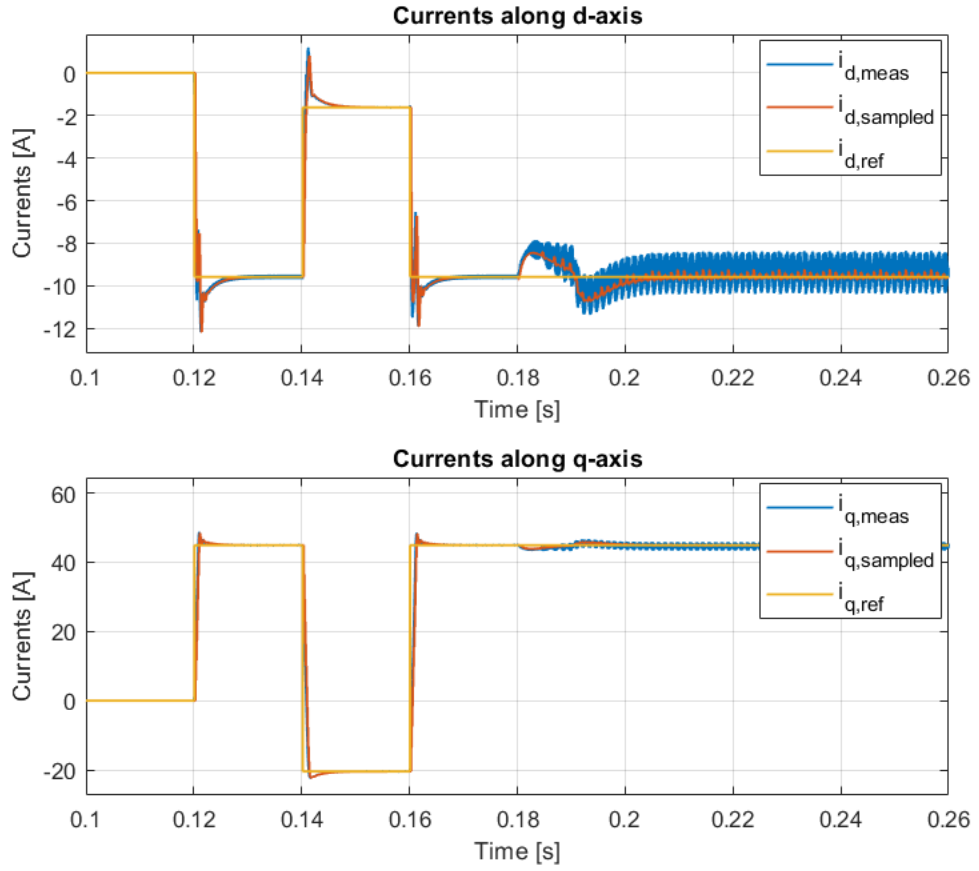


Figure 3.15: Currents in (d,q) frame: it is shown the "real" motor current, the one used in the control after the feedback currents transformation, the reference.

The last graph that is needed to check the flux-observer algorithm is the one below. In Fig.3.16 the fluxes in (d,q) sub-plane are shown in order to compare the fluxes taken from the motor model (in the simulation only it is possible) with the observed ones.

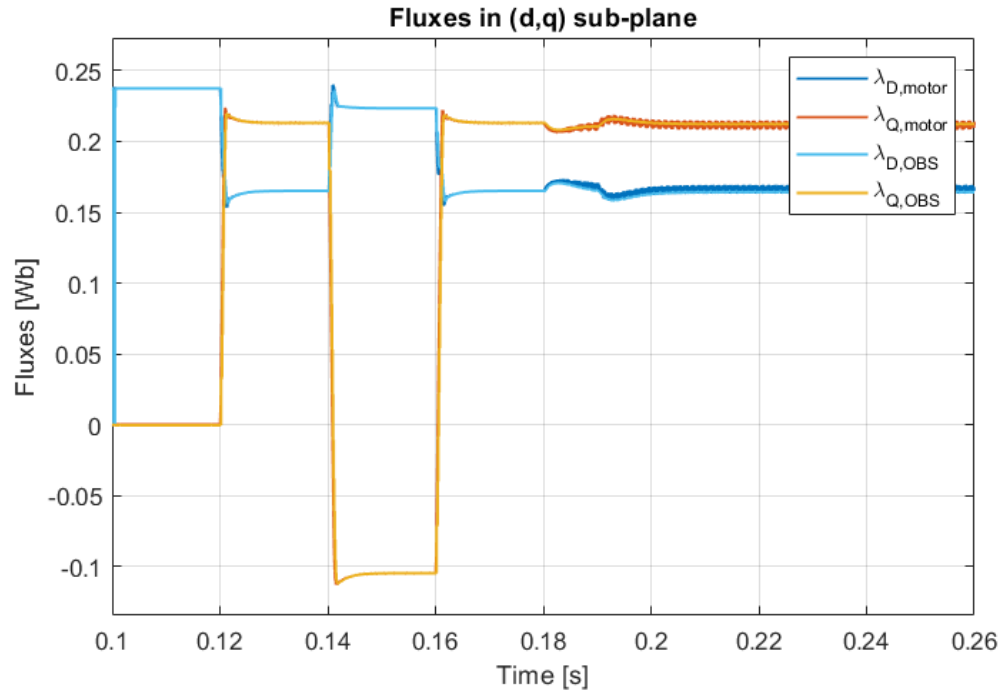


Figure 3.16: Fluxes in (d,q) frame: it is shown the "real" motor fluxes and the observed ones.

As it comes up from Fig.3.16, the fluxes are well-observed, allowing to use the correct feed-forward value for the control.

3.5 Simulation of the 6-phase motor drive

3.5.1 Modifications to the original model

When a six-phase system is taken into account, other aspects have to be considered to make the motor model consistent with reality. This section will explain each modification made to the original threephase model to allow readers to reproduce it easily.

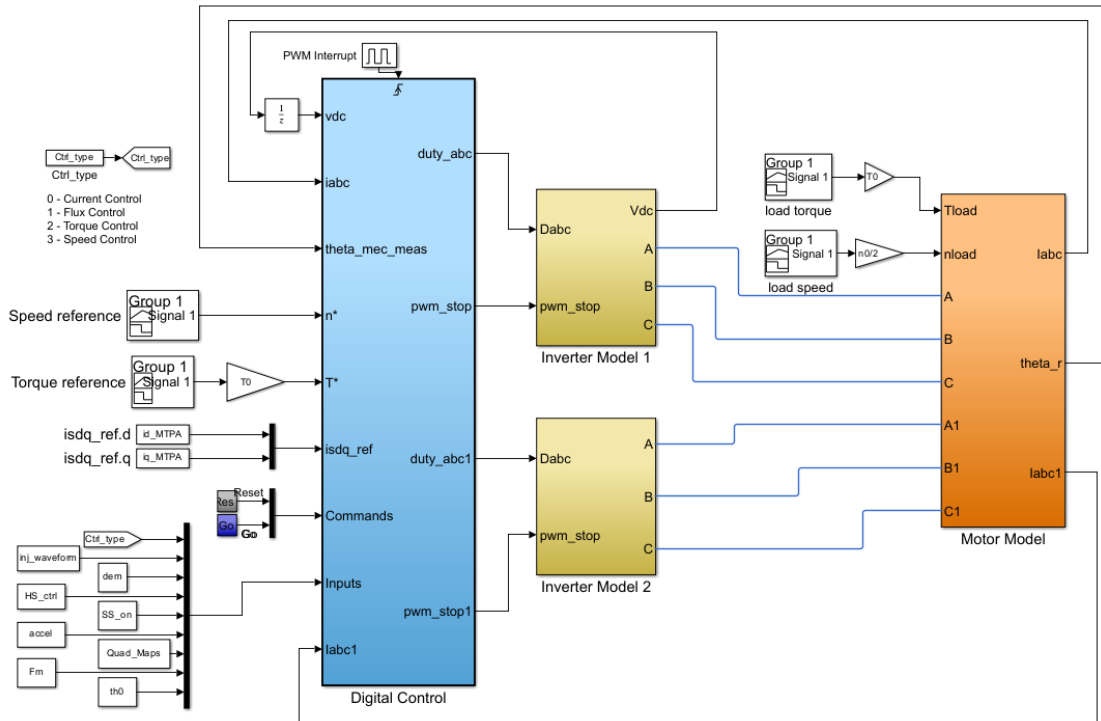


Figure 3.17: Sixphase machine Simulink model overview.

Firstly, as shown in Fig.3.17, the single inverter structure is replaced with two identical ones. Both inverters are independent of each other, and the only difference is that the DC-link voltage information is sent to the control from only one of them, since they will use a common voltage bus as a supply.

The second main modification adopted in the new model is the whole motor model block, which requires some changes. The mechanical load block remains the same. Starting from the electrical circuit, two isolated stars are inserted, each with a separated neutral point. In this way, the zero sequence mode quantities can be neglected. Each three-phase system is supplied by an inverter. The system just described is in the figure below.

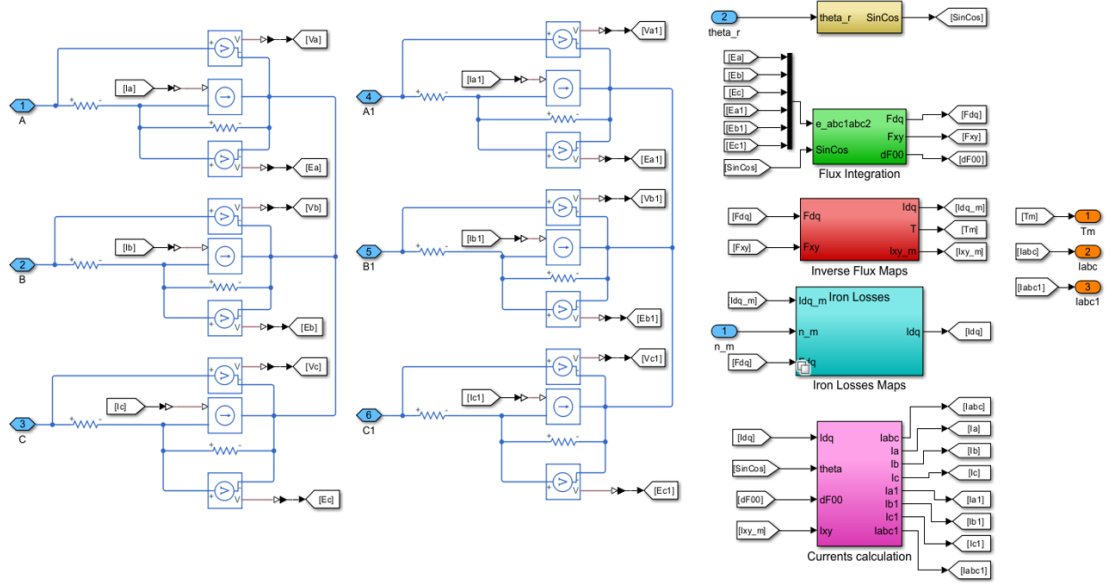


Figure 3.18: Motor model in the sixphase motor: the analogous blocks of threephase version are present, but with some modification explained in this section.

The approach used to obtain the several needed quantities is the same as the threephase model example given before in the previous section. Once the back-electromotive forces are obtained by measuring the voltage drop across the iron resistance, they are transformed to $(\alpha, \beta, x, y, o_1, o_2)$ using the T_6 transformation described in the theory chapter. As shown in Fig.3.19, only the transformation matrix and the number of represented quantities are changed in the "Flux Integration" block, since there are now six variables instead of three.

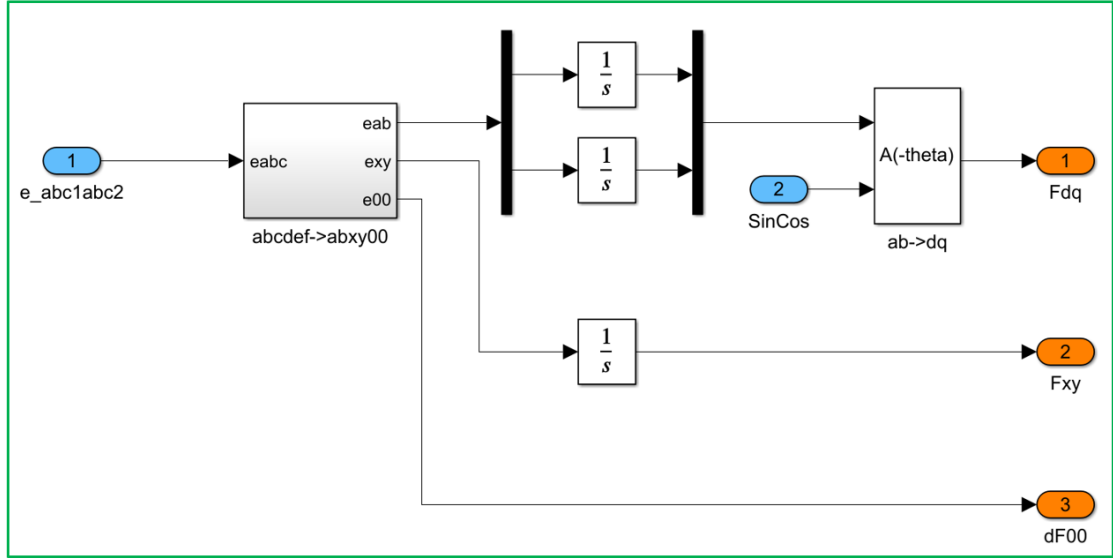


Figure 3.19: Motor model in the sixphase motor: the analogous blocks of threephase version are present, but with some modification explained in this section.

