POLITECNICO DI TORINO

Master of Science-level of the Bologna Process in
Mechatronic Engineering

Final Project Work

Model - Based Design of a Control Algorithm for ePowertrain

Advisors
Prof. Massimo VIOLANTE
Prof. Iustin Radu BOJOI

Candidate
Filippo SANTONOCITO
Student ID: 242743

DECEMBER 2019

To my family

Acknowledgements

I cannot begin to express my thanks to Prof. Violante and Prof. Bojoi for giving me the opportunity to work on this thesis, taking on this topic which is, nowadays, a very current subject in the engineering field.

I am also grateful to the people of the Enertronics Laboratory for hosting and supporting me during the last thesis period and the testing phases.

This thesis would not have been possible without my family, without their economic and moral support, without their love and advices.

I would also like to extend my deepest gratitude to Team DIANA and all its members for the days and the nights spent in the laboratory working on the rovers. For sure, this has been one of most incredible experiences I have ever done in my life so far, and I will always be thankful for what I have learned from both technical and human aspects.

Thanks should also go to all my friends and colleagues who have always encouraged me during my years in Turin.

In closing, I would like to extend my dedication to my grandparents Enza, Filippo, Simone and Teresa, with whom I was not able to spend the last years of their lives, and to who is still waiting for me to come back. I am sure you would have been proud of me.

Summary

With the recent progresses in the power electronic and computer engineering fields, the development of devices able to support high performances control systems has been carried out. One of the main application fields is the automotive industry, in which manufacturers have started to introduce, alongside to internal combustion engines, electric drives creating the basis, first for hybrid vehicles and then for full electric vehicles. To face a more demanding market, new development techniques for control softwares have been created aiming to reduce the time required for algorithm design such as Model-Based Design.

The first idea behind this thesis work has been to compare the performances of a motor control algorithm, developed with two different approaches: the user code C and the autogeneration of code, following the Model-Based design approach, for the target device, a custom board designed by Ideas&Motion s.r.l. These performances have been evaluated with a Hardware-in-the-Loop test, employing a RT Box 1 by Plexim.

The thesis work has been first conducted with a review of the literature concerning DC/AC converters and AC machines and their models, as well as the state-of-the-art of electric drives control and, later, with the motor control algorithm implementation.

In conclusion, results, problems, final considerations and further improvements will be pointed out.

Contents

List of Tables

List of Figures

Abbreviations

List of Abbreviations

- **ADC**: Analog-to-Digital Converter
- **BEM**: Balanced Envelop Modulation
- **BSP**: Board Support Package
- **CSI**: Current Source Inverter
- **DAC**: Digital-to-Analog Converter
- **DFOC**: Direct Field-Oriented Control
- **DFVC**: Direct-Flux Vector Control
- **DPWM**: Discontinuous Pulse Width Modulation
- **EMC**: Electro-Magnetic Coupling
- **EMF**: ElectroMotive Force
- **FOC**: Field-Oriented Control
- **GPOS**: General Purpose Operating System
- **IFOC**: Indirect-Flux Vector Control

- **IGBT**: Insulated Gate Bipolar Transistor
- **IM**: Induction Machine
- **IPM**: Interior Permanent Magnet
- **LUT**: Look-Up Table
- **MBD**: Model-Based Design
- **MOSFET**: Metal Oxide Semiconductor Field Effect Transistor
- **MTPA**: Maximum Torque Per Ampere
- **MTPV**: Macimum Torque Per Voltage
- **PMASR**: PM-Assisted Synchronous Reluctance
- **PMSM**: Permanent Magnet Synchronous Machine
- **PWM**: Pulse Width Modulation
- **RMS**: Root Mean Square
- **RTOS**: Real-Time Operating System
- **SAR**: Successive Approximation Register
- **SOC**: Start Of Conversion
- **SPM**: Surface-mount Permanent Magnet
- **SPWM**: Sinusoidal Pulse Width Modulation
- **SVPWM**: Space Vector Pulse Width Modulation
- **SoC**: System on Chip
- **SyR**: Synchronous Reluctance
- **VSI**: Voltage Source Inverter

Chapter 1

Introduction and Objectives

The recent progresses in the power electronics and computer engineering fields and the following costs' reduction of electronic components allowed a wide spread employment of electronic devices in the market. The advances in the aforementioned field made it possible also to handle increasingly larger voltages and currents and to obtain faster power switches.

As a consequence, the development of more sophisticated electric motor control techniques has been possible. But why is it necessary to control a motor?

Without any control, electric motors can basically operate at constant speed, but applications, as the automotive field, require to be able to control quantities like speed.

A control system can thus compute and provide an adequate control signal to a system which is able to drive an electric motor by comparing, i.e. the target speed (the speed to be achieved) to the current speed measured by a sensor.

Focusing on the automotive market, the electric motors have been first applied to internal combustion engines (ICEs) to reduce fuel consumption, giving start to the first generation of hybrid-electric vehicles (HEV) [1].

The trend in figure 1.1 shows how, nowadays, the more and more electric vehicles are present on the roads and their numbers is going to equal the one of ICE vehicles in the following twenty years.

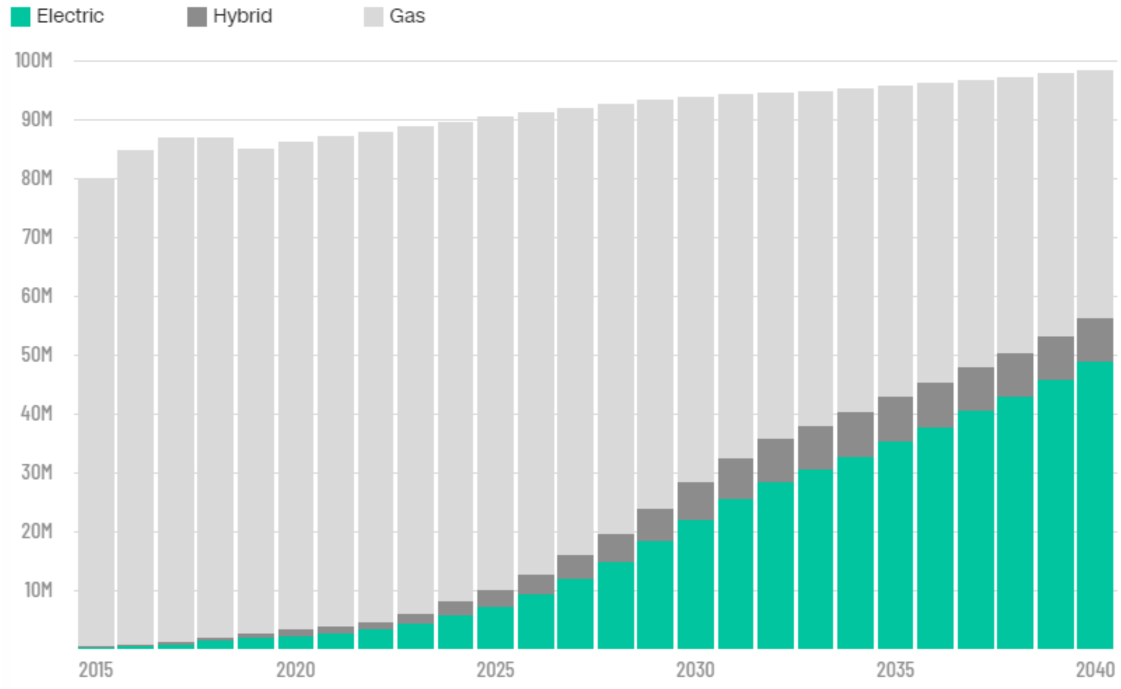


Figure 1.1: EV sales trend (Source: Bloomberg New Energy Finance)

The figure above is proof that a faster development process is required to keep up with the automotive market's changes, thus new strategies shall be adopted to make control techniques as more precise and versatile as possible.

One of the widely used approaches in the automotive field, for control software development is the Model-Based Design (MBD) that will be described in the following chapter, pointing out its advantages and drawbacks.

This method will be employed in the development of the control algorithm for electric powertrain (ePowertrain) in this thesis work.

This approach allows also to auto-generate code for the target device described in Section 4.1 equipped with a microcontroller running a Real-Time Operating System (RTOS).

The previously mentioned algorithm will be implemented in the same device, using the same RTOS, in user-code C.

The goal of this thesis work is thus to compare these two different approaches from different aspects: from the ease of implementation to the control algorithm performances.

The evaluation will be carried out with Hardware-in-the-Loop (HIL) tests using the real-time machine described in section 4.2.

The thesis has been carried out in collaboration with the Power Electronics Innovation Center (PEIC) of the Polytechnic University of Turin, employing a prototype board developed by the company Ideas&Motion s.r.l. (described in section 4.1).

This thesis is structured in different parts: chapter 2 will be dedicated to the literature review, describing and analysing the state-of-the-art of the control algorithm design techniques, with a particular focus on the automotive field and its regulations; later in the same chapter, an overview of the power electronic converters and AC motors modelling will be provided, with a particular focus on synchronous machine and especially interior permanent magnet (IPM) machines since they are mainly employed in traction applications. Then, the main control strategies that will be also used during the testing phases will be described.

Chapter 3 and chapter 4 are dedicated to the description of the technologies used from both software and hardware sides. Particular attention will be dedicated to the Board where the controller will be implemented, describing the employed peripherals and the real-time machine RT Box by Plexim, where the HIL tests will be performed.

Chapter 5 describes all practical aspects of the thesis, focusing on how the tests have been performed, showing and commenting the obtained results.

In conclusion, chapter 6 will point out the differences between the two developed approaches, analysing advantages and drawbacks, and defining the most convenient solution.

Chapter 2

Theoretical Background

In this chapter the key concepts and elements used in this thesis work will be addressed.

It will be presented the idea behind the Model – Based Design analysing both advantages and drawbacks, clarifying what code generation is and the main objective and applications of real-time operating systems.

Through a literature review the different models and theories related to eDrives and their control will be discussed comparing and evaluating the different approaches.

2.1 Model - Based Design and Code Generation

Model-Based Design is an easier and faster approach for the development of control systems, signal processing, communications and other dynamic systems employing models during the development process. It is widely used in all types of industries from aerospace to automotive, from industrial to consumer electronics applications.

Model - Based Design is built on eight essential ideas [2]:

- Executable Specification

  This type of modelling approach is characterised by the fact that it includes all design informations, requirements, system components and also test scenarios. For instance, the model could embed the use cases that the embedded software should face in the real application. This approach brings unambiguity in the informations communication between developers and end users.

- System-Level Simulation

  By employing models, system-level simulation can be performed allowing the simulation of the whole system analysing performances and interactions between the different system parts. One typical example is when a multi-domain model is used, as in the automotive field when vehicle dynamic control systems should be simulated.

- What-If Analysis

  It is a simulation method employed to investigate possible alternative solutions to the ones adopted in the current model, for example focusing on a single component.

- Model Elaboration

  As will be described later, in the dedicated section, when modelling a system, the starting point is always a simpler model based on reproducing the essential characteristics of a system. In order to make the model behaviour closer to the real system, improvements are made, for example adding effects that have been previously neglected. It is worth underlining that improvements may also concern the computational point of view, for instance, converting from continuous time to discrete time.

- Virtual Prototyping

  This technique allows to simulate systems that are not yet available because, for instance, they need to be produced, as it typical happen with mechanical components. Using a model instead of a physical prototype allows to reduce costs and save time.

- Continuous Test and Verification

  By adopting Model-Based Design, it is possible to continuously perform tests to identify faults or errors, before the production of the hardware or the embedded software, thus increasing quality.

- Automation

  Employing Model-Based Design aspects like automation are improved: first of all, Model-Based Design allows the auto generation of code, generation of reports related to design descriptions and test results, prove of code correctness. All these elements make the development process faster and the management of complex systems easier.

- Knowledge Capture and Management

  Since models are developed based on the company knowledge, the team members' past experience and so on and so forth, models can be considered as a common language for knowledge transfer inside a team, improving communication and relationships.

One of the common methodology adopted in Model-Based Design is the V-shaped Development Flow. This organisation is typically used in automotive applications.

The key concept behind the V-model is that to each development phase there is a corresponding testing phase.

On the other side, its main drawback is that the system design is carried out only at the beginning of the project and it often occurs that in complex projects some components may not be known a priori.

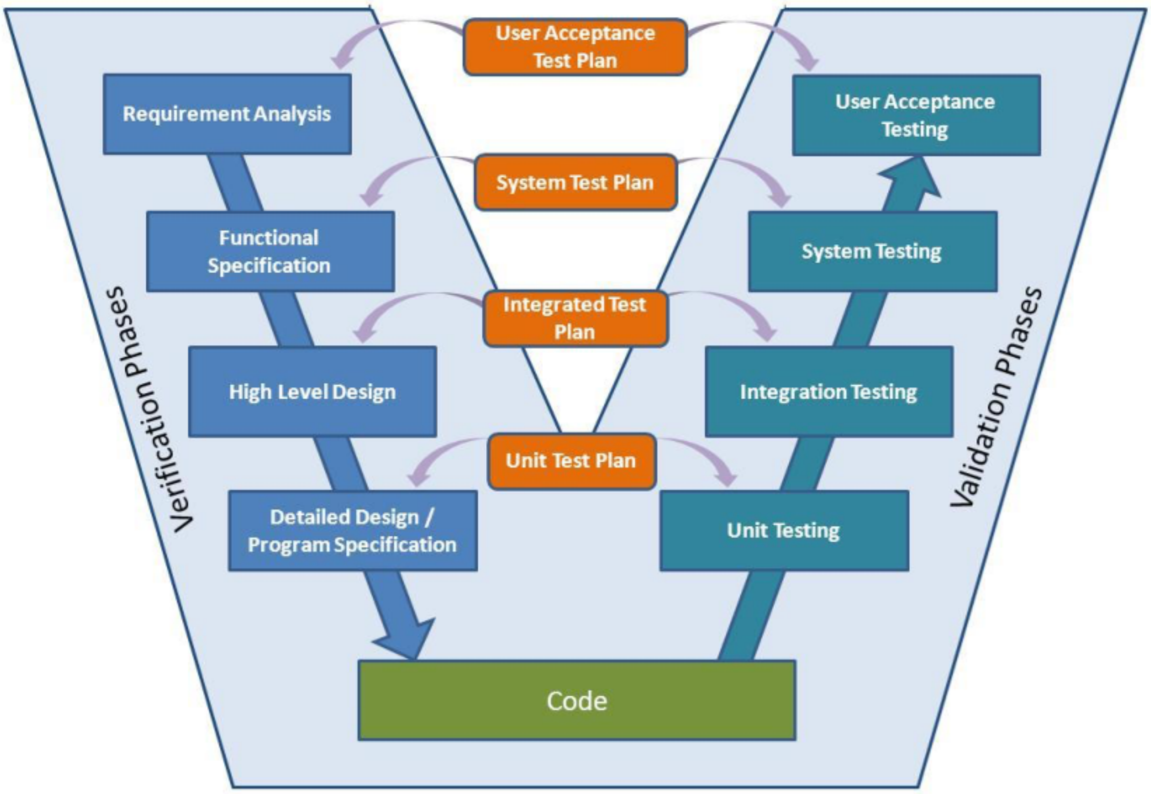


Figure 2.1: V-shaped Development Flow (Source: [3])

2.1.1 Advantages and Drawbacks of Model – Based Design

The main advantages deriving from the use of Model-Based Design are not only related to the technical aspect of an engineering project (as explained in the previous section), but also related to the business aspect.

To measure the value of the Model–Based Design one typical criterion used in the financial sector is the ROI (Return On Investment). It is the ratio between net profit and cost of investment. Basically, it is the measure of the efficiency of an investment relating profits to invested capital. For example, Model–Based Design allows to avoid code failures, and this translates in cost savings.

Model–Based Design also presents drawbacks: one of the main problems is that people may not be used to Model–Based Design approach since usually software developers are used to code in a precise language with the related tools, and to work with blocks and other graphical elements it is not exactly the same. This translates in time to be spent to train developers to get familiar with the new tools.

As an example, the architecture of a model using this approach is different and does not allow the use of objects, for-loops, while-loops as C code does.

It is also worth mentioning the MathWorks Automotive Advisory Board (MAAB), a control algorithm modelling design guide which describes templates and rules to be adopted when using the Model–Based Design approach in the automotive field.

Another reason why Model–Based Design approach may not be the best solution is that when the board support package for the target hardware has to be developed and, in this case, a long-term perspective should be adopted to take advantage of this approach. Along with this problem, it could happen that the code may not reflect perfectly the correct behaviour of the model.

The models characterising the Model–Based Design consist of every component related to the system behaviour (algorithms, control logic, physical components) represented through block diagrams, textual programs (see Stateflow) and other graphical components so that software developers can understand in advance whether the system works testing and verifying the design using simulations (see MIL, SIL and PIL).

Starting from the functional requirements a physical model is developed and then simulated in the modelling framework without using any physical hardware component; this is what is performed in the Model-in-the-Loop testing phase and it allows to perform testing during preliminary development cycle verifying the accuracy of the model (both plant and control system).

When the model development is completed, many outputs can be obtained such as C code, HDL code, as an example, through automatic code generation.

The following step, once the MIL has been successfully completed, is the Software-in-the-Loop (SIL) test to validate the behaviour of the code (auto generated or not) that should be used in the controller by providing appropriate stimuli (input signals) and displaying the results from the model. In this phase still no physical hardware is employed.

Before the Hardware-in-the-Loop, the testing phase requires a Processor-in-the-Loop (PIL) phase, where the model is validated with the code cross–compiled and executed on the target processor. Typically, BSP are used during the SIL and PIL providing target-oriented blocks for the code auto generation.

These tests aim to reveal if the target hardware behaves in the expected way and the code is correctly compiled.

The typical workflow adopted for Model–Based Design is shown in the following picture:

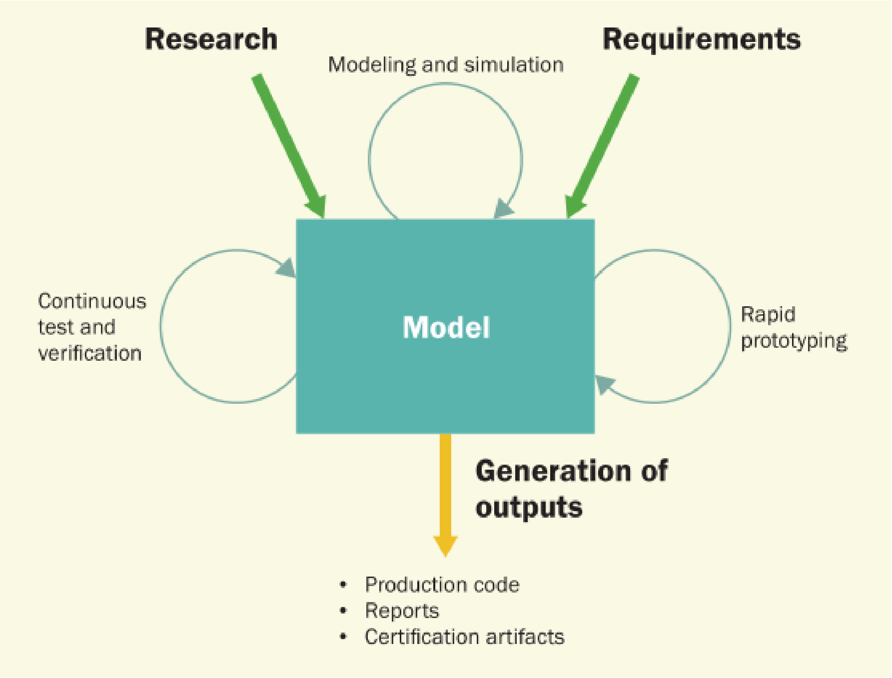


Figure 2.2: Model - Based Design Workflow (Source: [2])

2.1.2 Code Generation

As explained in the previous paragraph, auto generated code is one of the possible outputs of the model developed using the Model–Based Design approach. Code generation is the process of generating low-level code starting from a high-level programming languages or modelling environment such as the one used in MATLAB/Simulink.

Considering a model developed in Simulink, Real-Time Workshop© Embedded Coder is used to generate code in Embedded-C format for embedded processors, where different architectures can be selected.

Starting from the Simulink model, when Real-Time Workshop is executed the build process generates the model description file in the *.rtw* format. This file includes all the information related to the model. Then the *.rtw* file is passed to the Target Language Compiler to convert the model description file into target-specific code. The Target Language Compiler, in addition to the *.rtw* file requires also the *.tlc* target-specific file.

The generated files are the following [4]:

- The body for the generated C source code (*model.c*);
- Header files (*model.h*);
- Header file *mode_private.h* defining parameters and data structures private to the generated code;
- A makefile, *model.mk*, for building the application.

In figure 2.3, the previously described process is shown using a schematic approach:

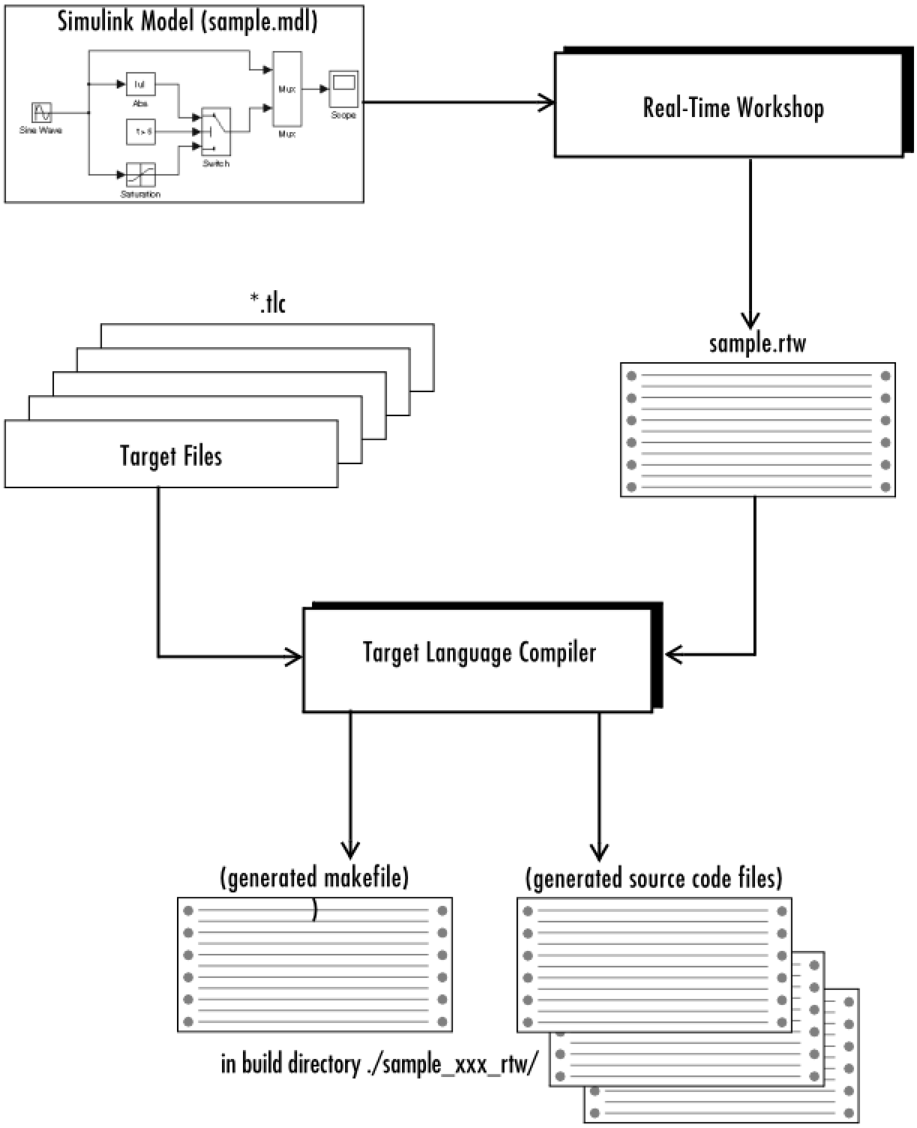


Figure 2.3: The Target Language Compiler Process (Source: [4])

2.2 Real - Time Operating Systems

Operating systems act as an interface between the hardware and the user of a computer. Typical operating systems such as Windows, MacOS, also called General Purpose Operating Systems (GPOS), are non-real-time systems and their purpose is just to correctly react to external events where multiple programs can be executed at the same time.

Considering a single processor it is not possible to talk about simultaneous execution of programs since only a single thread can be executed at the same time. A well-designed scheduler can anyway provide this illusion by immediately switching between the programs in execution.

Safety critical applications (such as automotive, aerospace, railway, defence applications) require an operating system able to react correctly and within a finite and defined period, named deadline. This type of operating systems are called real-time operating systems (RTOS).

The main purpose of a RTOS is to provide a deterministic execution pattern, achieved by using a predictable scheduler. Typically, to each task a priority is assigned determining which should be first executed.

Different scheduling algorithm can be implemented depending on the desired efficiency and processor capabilities.

Three types of real-time operating system can be distinguished [5]:

- Soft real-time, if deadline meeting is desirable just for performance reasons, but missing it will not cause any damage;
- Firm real-time, if deadline meeting will make the result not usable, but still no damage derives from it;
- Hard real-time, when the deadline must be respected otherwise the system performance will fail.

What should be clear is that the purpose of real-time operating systems is not to be fast, which means to minimize the execution time, instead to guarantee the timing behaviour of each process.

What distinguish GPOS and RTOS is also the hardware architecture: typically, GPOS are used on architecture like PC, servers and mainframes, while RTOS are light weight operating systems employed in small architecture such as SoC, microcontrollers and microprocessors or generally embedded systems.

2.3 Power Electronic Converters Modelling

To analyse, control and simulate a system it is necessary to know its physical behaviour obtained from a mathematical knowledge of mass and energy conservation laws, where the energy accumulation variables are identified as state variables [6].

Once the model has been obtained it is possible to simulate converter dynamic behaviour and implement control laws.

Software as MATLAB® can perform precise and reliable simulations of the time-domain behaviour of converters but results are not general. It can simply provide a description based on the input-output variables evolution [6].

Thus, for control purposes it is required an analytical model of the system, based on the knowledge of the physical behaviour of the converter itself.

Many model types can be employed but some general assumptions can be adopted [6]:

- Switches are considered ideal, so zero-value resistance during conduction, or ON state, and infinite-value resistance during the OFF state;
- Generators are assumed to be perfect;
- Passive elements are considered linear and invariant.

These assumptions are made to model only the dominant behaviour of a system, neglecting what are considered insignificant aspects. The elements that have been simplified, may be later introduced to obtain a refined model of the system.

Basing on these approximations, many type of models can be considered, in this treatise only the following will be addressed (with a focus on power electronic converters):

- State-Space Averaged Model [7]

  The state-space average model aims to derive the small-signal averaged equations of PWM switching converters. One of the main advantages of this method is that the obtained results do not lose generality and if the system can be described by state equations, thus a small-signal averaged model can always be derived.

  Writing the state equations of a system means to write the differential equations characterising its behaviour. Considering a linear system, the derivatives of the state variables can be expressed as a linear combination of the system independent inputs and the state variables themselves.

  In order to solve the obtained differential equations, the knowledge of the initial state of the state variables is mandatory.

  The state-space representation of a system in compact matrix form is:

$$K\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.1a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.1b}$$

  being $\mathbf{x}(t)$ the state variables vector, $\mathbf{u}(t)$ the independent inputs vector, the matrix $\mathbf{K}$, a matrix containing elements such as capacitance or inductance values; $\mathbf{A}$ and $\mathbf{B}$ matrices containing constants of proportionality. While $\mathbf{y}(t)$

is the output vector expressed also as a linear combination of the state and the inputs vector; **C** and **D** are matrices containing constants of proportionality.

Now, to build a model, lets consider a PWM converter (operating in continuous conduction mode, for sake of simplicity) during different subintervals T_{sw}, where the circuit has different configurations. Thus, in the first subinterval the system will be described by the state-space representation:

$$K\frac{d\mathbf{x}(t)}{dt} = \mathbf{A_n}\mathbf{x}(t) + \mathbf{B_n}\mathbf{u}(t) \tag{2.2a}$$

$$\mathbf{y}(t) = \mathbf{C_n}\mathbf{x}(t) + \mathbf{D_n}\mathbf{u}(t) \tag{2.2b}$$

while in the following subinterval, the representation of the circuit will be:

$$K\frac{d\mathbf{x}(t)}{dt} = \mathbf{A_{n+1}}\mathbf{x}(t) + \mathbf{B_{n+1}}\mathbf{u}(t) \tag{2.3a}$$

$$\mathbf{y}(t) = \mathbf{C_{n+1}}\mathbf{x}(t) + \mathbf{D_{n+1}}\mathbf{u}(t) \tag{2.3b}$$

Since in the two different subintervals, the circuit has not the same configuration, state equation matrices may also differ.

Once the equations related to each circuit configurations have been written, the results of the state-space averaging are the state equations of the equilibrium and the small-signal AC models [7].

Assuming the natural frequencies of the converter and the frequencies of variations of the converter inputs are much slower than the switching frequency [7], the state-space averaged model of the converter in equilibrium is described by:

$$0 = \mathbf{AX} + \mathbf{BU} \tag{2.4a}$$

$$\mathbf{Y} = \mathbf{CX} + \mathbf{DU} \tag{2.4b}$$

where the averaged matrices are:

$$\mathbf{A} = D\mathbf{A_n} + D^{'}\mathbf{A_{n+1}} \tag{2.5a}$$
$$\mathbf{B} = D\mathbf{B_n} + D^{'}\mathbf{B_{n+1}} \tag{2.5b}$$
$$\mathbf{C} = D\mathbf{C_n} + D^{'}\mathbf{C_{n+1}} \tag{2.5c}$$
$$\mathbf{D} = D\mathbf{D_n} + D^{'}\mathbf{D_{n+1}} \tag{2.5d}$$

being $\mathbf{X}$ the equilibrium DC state vector, $\mathbf{U}$ the equilibrium DC input vector, $\mathbf{Y}$ the equilibrium DC output vector and D the equilibrium DC duty-cycle and they represent the equilibrium values of the averaged vectors.

By solving equation (2.4), the equilibrium state and output vectors can be obtained:

$$\mathbf{X} = -\mathbf{A^{-1}BU} \tag{2.6a}$$
$$\mathbf{Y} = (-\mathbf{CA^{-1}B} + \mathbf{D})\mathbf{U} \tag{2.6b}$$

Thus, the state equations of the small-signal AC model are:

$$K\frac{d\widehat{\mathbf{x}}(t)}{dt} = \mathbf{A}\widehat{\mathbf{x}}(t) + \mathbf{B}\widehat{\mathbf{u}}(t) + \{(\mathbf{A_n} - \mathbf{A_{n+1}})\mathbf{X} + (\mathbf{B_n} - \mathbf{B_{n+1}})\mathbf{U}\}\hat{d}(t) \tag{2.7a}$$
$$\widehat{\mathbf{y}}(t) = \mathbf{C}\widehat{\mathbf{x}}(t) + \mathbf{D}\widehat{\mathbf{u}}(t) + \{(\mathbf{C_n} - \mathbf{C_{n+1}})\mathbf{X} + (\mathbf{D_n} - \mathbf{D_{n+1}})\mathbf{U}\}\hat{d}(t) \tag{2.7b}$$

where $\widehat{\mathbf{x}}(t)$, $\widehat{\mathbf{u}}(t)$, $\widehat{\mathbf{y}}(t)$ and $\hat{d}(t)$ are small AC variations about the equilibrium solution, defined in equations (2.2) and (2.7) [7].

- Averaged Switch Model [7]

  In the averaged switch model, the converter waveform are averaged rather than the converter state equations and a graphical approach can be used.

One of the main advantages of this method is its generality since it can be applied to different converter types.

The main concept behind averaged switch models is the circuit averaging, achieved by replacing converter switches with voltage and current sources, obtaining a time-invariant circuit.

Once the substitution has been performed, it is possible to compute the average value of the converter waveforms in the selected switching period T_{sw}. Here, one basic assumption shall be respected: the switching period is much smaller than the natural time constants of the converter. A good practice is also to average the converter waveforms over the switching period T_{sw}. In this way the model would be able to predict the low-frequency behaviour of the system while neglecting the high-frequency switching harmonics [7].

The following step is to express the switch output dependent quantities as functions of the independent inputs and the duty-cycle *d(t)*.

Once the averaging step has been completed, only the DC and low-frequency AC components are present [7] and a general model - for frequency lower than the switching one - has been obtained.

It is worth mentioning that the averaging procedure acts only on the switching components of a converter circuit, leaving the rest of the circuit unchanged, and that, with this approach, switching losses can be also modelled obtaining first a large-signal averaged switched model, which can be than perturbed and linearised, obtaining a small-signal AC model as explained before.

If non-linear elements are present in the averaged circuit model, they can be perturbed and linearised. The obtained result is always a small-signal AC model [7].

The key concept used in the described modelling approaches is the *moving average value*. To clarify what the moving average value is, lets consider the signal *f(t)* shown in figure 2.4:

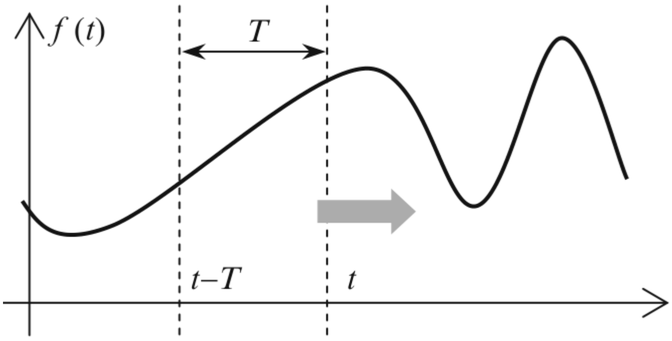


Figure 2.4: Moving average illustration (Source: [6])

it is averaged on a time interval T and its moving average is defined as:

$$\langle f(t)\rangle_0(t) = \frac{1}{T}\int_{t-T}^{t} f(\tau)\,d\tau \tag{2.8}$$

Differently from the concept of average value of a signal, the moving average is time-dependent since the time interval moves on the time axis.

If it is assumed that the signal *f(t)* is periodic and reaches its steady-state regime, the moving average becomes identical to the classical average [6].

As example, lets perform the analysis of switch-mode converters model, focusing on one of the poles, considered as their basic building block [1].

Each pole can be seen as a two-port component and consists of an on-off switch: on one side the capacitive port due to the presence of a capacitor acts (for a small time interval) as a voltage source (for this reason the port is also called voltage port); on the other side, an inductive port with an inductor is present, which acts (for a small time interval) as a current source (for this reason the port is also called current port).

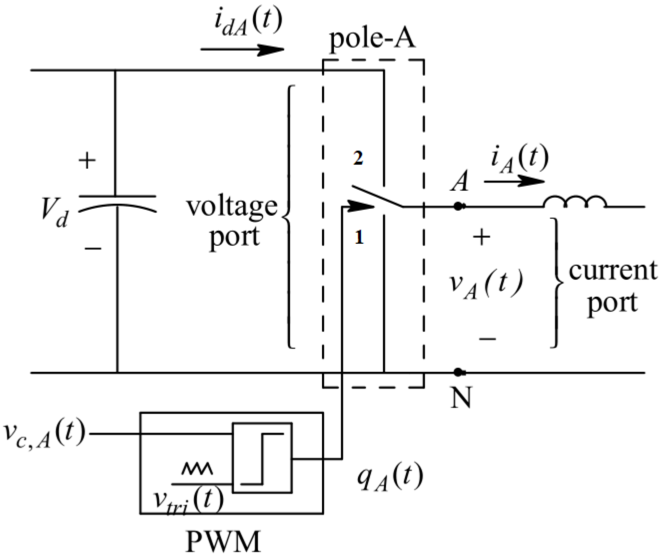


Figure 2.5: The canonical switching cell (Source: [1])

lets now add a voltage source V_{in} in parallel to the capacitive port and lets assume the switch is driven by a PWM with switching frequency f_{sw}. The aim is to obtain an output voltage on the inductive port whose average value is the desired one.

The switch is driven by a commutation function *q(t)* obtained from a PWM, using a triangular wave $v_{tri}(t)$ - with frequency f_{sw} - as carrier (for sake of simplicity with unitary amplitude) and a modulating signal $v_c(t)$, computed from the desired (or reference) voltage value. The two waveforms are compared, and the comparator output is mathematically described as follows:

if

$$v_c(t) > v_{tri}(t) \Rightarrow q(t) = 1 \Rightarrow \textit{switch ON} \Rightarrow v_A(t) = V_{in} \tag{2.9}$$

otherwise,

$$q(t) = 0 \Rightarrow \textit{switch OFF} \Rightarrow v_A(t) = 0 \tag{2.10}$$

When the switch is in position 2, the current flows from the input port to the output port through the inductor.

When the switch is in position 1, the current flows only in the secondary part of the circuit.

The output voltage is the $v_A(t)$ average value depending on the time the switch has been in position 2, called T_{ON}, so from the ratio $\frac{T_{ON}}{T_{sw}}$ it is possible to define the duty ratio.

By varying the duty-cycle within each period T_{sw}, different $v_A(t)$ average values can be obtained, thus it will be defined as moving average and computed as follows:

$$\bar{v}_A = \frac{1}{T_{sw}} \int_0^{T_{sw}} v_A(t)dt = \frac{T_{ON}}{T_{sw}} \cdot V_{in} = d \cdot V_{in} \tag{2.11}$$

hence, the output voltage linearly depends on the duty-cycle. To show how the output voltage also depends on the control voltage (the modulating signal), lets do the following considerations [1]:

if

$$v_c(t) = \hat{V}_{tri} \Rightarrow \text{the duty-cycle } d = 1 \Rightarrow \bar{v}_A = V_{in} \tag{2.12}$$

and if

$$v_c(t) = -\hat{V}_{tri} \Rightarrow \text{the duty-cycle } d = 0 \Rightarrow \bar{v}_A = 0 \tag{2.13}$$

From this, the following relation between control voltage and duty-cycle can be established:

$$\frac{\Delta d}{\Delta v_c} = \left(\frac{1}{2\hat{V_{tri}}}\right) \tag{2.14}$$

and

$$d = \left(\frac{1}{2\hat{V_{tri}}}\right) \cdot v_c + \text{offset} \tag{2.15}$$

by substituting to the control voltage value equation (2.12) or equation (2.13), the obtained offset is $\frac{1}{2}$.

The duty-cycle results to be [1]:

$$d = \frac{1}{2} + \frac{1}{2} \cdot \frac{v_c}{\hat{V}_{tri}} \tag{2.16}$$

Since the control voltage is computed at every T_{sw}, thus is time varying, also the duty-cycle can be defined as a time-dependent function.

What is also interesting to focus on, is that the output current remains smooth thanks to the series inductance at the output port. The output current behaviour will be analysed below.

Referring to figure 2.5, when the switch is in position 2, the current flows through the inductor, which will start to charge; the current will increase gradually during the T_{ON} period since the inductor tries to oppose to the change in current.

When the switch moves to position 1, the current is chopped, but the current in the inductor will not change abruptly, rather it will discharge gradually.

The output current average value can be derived as follows:

$$\bar{i}_{dA}(t) = d_A(t) \cdot \bar{i}_A(t) \tag{2.17}$$

From all the previous considerations, it can be observed that the quantities at the input and output ports are related to the duty-cycle. This allows to describe the power pole as an ideal transformer where input and output quantities are related by a coupling factor and both DC and AC currents and voltages can be transmitted across it [1].