Then, the process to reach the currents and torque is identical to the three-phase model, except for the scalar factor that multiplies the cross-product between fluxes and currents along the (d,q) domain. The factor that can be seen in equation 1.21 is no longer $3/2$ but 3 , like in equation 1.22, because there are six phases, and the ratio $6/2 = 3$. Additionally, in the same block, another operation is added because of the new (x,y) sub-plane, as shown in Fig.3.20. The multiplication for the inverse of leakage inductance is included to satisfy the equation 1.19 seen at the beginning of this work.

The value of "L_sigma" in Fig.3.20 has been obtained from a FEMM offline analysis, in which only a threephase system has turned on with a constant current value equal for the three windings of a star. The simulation is not a time-harmonic one and the obtained value for the leakage inductance is reported in the following table, where also other fundamental parameters are shown.

After the model obtains the currents, if the iron-losses flag has been activated, the iron-losses computation starts, and the current contribution due to R_{Fe} is calculated. In the "Iron losses" block, the multiplication factor before the MATLAB function has to be changed to $1/3$, in accordance with what was said before about the torque, as evidenced in Fig.3.21. In fact, in the drive-control theory often a transformation which use the proper phasor value is adopted and the power is overrated. So, since the single threephase stare power has to be inserted in the MATLAB function to compute the currents components, the whole system losses

have to be divided for the scalar factor as in the previous figure.

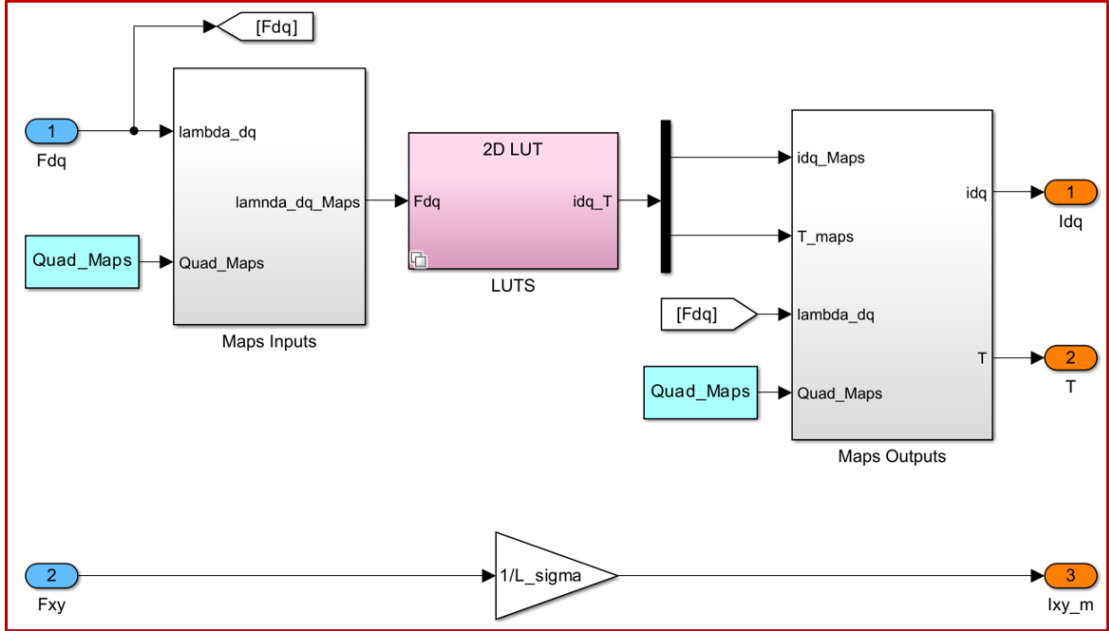


Figure 3.20: Motor model in the sixphase motor: the analogous blocks of threephase version are present, but with some modification explained in this section.

Inductance	value
R_s [Ω]	1.58
L_d [mH]	8.79
L_q [mH]	6.87
$L_{leakage}$ [mH]	5

Table 3.2: Electrical parameters' values estimated: R_s , L_d and L_q are obtained with Syre study, while the leakage one is the result of one FEMM simulation imposing a constant and common value to a threephase set turned on.

Finally, to come back to the initial $(a_1, b_1, c_1, a_2, b_2, c_2)$ domain, the inverse T_6 transformation is adopted in the model instead of the standard inverse Clarke transformation, as reported below.

and the flux-observer. While the former only changes some variable names but has the same structure as the original system, the latter has to be duplicated, as shown in Fig.1.17 in the first chapter.

The code in sixphase machine uses the same functions already shown before, and the first part concerning the speed computation is exactly the same of a standard threephase system. As can be notice in listing 3.12, after the speed and sin/cosine computation, there is the choice between the different control type. Since the common mode is the responsible for the energy conversion, as previously explained in the control theory chapter, the LUTs are related to the latter as reference, while the differential reference mode is set to zero, in order to avoid any unbalancing between the different threephase systems.

Once that the references are obtained, using the proper Clarke transformation for each threephase system, like in equation 1.1, representing the quantities in (α, β) domain is possible, and then rotate them with the well known rotational transformation: the latter has to be the same for all the systems, since the (d,q) sub-plane is shared between the two sets. Finally, using the matrix in equation 1.32, the feedback currents in common and differential coordinates are extracted.

Listing 3.12: Reference imposition and common and differential current feedback calculation.

```

1  case START:
2
3      //-----Speed Computation-----//
4
5
6      //-----Control Type -----//
7
8      switch (Ctrl_type){
9
10         case 0: //CurrentControl
11             isdq_ref_cm.d = isdq_ext.d;    //External signal
12             isdq_ref_cm.q = isdq_ext.q;    //External signal
13             break;
14
15         case 2: //TorqueControl
16
17             //Common Mode Reference
18             ReadLut(&ID_REF[0], fabs(T_ext), TMAX, TMIN, DT,
19             INV_DT, &isdq_ref_cm.d);
20             ReadLut(&IQ_REF[0], fabs(T_ext), TMAX, TMIN, DT,
21             INV_DT, &isdq_ref_cm.q);
22
23             isdq_ref_dm.d=0;

```

```

22         isdq_ref_dm.q=0;
23         switch (Quad_Maps){
24             case 0:
25                 isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
26                 break;
27
28             case 1:
29                 isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
30                 break;
31
32             case 2:
33                 isdq_ref_cm.q = sgn(T_ext)*isdq_ref_cm.q;
34                 break;
35         }
36         break;
37
38         case 3: //SpeedControl
39             omega_ref_in = n_ref_in * RPM2RAD;
40             ramp(omega_ref_in, accel * RPM2RAD*Ts, &
omega_ref_ramp);
41             sp_var.ref = omega_ref_ramp;
42             sp_var.fbk = omega_elt/PP;
43             sp_par.lim = Tmax_mot; //T_rated
44             kp_w = OMEGA_BW*J;
45             ki_w = pow(OMEGA_BW,2)*J/10;
46             sp_par.ki = ki_w*Ts;
47             sp_par.kp = kp_w;
48             PIReg(&sp_par, &sp_var);
49             T_ext = sp_var.out;
50
51             ReadLut(&ID_REF[0], fabs(T_ext), TMAX, TMIN, DT,
INV_DT, &isdq_ref_cm.d);
52             ReadLut(&IQ_REF[0], fabs(T_ext), TMAX, TMIN, DT,
INV_DT, &isdq_ref_cm.q);
53             isdq_refdm.d=0;
54             isdq_refdm.q=0;
55             switch (Quad_Maps){
56                 case 0:
57                     isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
58                     break;
59
60                 case 1:
61                     isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
62                     break;
63
64                 case 2:
65                     isdq_ref_cm.q = sgn(T_ext)*isdq_ref_cm.q;
66                     break;
67             }

```

```

68         break;
69     }
70
71     _clarke(isabc1, isab1);
72     _clarke1(isabc2, isab2);
73     _rot(isab1, SinCos_elt, isdq1);
74     _rot(isab2, SinCos_elt, isdq2);
75
76     //Common and Differential feedback current calculation
77     is_cm.d= (isdq1.d + isdq2.d)*0.5;
78     is_cm.q= (isdq1.q + isdq2.q)*0.5;
79     is_dm.d= (isdq1.d - isdq2.d)*0.5;
80     is_dm.q= (isdq1.q - isdq2.q)*0.5;

```

Then, the flux-observer function is used (and it is shown after) and the current loop parameters are set for the common-differential coordinates. Following the scheme in Fig.1.16, after the each current loop related to the common and differential mode, the feed-forward is applied to each variable in order to improve the dynamic response. The next step is the decoupling algorithm for the sixphase machine which allows the real reference voltage calculation for each threephase system (here called "1" and "2").

Again, a phase-advancing logic is considered to compensate the implementation delay and the real duties are obtained using different Clarke transformations and the rotational one: switching from (d,q) plane to (a,b,c) one is possible. In this case too, two different inverse Clarke transformation are considered, depending on the electric displacement angle of the threephase system respect to α reference. At the end, after the duties computation, safety control is used to shut down the entire system in the event of a failure.

Listing 3.13: Current loops and duty-cycles computation.

```

1      //Saving the previous reference voltages for the flux-
observer
2      vsab_km1_1 = vsab_ref1;
3      vsab_km1_2 = vsab_ref2;
4
5      FluxObserver();
6
7      //————Current Vector Control————//
8
9      //Setting of common and differential mode parameters
10
11     //d-axis
12     kp_cmd = OMEGA_BI*Ld_inic;
13     ki_cmd = OMEGA_BI*RS;

```



```

14 kp_dmd = OMEGA_BI*L_sigma;
15 ki_dmd = OMEGA_BI*RS;
16 //q-axis
17 kp_cmq = OMEGA_BI*Lq_inic;
18 ki_cmq = OMEGA_BI*RS;
19 kp_dmq = OMEGA_BI*L_sigma;
20 ki_dmq = OMEGA_BI*RS;
21
22 id_par_cm.kp = kp_cmd;
23 id_par_cm.ki = ki_cmd*Ts;
24 iq_par_cm.kp = kp_cmq;
25 iq_par_cm.ki = ki_cmq*Ts;
26 id_par_dm.kp = kp_dmd;
27 id_par_dm.ki = ki_dmd*Ts;
28 iq_par_dm.kp = kp_dmq;
29 iq_par_dm.ki = ki_dmq*Ts;
30
31
32 //Common mode current loop
33 Current_loop(vdc, Imax_mot, isdq_ref_cm, is_cm, &id_par_cm, &
34 id_var_cm, &iq_par_cm, &iq_var_cm, &vdq_cm_ref);
35
36 //Imposing the feed-forward contribution on common
37 current mode
38 vdq_cm_ref.d += RS*is_cm.d - omega_elt*lambda_dq.q;
39 vdq_cm_ref.q += RS*is_cm.q + omega_elt*lambda_dq.d;
40
41 //Differential mode current loop
42 Current_loop(vdc, Imax_mot, isdq_ref_dm, is_dm, &
43 id_par_dm, &id_var_dm, &iq_par_dm, &iq_var_dm, &vdq_dm_ref);
44
45 //Imposing the feed-forward contribution on differential
46 current mode
47 vdq_dm_ref.d += (RS*is_dm.d) - omega_elt*L_sigma*is_dm.q;
48 vdq_dm_ref.q += (RS*is_dm.q) + omega_elt*L_sigma*is_dm.d;
49
50 //Decoupling
51 vsdq_ref1.d = vdq_cm_ref.d + vdq_dm_ref.d;
52 vsdq_ref1.q = vdq_cm_ref.q + vdq_dm_ref.q;
53 vsdq_ref2.d = vdq_cm_ref.d - vdq_dm_ref.d;
54 vsdq_ref2.q = vdq_cm_ref.q - vdq_dm_ref.q;
55
56 //Phase advance
57 dTheta = 1.5*f*omega_elt*Ts;
58 SinCos_elt_dTheta.sin = SinCos_elt.sin*cosf(dTheta) +
59 SinCos_elt.cos*sinf(dTheta);
60 SinCos_elt_dTheta.cos = SinCos_elt.cos*cosf(dTheta) -
61 SinCos_elt.sin*sinf(dTheta);

```

```

57
58      //Duty-cycles computation
59      __invrot(vsdq_ref1, SinCos_elt_dTheta, vsab_ref1);
60      __invclarke(vsab_ref1, vsabc_ref1);
61      PWMduty(vsabc_ref1, vdc,&duty_abc1, deadtime, isabc1);
62
63      __invrot(vsdq_ref2, SinCos_elt_dTheta, vsab_ref2);
64      __invclarke1(vsab_ref2, vsabc_ref2);
65      PWMduty(vsabc_ref2, vdc,&duty_abc2, deadtime, isabc2);
66
67      //Safe control
68      CurrentProtection(isabc1, &State, &pwm_stop1);
69      if(pwm_stop1){
70          duty_abc1.a=0.0f;
71          duty_abc1.b=0.0f;
72          duty_abc1.c=0.0f;
73      }
74      CurrentProtection(isabc2, &State, &pwm_stop2);
75      if(pwm_stop2){
76          duty_abc2.a=0.0f;
77          duty_abc2.b=0.0f;
78          duty_abc2.c=0.0f;
79      }

```

The last thing to be shown before the final results is the flux-observer function, in listing 3.14.