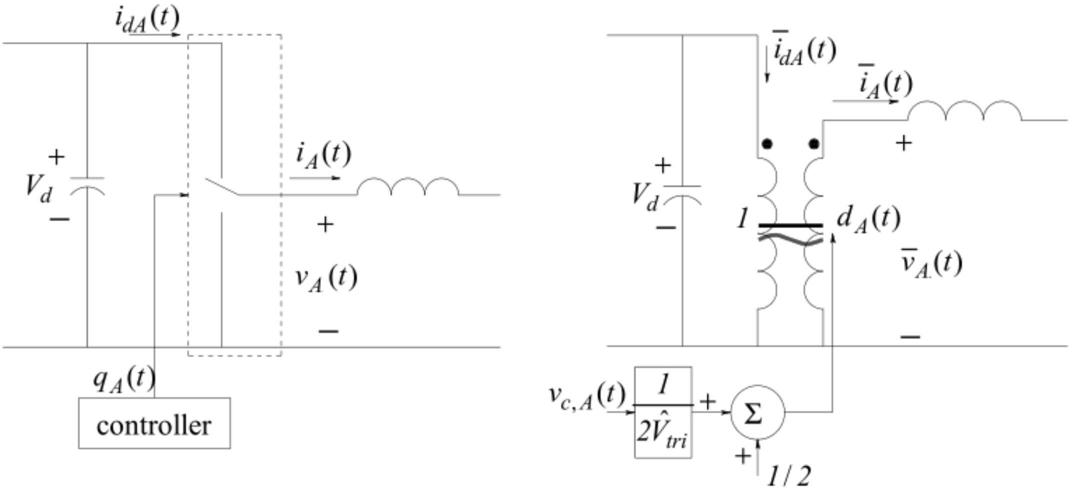


Figure 2.6: Equivalent ideal transformer of a single switching cell (Source: [1])

2.4 DC/AC Conversion

The power electronic converters able to obtain from a DC source a single or polyphase variable-voltage and variable-frequency power supply are called inverters.

In the following sections, single-phase and three-phase inverter will be addressed.

Inverters can be classified as voltage source inverters (VSIs) and current source inverters (CSIs): the first is fed by a DC voltage source, the latter by a DC current source and nowadays employed only for high power AC motor.

Inverters are composed of basic elements which are the main power control devices. Typically, they are insulated gate bipolar transistors (IGBTs) or power MOSFETs acting as switches.

Focusing on VSIs, they are normally controlled by a square-wave or by a pulse width modulation (PWM) and this controls the inverter's power switches.

Considering the application developed in this thesis work, only PWM inverters will be discussed.

2.4.1 2-Level Single-Phase Inverter

The simplest configuration of a VSI is a 2-level single-phase PWM inverter, obtained with a single switching pole. It consists of two power switches connected in series; each switch has a diode connected in antiparallel. The single switching pole is bidirectional in current.

It can be implemented by considering a single DC source or a double DC source of $\pm 0.5\,V_{dc}$ referred to the midpoint O. A capacitor is connected in parallel to each DC source.

Considering sufficiently large capacitances, it is possible to assume that the potential at midpoint O remains constant [1].

The load is connected between the midpoint O and the output of the switching pole.

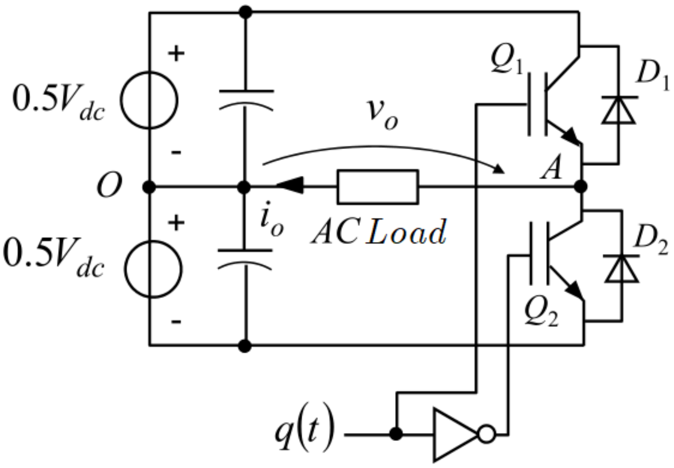


Figure 2.7: Single-phase inverter with single switching pole (Source: [8])

What determines the output voltage is the DC input voltage and the commutation function controlling the power switches gates.

For the inverter under analysis, the output voltage will be determined as follows:

$$v_o(t) = [2q(t) - 1]\frac{V_{dc}}{2} \tag{2.18}$$

The commutation function *q(t)* is the function obtained from a sinusoidal PWM with bipolar voltage switching.

As described in the previous section, this type of PWM uses a triangular wave - with frequency f_s equal to the inverter switching frequency f_{sw} - as carrier (for sake of simplicity with unitary amplitude) and a modulating signal computed from the desired (or reference) voltage value.

The modulating signal, also called command signal, must be sinusoidal.

To obtain an AC output voltage means to set an instantaneous voltage with sinusoidal moving average; if the moving average voltage is:

$$\bar{v}_o^*(t) = \hat{V}^* sin(\omega t) = \frac{V_{dc}}{2} \cdot \frac{v_c(t)}{\hat{V}_{tri}} \tag{2.19}$$

where $\omega = 2\pi f_o$, being f_o the desired output frequency, $\hat{V}$ the peak value of the desired output signal and $\hat{V}_{tri}$ the peak value of the triangular carrier signal. The superscript * indicates the desired value.

From this, the command signal is computed:

$$v_c(t) = \frac{\hat{V}^*}{\frac{V_{dc}}{2}} \cdot \hat{V}_{tri} \cdot sin(\omega t) \tag{2.20}$$

In this way, the ON period and the OFF period of the power switches can be determined by the crossover points during the command signal and the triangular carrier signal comparison.

Using a sinusoidal signal, constant changes on the duty cycle of the switches during each half-period are performed, thus reducing the output harmonic content.

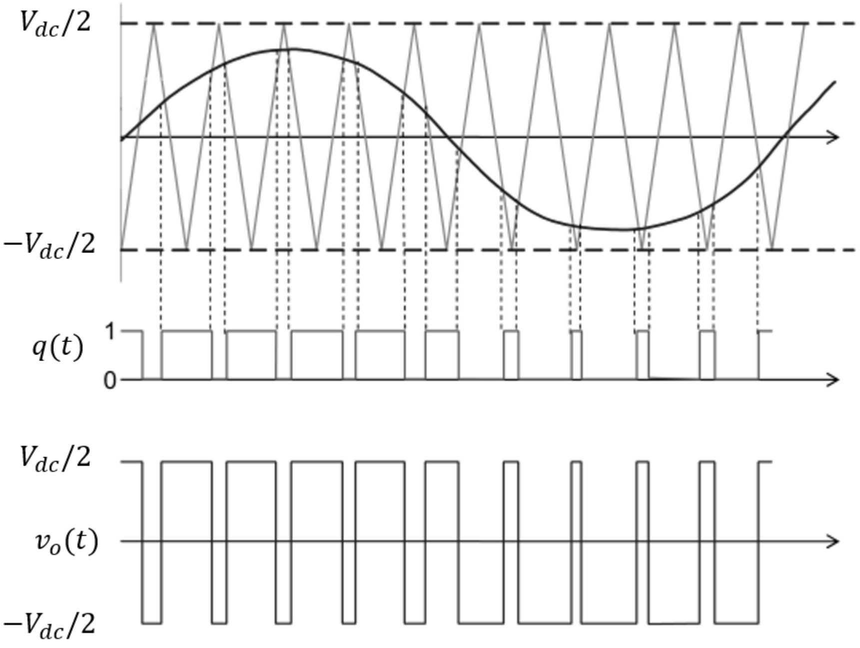


Figure 2.8: Sinusoidal PWM with unitary triangular wave as carrier (Source: [9])

Defining as

$$\hat{V}_c = \frac{\hat{V}^*}{\frac{V_{dc}}{2}} \cdot \hat{V}_{tri} \tag{2.21}$$

the command signal peak value, equation (2.20) can be written as:

$$v_c(t) = \hat{V}_c sin(\omega t) \tag{2.22}$$

From (2.21) the following parameter, called *amplitude modulation factor* can be defined:

$$\frac{\hat{V}_c}{\hat{V}_{tri}} = \frac{\hat{V}^*}{\frac{V_{dc}}{2}} = m_a \tag{2.23}$$

The following ratio defines the *frequency modulation factor*:

$$m_f = \frac{f_{sw}}{f_o} \tag{2.24}$$

where f_{sw} is the switching frequency and f_o is the reference output frequency.

For values of $m_f > 20$, it is possible to approximate the moving average output voltage of equation (2.19) as the desired voltage.

Knowing the command voltage, it is also possible to compute the duty-cycle of the switching pole, as previously defined in equation (2.16):

$$d = \frac{1}{2} + \frac{1}{2} \cdot v_c(t) \tag{2.25}$$

Considering a simple RL load, it is possible to observe that the output voltage for a two-level single phase PWM inverter is a pulse train of amplitude $\pm 0.5\mathrm{V_{dc}}$, where the fundamental frequency is the desired output frequency. Harmonics due to commutation are also present, having frequency:

$$f_h = k_1 \cdot f_s \pm k_2 \cdot f_o \qquad \mathrm{k_1,\, k_2} = 0,\, 1,\, 2,\, 3,\, \ldots \tag{2.26}$$

For what concerns the load current, it consists of a fundamental generated by the output voltage fundamental and a triangular ripple due to commutation.

The problem related with the load current is that the ripple is not constant during the fundamental period and the peak-to-peak value is obtained when the command signal is null thus when the duty-cycle is 0.5:

$$\Delta i_{o,max} = \frac{\frac{1}{2} \cdot T_s \cdot \frac{1}{2} \cdot V_{dc}}{L} = \frac{1}{4} \cdot \frac{V_{dc}}{L \cdot f_s} \tag{2.27}$$

assuming $2\pi f_s L \gg R$.

Focusing on the PWM modulation, it is possible to distinguish three situations basing on the amplitude modulation factor m_a:

- Linear modulation

  In this case the factor m_a is $0 \leq m_a \leq 1$, and the output voltage peak amplitude is $\hat{V}^* \leq \frac{V_{dc}}{2}$. The fundamental component of the output voltage corresponds to the moving average voltage and the generated harmonics are present at high frequency only. In this linear operating region, the fundamental voltage peak value is equal to the reference voltage peak value.

Since:

$$\bar{v}_o(t) \approx v_{o,1}(t) = \hat{V}_1 sin(\omega t) \tag{2.28}$$

and, from equation (2.23)

$$\hat{V}_1 = \frac{\hat{V}_c}{\hat{V}_{tri}} \cdot \frac{V_{dc}}{2} = m_a \cdot \frac{V_{dc}}{2} \tag{2.29}$$

the *root mean square (rms)* of the output voltage is obtained as follows:

$$V_1 = m_a \cdot \frac{V_{dc}}{2 \cdot \sqrt{2}} \quad [V_{rms}] \tag{2.30}$$

and the limit case correspond to the amplitude modulation factor $m_a = 1$:

$$V_1 \Big|_{m=1} = \frac{V_{dc}}{2 \cdot \sqrt{2}} = 0.353 \cdot V_{dc} \tag{2.31}$$

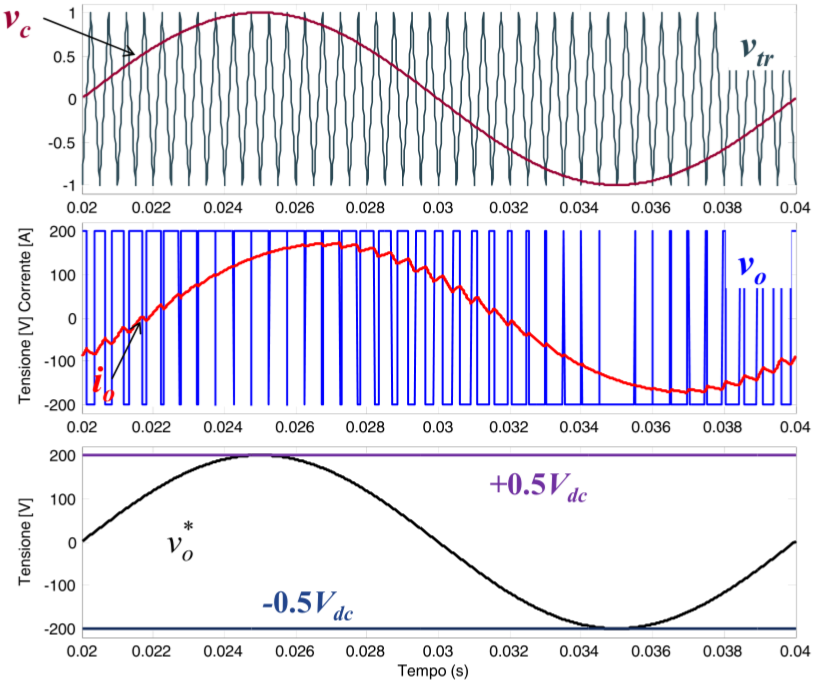


Figure 2.9: Sinusoidal PWM with $m_a = 1$ (Source: [8])

- Overmodulation

  In this case, $m_a > 1$ and the the *rms* of the output voltage is greater than $0.353 \cdot V_{dc}$.

  Here the command is saturated, and the duty-cycle becomes a trapezoidal wave since, referring to equation (2.25), the command is also a trapezoidal wave. During the saturation interval, no commutations occur.

  Now, also odd voltage harmonics are present at low frequency causing distortions in the load current.

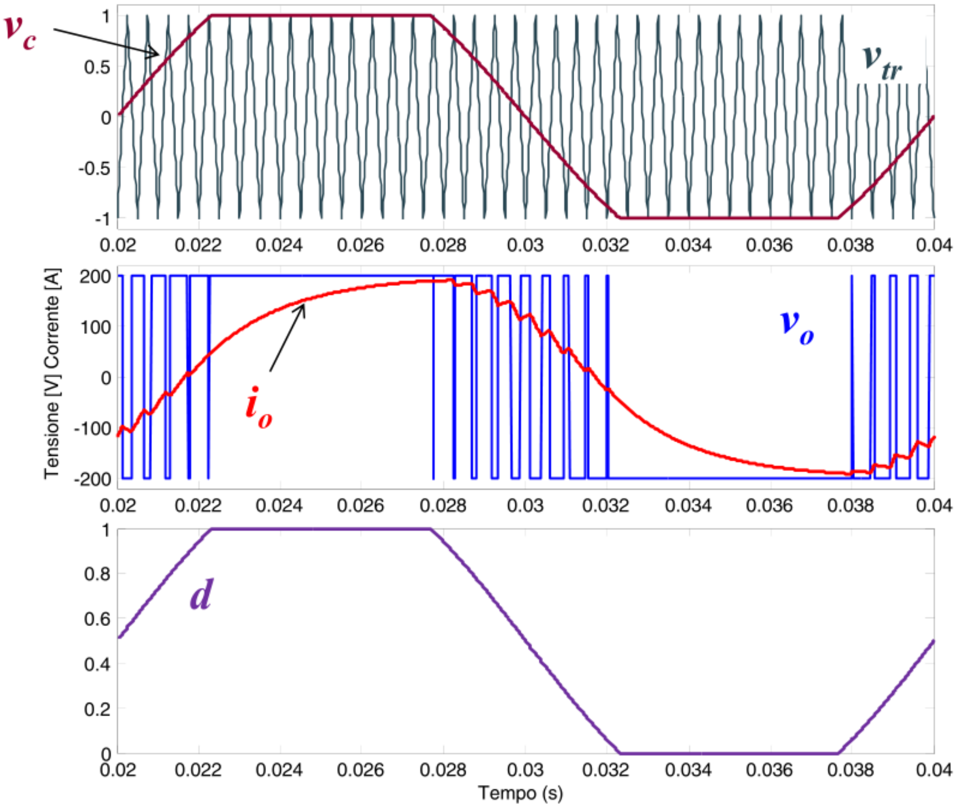


Figure 2.10: Sinusoidal PWM with $m_a = 1.5$ (Source: [8])

- Square wave modulation

  In this case m_a » 1 and the command (and so the duty-cycle) is a square wave. It means that only two commutations occur during the fundamental period, hence the output voltage is also a square wave, having amplitude $0.5 \cdot V_{dc}$.

The maximum value of the fundamental output voltage is $0.45\cdot\ V_{dc}$ (amplitude of the fundamental of a generic square wave):

$$V_1 = \frac{1}{\pi} \cdot \frac{V_{dc}}{2 \cdot \sqrt{2}} = \frac{\sqrt{2} \cdot V_{dc}}{\pi} \quad [V_{rms}] \tag{2.32}$$

Here the current is subjected to the maximum distortion.

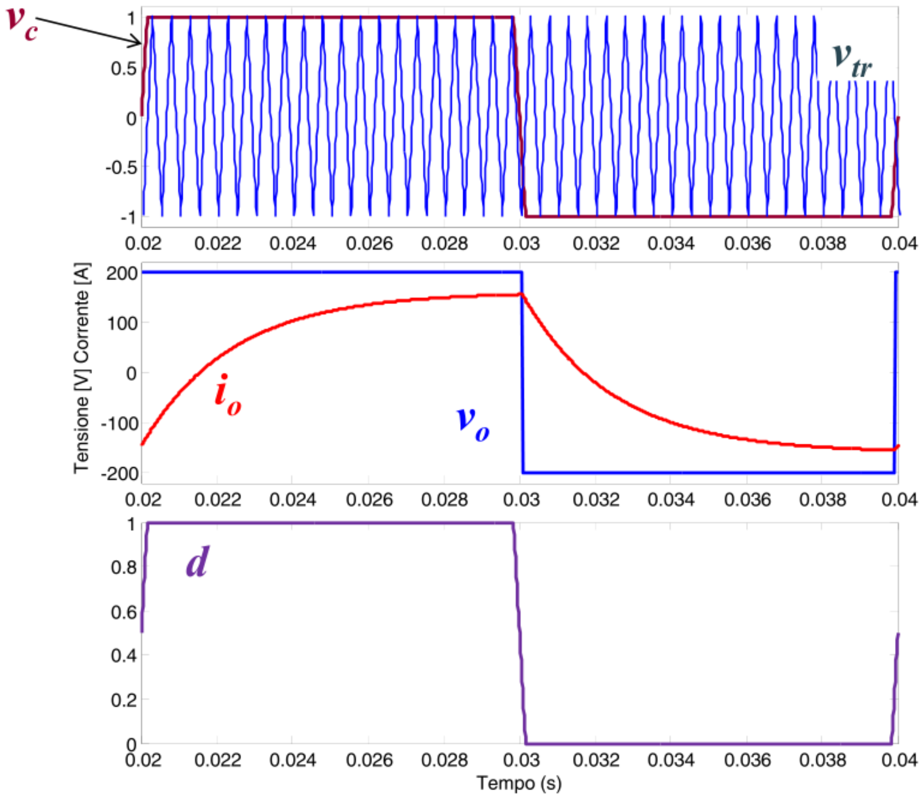


Figure 2.11: Sinusoidal PWM in six-step operation (Source: [8])

2.4.2 2-Level Three-Phase PWM Inverter

A 2-level three-phase PWM inverter can be considered an extension of a single-phase PWM inverter, with 3 switching poles and a single DC source. Even in this case, it is possible to consider a double DC source with virtual midpoint O and consider the output voltage of each phase with respect to this point.

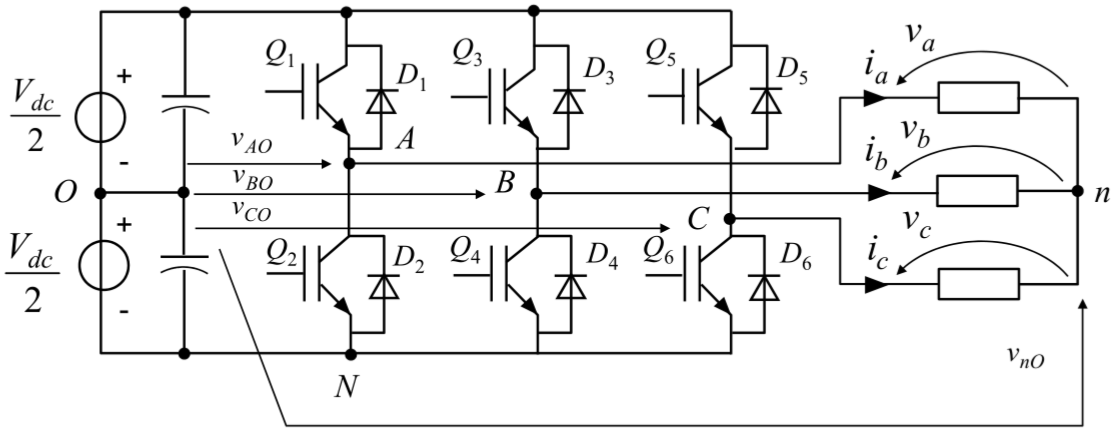


Figure 2.12: Three-phase inverter (Source: [8])

A three-phase inverter generates a three-phase AC voltages symmetrical system (the three inverter legs output voltages have a phase shift of 120°).