As can be seen, the scheme evidenced in Fig.1.17 is reproduced below in C-language. After the interpolation of the flux across the magnetic LUTs, the value is saved for the next step and the previous flux is compared with the last observed one in order to create the feedback quantity. It is clear that each threephase star has an own feedback, due to the comparison between the magnetic model flux and the proper observed value.

After, using the reference (α, β) voltages and the voltage drop on the series-resistances, the observed flux for each set is obtained and the their average is computed in order to obtain an unique value in the (d,q) domain.

Listing 3.14: "FluxObserver" function for the sixphase machine flux observation .

```

1
2 void FluxObserver(void) {
3     //Saving the flux from the previous digital step
4     lambda_CM_ab_kml = lambda_CM_ab;
5
6     //Extrapolation of the flux from the LUT depending on the
    machine type

```

```

7      switch (Quad_Maps){
8      case 0:
9          ReadLut2d(&FD_LUT[0][0], fabs(is_cm.d), fabs(is_cm.q),
10         DIDD, INV_DIDD, DIQD, INV_DIQD, ID_TAB_MAX, ID_TAB_MIN,
11         IQ_TAB_MAX, IQ_TAB_MIN, n_size, &lambda_CM_dq.d);
12         ReadLut2d(&FQ_LUT[0][0], fabs(is_cm.q), fabs(is_cm.d),
13         DIQQ, INV_DIQQ, DIQD, INV_DIQD, IQ_TAB_MAX, IQ_TAB_MIN,
14         ID_TAB_MAX, ID_TAB_MIN, n_size, &lambda_CM_dq.q);
15         if (isdq.d < 0)
16             lambda_CM_dq.d = -lambda_CM_dq.d;
17         if (isdq.q < 0)
18             lambda_CM_dq.q = -lambda_CM_dq.q;
19         break;
20
21     case 1:
22         ReadLut2d(&FD_LUT[0][0], fabs(isdq.d), isdq.q, DIDD,
23         INV_DIDD, DIQD, INV_DIQD, ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX,
24         IQ_TAB_MIN, n_size, &lambda_CM_dq.d);
25         ReadLut2d(&FQ_LUT[0][0], isdq.q, fabs(isdq.d), DIQQ,
26         INV_DIQQ, DIDQ, INV_DIDQ, IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX,
27         ID_TAB_MIN, n_size, &lambda_CM_dq.q);
28         if (isdq.d < 0)
29             lambda_CM_dq.d = -lambda_CM_dq.d;
30         break;
31
32     case 2:
33         ReadLut2d(&FD_LUT[0][0], is_cm.d, fabs(is_cm.q), DIDD,
34         INV_DIDD, DIQD, INV_DIQD, ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX,
35         IQ_TAB_MIN, n_size, &lambda_CM_dq.d);
36         ReadLut2d(&FQ_LUT[0][0], fabs(is_cm.q), is_cm.d, DIQQ,
37         INV_DIQQ, DIDQ, INV_DIDQ, IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX,
38         ID_TAB_MIN, n_size, &lambda_CM_dq.q);
39         if (isdq.q < 0)
40             lambda_CM_dq.q = -lambda_CM_dq.q;
41         break;
42     }
43
44     //Obtaining the flux from the magnetic model for the next
45     digital step
46     _invrot(lambda_CM_dq, SinCos_elt_dTheta, lambda_CM_ab);
47
48     //Feedback computation for each system
49     feedback_OBS1.alpha = lambda_CM_ab_km1.alpha - lambda_OBS1.
50     alpha;
51     feedback_OBS1.beta   = lambda_CM_ab_km1.beta - lambda_OBS1.
52     beta;
53     feedback_OBS2.alpha = lambda_CM_ab_km1.alpha - lambda_OBS2.
54     alpha;

```

```

40 feedback_OBS2.beta      = lambda_CM_ab_km1.beta - lambda_OBS2.
    beta;
41
42 // Integration
43 lambda_OBS1.alpha += Ts*(vsab_km1_1.alpha - RS*isab1.alpha +
    KOBS*feedback_OBS.alpha);
44 lambda_OBS1.beta += Ts*(vsab_km1_1.beta - RS*isab1.beta +
    KOBS*feedback_OBS1.beta);
45 lambda_OBS1.amp = sqrt(pow(lambda_OBS1.alpha,2) + pow(
    lambda_OBS1.beta,2));
46 lambda_OBS2.alpha += Ts*(vsab_km1_2.alpha - RS*isab1.alpha
    + KOBS*feedback_OBS2.alpha);
47 lambda_OBS2.beta += Ts*(vsab_km1_2.beta - RS*isab1.beta +
    KOBS*feedback_OBS1.beta);
48 lambda_OBS2.amp = sqrt(pow(lambda_OBS2.alpha,2) + pow(
    lambda_OBS2.beta,2));
49
50 aux_ab.alpha = 0.5*(lambda_OBS1.alpha + lambda_OBS2.alpha);
    //Common mode along alpha-axis
51 aux_ab.beta = 0.5*(lambda_OBS1.beta + lambda_OBS2.beta); //
    Common mode along beta-axis
52 _rot(aux_ab, SinCos_elt_dTheta, lambda_dq); //Obtaining the (d,q)
    flux
53 }

```

3.5.2 Results

The logic flow to highlight the results is the same adopted before in the threephase system. In order to obtain a coherent comparison with the original motor, the same settings of table 3.1 have been used.

So, firstly the mechanical behaviour has been studied, taking into account the same path of the threephase machine to be able to compare the both consistently. At the beginning a torque reference positive and negative is applied, in order to check the torque control mode and then a ramp speed has been considered in order to observe the control behaviour dynamically, with a dragged load. In Fig.3.23 there is an example.

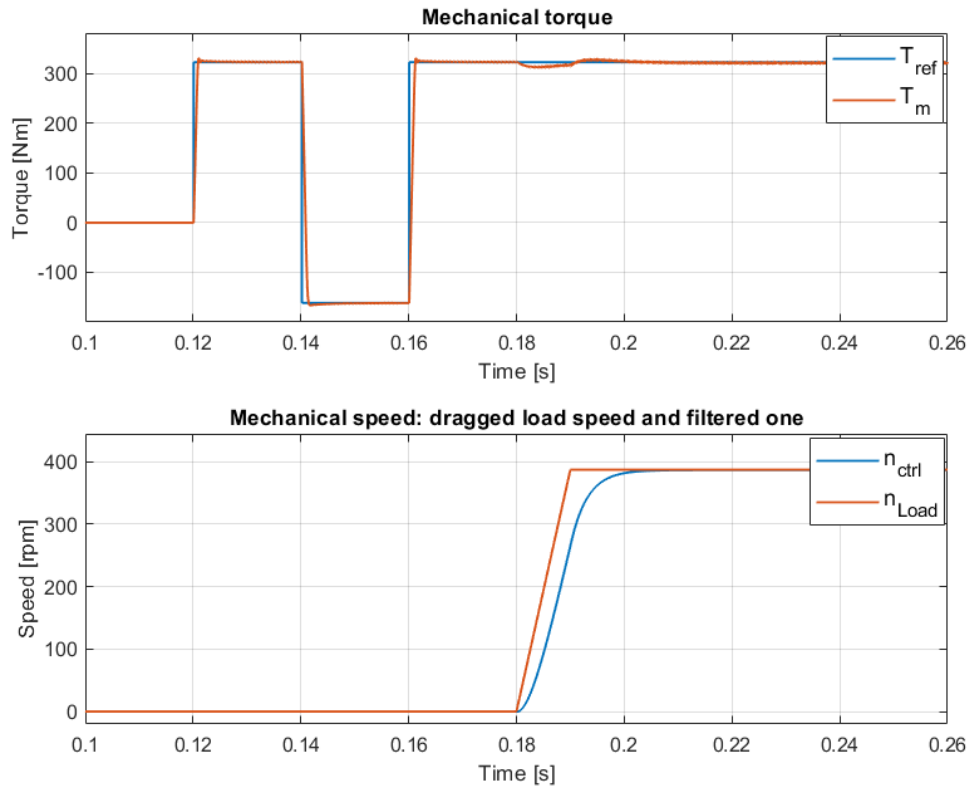


Figure 3.23: From the top to the bottom: Torque reference and the torque applied by the motor, while in the second graph the load speed and the filtered one with a digital filter are represented.

A lower overshoot can be noticed if the previous torque diagram is compared with the threephase version. However, the control seems working well even with high dynamic requirements (step torque references) and with a speed that changes

from zero to the rated value quickly.

Secondly, it is helpful analysing the duty-cycles with the aim to check the correctness of the control. Below is given a graph with the six duties to be compared with what is expected for a sixphase machine.

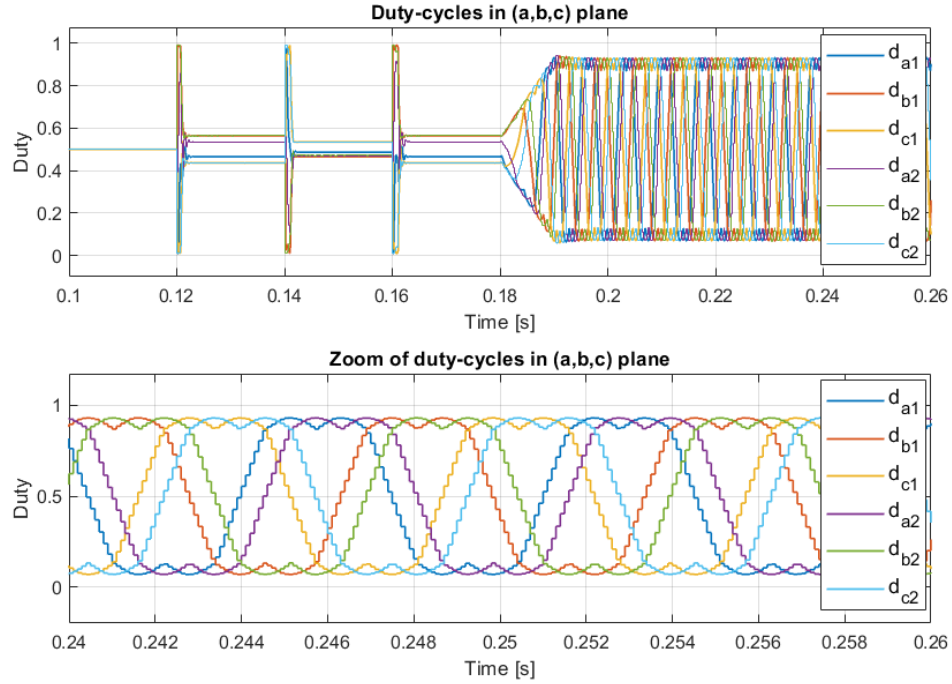


Figure 3.24: Threephase machine duty-cycles implementing the third harmonic injection. In the bottom graph a zoom at steady state is present.

As can be evidenced in the previous figure, the duty of each threephase system is displaced from the other of 30° electrical angle and there is the typical wave-form of the third harmonic injection.

As proof of the correctness of the control strategy, in Fig.3.25 there are the currents waveform in the $(a_1, b_1, c_1, a_2, b_2, c_2)$ plane. They are balanced, as can be seen in the first graph, and they have the right displacement, as it is clear in the second diagram on the bottom side of Fig.3.25. In addition, in the latter graph there are also the sampled currents of a threephase set, to show that the sampling phenomenon follows the real wave-forms with a negligible delay.

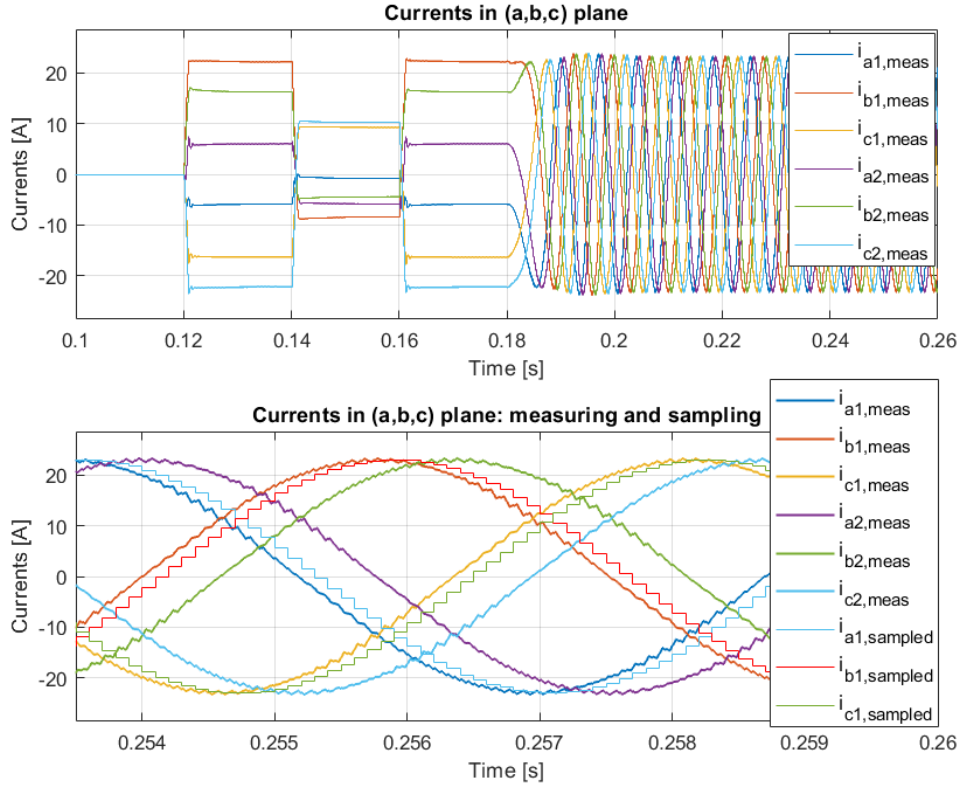


Figure 3.25: From the top to the bottom: $i_{abc,1}$ and $i_{abc,2}$ with 30° displacement electrical degrees, and a zoom to highlight the difference between the physic quantities and the sampled ones.

The next step is to check the correctness of the rotational transformation and observing if the currents reference follows the MTPA trajectory as wished. In Fig.3.26 there is the graph with measured currents along d and q axis, with the related reference and sampled wave-forms. The current loop is well-designed and the currents follows the reference as required.

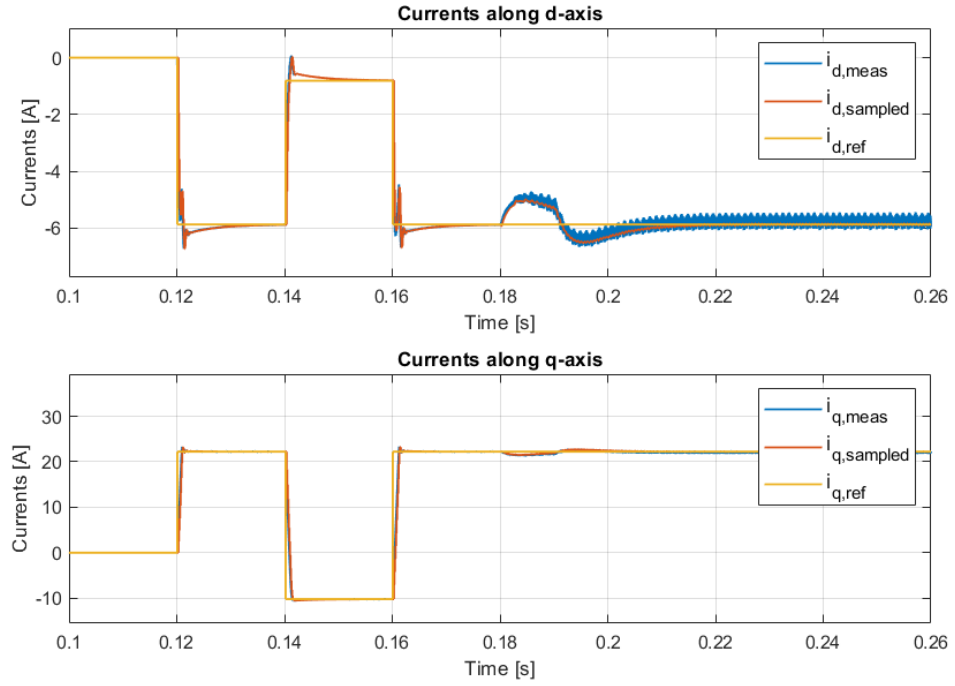


Figure 3.26: Currents in (d,q) frame: it is shown the "real" motor current, the one used in the control after the feedback currents transformation, the reference.

In the previous graph can be observed a smother overshoot comparing it with the threephase control, since the inverter has to manage an halved current and there is an higher inductance.

Finally, the flux observer is analysed and in Fig.3.27 is demonstrated the correct implementation of the algorithm described in the previous sections. In fact, the fluxes in (d,q) domain observed are quite coincident with the real fluxes simulated in the motor model.

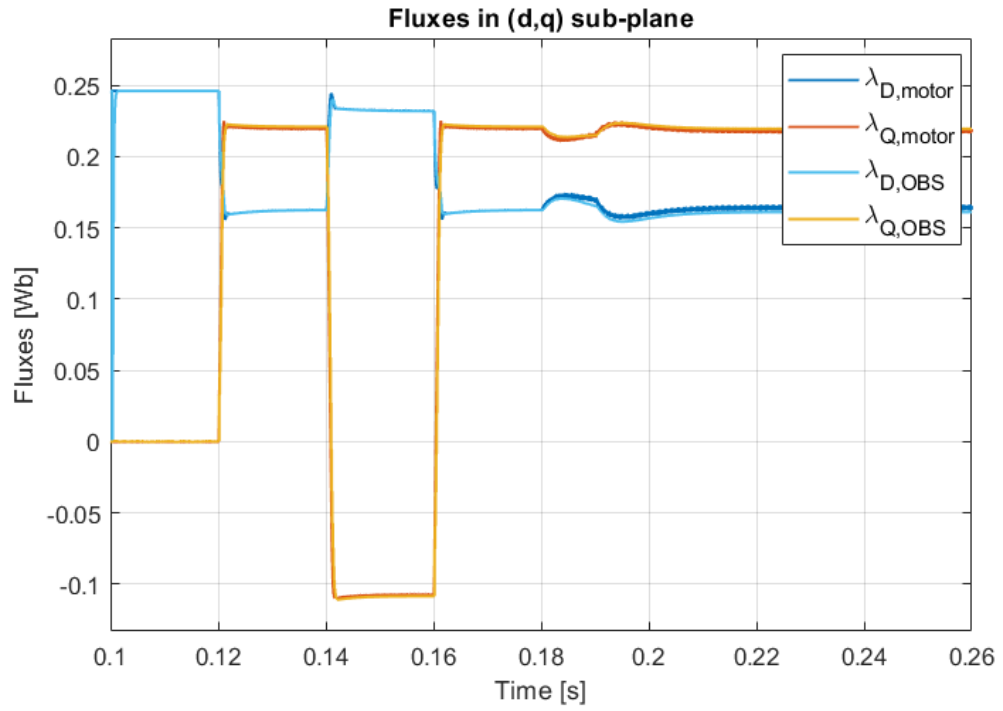


Figure 3.27: Currents in (d,q) frame: it is shown the "real" motor current, the one used in the control after the feedback currents transformation, the reference.

3.6 Simulation of the 12-phase motor drive

3.6.1 Modifications to the original mode

The twelve motors is analogous to the sixphase one in terms of control technique and simulation structure, but it needed some new modifications since there are four threephase systems. So, the main changes to the Simulink model will be briefly described and the results will be shown in the following section.

The overview on the model is shown in Fig.3.28, where the four independent inverters can be noticed: they have the same principle of the sixphase machine, so they are all the same block and only one of them sends the bus voltage information to the control because the DC-link is shared between the different power electronics converter.

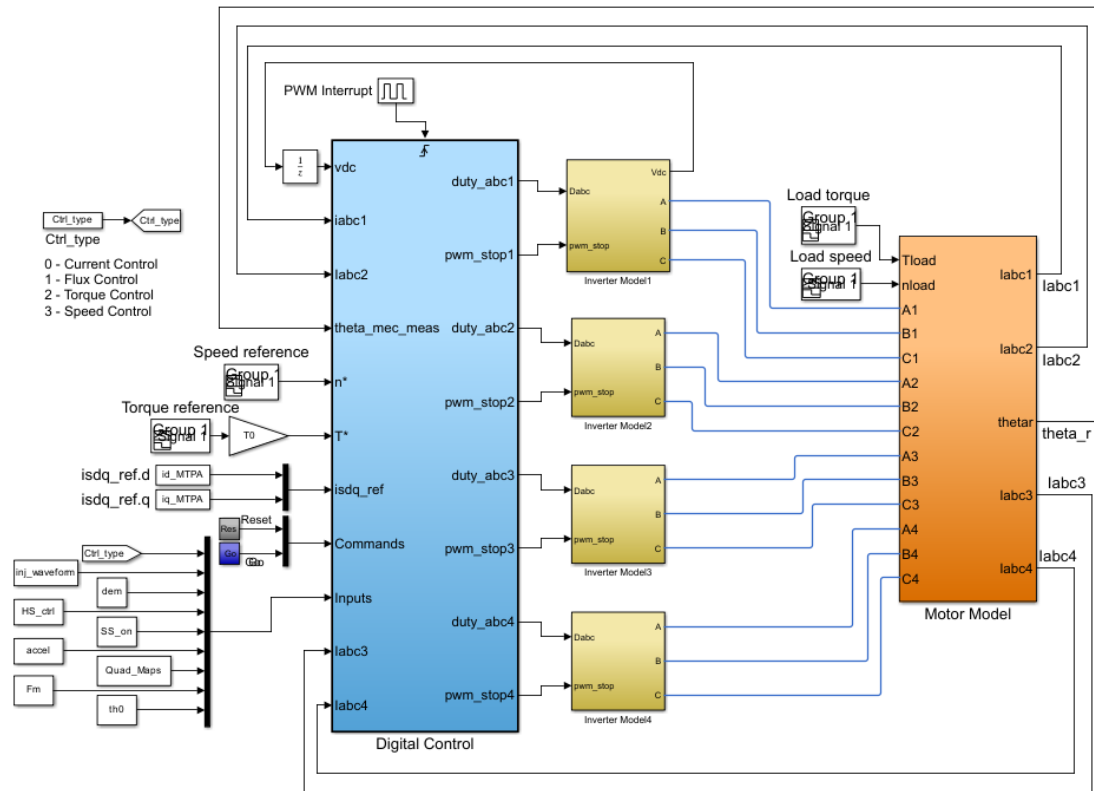


Figure 3.28: Simulink model overview about the twelvephase model.

Starting from the motor model which represents the motor and his dynamics, four threephase isolated stars have been considered, as is clear in Fig.3.29.

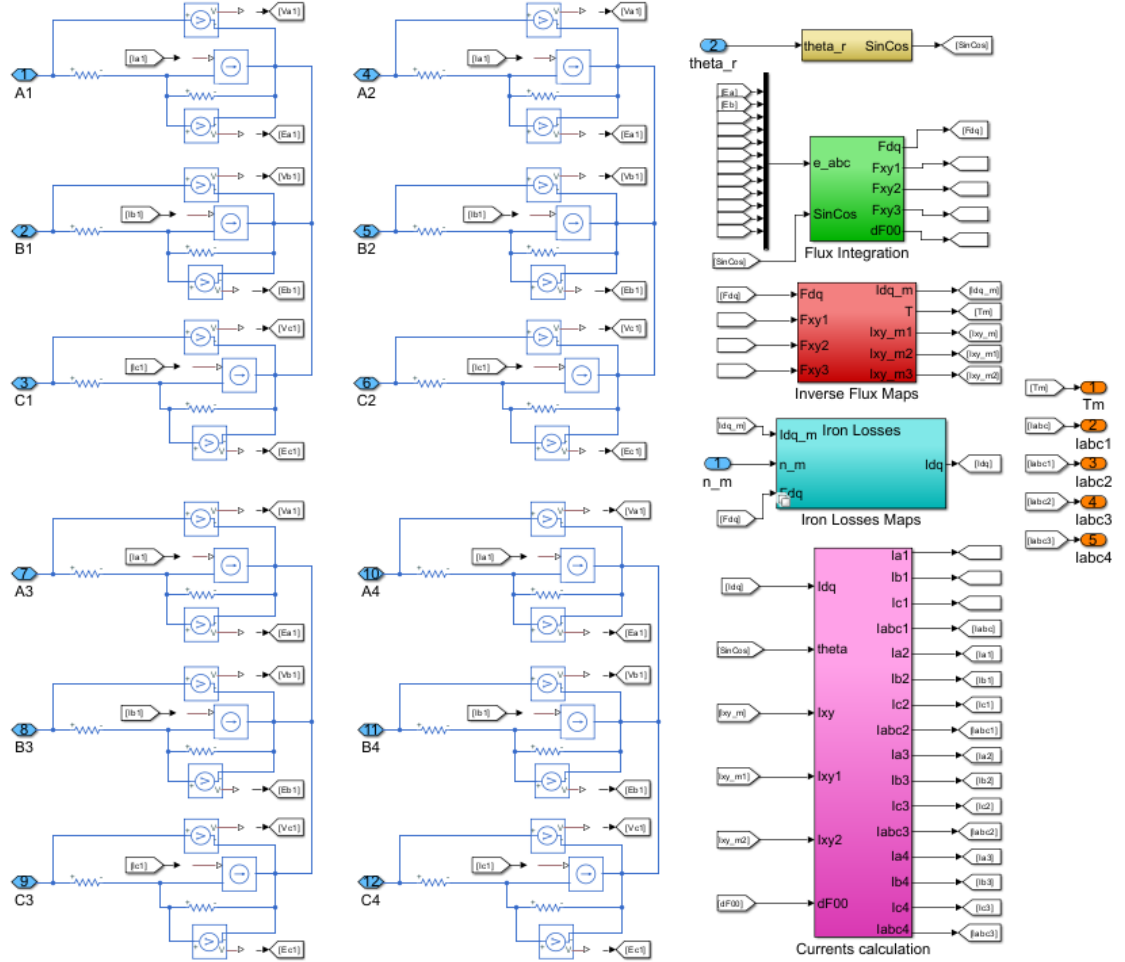


Figure 3.29: Motor model of twelvephase machine, with four threephase systems with isolated neutral points.