To obtain a three-phase system also the command signals shall be a three-phase symmetrical system:

$$\begin{cases} v_{cA}(t) = \frac{\hat{V}^*}{(\frac{V_{dc}}{2})} \cdot \hat{V}_{tri} \cdot sin(\omega t) \\ v_{cB}(t) = \frac{\hat{V}^*}{(\frac{V_{dc}}{2})} \cdot \hat{V}_{tri} \cdot sin(\omega t - \frac{2\pi}{3}) \\ v_{cC}(t) = \frac{\hat{V}^*}{(\frac{V_{dc}}{2})} \cdot \hat{V}_{tri} \cdot sin(\omega t + \frac{2\pi}{3}) \end{cases} \tag{2.33}$$

from which the moving average values of the three inverter legs output voltages with respect to the midpoint O are:

$$\begin{cases} \bar{v}_{AO}(t) = \frac{V_{dc}}{2} \cdot \frac{v_{cA}(t)}{\hat{V}_{tri}} \\ \bar{v}_{BO}(t) = \frac{V_{dc}}{2} \cdot \frac{v_{cB}(t)}{\hat{V}_{tri}} \\ \bar{v}_{CO}(t) = \frac{V_{dc}}{2} \cdot \frac{v_{cC}(t)}{\hat{V}_{tri}} \end{cases} \tag{2.34}$$

The principle on which the command signal generation is based is the same as explained in the previous paragraph for the single-phase inverter.

Assuming a generic balanced star connected AC load, it is important to distinguish between the inverter output voltage with respect to the midpoint O and the three-phase AC load phase voltages referred to the neutral point n.

The phase voltages are obtained as follows:

$$\begin{cases} v_a(t) = v_{AO}(t) - v_{nO} \\ v_b(t) = v_{BO}(t) - v_{nO} \\ v_c(t) = v_{CO}(t) - v_{nO} \end{cases} \tag{2.35}$$

where $v_{AO}(t)$, $v_{BO}(t)$ and $v_{CO}(t)$ are the output voltages of the three inverter legs, while v_{nO} is the common mode voltage and it is computed as follows:

$$v_{nO}(t) = \frac{1}{3} \cdot [v_{AO}(t) + v_{BO}(t) + v_{CO}(t)] \tag{2.36}$$

If considering the moving average voltages,

$$\begin{cases} \bar{v}_a(t) = \bar{v}_{AO}(t) - \bar{v}_{nO} \\ \bar{v}_b(t) = \bar{v}_{BO}(t) - \bar{v}_{nO} \\ \bar{v}_c(t) = \bar{v}_{CO}(t) - \bar{v}_{nO} \end{cases} \tag{2.37}$$

and, since the moving average value of the common mode voltage is null

$$\bar{v}_{nO} = \frac{1}{3} \sum_{k=a,b,c} \bar{v}_{kO}(t) = 0 \tag{2.38}$$

thus, from equation (2.37) the moving average value of the phase voltages result to be equal to the inverter legs output voltages of equation (2.34).

Considering the same modulator implemented for the single-phase inverter with unitary triangular carrier, the duty-cycles of the switching poles are:

$$\begin{cases} d_A(t) = \frac{1}{2} + \frac{1}{2} \cdot v_{cA}(t) \\ d_B(t) = \frac{1}{2} + \frac{1}{2} \cdot v_{cB}(t) \\ d_C(t) = \frac{1}{2} + \frac{1}{2} \cdot v_{cC}(t) \end{cases} \tag{2.39}$$

As result, the three legs output voltages are a bipolar voltage pulse train having amplitude $\pm 0.5 \cdot V_{dc}$ and 120° phase shift.

Referring to the equation (2.35), the load phase voltages are instead a voltage pulse train with amplitudes $\pm\frac{2}{3}V_{dc}$, $\pm\frac{1}{3}V_{dc}$ and 0.

The load phase voltages present:

- A sinusoidal fundamental component:

$$v_{a,1}(t) = \hat{V}_1 \cdot sin(\omega t) \tag{2.40}$$

- Harmonics due to converter commutation phenomenon with frequencies:

$$f_h = k_1 f_s \pm k_2 f_2 \quad k_1, k_2 = 0, 1, 2, 3, \ldots \tag{2.41}$$

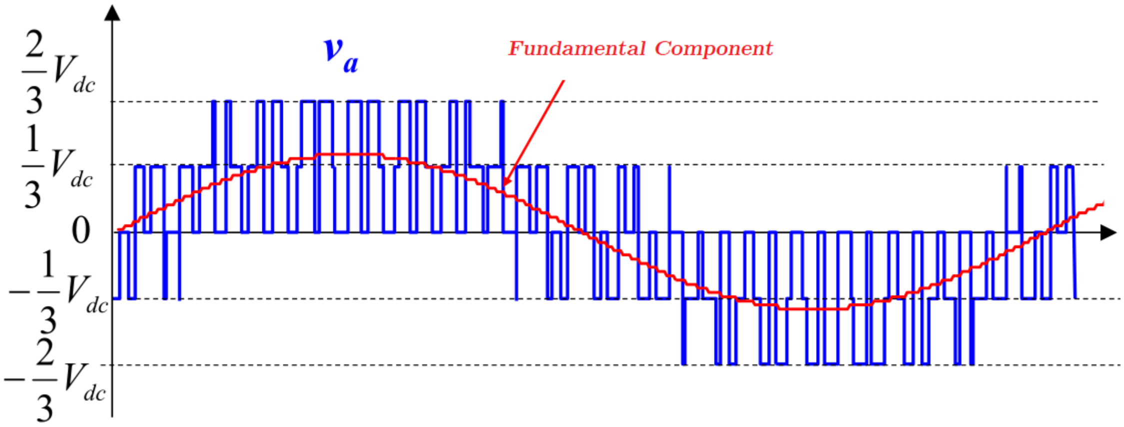


Figure 2.13: Load phase voltage of a three-phase inverter (Source: [8])

The load currents have quasi-sinusoidal waveform with triangular ripple caused by commutations.

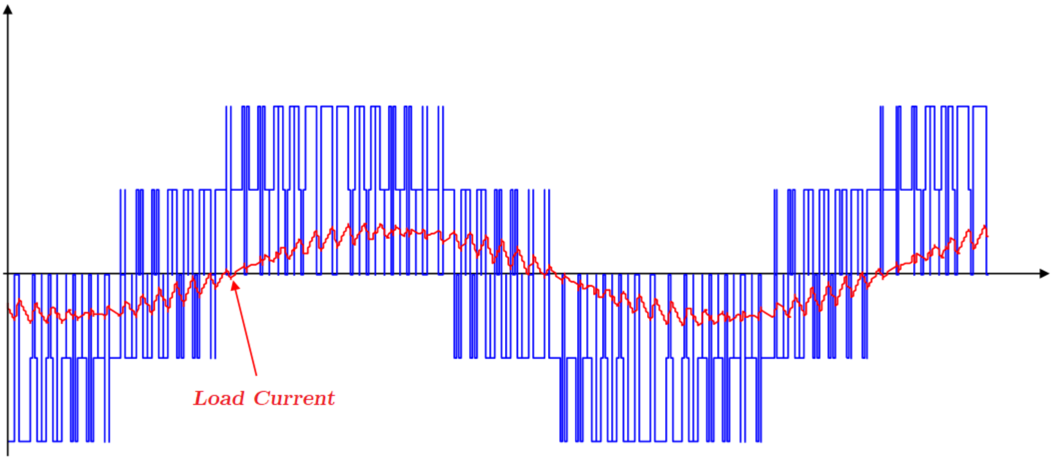


Figure 2.14: Load phase current of a three-phase inverter (Source: [8])

Even for the three-phase PWM inverter it is possible to analyse the modulation depending on the amplitude modulation factor, underlining the differences with the single-phase inverter: in linear modulation the same considerations done for the single-phase inverter hold for the three-phase inverter.

What it is worth underlining is that in overmodulation and in square-wave modulation (also called six-step modulation since only two commutations per phase occur), no odd voltage harmonics multiple of 3 are present. Since third harmonics and their multiple are present in both output leg voltage and common mode voltage, applying equation (2.35), they are cancelled.

In the six-step operation, the maximum value of the voltage fundamental is obtained, but also the distortion is maximum due to the presence of low-frequency harmonics.

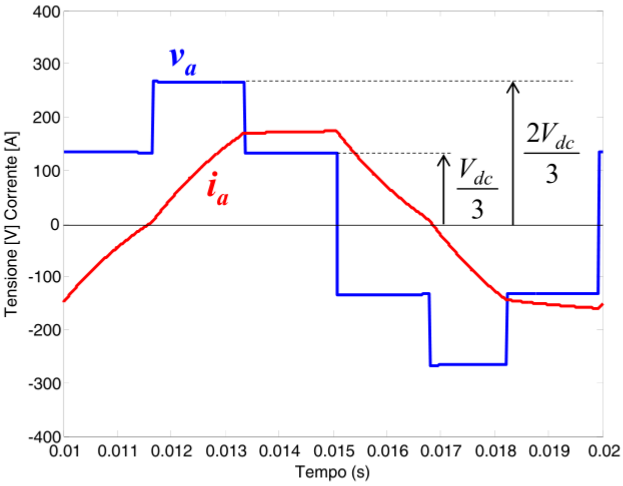


Figure 2.15: Load phase current of a three-phase inverter in six-step operating mode (Source: [8])

As discussed before, in the inverter output voltages (both single-phase and three-phase) only odd harmonics are present. The reason why no even harmonics can be observed is because, even considering the worst case scenario where non-linear components are used, the distorted waves always maintain a symmetry (the positive half-period is equal and opposite to the negative half-period).

2.4.3 Advanced PWM Techniques

Up to now the sinusoidal PWM have been analysed pointing out the main characteristics of the modulation, but it presents also some limits: when working in the linear modulation region, where the full range of the DC-link voltage cannot be used, switching losses cannot be limited.

For these reasons in a three-phase inverter, advanced modulation techniques can be implemented to improve the performance of the converter.

They are based on the common mode voltage, a component which is always present in analog signals.

Thus, the normalised (with respect to the DC-link voltage) reference voltages are modified by adding the common mode voltage resulting in new reference voltages for the modulator:

$$\begin{cases} v_a^{**}(t) = v_a^*(t) + v_{CM}(t) \\ v_b^{**}(t) = v_b^*(t) + v_{CM}(t) \\ v_c^{**}(t) = v_c^*(t) + v_{CM}(t) \end{cases} \tag{2.42}$$

Basically, two main techniques can be adopted:

- PWM-BEM (PWM – Balanced Envelop Modulation)

  This technique is based on the third harmonic injection in the reference signal where the common mode voltage is computed as follows:

$$v_{CM}(t) = -\frac{1}{2}[max(v_{a,norm}^*, v_{b,norm}^*, v_{c,norm}^*) + min(v_{a,norm}^*, v_{b,norm}^*, v_{c,norm}^*)] \tag{2.43}$$

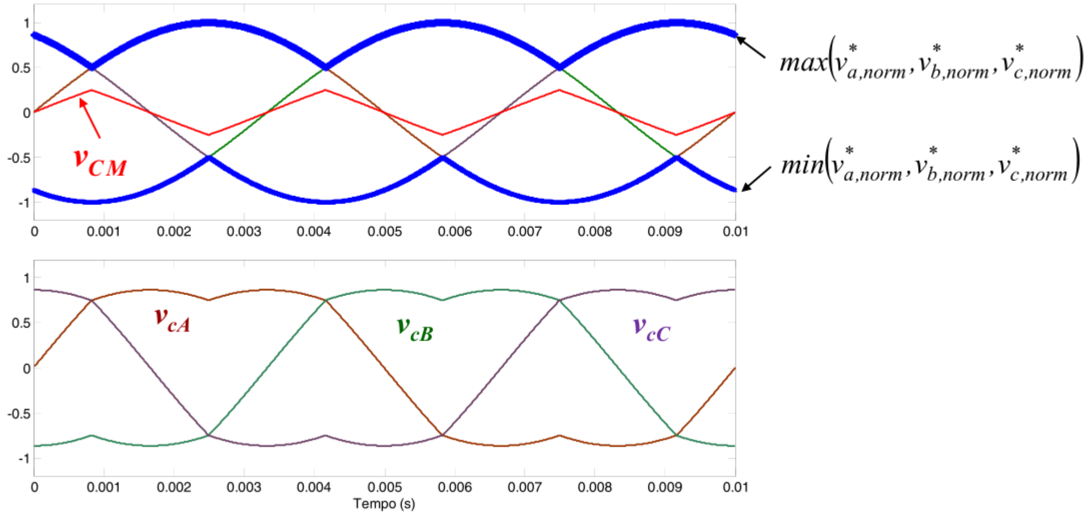


Figure 2.16: PWM with third harmonic injection (Source: [8])

The reason for choosing the third harmonic is that, in a three-phase system, it has the same time function (zero sequence components) in all the phases, thus it cannot produce current [10].

This technique allows to use 15% more of the linear modulation region with respect to the sinusoidal modulation, because the command saturation occurs for a value of the phase reference voltage of:

$$\frac{2}{\sqrt{3}} \cdot \frac{V_{dc}}{2} = 1.15 \cdot \frac{V_{dc}}{2} \tag{2.44}$$

The main advantage of this technique is related to the control of the converter, providing a wider linear operating range.

The obtained duty-cycles are the same obtained by a Space-Vector PWM (SVPWM).

- DPWM (Discontinuous PWM)

  This technique is aimed at reducing switching losses. This is achieved by computing the common mode voltage in such a way that no commutation of one phase occurs for 120° electrical degrees.

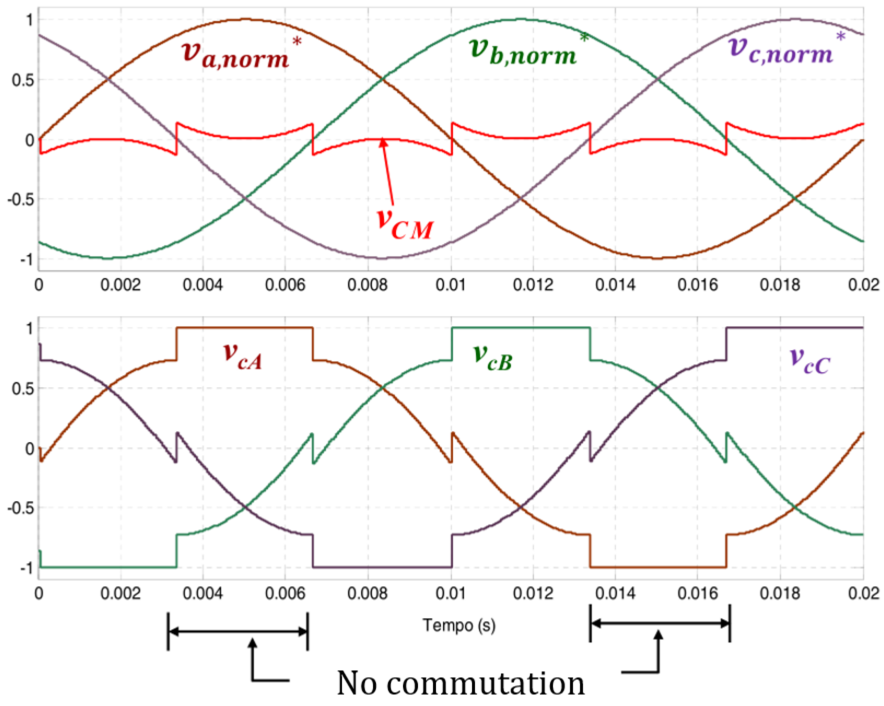


Figure 2.17: Discontinuous PWM (Source: [8])

2.4.4 Inverter Power Balance

To evaluate inverter performances from a power efficiency point of view it is necessary to refer to the average dynamic model of the inverter, focusing on the switching poles modelled as described in section 2.3 .

Assuming the inverter under analysis is a three-phase PWM (with unitary triangular carrier) inverter with single DC source and R-L load, to perform a power balance, the average current absorbed by the power supply $i_d(t)$ shall be computed.

The total absorbed current is:

$$\bar{i}_d(t) = \bar{i}_{dA}(t) + \bar{i}_{dB}(t) + \bar{i}_{dC}(t) \tag{2.45}$$

where

$$\begin{cases} \bar{i}_{dA}(t) = d_A(t)\bar{i}_{dA}(t) \\ \bar{i}_{dB}(t) = d_B(t)\bar{i}_{dB}(t) \\ \bar{i}_{dC}(t) = d_C(t)\bar{i}_{dC}(t) \end{cases} \tag{2.46}$$

Since the moving average values of the phase voltage result to be equal to (2.34), by making explicit the three command voltages and substituting equation (2.37) in equation (2.39), the duty-cycles can be written as:

$$\begin{cases} d_A(t) = \frac{1}{2} + \frac{\bar{v}_a}{V_{dc}} \\ d_B(t) = \frac{1}{2} + \frac{\bar{v}_b}{V_{dc}} \\ d_C(t) = \frac{1}{2} + \frac{\bar{v}_c}{V_{dc}} \end{cases} \tag{2.47}$$

Substituting these expressions in (2.46) and referring to the condition characterising a symmetric three-phase current system:

$$\bar{i}_a(t) + \bar{i}_b(t) + \bar{i}_c(t) = 0 \tag{2.48}$$

the three absorbed currents can be written as:

$$\begin{cases} \bar{i}_{dA}(t) = \frac{1}{2} \cdot \bar{i}_a(t) + \frac{\bar{v}_a \cdot \bar{i}_a}{V_{dc}} \\ \bar{i}_{dB}(t) = \frac{1}{2} \cdot \bar{i}_b(t) + \frac{\bar{v}_b \cdot \bar{i}_b}{V_{dc}} \\ \bar{i}_{dC}(t) = \frac{1}{2} \cdot \bar{i}_c(t) + \frac{\bar{v}_c \cdot \bar{i}_c}{V_{dc}} \end{cases} \tag{2.49}$$

Thus, the absorbed power is:

$$P_{dc} = V_{dc} \cdot \bar{i}_d(t) = \bar{v}_a \cdot \bar{i}_a + \bar{v}_b \cdot \bar{i}_b + \bar{v}_c \cdot \bar{i}_c \tag{2.50}$$

Since:

$$\begin{cases} \bar{v}_a(t) = \hat{V} \cdot sin(\omega t) \\ \bar{v}_b(t) = \hat{V} \cdot sin(\omega t - \frac{2\pi}{3}) \\ \bar{v}_c(t) = \hat{V} \cdot sin(\omega t + \frac{2\pi}{3}) \end{cases} \tag{2.51}$$

and

$$\begin{cases} \bar{i}_a(t) = \hat{I} \cdot sin(\omega t - \Phi) \\ \bar{i}_b(t) = \hat{I} \cdot sin(\omega t - \frac{2\pi}{3} - \Phi) \\ \bar{i}_c(t) = \hat{I} \cdot sin(\omega t + \frac{2\pi}{3} - \Phi) \end{cases} \tag{2.52}$$

thus, the absorbed power can be computed as:

$$P_{dc} = \hat{V} \cdot \hat{I} \cdot \sum_{k=0,1,2} [sin(\omega t - k\frac{2\pi}{3}) \cdot sin(\omega t - k\frac{2\pi}{3} - \Phi)] \tag{2.53}$$

and the following is obtained:

$$P_{dc} = \frac{3}{2} \cdot \hat{V} \cdot \hat{I} = 3 \cdot V_{rms} \cdot I_{rms} \cdot cos\Phi \tag{2.54}$$

When considering a real inverter, the absorbed power also takes into account for the losses of the power converter (switching and conduction losses). Therefore, the efficiency can be defined as:

$$\eta = \frac{3 \cdot V_{rms} \cdot I_{rms} \cdot cos\Phi}{P_{dc}} = \frac{P_{active,load}}{P_{dc}} \tag{2.55}$$

2.4.5 Effects of Inverters on AC Motors

As previously mentioned, inverters are used to drive AC load as an AC motor.

Differently from what happens in simulation, where all the following aspects are not always considered, in the interaction between inverter and an AC motor, different issues arise.

The main problem is related to the voltage the inverter applies to the motor, which is not ideal but presents a fundamental voltage component and harmonics due to commutation; it is also depending on the modulation (if overmodulated or working in six-step).