As seen before for the sixphase machine, the Clarke transformation is replaced now with the T_{12} that allows to describe a twelvephase system in the $(\alpha, \beta, x_1, y_1, x_2, y_2, x_3, y_3, o_1, o_2, o_3, o_4)$ sub-plane. In fact, if compared with the original Simulink model, the "Flux Integration" block is modified as in the figure below, where what just mentioned before is modelled.

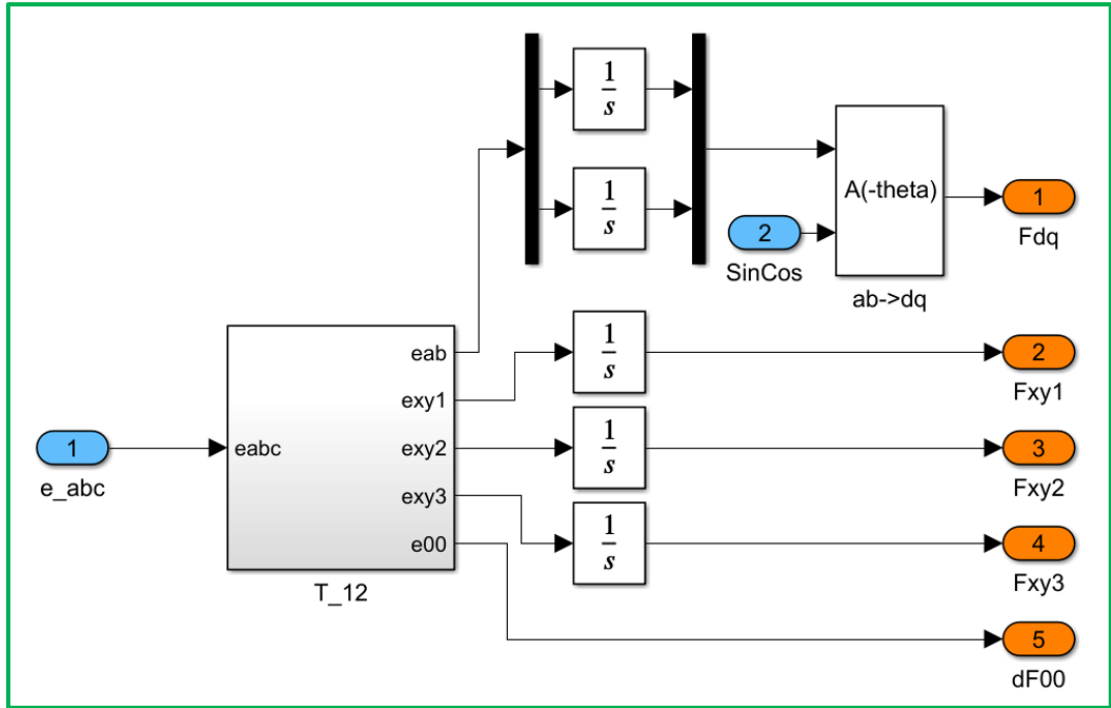


Figure 3.30: "Flux Integration" block, where the T_{12} is implemented and the (α, β) coordinates are rotated to the (d, q) ones.

Once that the fluxes have been obtained, using the LUTs already seen, the currents can be computed in the "Inverse Flux Maps" block and then the torque is calculated with the cross-multiplication between the flux and the current in the (d, q) domain. As regards the latter, the product used for the torque is not multiplied anymore for $3/2$ as for the threephase machine, but the new factor is "6", since the phases are twelve and $12/2 = 6$.

In Fig.3.31 there is an example.

As can be seen in Fig.3.31, the leakage fluxes are divided with the ratio $1/L_{\sigma}$ in order to obtain the currents related to the other sub-planes. The parameters value used are reported together in following table.

Then, for the iron-losses computation only the factor evidenced in the Fig.3.21 has to be changed in $1/6$ to consider the fact there are twelve phases, as for the torque. Finally, there is the block which obtains the real currents, the "Current calculation" one, in which is considered the inverse transformation of T_{12} . In Fig.3.32 there is what has been said.

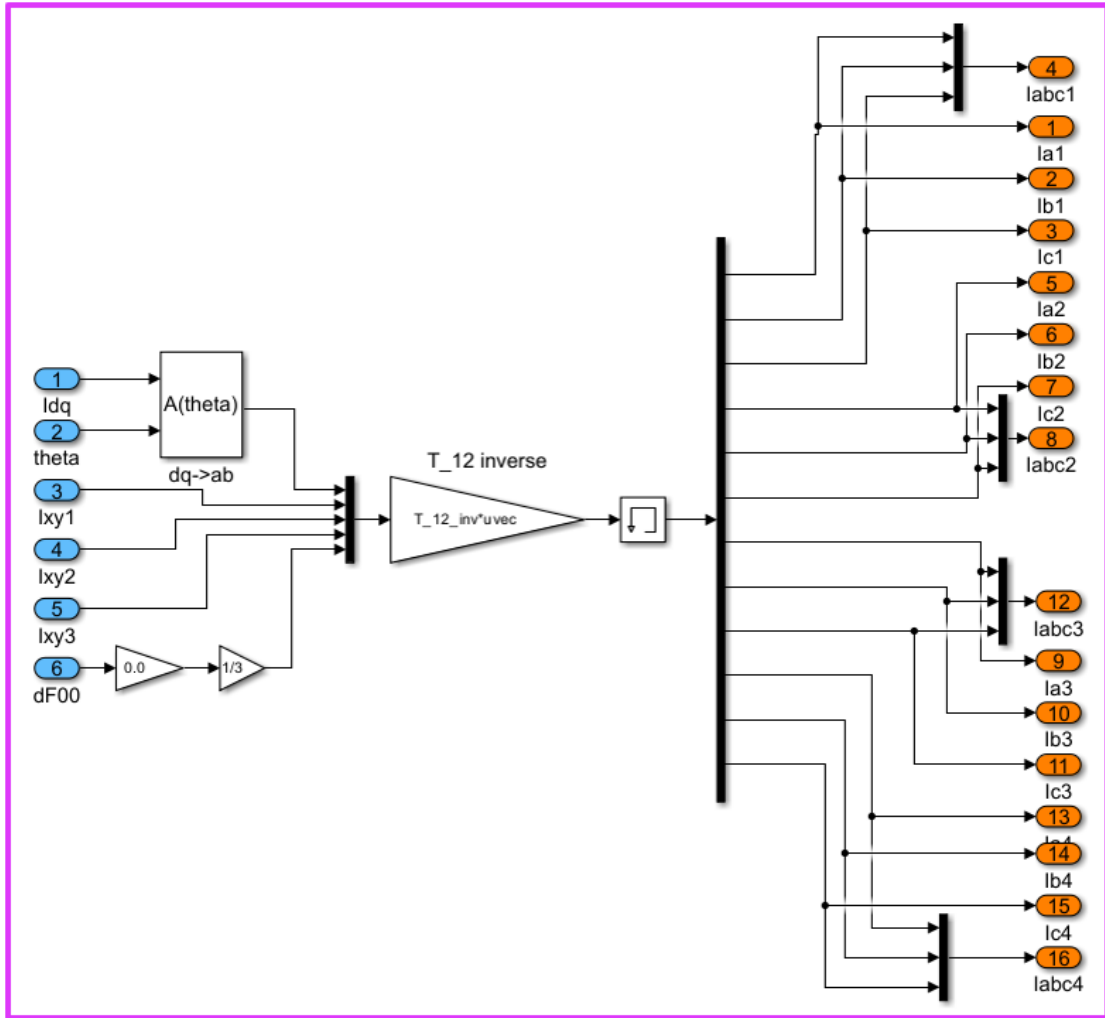


Figure 3.32: "Currents calculation" block, in which the twelvephase currents are computed.

The last block to show is the control code adopted for the twelvephase motor. Before it was introduced that the twelvephase motor control strategy is based on the same principle of the sixphase one. For this reason, only the code will be shown, since only the variables number is changed and the scheme is described in the theory chapter in Fig.1.16.

Listing 3.15: C-language cose adopted for the twelvephase motor. The commissioning part was neglected and the speed computation too. The latter has been already described in the previous sections.

```

1 //-----Control Type -----//
2
3
4     switch (Ctrl_type){
5
6         case 0: //CurrentControl
7             isdq_ref_cm.d = isdq_ext.d; //External reference
8             isdq_ref_cm.q = isdq_ext.q; //External reference
9             break;
10
11         case 2: //TorqueControl
12             ReadLut(&ID_REF[0], fabs(T_ext), TMAX, TMIN, DT,
13             INV_DT, &isdq_ref_cm.d);
14             ReadLut(&IQ_REF[0], fabs(T_ext), TMAX, TMIN, DT,
15             INV_DT, &isdq_ref_cm.q);
16
17             switch (Quad_Maps){
18                 case 0:
19                     isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
20                     break;
21
22                 case 1:
23                     isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
24                     break;
25
26                 case 2:
27                     isdq_ref_cm.q = sgn(T_ext)*isdq_ref_cm.q;
28                     break;
29             }
30
31             break;
32
33         case 3: //SpeedControl
34             omega_ref_in = n_ref_in * RPM2RAD;
35             ramp(omega_ref_in, accel * RPM2RAD*Ts, &
36             omega_ref_ramp);
37
38             sp_var.ref = omega_ref_ramp;
39             sp_var.fbk = omega_elt/PP;
40             sp_par.lim = T_rated;
41             kp_w = 2*OMEGA_BW*J;
42             ki_w = pow(OMEGA_BW,2)*J;
43             sp_par.ki = ki_w*Ts;
44             sp_par.kp = kp_w;
45             PIReg(&sp_par, &sp_var);

```

```

41         T_ext          = sp_var.out;
42
43         ReadLut(&ID_REF[0], fabs(T_ext), TMAX, TMIN, DT,
44 INV_DT, &isdq_ref_cm.d);
45         ReadLut(&IQ_REF[0], fabs(T_ext), TMAX, TMIN, DT,
46 INV_DT, &isdq_ref_cm.q);
47
48         switch (Quad_Maps){
49             case 0:
50                 isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
51                 break;
52             case 1:
53                 isdq_ref_cm.d = sgn(T_ext)*isdq_ref_cm.d;
54                 break;
55             case 2:
56                 isdq_ref_cm.q = sgn(T_ext)*isdq_ref_cm.q;
57                 break;
58         }
59         break;
60     }
61
62     isdq_ref_dm1.d=0;
63     isdq_ref_dm1.q=0;
64     isdq_ref_dm2.d=0;
65     isdq_ref_dm2.q=0;
66     isdq_ref_dm3.d=0;
67     isdq_ref_dm3.q=0;
68
69     //Clarke transformation for each threephase star
70     _clarke1(isabc1, isab1);
71     _clarke2(isabc2, isab2);
72     _clarke3(isabc3, isab3);
73     _clarke4(isabc4, isab4);
74
75     //Rotational transformation for each threephase star
76     _rot(isab1, SinCos_elt, isdq1);
77     _rot(isab2, SinCos_elt, isdq2);
78     _rot(isab3, SinCos_elt, isdq3);
79     _rot(isab4, SinCos_elt, isdq4);
80
81     //Computation of common and differential mode using the
82     rotated feedback quantities
83     is_cm.d= (isdq1.d+isdq2.d+isdq3.d+isdq4.d)*0.25;
84     is_cm.q= (isdq1.q+isdq2.q+isdq3.q+isdq4.q)*0.25;
85     is_dm1.d= (isdq1.d-isdq2.d)*0.5;
86     is_dm1.q= (isdq1.q-isdq2.q)*0.5;
87     is_dm2.d= (isdq2.d-isdq3.d)*0.5;

```



```

87         is_dm2.q= (isdq2.q-isdq3.q)*0.5;
88         is_dm3.d= (isdq3.d-isdq4.d)*0.5;
89         is_dm3.q= (isdq3.q-isdq4.q)*0.5;
90
91         //Saving previous reference voltages for the flux-
observer
92         vsab_km1_1 = vsab_ref1;
93         vsab_km1_2= vsab_ref2;
94         vsab_km12_3 = vsab_ref3;
95         vsab_km13_4 = vsab_ref4;
96
97         FluxObserver();
98
99         //—————Current Vector Control—————//
100
101         //Setting PI parameters
102         //D-axis
103         kp_cmd = OMEGA_BI*Ld_inic;
104         ki_cmd =OMEGA_BI*RS;
105         kp_dmd = OMEGA_BI*L_sigma;
106         ki_dmd = OMEGA_BI*RS;
107         //Q-axis
108         kp_cmq = OMEGA_BI*Lq_inic;
109         ki_cmq = OMEGA_BI*RS;
110         kp_dmq = OMEGA_BI*L_sigma;
111         ki_dmq = OMEGA_BI*RS;
112
113         id_par1.kp    = kp_cmd;
114         id_par1.ki    = ki_cmd*Ts;
115         iq_par1.kp    = kp_cmq;
116         iq_par1.ki    = ki_cmq*Ts;
117         id_par2.kp    = kp_dmd;
118         id_par2.ki    = ki_dmd*Ts;
119         iq_par2.kp    = kp_dmq;
120         iq_par2.ki    = ki_dmq*Ts;
121         id_par3.kp    = kp_dmd;
122         id_par3.ki    = ki_dmd*Ts;
123         iq_par3.kp    = kp_dmq;
124         iq_par3.ki    = ki_dmq*Ts;
125         id_par4.kp    = kp_dmd;
126         id_par4.ki    = ki_dmd*Ts;
127         iq_par4.kp    = kp_dmq;
128         iq_par4.ki    = ki_dmq*Ts;
129
130         //Current loop
131         Current_loop(vdc, Imax_mot, isdq_ref_cm, is_cm, &id_par1, &
id_var1, &iq_par1, &iq_var1, &vsdq_cm_ref);
132         Current_loop(vdc, Imax_mot, isdq_ref_dm1, is_dm1, &
id_par2, &id_var2, &iq_par2, &iq_var2, &vsdq_dm_ref1);

```

```

133     Current_loop(vdc, Imax_mot, isdq_ref_dm2, is_dm2, &
134     id_par3, &id_var3, &iq_par3, &iq_var3, &vsdq_dm_ref2);
135     Current_loop(vdc, Imax_mot, isdq_ref_dm3, is_dm3, &
136     id_par4, &id_var4, &iq_par4, &iq_var4, &vsdq_dm_ref3);
137
138     //Feed-forward
139     vsdq_cm_ref.d += RS*is_cm.d - omega_elt*lambda_dq.q;
140     vsdq_cm_ref.q += RS*is_cm.q + omega_elt*lambda_dq.d;
141
142     vsdq_dm_ref1.d += (RS*is_dm1.d) - omega_elt*L_sigma*
143     is_dm1.q;
144     vsdq_dm_ref1.q += (RS*is_dm1.q) + omega_elt*L_sigma*is_dm1.d;
145
146     vsdq_dm_ref2.d += (RS*is_dm2.d) - omega_elt*L_sigma*
147     is_dm2.q;
148     vsdq_dm_ref2.q += (RS*is_dm2.q) + omega_elt*L_sigma*is_dm2.d;
149
150     vsdq_dm_ref3.d += (RS*is_dm3.d) - omega_elt*L_sigma*
151     is_dm3.q;
152     vsdq_dm_ref3.q += (RS*is_dm3.q) + omega_elt*L_sigma*is_dm3.d;
153
154     //Decoupling: reference voltages to compute duty-cycles
155     are obtained
156     vsdq_ref1.d = vsdq_cm_ref.d + 1.5f*vsdq_dm_ref1.d + 1.0f*
157     vsdq_dm_ref2.d + 0.5f*vsdq_dm_ref3.d;
158     vsdq_ref1.q = vsdq_cm_ref.q + 1.5f*vsdq_dm_ref1.q + 1.0f*
159     vsdq_dm_ref2.q + 0.5f*vsdq_dm_ref3.q;
160
161     vsdq_ref2.d = vsdq_cm_ref.d - 0.5f*vsdq_dm_ref1.d + 1.0f*
162     vsdq_dm_ref2.d + 0.5f*vsdq_dm_ref3.d;
163     vsdq_ref2.q = vsdq_cm_ref.q - 0.5f*vsdq_dm_ref1.q + 1.0f*
164     vsdq_dm_ref2.q + 0.5f*vsdq_dm_ref3.q;
165
166     vsdq_ref3.d = vsdq_cm_ref.d - 0.5f*vsdq_dm_ref1.d - 1.0f*
167     vsdq_dm_ref2.d + 0.5f*vsdq_dm_ref3.d;
168     vsdq_ref3.q = vsdq_cm_ref.q - 0.5f*vsdq_dm_ref1.q - 1.0f*
169     vsdq_dm_ref2.q + 0.5f*vsdq_dm_ref3.q;
170
171     vsdq_ref4.d = vsdq_cm_ref.d - 0.5f*vsdq_dm_ref1.d - 1.0f*
172     vsdq_dm_ref2.d - 1.5f*vsdq_dm_ref3.d;
173     vsdq_ref4.q = vsdq_cm_ref.q - 0.5f*vsdq_dm_ref1.q - 1.0f*
174     vsdq_dm_ref2.q - 1.5f*vsdq_dm_ref3.q;
175
176     //Phase advance
177     dTheta = 1.5f*omega_elt*Ts;
178     SinCos_elt_dTheta.sin = SinCos_elt.sin*cosf(dTheta) +
179     SinCos_elt.cos*sinf(dTheta);

```

```

166         SinCos_elt_dTheta.cos = SinCos_elt.cos*cosf(dTheta) -
SinCos_elt.sin*sinf(dTheta);
167
168         //Duty-cycles computation rotating the (d,q) reference
voltages in the realted (a,b,c) plane
169         _invrot(vsdq_ref, SinCos_elt_dTheta, vsab_ref);
170         _invclarke1(vsab_ref1, vsabc_ref1);
171         PWMduty(vsabc_ref1, vdc, &duty_abc1, deadtime, isabc1);
172
173         _invrot(vsdq_ref2, SinCos_elt_dTheta, vsab_ref2);
174         _invclarke2(vsab_ref2, vsabc_ref2);
175         PWMduty(vsabc_ref2, vdc, &duty_abc2, deadtime, isabc2);
176
177         _invrot(vsdq_ref3, SinCos_elt_dTheta, vsab_ref3);
178         _invclarke3(vsab_ref3, vsabc_ref3);
179         PWMduty(vsabc_ref3, vdc, &duty_abc3, deadtime, isabc3);
180
181         _invrot(vsdq_ref4, SinCos_elt_dTheta, vsab_ref4);
182         _invclarke4(vsab_ref4, vsabc_ref4);
183         PWMduty(vsabc_ref4, vdc, &duty_abc4, deadtime, isabc4);
184
185     }
186
187     //duty cycles saturation
188     if (duty_abc1.a > 0.99f) duty_abc1.a=0.99f;
189     if (duty_abc1.b > 0.99f) duty_abc1.b=0.99f;
190     if (duty_abc1.c > 0.99f) duty_abc1.c=0.99f;
191
192     if (duty_abc1.a < 0.01f) duty_abc1.a=0.01f;
193     if (duty_abc1.b < 0.01f) duty_abc1.b=0.01f;
194     if (duty_abc1.c < 0.01f) duty_abc1.c=0.01f;
195
196     if (duty_abc2.a > 0.99f) duty_abc2.a=0.99f;
197     if (duty_abc2.b > 0.99f) duty_abc2.b=0.99f;
198     if (duty_abc2.c > 0.99f) duty_abc2.c=0.99f;
199
200     if (duty_abc2.a < 0.01f) duty_abc2.a=0.01f;
201     if (duty_abc2.b < 0.01f) duty_abc2.b=0.01f;
202     if (duty_abc2.c < 0.01f) duty_abc2.c=0.01f;
203
204     if (duty_abc3.a > 0.99f) duty_abc3.a=0.99f;
205     if (duty_abc3.b > 0.99f) duty_abc3.b=0.99f;
206     if (duty_abc3.c > 0.99f) duty_abc3.c=0.99f;
207
208     if (duty_abc3.a < 0.01f) duty_abc3.a=0.01f;
209     if (duty_abc3.b < 0.01f) duty_abc3.b=0.01f;
210     if (duty_abc3.c < 0.01f) duty_abc3.c=0.01f;
211
212     if (duty_abc4.a > 0.99f) duty_abc4.a=0.99f;

```

```

213     if (duty_abc4.b > 0.99f) duty_abc4.b=0.99f;
214     if (duty_abc4.c > 0.99f) duty_abc4.c=0.99f;
215
216     if (duty_abc4.a < 0.01f) duty_abc4.a=0.01f;
217     if (duty_abc4.b < 0.01f) duty_abc4.b=0.01f;
218     if (duty_abc4.c < 0.01f) duty_abc4.c=0.01f;
219
220     //Safe control
221     CurrentProtection(isabc1, &State, &pwm_stop1);
222     CurrentProtection(isabc2, &State, &pwm_stop2);
223     CurrentProtection(isabc3, &State, &pwm_stop3);
224     CurrentProtection(isabc4, &State, &pwm_stop4);
225
226     if(pwm_stop1){
227         duty_abc1.a=0.0f;
228         duty_abc1.b=0.0f;
229         duty_abc1.c=0.0f;}
230     if(pwm_stop2){
231         duty_abc2.a=0.0f;
232         duty_abc2.b=0.0f;
233         duty_abc2.c=0.0f;}
234     if(pwm_stop3){
235         duty_abc3.a=0.0f;
236         duty_abc3.b=0.0f;
237         duty_abc3.c=0.0f;}
238     if(pwm_stop4){
239         duty_abc.a=0.0f;
240         duty_abc.b=0.0f;
241         duty_abc.c=0.0f;}

```

While the flux-observer function is below in listing 3.16.

Listing 3.16: "FluxObserver" function for a twelvephase system.

```

1 void FluxObserver(void) {
2
3     //Saving the lambda estimated in step (k-1)
4     lambda_CM_ab_kml = lambda_CM_ab;
5
6     switch (Quad_Maps){
7     case 0:
8         ReadLut2d(&FD_LUT[0][0], fabs(is_cm.d), fabs(is_cm.q),
9         DIDD, INV_DIDD, DIQD, INV_DIQD, ID_TAB_MAX, ID_TAB_MIN,
10        IQ_TAB_MAX, IQ_TAB_MIN, n_size, &lambda_CM_dq.d);
11        ReadLut2d(&FQ_LUT[0][0], fabs(is_cm.q), fabs(is_cm.d),
12        DIQQ, INV_DIQQ, DIQD, INV_DIQD, IQ_TAB_MAX, IQ_TAB_MIN,
13        ID_TAB_MAX, ID_TAB_MIN, n_size, &lambda_CM_dq.q);
14        if (isdq.d < 0)
15            lambda_CM_dq.d = -lambda_CM_dq.d;
16        if (isdq.q < 0)

```

```

13         lambda_CM_dq.q = -lambda_CM_dq.q;
14     break;
15
16     case 1:
17         ReadLut2d(&FD_LUT[0][0], fabs(is_cm.d), is_cm.q, DIDD,
18         INV_DIDD, DIQD, INV_DIQD, ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX,
19         IQ_TAB_MIN, n_size, &lambda_CM_dq.d);
20         ReadLut2d(&FQ_LUT[0][0], is_cm.q, fabs(is_cm.d), DIQQ,
21         INV_DIQQ, DIDQ, INV_DIDQ, IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX,
22         ID_TAB_MIN, n_size, &lambda_CM_dq.q);
23         if (isdq.d < 0)
24             lambda_CM_dq.d = -lambda_CM_dq.d;
25     break;
26
27     case 2:
28         ReadLut2d(&FD_LUT[0][0], is_cm.d, fabs(is_cm.q), DIDD,
29         INV_DIDD, DIQD, INV_DIQD, ID_TAB_MAX, ID_TAB_MIN, IQ_TAB_MAX,
30         IQ_TAB_MIN, n_size, &lambda_CM_dq.d);
31         ReadLut2d(&FQ_LUT[0][0], fabs(is_cm.q), is_cm.d, DIQQ,
32         INV_DIQQ, DIDQ, INV_DIDQ, IQ_TAB_MAX, IQ_TAB_MIN, ID_TAB_MAX,
33         ID_TAB_MIN, n_size, &lambda_CM_dq.q);
34         if (is_cm.q < 0)
35             lambda_CM_dq.q = -lambda_CM_dq.q;
36     break;
37 }
38
39 //Rotating (d,q) lambda for the next step
40 _invrot(lambda_CM_dq, SinCos_elt_dTheta, lambda_CM_ab);
41
42 //Feedback calculation: they are the difference between the
43 estimated flux and the observed one
44 feedback_OBS1.alpha = lambda_CM_ab_km1.alpha -
45 lambda_OBS1.alpha;
46 feedback_OBS1.beta = lambda_CM_ab_km1.beta -
47 lambda_OBS1.beta;
48
49 feedback_OBS2.alpha = lambda_CM_ab_km1.alpha -
50 lambda_OBS2.alpha;
51 feedback_OBS2.beta = lambda_CM_ab_km1.beta -
52 lambda_OBS2.beta;
53
54 feedback_OBS3.alpha = lambda_CM_ab_km1.alpha -
55 lambda_OBS3.alpha;
56 feedback_OBS3.beta = lambda_CM_ab_km1.beta -
57 lambda_OBS3.beta;
58
59 feedback_OBS4.alpha = lambda_CM_ab_km1.alpha -
60 lambda_OBS4.alpha;