The distorted voltages applied from the inverter to the motor cause torque ripple, efficiency decrease, losses and EMC issues.

In the following, the previously mentioned problems will be described:

- Torque Ripple

  When working in linear modulation, a PWM inverter output currents present triangular ripple due to commutations and these cause torque ripple, influencing also the motor speed. To reduce these effects, a higher PWM switching frequency could be adopted, although increasing the probability of insulation failure, since it is directly proportional to the switching frequency [11]. The situation is worst when working in overmodulation and six-step, since high torque undulations with a frequency being six time the fundamental frequency are produced [11]. This causes vibrations that damage the motor-transmission mechanical system.

- Motor Losses

  Efficiency in AC motors is reduced by many factors: in fact, due to commutation ripple, the current RMS increases causing Joule losses, while an increment of RMS of the voltage that the inverter feeds to the motor, causes iron losses.

- EMC

  The phase voltages at the inverter output are characterised by pulses having different amplitude and with derivatives, with respect to time, that can be greater than 1000 V/µs and this becomes a prevailing factor when considering cables connecting the inverter and the motor. Cables can be considered as transmission lines, thus they present impedances distributed in sections of inductances/capacitances series/parallel connected [11].

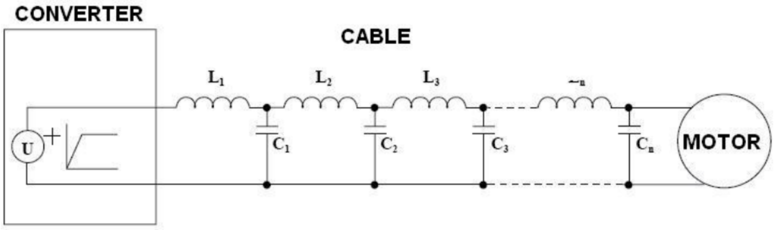


Figure 2.18: Connection between inverter and an AC motor represented as transmission line (Source: [11])

  Therefore, the signal arriving at the motor is subjected to reflections causing overvoltage (since the motor high frequency impedance is greater than the cable impedance [11]) also related to the cable length. As a result, the motor insulation life, and thus motor life, is reduced.

- Common Mode Voltage

  Another element causing disturbances in the inverter-motor interaction is the common mode voltage (refer to equation (2.36)) from the inverter. Its instantaneous value depends on the amplitude modulation factor m_a. Focusing on the worst case, when $m_a = 0$, the common mode voltage produces common mode capacitive currents in the ground-connected parasitic capacitances, causing current loops damaging the motor bearing [8]. A possible solution could be to introduce common mode filters.

2.5 3-Phase Motor Modelling

Three-phase machines, also called AC machines, can be classified in two categories: asynchronous (or induction) motors (IM) and synchronous motors which include different motor type, depending on the rotor: the Surface-mounted Permanent Magnet (SPM) motors, the Internal Permanent Magnet (IPM) motors, the PM-assisted Synchronous Reluctance (PM-SyR) motors and Synchronous Reluctance (SyR) motors.

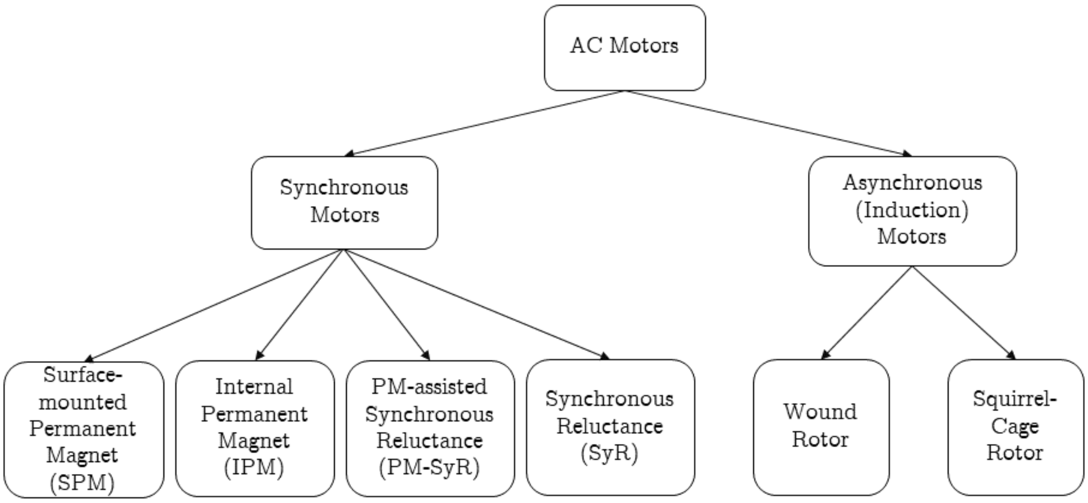


Figure 2.19: AC Motor Classification

Any electrical rotating machine is composed of two main parts:

- Windings, the coils carrying currents classified as distributed windings (where end turns overlap) and concentrated windings (where end turns do not overlap). The two different winding configurations produce, when current flows through them, magnetic fields having different features: in distributed windings a magnetic field having a quasi-sinusoidal distribution at the air gap is produced; concentrated windings, cheaper than the previously mentioned, produce harmonics fields and are employed in applications where the required power is not high.

- Magnetic Core which consists of two concentric cylindrical parts made of electrical steel laminations, separated by an air gap: stator and rotor. Focusing on AC machines, typically the windings are housed inside the stationary part, the stator and defined as stator windings. For this reason, stator laminations have slots. Induction motor rotors are also characterised by slots to house the rotor windings.

In the following, a briefly description of the basic concepts related to AC machines functioning will be addressed.

2.5.1 Asynchronous Motor

As described before, the induction motor, or asynchronous motor, is composed of a stator, housing three-phase windings (with a spatial displacement of 120 electrical degrees) in which a three-phase currents system flow, generating a rotating magnetic field with an electrical angular speed $\omega_s = 2\pi f_s$, where f_s in the frequency of AC stator currents.

The stator windings are magnetically coupled with the rotor windings, thus, currents are induced in the short-circuited rotor windings (independently from the rotor type which can be: wound rotor, typically used in high power applications or squirrel cage rotor composed of conductive bars short-circuited at both ends by rings forming a cage structure).

For these reasons the rotor cage can be compared to a rotating three-phase winding and the induction motor can be modelled as a rotating transformer.

In IM, the torque is produced due to the interaction between the magnetic field generated in the stator and rotor currents.

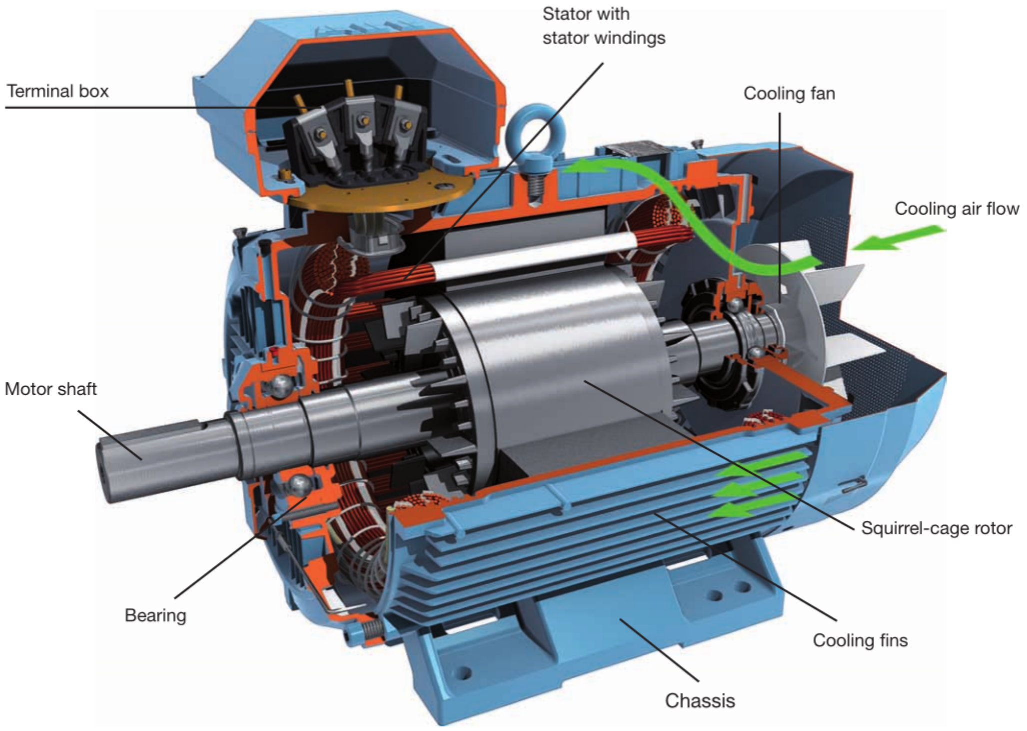


Figure 2.20: Cross section of an asynchronous motor (Source: [12])

Assuming the rotor is rotating with an angular speed of ω_m, the electromotive force (e.m.f.) induced in the rotor will be defined as:

$$E_r = K_r \cdot \Phi \cdot (\omega_s - \omega_m) \tag{2.56}$$

where K_r is the rotor windings coefficient and Φ is the machine flux. It is worth underlining the difference between rotor speed relatively to the stator rotating magnetic field: the basic functioning behind IM is related to the fact that the rotor tends to synchronise to the excitation field, but rotor rotating speed will always be different (except in no-load conditions where $\omega_\mathrm{s} = \omega_\mathrm{m}$) with respect to the stator rotating field speed, from this the name asynchronous.

The difference in rotational speed previously defined in equation 2.56, normalised with respect to the stator rotating field speed, is defined as *slip*:

$$s = \frac{(\omega_s - \omega_m)}{\omega_s} \tag{2.57}$$

When the $s = 1$, the rotor speed is null, and the rotor windings are coupled with the magnetic field at the stator current frequency. This corresponds to the starting condition of an IM.

If $s = 0$, no relative motion occurs, and the rotor angular speed is synchronous with the rotating magnetic field angular speed resulting in no torque production.

It is important to evidence that even if the rotor slips with respect to the rotating field, all electrical and magnetic quantities do rotate synchronously [13].

2.5.2 Synchronous Motor

A synchronous machine is an AC machine where the rotor rotates in synchronism with the rotating magnetic field generated by the three-phase currents system in the stator.

For sake of simplicity, lets consider a two-poles AC machine with three distributed windings, using one slot per pole. The positive phase currents are defined in accordance with the definition of the magnetic axis, having spatial displacement of 120 electrical degrees [13].

Since each phase produces a magneto-motive force (m.m.f.), employing a three-phase currents system, a rotating field can be generated. The north pole N of the m.m.f. is oriented along the related phase magnetic axis.

The fundamental component of the rotating field is sinusoidal, its rotating speed is called *synchronous speed* and it is related to the power supply frequency f_s and the number of pole pairs of the machine p as follows:

$$\frac{\omega}{p}\,[rad/s] \quad \rightarrow \quad n = \frac{60 \cdot f_s}{p}\,[rpm] \tag{2.58}$$

The rotor rotating speed is defined as:

$$\omega_m = \frac{\omega}{p}\,[rad/s] \tag{2.59}$$

In the following, an overview of the previously mentioned synchronous machines will be set out.

Surface-mounted Permanent Magnet Motor

The Surface-mounted Permanent Magnet machine can be characterised by both distributed or concentrated three-phase stator windings and a rotor, where magnets are placed on its surface.

Typically, the permanent magnets are rare-earth magnets (*NdFeB*, Neodymium Iron Boron or *SmCo*, Samarium Cobalt), necessary to obtain high torque density.

Rotor, in SPM machines, has an isotropic structure with respect to the (d,q) rotor axis, so the magnetic inductances along d-axis and q-axis are the same:

$$L_d = L_q = L_s \tag{2.60}$$

According to the typical convention applied for the SPM machines, the d-axis is defined by the permanent magnets north pole N, while the q-axis is in quadrature (90 electrical degrees away).

In SPM machines, the torque is an alignment torque generated by the interaction between the stator currents and the rotor flux:

$$T_{em} = \frac{3}{2} \cdot p \cdot \lambda_{dq} \wedge \mathbf{i}_{dq} = \frac{3}{2} \cdot p \cdot (\lambda_d \cdot i_q - \lambda_q \cdot i_d) \tag{2.61}$$

Substituting the fluxes obtained from the magnetic model (refer to section 2.5.3) of the SPM machine, the resulting electromagnetic torque is:

$$T_{em} = \frac{3}{2} \cdot p \cdot \lambda_m \cdot i_q \tag{2.62}$$

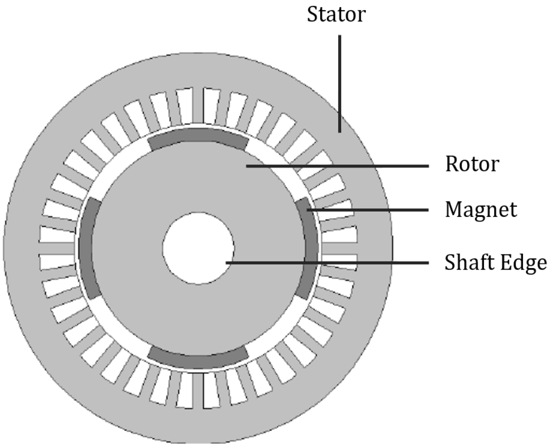


Figure 2.21: Cross-sectional view of a Surface-mounted Permanent Magnet Motor (Source: [14])

Synchronous Reluctance Motor

The main problem related to SPM machines is that rare-earth costs are subjected to oscillations.

Instead of employing permanent magnets, to reduce the cost, it can be taken advantage of the magnetic anisotropy using rotors where the magnetic reluctances along the magnetic axis are different.

The AC machines employing salient-pole rotors are called Synchronous Reluctance Motors or SynchRel or SyR.

The three-phase stator windings are always distributed windings.

According to the literature, the most common convention considers the *d*-axis defined as the minimum reluctance axis (so the maximum inductance axis), while the *q*-axis is the maximum reluctance axis (so the minimum inductance axis) and in quadrature with respect to the *d*-axis such that:

$$\Re_d < \Re_q \tag{2.63}$$

From this, the anisotropy factor can be defined:

$$\frac{L_d}{L_q} \tag{2.64}$$

In SyR machines, the electromagnetic torque is produced only due to rotor magnetic anisotropy, hence the only contribution is the reluctance torque. Substituting the magnetic model of the SyR machine (refer to equation (2.61 in section 2.5.3)), the electromagnetic torque is defined as:

$$T_{em} = \frac{3}{2} \cdot p \cdot (L_d - L_q) \cdot i_d \cdot i_q \tag{2.65}$$

It can be noticed that the torque is proportional to the difference between the inductances along *d*-axis and *q*-axis, thus the higher is the anisotropy factor, the higher will be the torque.

SyR machines with salient-pole rotor do not allow to reach high anisotropy factor values (typically $< 2 - 2.5$) and its performances result to be worst with respect to an IM.

To improve SyR performances, a more complex structure could be employed with rotor having 3 or 4 flux-barriers and hence higher anisotropy factor (typically between 6 and 10).

Such a structure needs to be mechanically validated with analysis; also the magnetic model becomes very complex due to magnetic saturation and cross-saturation phenomena.

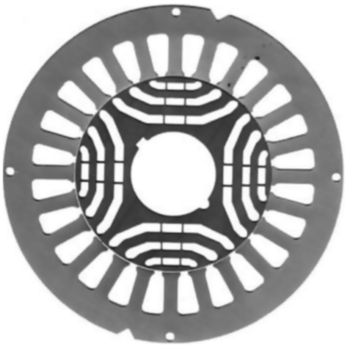


Figure 2.22: Cross-sectional view of a Synchronous Reluctance Motor (Source: [13])

PM-assisted Synchronous Reluctance Motor

Another problem related to the SyR machines are the performances in flux-weakening operations, where a decreasing power operating zone, before reaching the maximum speed, can be identified.

To solve this, permanent magnets are introduced in the rotor structure.

SyR machines using this type of rotor are called PM-assisted Synchronous Reluctance Motors (PM-SyR or PMASR). Magnets are added along the q-axis (this is related to the used convention) and they are typically ferrite (low cost magnets) generating a low flux λ_m.

In PMASR motors, the torque is the result of two contributions: the reluctance torque, being the dominant one and the alignment torque:

$$T_{em} = \frac{3}{2} \cdot p \cdot [(L_d - L_q) \cdot i_d \cdot i_q + \lambda_m \cdot i_d] \tag{2.66}$$

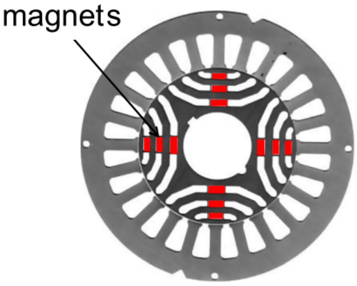


Figure 2.23: Cross-sectional view of a PM-assisted Synchronous Reluctance Motor (Source: [15])

PM-assisted SyR motors belongs to a group of motor called Internal Permanent Magnet (IPM) motors, in particular they are Multi-Layer IPM machines.

For nowadays EV applications, where the flux weakening region is the typical working one, the main design aim is to maximise the rotor anisotropy, reducing the amount of permanent magnet employed in the motor structure. These requirements make the multi-layer rotor structure the preferred one.

Internal Permanent Magnet Motor

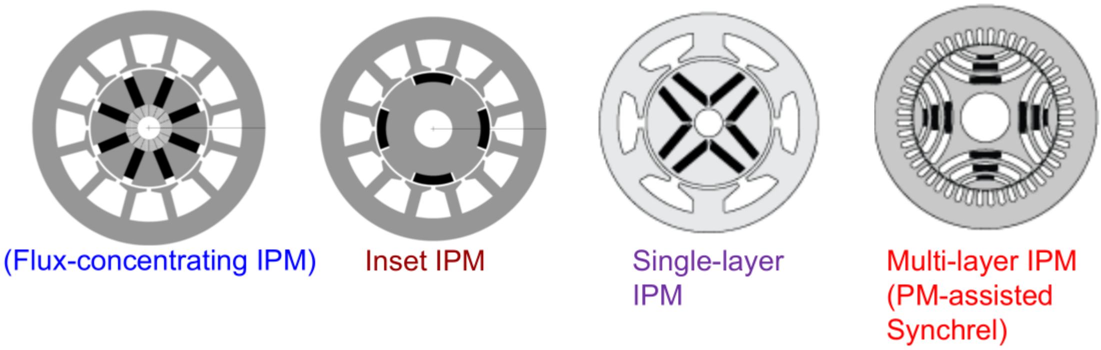


Figure 2.24: Classification of Internal Permanent Magnet Motors (Source [15])

For what concerns conventions, the d-axis is usually placed in correspondence of the main flux, thus in the direction of the minimum reluctance, while the q-axis is chosen in the direction of the PM north pole N. An alternative reference frame, frequently used in the literature is the one using the same convention of SPMSM, with the d-axis chosen in the direction of the PM north pole N and indicated in the figure below as (d', q'):

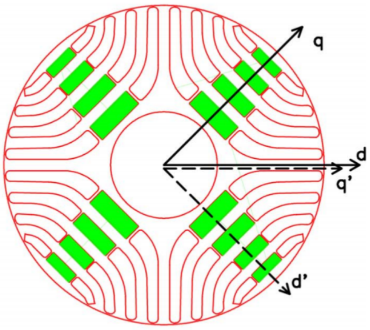


Figure 2.25: Possible IPM reference frames (Source: [16])

Considering the torque equation 2.66, for Flux-concentrating IPM, Inset IPM and Single-layer IPM, where the flux is mainly generated by rare-earth or ferrite magnets, the alignment torque is the dominant contribution in the torque generation; for Multi-layer IPM, the dominant contribution is given by the reluctance torque.

2.5.3 AC Machine Models

In section 2.3 the importance and the development of a mathematical model has been clarified.

In this section, a focus on AC motor modelling will be proposed.

For sake of generality, the following consideration will be in a first moment related to AC motors in a three-phase coordinate system. Later in the section, detailed electrical and magnetic models for each of the motors previously described will be addressed.