```

```

45     feedback_OBS4.beta      =   lambda_CM_ab_km1.beta -
lambda_OBS4.beta ;
46
47     // Integration
48     lambda_OBS1.alpha   += Ts*(vsab_km1_1.alpha - RS*isab1.alpha +
KOBS*feedback_OBS1.alpha) ;
49     lambda_OBS1.beta   += Ts*(vsab_km1_1.beta - RS*isab1.beta +
KOBS*feedback_OBS1.beta) ;
50     lambda_OBS1.amp     = sqrt(pow(lambda_OBS1.alpha,2) + pow(
lambda_OBS1.beta,2)) ;
51
52     lambda_OBS2.alpha   += Ts*(vsab_km1_2.alpha - RS*isab2.alpha
+ KOBS*feedback_OBS2.alpha) ;
53     lambda_OBS2.beta   += Ts*(vsab_km1_2.beta - RS*isab2.beta +
KOBS*feedback_OBS2.beta) ;
54     lambda_OBS2.amp     = sqrt(pow(lambda_OBS2.alpha,2) + pow(
lambda_OBS2.beta,2)) ;
55
56     lambda_OBS3.alpha   += Ts*(vsab_km1_3.alpha - RS*isab3.alpha
+ KOBS*feedback_OBS3.alpha) ;
57     lambda_OBS3.beta   += Ts*(vsab_km1_3.beta - RS*isab3.beta +
KOBS*feedback_OBS3.beta) ;
58     lambda_OBS3.amp     = sqrt(pow(lambda_OBS3.alpha,2) + pow(
lambda_OBS3.beta,2)) ;
59
60     lambda_OBS4.alpha   += Ts*(vsab_km1_4.alpha - RS*isab4.alpha
+ KOBS*feedback_OBS4.alpha) ;
61     lambda_OBS4.beta   += Ts*(vsab_km1_4.beta - RS*isab4.beta +
KOBS*feedback_OBS4.beta) ;
62     lambda_OBS4.amp     = sqrt(pow(lambda_OBS.alpha,2) + pow(
lambda_OBS4.beta,2)) ;
63
64     //Computation of the average of the four fluxes obtained
before and rotation of it to obtain a single value in (d,q) plane
65     aux_ab.alpha=0.25*(lambda_OBS1.alpha+lambda_OBS2.alpha+
lambda_OBS3.alpha+lambda_OBS4.alpha) ;
66     aux_ab.beta=0.25*(lambda_OBS1.beta+lambda_OBS2.beta+
lambda_OBS3.beta+lambda_OBS4.beta) ;
67
68     _rot(aux_ab, SinCos_elt_dTheta, lambda_dq);
69
70 }

```

3.6.2 Results

In order to have a coherent comparison with the previous versions, the same setting adopted before have been use for the twelvephase too. In table 3.1 the values there are and the physical parameters to use are shown in table 3.3.

The first result shown is the torque, in Fig.3.33, that follows the reference as expected. The dynamics imposed are the same of the previous cases, in order to have a consistent argumentation.

It is clear that the dynamic behaviour is better if compared with the previous motors and the overshoot are lower than threephase machine.

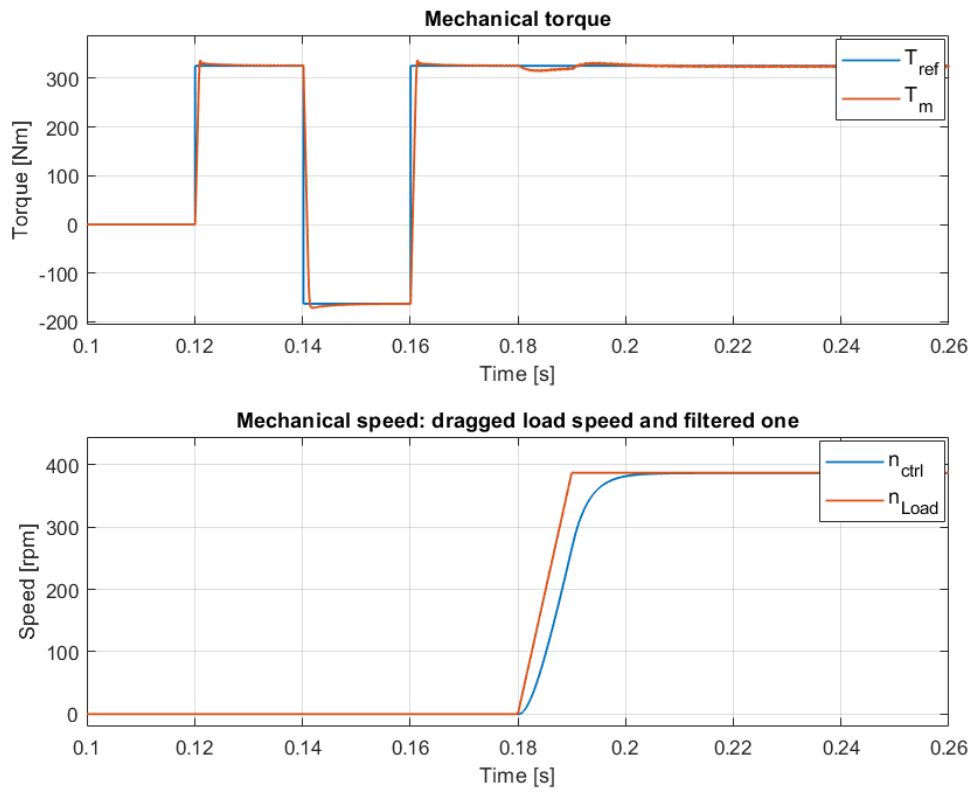


Figure 3.33: From the top to the bottom: motor torque and related reference, dragged load speed and the filtered one with the control.

The second control's check are the duty.cycles, as below, that are shifted of 15° electrical degrees and have the typical shape of the third harmonic injection technique.

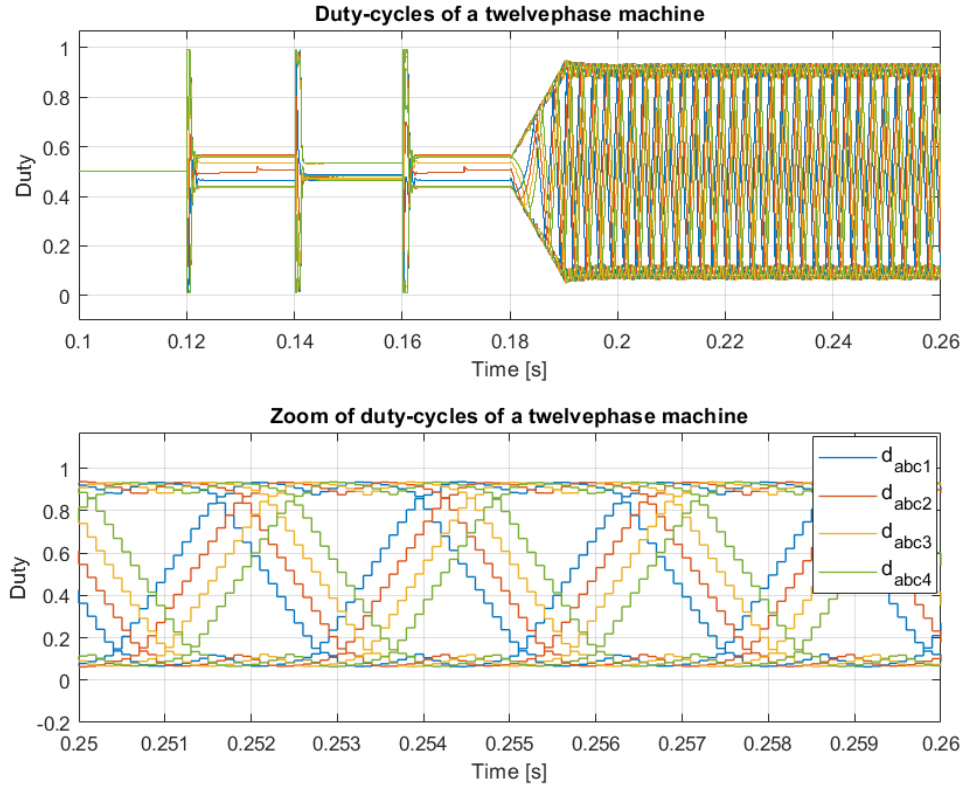


Figure 3.34: Duty-cycles adopted for the twelvephase motor. The third harmonic injection is recognizable. The second graph shows with the same color the three duty-cycles related to the same threephase system.

Then, the currents can be shown in Fig.3.35. The typical displacement of the asymmetrical twelvephase machine is evidenced and the system is balanced as expected, so the control does not introduces any unwanted harmonic behaviour.

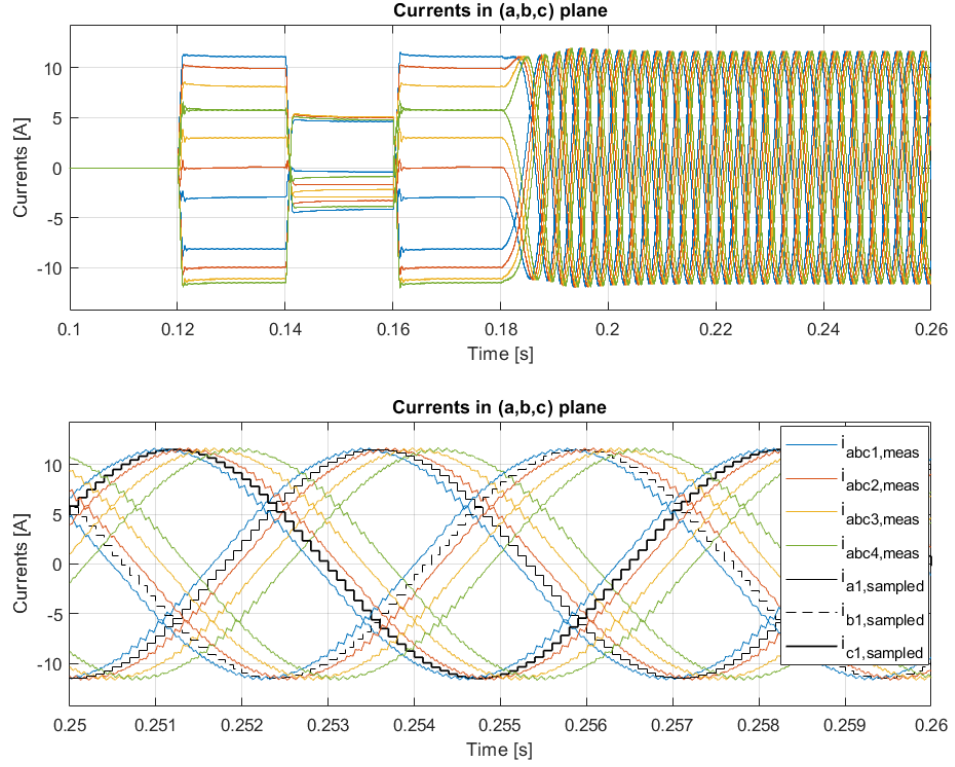


Figure 3.35: Currents of twelvephase motor with the typical displacement of 15° electrical degrees of an asymmetrical machine. The currents related to the same star are have been represented with the same color. The second graph shows also the sampling phenomenon delay for one threephase system only, for reasons of graphic clarity.

The final check about the control code in listing 3.15 is the rotational transformation related with the current loop. In Fig.3.36 there is the proof that the dynamic behaviour is good and that the integral part of the controller works well.

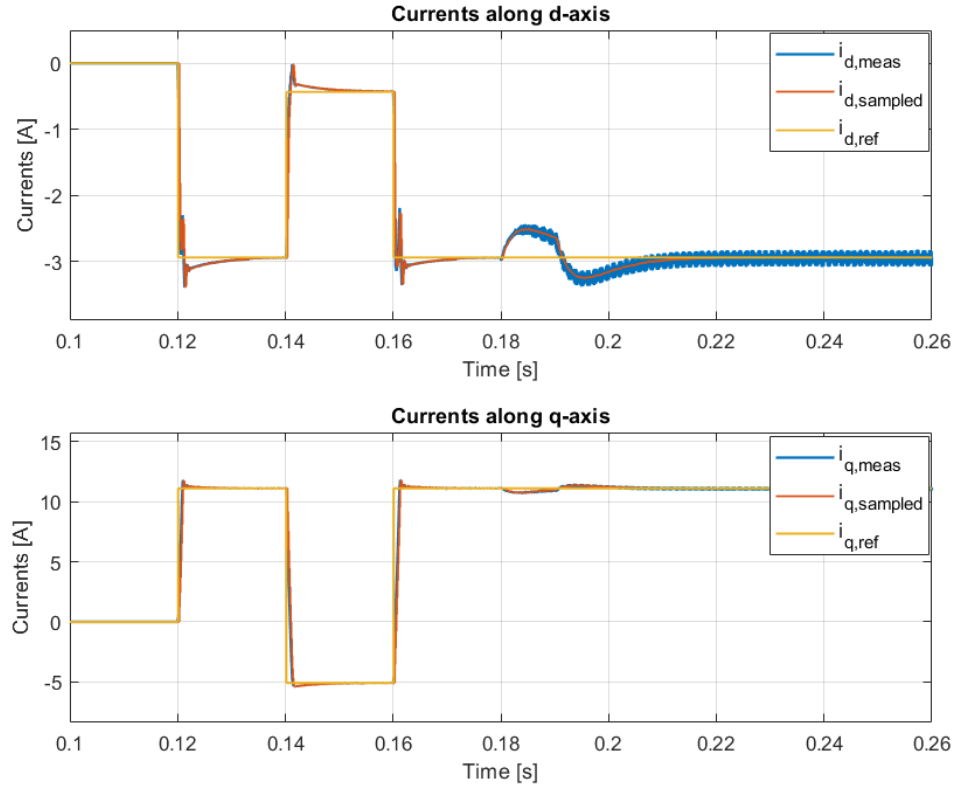


Figure 3.36: From the top to the bottom: physical current, sampled one and its reference along d-axis and the same quantities for the q-axis.