In a three-phase reference frame, the variables used for machine models are:

- Stator phase voltages defined in the stator reference frame: v_{as}, v_{bs}, v_{cs}

- Stator phase currents defined in the stator reference frame: i_{as}, i_{bs}, i_{cs}
- Rotor phase currents (for IM only) defined in the rotor reference frame: i_{ar}, i_{br}, i_{cr}
- Stator phase flux linkages with the stator windings defined in the stator reference frame: λ_{as}, λ_{bs}, λ_{cs}
- Rotor phase flux linkages with the stator windings (for IM only) defined in the rotor reference frame: λ_{ar}, λ_{br}, λ_{cr}

In both steady-state and transient conditions, it is useful to apply transformations on the electrical and magnetic quantities to simplify equations: a first transformation is applied to obtain from a three-phase system a two-phase equivalent system using Clarke transformation. A 3x3 transformation matrix $[T]$ is applied on the three-phase quantities obtaining:

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} = \begin{bmatrix} T \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \tag{2.67}$$

In the case of electrical machines, the zero-sequence component is not involved in the electromechanical energy conversion, thus can be neglected [17].

With this transformation, symmetrical three-phase sinusoidal variables are represented as a rotating vector in a complex plane with angular speed $\omega = 2\pi f$, being f the frequency (ω can be positive or negative).

Equations in a two-phase system can be further simplified by applying another transformation, the Park transformation, which moves the two-phase system in a fixed reference frame (typically the stator reference frame) to a new reference frame

(d,q) rotating with the synchronous speed ω previously defined. The transformation is performed by applying a 2x2 matrix:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} R(\theta) \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \tag{2.68}$$

in case of a positive rotation, where the matrix $[R(\theta)]$ is the direct rotational matrix, or

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} R(-\theta) \end{bmatrix} \cdot \begin{bmatrix} x_d \\ x_q \end{bmatrix} \tag{2.69}$$

in the case of negative rotation, where the matrix $[R(-\theta)]$ is the inverse rotational matrix.

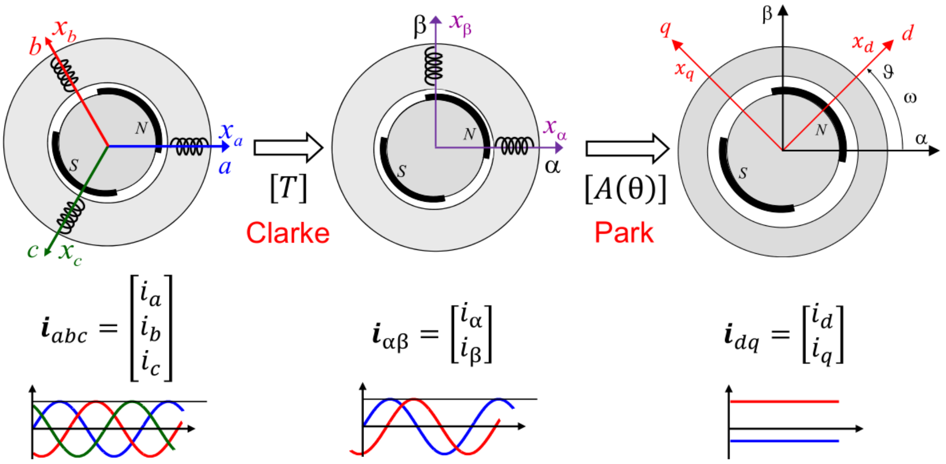


Figure 2.26: Example of phase currents of a SPM machine in the different described reference frames (Source: [17])

Referring to the models of AC machines, for sake of simplicity they are expressed in the (d,q) reference frame, where the three-phase quantities become DC quantities.

It is worth underlining the difference between synchronous machine (d,q) reference frame, defined by the established conventions, and asynchronous machine, where the (d,q) reference frame is defined by the rotor flux vector λ_r which needs to be estimated [13].

Electrical Models

In the following, the electrical models of the synchronous and asynchronous machines will be presented.

Consider a three-phase synchronous machine, the voltages equations are:

$$\begin{cases} v_a = R_s \cdot i_a + \frac{d\lambda_a}{dt} \\ v_b = R_s \cdot i_b + \frac{d\lambda_b}{dt} \\ v_c = R_s \cdot i_c + \frac{d\lambda_c}{dt} \end{cases} \tag{2.70}$$

For practical reasons, they can be written also in matrix form:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_s \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{2.71}$$

Applying the direct Clarke transformation:

$$\begin{bmatrix} T \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_s \cdot \begin{bmatrix} T \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} T \end{bmatrix} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{2.72}$$

thus,

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = R_s \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} \tag{2.73}$$

equivalent to the vector form:

$$\mathbf{v}_{\alpha\beta} = R_s \cdot \mathbf{i}_{\alpha\beta} + \frac{d}{dt}\lambda_{\alpha\beta} \tag{2.74}$$

where $\mathbf{v}$, $\mathbf{i}$ and λ are stator voltage vector, stator current vector and stator flux linkage vector defined in the stator (α,β) reference frame, respectively.

Applying the direct Park transformation to equation (2.73):

$$\begin{bmatrix} R[\theta] \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = R_s \cdot \begin{bmatrix} R[\theta] \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} R[\theta] \end{bmatrix} \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} \tag{2.75}$$

the voltage equations in the rotor (d,q) reference frame are obtained in matrix form,

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R_s \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} + \omega \cdot \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} \tag{2.76}$$

equivalent to the vector form:

$$\mathbf{v}_{dq} = R_s \cdot \mathbf{i}_{dq} + \frac{d}{dt}\lambda_{dq} + \omega \mathbf{J} \lambda_{dq} \tag{2.77}$$

For asynchronous machines, the voltages equations are more complicated because also rotor voltage equations are present:

$$\begin{cases} v_a = R_s \cdot i_a + \frac{d\lambda_a}{dt} \\ v_b = R_s \cdot i_b + \frac{d\lambda_b}{dt} \\ v_c = R_s \cdot i_c + \frac{d\lambda_c}{dt} \end{cases} \qquad \begin{cases} 0 = R_r \cdot i_{ar} + \frac{d\lambda_{ar}}{dt} \\ 0 = R_r \cdot i_{br} + \frac{d\lambda_{br}}{dt} \\ 0 = R_r \cdot i_{cr} + \frac{d\lambda_{cr}}{dt} \end{cases} \tag{2.78}$$

where R_r is the rotor phase resistance.

Applying direct Clarke transformation to the stator voltage equations, they are now referred to a reference frame (α,β) fixed with the stator:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = R_s \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} \quad \text{or} \quad \mathbf{v}_{\alpha\beta} = R_s \cdot \mathbf{i}_{\alpha\beta} + \frac{d}{dt}\lambda_{\alpha\beta} \tag{2.79}$$

and applying direct Clarke transformation to the rotor voltage equations, they are now referred to a rotating reference frame (α_r, β_r) synchronous with the rotor:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = R_r \cdot \begin{bmatrix} i_{r\alpha r} \\ i_{r\beta r} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_{r\alpha r} \\ \lambda_{r\beta r} \end{bmatrix} \quad \text{or} \quad \mathbf{0}_{\alpha\beta\mathbf{r}} = R_r \cdot \mathbf{i}_{\mathbf{r}\alpha\beta\mathbf{r}} + \frac{d}{dt}\lambda_{\mathbf{r}\alpha\beta}\mathbf{r} \tag{2.80}$$

Since the rotor reference frame is rotated of an angle θ_r with respect to the stator, by applying a inverse Park transformation (rotation matrix of an angle -θ_r), it is possible to refer the rotor voltage equations in the (α_r, β_r) reference frame to the stator reference frame (α, β).

Thus, the voltage equations for an IM, in the stator fixed frame are:

$$\begin{cases} \mathbf{v}_{\alpha\beta} = R_s \cdot \mathbf{i}_{\alpha\beta} + \frac{d}{dt}\lambda_{\alpha\beta} \\ \mathbf{0}_{\alpha\beta} = R_r \cdot \mathbf{i}_{\mathbf{r}\alpha\beta} + \frac{d}{dt}\lambda_{\mathbf{r}\alpha\beta} - \mathbf{J}\omega_r\lambda_{\mathbf{r}\alpha\beta} \end{cases} \tag{2.81}$$

where now R_r is the rotor equivalent phase resistance equivalent to the stator, ω_r is the electrical rotor speed and

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.82}$$

By applying the direct Park transformation, the voltages equations from fixed reference frame (α, β) are now defined in a synchronous reference frame defined by the rotor flux vector λ_r. To apply this transformation, the angle θ, as previously mentioned, must be estimated.

The obtained equations are:

$$\begin{cases} \mathbf{v}_{dq} = R_s \cdot \mathbf{i}_{dq} + \frac{d}{dt}\lambda_{dq} + \mathbf{J}\omega\lambda_{dq} \\ \mathbf{0}_{dq} = R_r \cdot \mathbf{i}_{rdq} + \frac{d}{dt}\lambda_{rdq} + \mathbf{J}(\omega - \omega_r)\lambda_{r\alpha\beta} \end{cases} \tag{2.83}$$

where *(ω - ω_r)* is defined as the electrical slip speed ω_{slip} (here for sake of simplicity the subscript s for the ω has been omitted, but it is equivalent to the ω_s defined at the beginning of the section).

Magnetic Models

The magnetic model represents the relation between current and flux [13]. As for electrical models, the magnetic model of synchronous machine will be first presented in phase coordinates and starting from an IPM motor magnetic model, a generic magnetic model for synchronous machines will be addressed.

Neglecting the mutual leakage couplings,

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = L_{ls} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} M_{aa} & M_{ab} & M_{ac} \\ M_{ba} & M_{bb} & M_{bc} \\ M_{ca} & M_{cb} & M_{cc} \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \lambda_m \cdot \begin{bmatrix} cos(\theta) \\ cos(\theta - \frac{2\pi}{3}) \\ cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{2.84}$$

where L_{ls} is the leakage phase inductance, M_{aa}, M_{bb} and M_{cc} are the magnetising self inductances and M_{ab}, M_{ac}, M_{ba}, M_{bc}, M_{ca}, M_{cb} are the magnetising mutual inductances.

Flux linkages depend on the following quantities (for IPM machines):

$$\begin{cases} \lambda_a = \lambda_a(i_a, i_b, i_c, \theta, \lambda_m) \\ \lambda_b = \lambda_b(i_a, i_b, i_c, \theta, \lambda_m) \\ \lambda_c = \lambda_c(i_a, i_b, i_c, \theta, \lambda_m) \end{cases} \tag{2.85}$$

where θ is the rotor position and λ_m is the magnets flux linkage, which is temperature dependent.

Since the rotor (except for SPM) is characterised by magnetic anisotropy, the magnetising inductances are not constant, but are rotor position dependent, thus a minimum and maximum value can be defined as follows:

$$M_{aa,min} = M_d = \frac{N^2}{\Re_d} \quad M_{aa,max} = M_q = \frac{N^2}{\Re_q} \tag{2.86}$$

Also, assuming sinusoidal winding distribution, the self magnetising inductances are depending on 2θ [13]:

$$\begin{cases} M_{aa} = M_{avg} + M_{\Delta} cos(2\theta) \\ M_{bb} = M_{avg} + M_{\Delta} cos(2\theta + \frac{2\pi}{3}) \\ M_{cc} = M_{avg} + M_{\Delta} cos(2\theta - \frac{2\pi}{3}) \end{cases} \tag{2.87}$$

where

$$M_{avg} = \frac{M_d + M_q}{2} \quad \text{is the average inductance} \tag{2.88}$$

$$M_{\Delta} = \frac{M_d - M_q}{2} \quad \text{is the anisotropy inductance} \tag{2.89}$$

While the mutual magnetising inductance are:

$$\begin{cases} M_{ab} = M_{ba} = -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta - \frac{2\pi}{3}) \\ M_{ac} = M_{ca} = -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta + \frac{2\pi}{3}) \\ M_{bc} = M_{cb} = -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta) \end{cases} \tag{2.90}$$

Defining the magnetising inductance matrix as:

$$[M(2\theta)] = \begin{bmatrix} L_{ls} + M_{avg} + M_{\Delta} cos(2\theta) & -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta - \frac{2\pi}{3}) & -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta + \frac{2\pi}{3}) \\ -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta - \frac{2\pi}{3}) & L_{ls} + M_{avg} + M_{\Delta} cos(2\theta + \frac{2\pi}{3}) & -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta) \\ -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta + \frac{2\pi}{3}) & -\frac{1}{2} M_{avg} + M_{\Delta} cos(2\theta) & L_{ls} + M_{avg} + M_{\Delta} cos(2\theta - \frac{2\pi}{3}) \end{bmatrix} \tag{2.91}$$

The magnetic model in the three-phase reference frame results to be very complex:

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = [M(2\theta)] \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \lambda_m \cdot \begin{bmatrix} cos(\theta) \\ cos(\theta - \frac{2\pi}{3}) \\ cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{2.92}$$

hence applying the direct Clarke transformation and neglecting the homopolar components,

$$\begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix} = \begin{bmatrix} L_{ls} + \frac{3}{2}M_{avg} + \frac{3}{2}M_\Delta cos(2\theta) & \frac{3}{2}M_\Delta sin(2\theta) \\ \frac{3}{2}M_\Delta sin(2\theta) & L_{ls} + \frac{3}{2}M_{avg} - \frac{3}{2}M_\Delta cos(2\theta) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \lambda_m \cdot \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix} \tag{2.93}$$

But the model still depends on 2θ; applying direct Park transformation, the model becomes independent from the rotor position:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_{ls} + \frac{3}{2}M_{avg} + \frac{3}{2}M_\Delta & 0 \\ 0 & L_{ls} + \frac{3}{2}M_{avg} - \frac{3}{2}M_\Delta \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \lambda_m \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.94}$$

and defining:

$$L_d = L_{ls} + \frac{3}{2}M_{avg} + \frac{3}{2}M_\Delta \tag{2.95}$$

$$L_q = L_{ls} + \frac{3}{2}M_{avg} - \frac{3}{2}M_\Delta \tag{2.96}$$

equation (2.94) simplifies in:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2.97}$$

where L_d is the d-axis inductance, L_q is the q-axis inductance and λ_m is the magnets flux.

It is important to underline that all the inductances are apparent inductances.

The magnetic model of IM is more complicated since the stator and flux linkages depend both on stator and rotor currents (as discussed for the electrical equations, the stator-rotor coupling is dependent on rotor position and the stator currents are referred to the fixed reference frame (α,β) while rotor currents are referred to the rotor rotating reference frame $(\alpha_\mathrm{r},\beta_r)$) [13]:

$$\begin{bmatrix}\lambda_a\\ \lambda_b\\ \lambda_c\end{bmatrix} = L_{ls}\cdot\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix} + \begin{bmatrix}M_{asas} & M_{asbs} & M_{ascs}\\ M_{bsas} & M_{bsbs} & M_{bscs}\\ M_{csas} & M_{csbs} & M_{cscs}\end{bmatrix}\cdot\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix} + \begin{bmatrix}M_{asar} & M_{asbr} & M_{ascr}\\ M_{bsar} & M_{bsbr} & M_{bscr}\\ M_{csar} & M_{csbr} & M_{cscr}\end{bmatrix}\cdot\begin{bmatrix}i_{ar}\\ i_{br}\\ i_{cr}\end{bmatrix} \tag{2.98}$$

$$\begin{bmatrix}\lambda_{ar}\\ \lambda_{br}\\ \lambda_{cr}\end{bmatrix} = L_{lr}\cdot\begin{bmatrix}i_{ar}\\ i_{br}\\ i_{cr}\end{bmatrix} + \begin{bmatrix}M_{aras} & M_{arbs} & M_{arcs}\\ M_{bras} & M_{brbs} & M_{brcs}\\ M_{cras} & M_{crbs} & M_{crcs}\end{bmatrix}\cdot\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix} + \begin{bmatrix}M_{arar} & M_{arbr} & M_{arcr}\\ M_{brar} & M_{brbr} & M_{brcr}\\ M_{crar} & M_{crbr} & M_{crcr}\end{bmatrix}\cdot\begin{bmatrix}i_{ar}\\ i_{br}\\ i_{cr}\end{bmatrix} \tag{2.99}$$

A more compact magnetic model can be used, as proposed in [13]:

$$\begin{cases}[\lambda]_{abc} = L_{ls}\cdot[i]_{abc} + M\cdot[B(0)]\cdot[i]_{abc} + M\cdot[B(\theta_r)]\cdot[i_r]_{abcr}\\ [\lambda_r]_{abcr} = L_{lr}\cdot[ir]_{abcr} + M\cdot[B(-\theta_r)]\cdot[i]_{abc} + M\cdot[B(0)]\cdot[i_r]_{abcr}\end{cases} \tag{2.100}$$

where the matrix *B(x)* is:

$$[B(x)] = \begin{bmatrix} cos(x) & cos(x+\frac{2\pi}{3}) & cos(x-\frac{2\pi}{3}) \\ cos(x-\frac{2\pi}{3}) & cos(x) & cos(x+\frac{2\pi}{3}) \\ cos(x+\frac{2\pi}{3}) & cos(x-\frac{2\pi}{3}) & cos(x) \end{bmatrix} \tag{2.101}$$

with $x =$ *0*, θ_r, -θ_r.

By applying the direct Clarke transformation and neglecting the homopolar components:

$$\begin{cases} [\lambda]_{\alpha\beta} = L_{ls} \cdot [i]_{\alpha\beta} + \frac{3}{2}M \cdot [i]_{\alpha\beta} + \frac{3}{2}M \cdot \begin{bmatrix} cos(\theta_r) & -sin(\theta_r) \\ sin(\theta_r) & cos(\theta_r) \end{bmatrix} \cdot [i_r]_{\alpha\beta r} \\ [\lambda_r]_{\alpha\beta r} = L_{lr} \cdot [ir]_{\alpha\beta r} + \frac{3}{2}M \cdot \begin{bmatrix} cos(\theta_r) & sin(\theta_r) \\ -sin(\theta_r) & cos(\theta_r) \end{bmatrix} \cdot [i]_{\alpha\beta} + \frac{3}{2}M \cdot [i_r]_{\alpha\beta r} \end{cases} \tag{2.102}$$

defining $L_m = \frac{3}{2} \cdot M$ as the magnetising inductance, $L_s = L_{ls} + L_m$ as the stator inductance and $L_r = L_{lr} + L_m$ as the rotor inductance, a more compact representation can be obtained:

$$\begin{cases} [\lambda]_{\alpha\beta} = L_s \cdot [i]_{\alpha\beta} + L_m \cdot [R(-\theta_r)] \cdot [i_r]_{\alpha\beta r} \\ [\lambda_r]_{\alpha\beta r} = L_m \cdot [R(\theta_r)] \cdot [i]_{\alpha\beta} + L_r \cdot [i_r]_{\alpha\beta r} \end{cases} \tag{2.103}$$

and applying an inverse Park transformation of an angle θ_r to the rotor quantities, the equations are now referred to the stationary frame (α,β):

$$\begin{cases} [\lambda]_{\alpha\beta} = L_s \cdot [i]_{\alpha\beta} + L_m \cdot [i_r]_{\alpha\beta} \\ [\lambda_r]_{\alpha\beta} = L_m \cdot [i]_{\alpha\beta} + L_r \cdot [i_r]_{\alpha\beta} \end{cases} \tag{2.104}$$

The direct Park transformation is applied,

$$\begin{cases} [\lambda]_{dq} = L_s \cdot [i]_{dq} + L_m \cdot [i_r]_{dq} \\ [\lambda_r]_{dq} = L_m \cdot [i]_{dq} + L_r \cdot [i_r]_{dq} \end{cases} \tag{2.105}$$

as can be noticed, the magnetic model does not depend on the reference frame, thus can be written in vector form as follows:

$$\begin{cases} \lambda_s = L_s \cdot \mathbf{i}_s + L_m \cdot \mathbf{i}_r \\ \lambda_r = L_m \cdot \mathbf{i}_s + L_r \cdot \mathbf{i}_r \end{cases} \tag{2.106}$$

Summarising, the magnetic models for all the presented AC machines are:

- SPM Magnetic Model

  The machine is theoretically isotropic, thus $L_s = L_d = L_q$, where L_s is defined as synchronous inductance.

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_s & 0 \\ 0 & L_s \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2.107}$$

- IPM Magnetic Model

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2.108}$$

- SynchRel Magnetic Model

  In the SynchRel machine no magnets are present, hence $\lambda_m = 0$.