At the end of this section the flux-observer algorithm is shown, in order to verify if the scheme in 1.18 works even for a twelvephase machine. Below the comparison between the real flux measured thanks to the simulation ambient and the observed one is present.

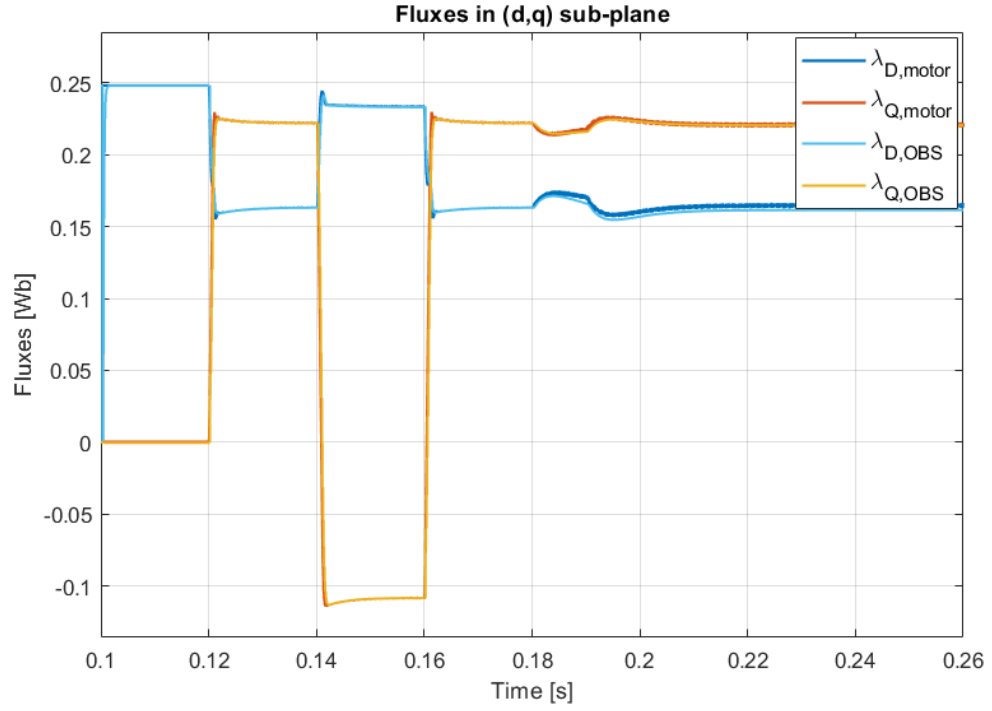


Figure 3.37: From the top to the bottom: physical flux and observed one along d-axis and the same quantities for the q-axis.

Chapter 4

Conclusions

At the end of this intensive work, several interesting conclusions can be drawn. A three-phase ETEL motor was reconfigured to sixphase and twelvephase versions without any modifications to its active parts, but by only working on the windings connection and control code.

FEM analysis was used to study the machine and the multi-threephase Syre's tool was independently verified with a FEMM drawing. Then, a method to modify each multi-threephase motor has been proposed and checked.

Furthermore, a common-differential mode control was merged with a MS approach, utilizing the VSD technique to represent the behaviour of the motor. Additionally, a decoupling method was suggested for the four-threephase system configuration, which is valid only for safe working conditions.

It was highlighted that as the number of three-phase systems increased, the torque slightly increased, and the torque ripple was drastically reduced as expected.

In addition, with the higher number of three-phase systems, a greater speed range was available while maintaining the same efficiency value.

Regarding the thermal aspects, multi-three-phase solutions had fewer iron losses due to the lower harmonic disturbance. Besides, the higher modularity present in sixphase and twelvephase solutions allowed to have a better thermal control, without concentrating the power electronics heating flow in a specific area.

Finally, with the new motors, a higher power factor and better efficiency level were achieved at the rated working point.

The control code was checked under high stress dynamically, with step references and rapid speed ramps.

This work demonstrates that proposing a multi-three-phase solution starting from a standard one provides advantages in terms of performance from all perspectives and facilitates the integration of everything into one solution, including the motor and power electronics.

Further points of improvement for the future could be:

- conducting real tests with a prototype motor on a benchmark.
- analyzing the fault working conditions by studying the operation with one or more stars turned off to highlight possible advantages in the working area.
- analyzing the multi-three-phase motor with a 3D CAD to explore additional three-dimensional aspects.
- continuing to study multi-three-phase solutions (as it was done during the internship period but not shown in this work) in which the windings are the same as the three-phase configuration, using a halved DC-link voltage (or a quarter). This could be useful for customers who need power electronics with high current but lower voltage.

Bibliography

- [1] Radu Bojoi, Sandro Rubino, Alberto Tenconi, and Silvio Vaschetto. «Multi-phase electrical machines and drives: A viable solution for energy generation and transportation electrification». In: *2016 International Conference and Exposition on Electrical and Power Engineering (EPE)*. 2016 International Conference and Exposition on Electrical and Power Engineering (EPE). Iasi, Romania: IEEE, Oct. 2016, pp. 632–639. ISBN: 9781509061297. DOI: 10.1109/ICEPE.2016.7781416. URL: <http://ieeexplore.ieee.org/document/7781416/> (visited on 01/03/2023) (cit. on pp. 1, 2, 4, 5, 7).
- [2] R. Nelson and P. Krause. «Induction Machine Analysis for Arbitrary Displacement Between Multiple Winding Sets». In: *IEEE Transactions on Power Apparatus and Systems* PAS-93.3 (May 1974), pp. 841–848. ISSN: 0018-9510. DOI: 10.1109/TPAS.1974.293983. URL: <http://ieeexplore.ieee.org/document/4075429/> (visited on 01/03/2023) (cit. on p. 1).
- [3] M. Barcaro, N. Bianchi, and F. Magnussen. «Faulty Operations of a PM Fractional-Slot Machine With a Dual Three-Phase Winding». In: *IEEE Transactions on Industrial Electronics* 58.9 (Sept. 2011), pp. 3825–3832. ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2087300 (cit. on pp. 1, 7).
- [4] R. Bojoi, A. Cavagnino, A. Tenconi, A. Tassarolo, and S. Vaschetto. «Multi-phase electrical machines and drives in the transportation electrification». In: *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. 2015, pp. 205–212. DOI: 10.1109/RTSI.2015.7325099 (cit. on pp. 1, 5, 7).
- [5] G. Pellegrino, A. Vagati, P. Guglielmi, and B. Boazzo. «Performance Comparison Between Surface-Mounted and Interior PM Motor Drives for Electric Vehicle Application». In: *IEEE Transactions on Industrial Electronics* 59.2 (Feb. 2012), pp. 803–811. ISSN: 0278-0046. DOI: 10.1109/TIE.2011.2151825 (cit. on p. 1).

- [6] M. Barcaro, N. Bianchi, and F. Magnussen. «Analysis and Tests of a Dual Three-Phase 12-Slot 10-Pole Permanent-Magnet Motor». In: *IEEE Transactions on Industry Applications* 46.6 (Nov. 2010), pp. 2355–2362. ISSN: 0093-9994. DOI: 10.1109/TIA.2010.2070784 (cit. on pp. 2, 6).
- [7] Nicola Bianchi, Michele Dai Pré, Luigi Alberti, and Emanuele Fornasiero. *Theory and Design of Fractional-Slot PM Machines*. IEEE IAS Tutorial Course notes, IAS'07 Annual Meeting. (info@cleup.it). CLEUP, Padova (Italy), New Orleans, LA, September 23, 2007. ISBN: 978-88-6129-122-5 (cit. on p. 2).
- [8] M. Barcaro, L. Alberti, and N. Bianchi. «Thermal analysis of dual three-phase machines under faulty operations». In: *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics Drives (SDEMPED)*, 2011. Sept. 2011, pp. 165–171. DOI: 10.1109/SDEMPED.2011.6063619 (cit. on p. 2).
- [9] N. Bianchi. *Electrical Machine Analysis Using Finite Elements*. Power Electronics and Applications Series. CRC Press, 2017. ISBN: 9781420057874. URL: <https://books.google.de/books?id=n5fMBQAAQBAJ> (cit. on p. 2).
- [10] E. Fornasiero. «Advanced Design of Direct Drive PM Machines». (pdf). Dissertation, 2728. Padova, Italy: University of Padova, 2004 (cit. on p. 2).
- [11] D. Meeker. «www.femm.info/wiki/HomePage». In: (cit. on p. 2).
- [12] Sandro Rubino, Obrad Dordevic, Eric Armando, Iustin Radu Bojoi, and Emil Levi. «A Novel Matrix Transformation for Decoupled Control of Modular Multiphase PMSM Drives». In: *IEEE Transactions on Power Electronics* 36.7 (2021), pp. 8088–8101. DOI: 10.1109/TPEL.2020.3043083 (cit. on pp. 2, 22, 27).
- [13] Yashan Hu, Z. Q. Zhu, and Milijana Odavic. «Comparison of Two-Individual Current Control and Vector Space Decomposition Control for Dual Three-Phase PMSM». In: *IEEE Transactions on Industry Applications* 53.5 (Sept. 2017), pp. 4483–4492. ISSN: 0093-9994, 1939-9367. DOI: 10.1109/TIA.2017.2703682. URL: <http://ieeexplore.ieee.org/document/7926384/> (visited on 01/03/2023) (cit. on pp. 2, 7).
- [14] Sandro Rubino, Radu Bojoi, Emil Levi, and Obrad Dordevic. «Vector Control of Multiple Three-Phase Permanent Magnet Motor Drives». In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society. Washington, DC: IEEE, Oct. 2018, pp. 5866–5871. ISBN: 9781509066841. DOI: 10.1109/IECON.2018.8591146. URL: <https://ieeexplore.ieee.org/document/8591146/> (visited on 01/03/2023) (cit. on p. 2).

- [15] Sandro Rubino, Obrad Dordevic, Radu Bojoi, and Emil Levi. «Modular Vector Control of Multi-Three-Phase Permanent Magnet Synchronous Motors». In: *IEEE Transactions on Industrial Electronics* 68.10 (Oct. 2021), pp. 9136–9147. ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2020.3026271. URL: <https://ieeexplore.ieee.org/document/9209077/> (visited on 01/03/2023) (cit. on p. 2).
- [16] Yifan Zhao and T.A. Lipo. «Space vector PWM control of dual three-phase induction machine using vector space decomposition». In: *IEEE Transactions on Industry Applications* 31.5 (Oct. 1995), pp. 1100–1109. ISSN: 00939994. DOI: 10.1109/28.464525. URL: <http://ieeexplore.ieee.org/document/464525/> (visited on 01/03/2023) (cit. on p. 2).
- [17] Eric Armando, Radu Iustin Bojoi, Paolo Guglielmi, Gianmario Pellegrino, and Michele Pastorelli. «Experimental Identification of the Magnetic Model of Synchronous Machines». In: *IEEE Transactions on Industry Applications* 49.5 (2013), pp. 2116–2125. DOI: 10.1109/TIA.2013.2258876 (cit. on pp. 2, 21).
- [18] J. Karttunen, S. Kallio, P. Peltoniemi, P. Silventoinen, and O. Pyrhönen. «Dual three-phase permanent magnet synchronous machine supplied by two independent voltage source inverters». In: *Automation and Motion International Symposium on Power Electronics Power Electronics, Electrical Drives*. Automation and Motion International Symposium on Power Electronics Power Electronics, Electrical Drives. June 2012, pp. 741–747. DOI: 10.1109/SPEEDAM.2012.6264448 (cit. on p. 3).
- [19] Simone Ferrari, Paolo Ragazzo, Gaetano Dilevrano, and Gianmario Pellegrino. «Flux and Loss Map Based Evaluation of the Efficiency Map of Synchronous Machines». In: *IEEE Transactions on Industry Applications* (2022), pp. 1–10. ISSN: 0093-9994, 1939-9367. DOI: 10.1109/TIA.2022.3221381. URL: <https://ieeexplore.ieee.org/document/9944887/> (visited on 02/22/2023) (cit. on pp. 50, 63).