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{2.109}$$

- PM-assisted SynchRel Magnetic Model

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ -\lambda_m \end{bmatrix} \tag{2.110}$$

Considering the synchronous machines only, apart from the non-saturated PMSM machines, saturation and cross-saturation phenomena make the flux-current behaviour more complex due to non-linearities, therefore models becomes much complicated.

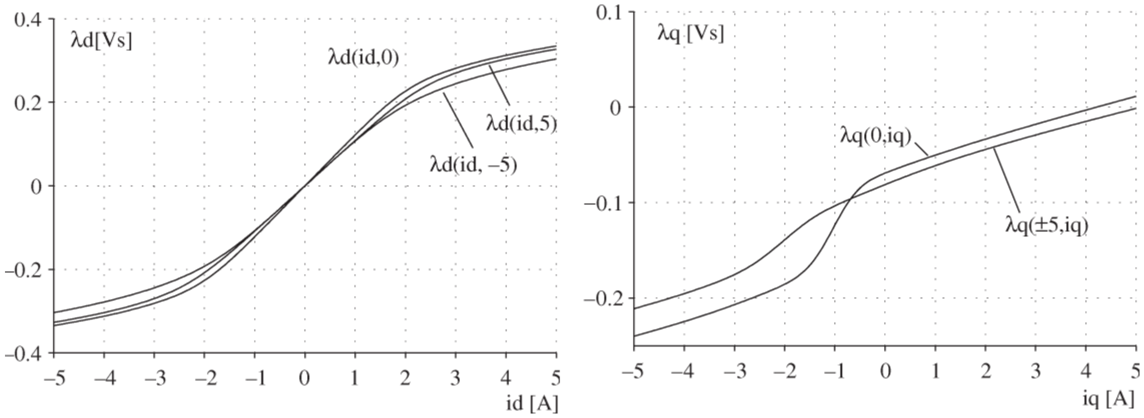
Referring to the rotor structure represented in figure 2.25, an example of flux-current relation is shown in figure 2.27.

Figure 2.27 shows the cross-saturation effect. In particular, in figures 2.27a and 2.27c the flux linkage λ_d with respect to the i_d current, assuming i_q as a parameter and the flux linkage λ_d with respect to the i_q current, assuming i_d as a parameter are presented, respectively.

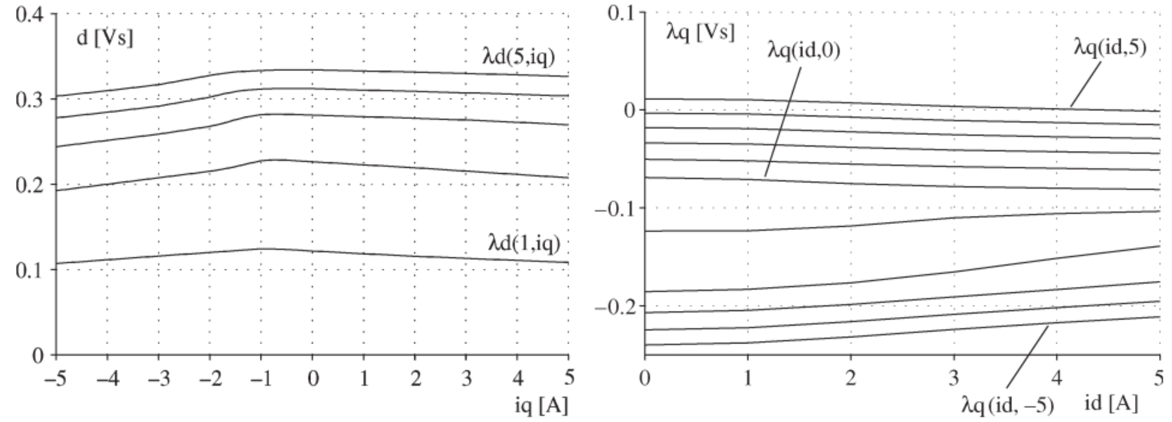
It can be observed how the cross-saturation effect is different depending on the values of i_q (positive or negative), even if in real applications i_q is always positive [16].

While, figures 2.27b and 2.27d represent the flux linkage λ_q with respect to the i_q current, assuming i_d as a parameter and the flux linkage λ_q with respect to the i_d current, assuming i_q as a parameter, respectively.

Even in these cases the effect of cross-saturation is highlighted, but the influence of i_q stands out when assuming negative values. This can be justified knowing that



(a) $\lambda_d(i_d, I_q)$ behaviour, for I_q = -5,0,+5 A. *(b)* $\lambda_q(I_d, i_q)$ behaviour, for I_d = -5,0,+5 A.



(c) $\lambda_d(I_d, i_q)$ behaviour, for I_d = 1,2,3,4,5 A. *(d)* $\lambda_q(i_d, I_q)$ behaviour, for I_q = -5,-4,-3,-2,-1,0,1,2,3,4,5 A.

Figure 2.27: Example of flux-current relationship for a PM-assisted Synchronous Reluctance machine (Source: [16])

magnets and stator magneto-motive force actions are summed and the flux changes its sign in the rotor ribs [16].

Assuming that the inductances shall all be considered as apparent inductances and defining L_{dq} and L_{qd} as the cross-saturation inductances, the models previously presented shall be modified as:

- Saturated SPM and IPM Magnetic Model

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & L_{dq} \\ L_{qd} & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2.111}$$

- Saturated SynchRel Magnetic Model

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & L_{dq} \\ L_{qd} & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{2.112}$$

- Saturated PM-assisted SynchRel Magnetic Model

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & L_{dq} \\ L_{qd} & L_q \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ -\lambda_m \end{bmatrix} \tag{2.113}$$

The vector form of the IM magnetic model presented in equation (2.106) can be modified in the following way, expliciting the relation between stator flux vector and rotor flux vector:

from

$$\lambda_r = L_m \cdot \mathbf{i}_s + L_r \cdot \mathbf{i}_r \tag{2.114}$$

the rotor current vector can be written as

$$\mathbf{i}_r = \frac{\lambda_r - L_m \cdot \mathbf{i}_s}{L_r} \tag{2.115}$$

and substituting in

$$\lambda_s = L_s \cdot \mathbf{i}_s + L_m \cdot \mathbf{i}_r \tag{2.116}$$

the stator flux vector becomes:

$$\lambda_s = L_s \cdot \mathbf{i}_s + \frac{L_m}{L_r} \cdot (\lambda_{\mathbf{r}} - L_m \cdot \mathbf{i}_s) \tag{2.117}$$

or

$$\lambda_s = (L_s - \frac{L_m^2}{L_r}) \cdot \mathbf{i}_s + \frac{L_m}{L_r} \cdot \lambda_r \tag{2.118}$$

Defining

$$\begin{cases} \sigma = 1 - \frac{L_m^2}{L_s \cdot L_r} & \text{as the total leakage factor} \\ k_r = \frac{L_m}{L_r} \simeq 1 & \text{as the rotor coupling factor} \end{cases} \tag{2.119}$$

the stator flux vector becomes:

$$\lambda_s = \sigma L_s \cdot \mathbf{i}_s + k_r \cdot \lambda_r \tag{2.120}$$

Dynamic Models in the (*d,q*) Frame

In this section, electrical and mechanical dynamics of the AC machines will be presented.

Starting from synchronous machines, two types of dynamic models can be defined:

- Flux-Based (d,q) Model

  Recalling voltage equation from equation 2.77 and the magnetic model from equation (2.111) to (2.113), where cross-saturation effects are considered, the latter can be written in a more compact form as:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = [\lambda(i_d, i_q)] \tag{2.121}$$

where [λ] includes PM flux (if exists) and other saturation effects. Considering also the generic torque equation:

$$T_{em} = \frac{3}{2} \cdot p \cdot \lambda \wedge \mathbf{i} \tag{2.122}$$

The flux-based model in (d,q) frame is:

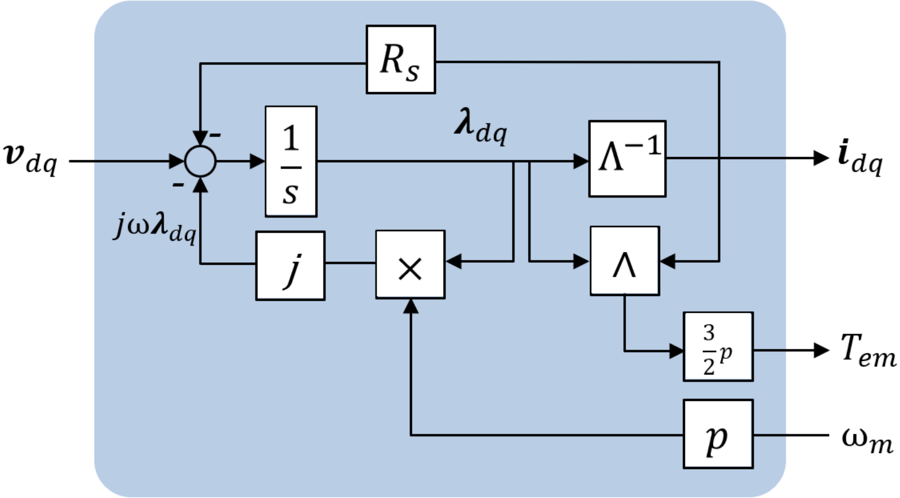


Figure 2.28: Flux-based (d,q) model of a synchronous machine

where the inputs are the voltages $\mathbf{v}_{dq}$ and the mechanical speed ω_m, while the outputs are the currents $\mathbf{i}_{dq}$ and the electromagnetic torque T_{em}.

- Current-based (d,q) Model

  As before, considering voltage equation from equation (2.77), it can be modified introducing incremental inductances:

$$\mathbf{v}_{dq} = R_s \cdot \mathbf{i}_{dq} + \frac{\partial \lambda_{dq}}{\partial \mathbf{i}_{dq}} \cdot \frac{d\mathbf{i}_{dq}}{dt} + \omega \mathbf{J}(\lambda_m + [L] \cdot \mathbf{i}_{dq}) \tag{2.123}$$

where $\frac{\partial \lambda_{\mathrm{dq}}}{\partial \mathbf{i}_{\mathrm{dq}}}$ represent the incremental inductance matrix obtained as described in [16] and it is indicated as:

$$[L]_{inc} = \begin{bmatrix} \frac{\partial \lambda_d}{\partial i_d} & \frac{\partial \lambda_d}{\partial i_q} \\ \frac{\partial \lambda_q}{\partial i_d} & \frac{\partial \lambda_q}{\partial i_q} \end{bmatrix} = \begin{bmatrix} l_d(i_d, i_q) & l_{dq}(i_d, i_q) \\ l_{qd}(i_d, i_q) & l_q(i_d, i_q) \end{bmatrix} \tag{2.124}$$

while $[L]$ represents the apparent inductance matrix. The considered magnetic model is:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = [L] \cdot \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2.125}$$

Referring to the same torque equation as in the previous model, the current-based model is:

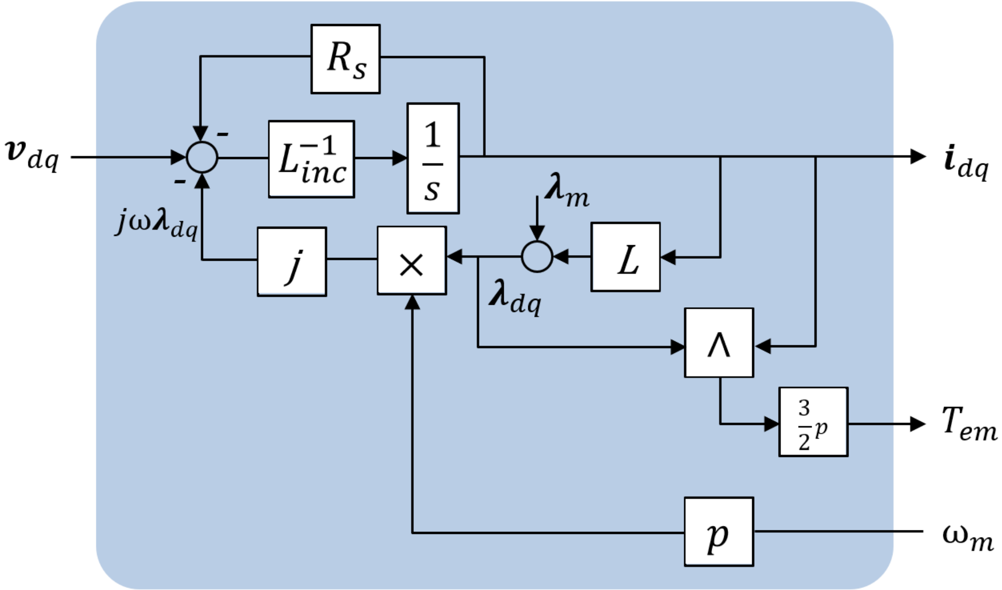


Figure 2.29: Current-based (d,q) model of a synchronous machine

where the inputs are the voltages $\mathbf{v}_{dq}$ and the mechanical speed ω_m, while the outputs are the currents $\mathbf{i}_{dq}$ and the electromagnetic torque T_{em}.

The current-based models result to be more complex compared to the flux-based models since both apparent and incremental inductances shall be taken into account.

Particular attention should be dedicated to the flux-based model of the induction machine in the (α,β) frame which is equivalent to a model implemented in the (d,q) frame since the IM is isotropic. It is obtained by eliminating the stator and rotor currents from the voltage equations defined in equation (2.81):

$$\begin{cases} \mathbf{v}_{\alpha\beta} = R_s \cdot \mathbf{i}_{\alpha\beta} + \frac{d}{dt}\lambda_{\alpha\beta} \\ \mathbf{0}_{\alpha\beta} = R_r \cdot \mathbf{i}_{\mathbf{r}\alpha\beta} + \frac{d}{dt}\lambda_{\mathbf{r}\alpha\beta} - j\omega_r\lambda_{\mathbf{r}\alpha\beta} \end{cases} \tag{2.126}$$

Therefore, stator and rotor currents are obtained from the magnetic model of equation (2.106).

Considering the magnetic model in equation (2.104)

$$\begin{cases} \lambda_{\alpha\beta} = L_s \cdot \mathbf{i}_{\alpha\beta} + L_m \cdot \mathbf{i}_{r\alpha\beta} \\ \lambda_{r\alpha\beta} = L_m \cdot \mathbf{i}_{\alpha\beta} + L_r \cdot \mathbf{i}_{r\alpha\beta} \end{cases} \tag{2.127}$$

it is possible to write:

$$\begin{cases} \mathbf{i}_{\alpha\beta} = \frac{\lambda_{\alpha\beta} - k_r \cdot \lambda_{r\alpha\beta}}{\sigma L_s} \\ \mathbf{i}_{r\alpha\beta} = \frac{\lambda_{r\alpha\beta} - k_s \cdot \lambda_{\alpha\beta}}{\sigma L_r} \end{cases} \tag{2.128}$$

From equation (2.81), the stator voltage equation becomes:

$$\mathbf{v}_{\alpha\beta} = \frac{R_s}{\sigma L_s} \cdot (\lambda_{\alpha\beta} - k_r \cdot \lambda_{r\alpha\beta}) + \frac{d\lambda_{\alpha\beta}}{dt} \tag{2.129}$$

thus

$$\tau_s^{'} \frac{d\lambda_{\alpha\beta}}{dt} + \lambda_{\alpha\beta} = k_r \cdot \lambda_{r\alpha\beta} + \tau_s^{'} \cdot \mathbf{v}_{\alpha\beta} \tag{2.130}$$

where

$$k_s = \frac{L_m}{L_s},\ k_r = \frac{L_m}{L_r},\ \tau_s^{'} = \frac{\sigma L_s}{R_s} \tag{2.131}$$

Considering again equation (2.81), the rotor voltage equation becomes:

$$\mathbf{0}_{\alpha\beta} = \frac{R_r}{\sigma L_r} \cdot (\lambda_{r\alpha\beta} - k_s \cdot \lambda_{\alpha\beta}) + \frac{d\lambda_{r\alpha\beta}}{dt} - \omega_r \mathbf{J} \lambda_{r\alpha\beta} \tag{2.132}$$

thus

$$\tau_r^{'} \frac{d\lambda_{r\alpha\beta}}{dt} + \lambda_{r\alpha\beta} = k_s \cdot \lambda_{\alpha\beta} + \tau_r^{'} \cdot \omega_r \mathbf{J} \lambda_{r\alpha\beta} \tag{2.133}$$

where

$$\tau_r^{'} = \frac{\sigma L_r}{R_r} \tag{2.134}$$

For what concerns the electromagnetic torque, from:

$$T_{em} = \frac{3}{2} \cdot p \cdot \lambda_{\alpha\beta} \wedge \mathbf{i}_{\alpha\beta} \tag{2.135}$$

and considering equation (2.128),

it becomes:

$$T_{em} = -\frac{3}{2} \cdot p \cdot \frac{k_r}{\sigma L_s} \lambda_{\alpha\beta} \wedge \lambda_{r\alpha\beta} \tag{2.136}$$

thus

$$T_{em} = \frac{3}{2} \cdot p \cdot \frac{k_r}{\sigma L_s} \lambda_{r\alpha\beta} \wedge \lambda_{\alpha\beta} \tag{2.137}$$

and

$$T_{em} = \frac{3}{2} \cdot p \cdot \frac{k_r}{\sigma L_s} (\lambda_{r\alpha} \lambda_{\beta} - \lambda_{r\beta} \lambda_{\alpha}) \tag{2.138}$$

The resulting model is:

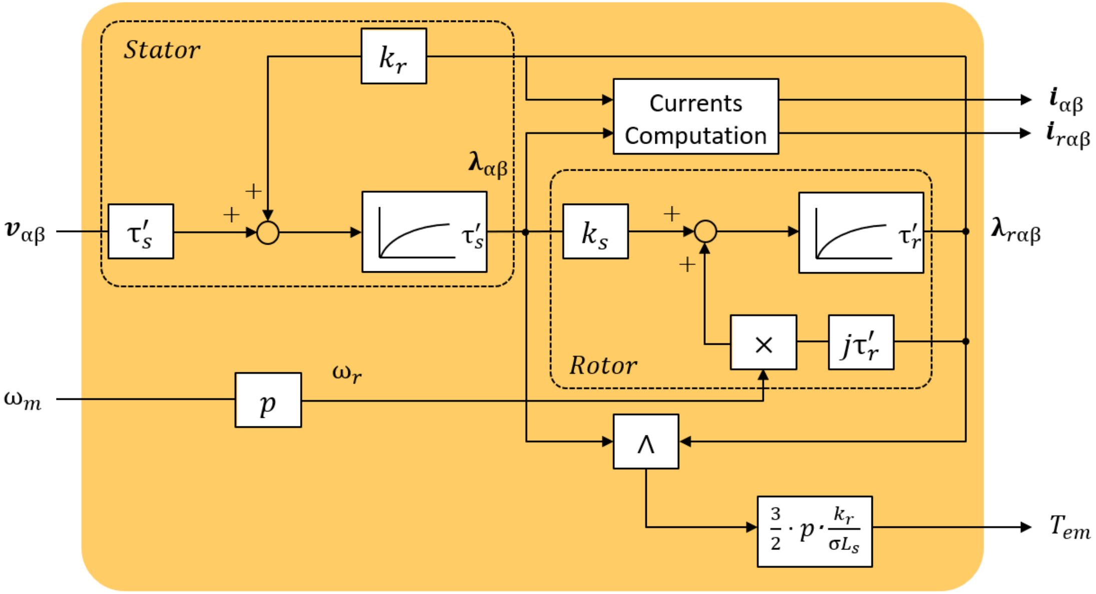


Figure 2.30: Flux-based (α,β) model of an induction machine

where the inputs are the voltages $\mathbf{v}_{\alpha\beta}$ and the mechanical speed ω_m, while the outputs are the currents $\mathbf{i}_{\alpha\beta}$, $\mathbf{i}_{r\alpha\beta}$ and the electromagnetic torque T_{em}.

2.6 Fundamentals of Control Systems

Nowadays control systems are a fundamental component of any dynamic system and any engineering sector. Control theory has been developed during the previous century and the commonly used today are classical control theory, modern control theory and robust control theory [18].

A typical control system is characterised by the following elements:

- Reference value: the desired value that the system shall reach;
- Sensors or observers: when a feedback control system is implemented (see later), the output of the system shall be monitored and compared with the

reference value. When a quantity measure cannot be achieved by direct observation, state observers can be implemented, and an estimation of the desired quantity obtained;

- Plant: the system to be controlled characterised by a transfer function, a mathematical model representing the input-output relationship between variables.
- Controller: even the controller is defined by a transfer function which has been properly designed according to defined specifications and desired results. The controller receives as input the difference between reference value and measured (or estimated) value, the error, and produces as output a value provided as input to the system to be controlled or to the actuator acting on the plant.

Two typical configurations of control systems are basically implemented:

- Open-loop

  In open-loop control systems, the reference value is directly imposed to the controller producing a defined output for each type of desired input. In this type of systems, calibration has an extreme importance since the output is not measured or compared with the reference input and this makes the system weak when disturbances are present. Elsewhere, open-loop system implementation is very simple and cheap, and it is widely used for applications where no particular precision is required.

- Closed-loop

  In closed-loop control systems (also called feedback control systems) as previously described, the reference value is compared to the system's measured (or estimated output) to obtain an error fed to the controller. Due to this configuration, a system is more robust with respect to an open-loop control system when external disturbances and or parameters variations or uncertainty are present. On the other side, when implementing a feedback control system, stability problems arise since such a system tends to over correct errors causing oscillations which may affect the plant behaviour or damage it [18]. Another drawback of closed-loop control systems is that their implementation is expensive due to the higher number of required components.

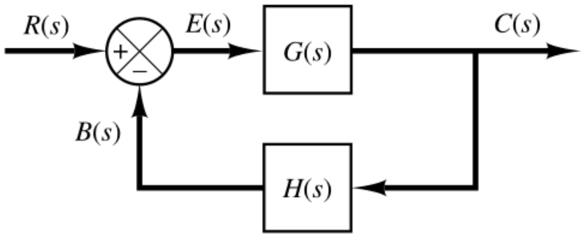


Figure 2.31: Closed-loop control system (Source: [18])

To evaluate a control system's performances, the response to different input signals is analysed. Typical test signals are the unit-step or the unit-ramp signals. The former represents one of the worst situations a dynamic system could experience since it is an abrupt change in the input; the latter can be compared to a linear change in the system input.

Assuming a simple second-order system, the typical step response is shown in figure 2.32.

It can be observed that the best system behaviour is when the output reaches the reference in the minimum possible time, in this case no oscillations are present, and it is said that the system is critically damped. When the response is slow the system is said to be over-damped while when the output reaches the reference with the presence of overshoot and oscillations, the system is said to be under-damped.

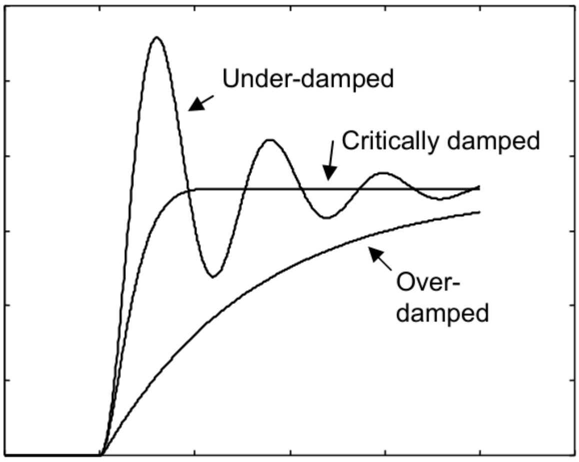


Figure 2.32: Step responses (Source: [18]).

2.6.1 PID Controllers

The most common control system, widely used in industrial applications for its ease in tuning and in application to any system, is the PID controller.

PID controllers have been developed in the 1940s and they have been subjected to many changes during the following years. Nowadays, they are typically implemented by using a microcontroller or microprocessor.

In this paragraph PID controller will be first presented from a mathematical point of view and later the digital implementation will be addressed.

The PID controller is based on the effects of three actions from which the name: Proportional, Integral and Derivative. The mathematical PID expression is:

$$u(t) = K_p\Big(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\Big) \tag{2.139}$$

as it can be observed the three terms are respectively proportional to the error, to the integral of the error and to the derivative of the error.

The parameters of the controller are the proportional gain K_p, the integral gain $K_i = \frac{K_p}{T_i}$, being T_i the integral time and the derivative gain $K_d = K_p \cdot\ T_d$, being T_d the derivative time, while *e(t)* is the error fed to the controller and obtained as the difference between the reference input and the output $e(t) = r(t) - y(t)$.

An alternative form of the PID expression is the representation as a transfer function in Laplace domain:

$$G(s) = K_p\Big(1 + \frac{1}{sT_i} + sT_d\Big) \tag{2.140}$$

It is also possible not to use all the three terms at the same time; as will be discussed later, in eDrives applications only PI controller are usually employed.

Each of the previously described terms has a precise effect on the PID output. By increasing the proportional gain, the error will decrease but this will increase system's oscillations. Considering also the integral action, increasing it (by reducing the integral time T_i), will make the system react faster, with respect to the use of the proportional action only, introducing a rejection to first-order disturbances. Even in this case, if the integral action increases, also oscillations increase.

By adding also the derivative action, the system will result to be damped as the derivative action increases by increasing the derivative time T_d till a certain value is reached. If T_d is too large, this will cause oscillations in the system, in particular if T_d is larger than 1/6 of the period [19].

Thus, the controller is based on the action of these terms whose parameters can be modified to change its behaviour. This procedure is called tuning and will be investigated in section 2.7.2 focusing on PI current controllers.

The reason why typically PI controller are used instead of PID, is because the differentiation is easily affected by noise and in eDrives applications this is what in most cases happens. Considering a step disturbance, its derivative is an impulse which will be present in the control system, and it is an undesired effect. Another reason why derivative action is not used is because it is very time consuming when the controller is implemented in a microcontroller or microprocessor.

Another important aspect related to PID controllers is the wind-up phenomenon affecting integral action. This is an effect caused by the interaction between integral action and saturations related to the actuators controlled by the PID. What happens when saturation of the control signal is reached is that the feedback system starts to behave as an open-loop system, but the error will continue to be integrated leading to very large values of the integral action, being very far from the steady-state and causing large transients [19].

To avoid wind-up phenomenon, one of the proposed methods is to introduce limits on the reference value variations, in this way the controller output will not reach actuators limits.

Another method is to disable the integral action when the output saturates or to limit the changing rate of the control signal.

Digital Implementation of PID Controllers

As stated before, nowadays PID controllers are mainly implemented in digital systems such as microcontrollers and microprocessors. In the following, particular attention will be dedicated to power electronic application controllers. In this dissertation only a first category of architecture will be discussed corresponding to the microcontrollers and digital signal processors (DSP), but field programmable gate arrays (FPGA) could be also used [20].

The use of digital controller implies also the use of analog to digital converters (ADC) since signals coming from an eDrive (i.e., DC-link voltage, phase currents, rotor angular position) are typically continuous-time signals; the output of the controller is discrete, thus it has to be converted again in an analog signal, so a feedback control system requires also digital to analog converters (DAC).

The use of ADC and DAC adds complexity to a system and delays, but the advantages are considerable: first of all flexibility; working with digital controllers, in fact allows reusability of the devices since the algorithm is implemented by a software which can be easily modified and loaded on the microcontroller memory.

In the previous section, PID controllers have been presented in the continuous-time domain, but in order to implement digital controllers, discretization is necessary and one of the main concept is the discretization strategy to be used when it is necessary to replace continuous-time computation of integrals with a numerical approximation.

The Z – transform shall be applied to work in the discrete time domain.

In the literature, the suggested integration methods are the Euler integration and trapezoidal integration method [21].

Focusing on the Euler approximation method, lets consider a curve:

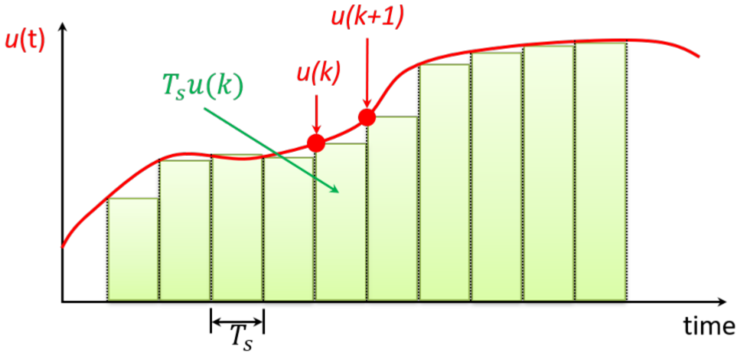


Figure 2.33: Euler approximation method: graphical interpretation (Source: [22]).

the area under this curve can be approximated as the sum of rectangular areas and defined as the discrete-time integral of *u(t)*.

Euler approximation method can be implemented in two different fashions: forward and backward integration.

The Euler approximations in the Z-domain are presented below [21]:

For control applications in real-time system, the backward Euler method is used because the sampled signal is always delayed with respect to the continuous-time signal.

What happens after discretization is that distortion phenomena can be observed,

Method	Z-form
Backward Euler	$s = \frac{z-1}{z \cdot T_S}$
Forward Euler	$s = \frac{z-1}{T_S}$

Table 2.1: Euler approximation methods

also called frequency warping effect [21]. Euler method is preferred when the ratio between the sampling frequency and the frequency of interest is greater than 20; if this condition is respected, the distortion will be lower than the 3%.

For sake of simplicity, a PI controller will be considered and its digital implementation, based on the backward Euler approximation is:

$$\begin{cases} m_I(k) = K_I \cdot T_s \cdot \epsilon_I(k) + m_I(k-1) \\ m(k) = m_p(k) + m_I(k) = K_P \cdot \epsilon_I(k) + m_I(k) \end{cases} \tag{2.141}$$

where T_s is the sampling time and the index k represent kT_s in a compact form.

It is worth underlining that the proportional term in the digital domain is the same as in the continuous time domain, while the integral term is simply obtained by multiplying the integral gain by the sampling time.

Therefore:

$$\begin{cases} K_{I,dig} = K_I \cdot T_s \\ K_{P,dig} = K_P \end{cases} \tag{2.142}$$

2.7 Control of AC Drives

Nowadays AC machines main applications are not limited to those operating at constant speed, but with the progresses in power electronics and computer science fields, they are covering wide areas: from industrial applications to traction applications where variable speed is required.

This stimulated the development of control strategies to face all the different operating situations of an AC motor.

The main drive functions are to control quantities like torque (or force), speed or position.

When speed control is applied, a reference speed shall be set and the reference torque is imposed by a regulator for the speed loop. This regulator is placed in cascade with the torque control loop.

Focusing on the torque control, it usually represents the inner loop of an AC drives control scheme; it receives the torque reference and provides as output the reference three-phase voltages for the inverter.

Torque reference may be also obtained (other than from the speed loop) from the torque equations; while voltage equations combined with the current models can be used to tune the PI current regulators in the torque control loop.

These control loops use the motor models in (d,q) reference frame presented in section 2.5.3.

In the following, the different variable frequency techniques will be presented. Basically, it is possible to distinguish between:

- Scalar Control

  This technique is defined as scalar because only one variable is controlled, thus allowing only one degree of freedom. It is a sort of steady-state control method where it is assumed that currents and voltages are stationary, and the dynamic is neglected.

  Typical scalar control strategies for AC drives are V/f control and slip control.

- Vector Control

This technique is based on two controlled variables, hence two degrees of freedom, represented as space vectors allowing high performances in both steady-state and transients. Depending on the controlled variables, different control strategies can be implemented: *Field–Oriented Control* (FOC) if the currents in the (d,q) reference frame are controlled, *Direct Flux Vector Control* (DFVC) if flux amplitude and quadrature currents in the stator reference frame are controlled (d_s,q_s) and *Direct Torque Control* (DTC) acting on the stator flux and the torque.

Another control strategy is available operating in open-loop for the outer one, but with a closed-loop current control, the I–f control.

2.7.1 V/f Control

The V/f control is a technique implemented for IM motors only, not for synchronous motors since it does not guarantee the synchronism between rotor angular velocity and the rotating magnetic field angular velocity (if this condition is not respected it may lead to instability of the synchronous machine).

As early stated, the V/f control is open-loop, thus no current regulators are adopted and the voltage is directly imposed and controlled as a vector. From a desired speed ω^*, a constant voltage reference is obtained in the (d,q) reference frame rotating at ω^*.

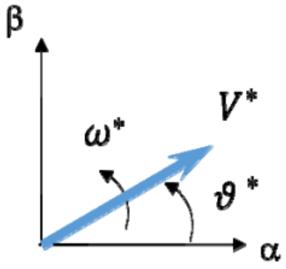


Figure 2.34: Voltage vector rotating at speed ω^* (Source: [22]).

Thus, the stator voltage is:

$$\begin{cases} v_d^* = V_0 + k\omega^* \\ v_q^* = 0 \end{cases} \tag{2.143}$$

where k is the rated stator flux and V_0 is used to compensate for the resistive voltage drop and to flux the machine even at standstill.

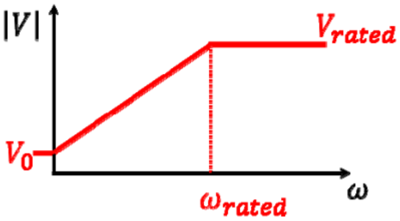


Figure 2.35: V/f control law (Source: [22]).

It can be observed that the voltage amplitude is saturated at a value corresponding to the rated voltage.

Since no feedback position or speed is adopted, the (d,q) reference frame coordinates are open-loop determined from the reference speed by integrating it.

The control scheme is shown below:

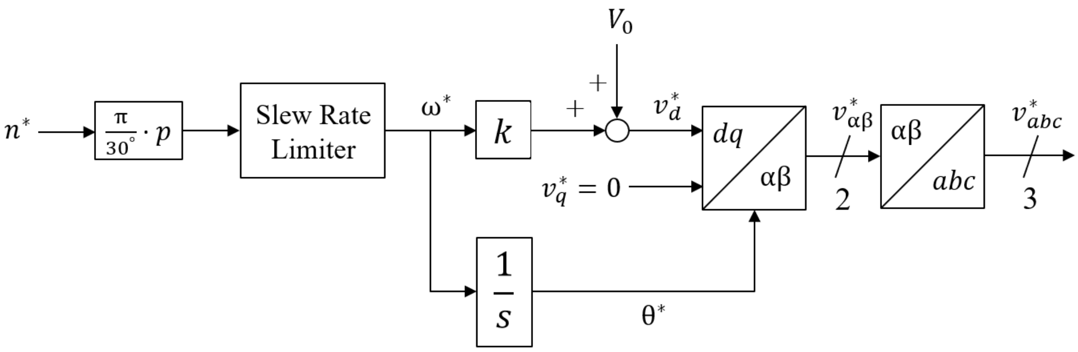


Figure 2.36: V/f control scheme

where the *ramp* block has to be intended as a slew rate limiter to avoid overcurrent.

Even if the voltage is vector controlled, it is based on amplitude and phase, which are both quantities depending on the reference frequency f_o, hence only one degree of freedom is present; for this reason the V/f control is a scalar control and not a vector control technique.

The V/f control is typically employed to test inverter correct operating conditions (i.e., modulation) and calibrations and it is performed by using as load an IM even if the final objective is to control a completely different AC drive as a synchronous machine.

It is also used for applications where no particular accuracy is required and it implements flux-weakening automatically when the condition $\omega > \omega_{\text{rated}}$ applies.

On the other side, this technique is very slow in terms of dynamic response and low efficient.

2.7.2 I-f Control

The first type of vector control addressed is the I-f control, where a reference current is imposed in the (d,q) reference frame with a constant amplitude value, rotating at a reference speed ω^*.

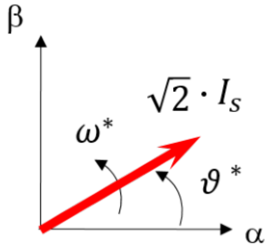


Figure 2.37: Current vector rotating at speed ω^* (Source: [22]).

The amplitude is guaranteed by the current closed-loop control, realised by means of PI regulators (one for the current component along the *d*-axis and one for the current component along *q*-axis). The speed is open-loop imposed (as for V/f control).

The reference values are used to impose a rotating vector with amplitude I_s [A] rms:

$$\begin{cases} i_d^* = \sqrt{2} \cdot I_s \\ i_q^* = 0 \end{cases} \tag{2.144}$$

The control scheme is shown below:

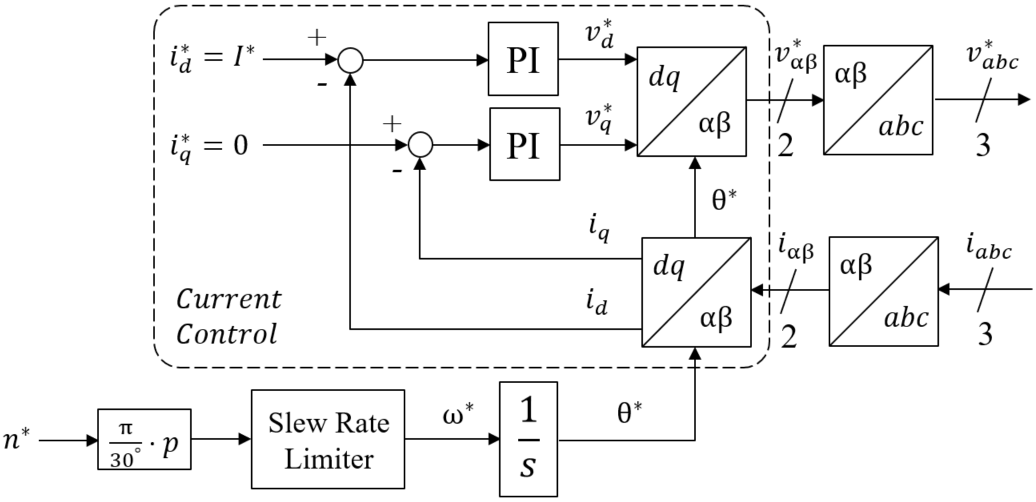


Figure 2.38: I-f control scheme

The performances of the I-f vector control are good if working in no-load conditions but it is subjected to high instability when under load. For this reason this type of control strategy is not used in practice but it is perfect to test (always in no-load conditions) the flux estimators and observers and in particular for tuning the PI current regulators (in (d,q) reference frame).

When working in no-load conditions, the current vector is in phase with the stator and rotor flux vectors (considering an IM), this allows to verify that the observed flux phase is correct. While, if considering SPM, the current is aligned to the magnet direction or to the maximum inductance direction if considering SynchRel machines.

An analogy between the I-f control and the Field-Oriented Control can be highlighted: in no-load conditions, the reference (d,q) directions are the same. The situation changes when under load.

In fact, under load, a phase shift is present between current and rotor flux, for IM. This lets the machine suddenly reach the limit torque; if the limit is overcome, the control is lost.

Without loss of generality, for synchronous machines, in an IPM machine, the presence of load torque causes a phase shift between current and rotor d-axis. Even in this case, once the limiting condition is reached (corresponding to the MTPA angle) and overcome, the control is lost.

Losing the control means that the imposed reference current is no more followed by the system.

PI Current Controller Tuning

In section 2.6.1 PID controllers have been introduced, here a real application will be presented.

As mentioned, the I-f control is also used to tune PI current regulators; the procedure will be described in the following, without loss of generality, considering an IPM machine (since it is the selected machine for this thesis work). It can be easily applied to all the other AC drives described in section 2.5, by simply changing the considered motor electrical and magnetic model.

From voltage equation (2.77) and the simplified IPM magnetic model from equation (2.108) (considering the SPMSM convention), it follows:

$$\begin{cases} v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \\ v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega \lambda_m + \omega L_d i_d \end{cases} \tag{2.145}$$

applying Laplace transform and collecting the current terms (the s-dependence has been neglected for simplicity)

$$\begin{cases} V_d = I_d(R_s + sL_d) - \omega L_q I_q \\ V_q = I_q(R_s + sL_d) + \omega\lambda_m + \omega L_d I_d \end{cases} \tag{2.146}$$

From the flux-based model described in section 2.5.3 and neglecting the saturation and cross-saturation effects, the IPM model can be represented as:

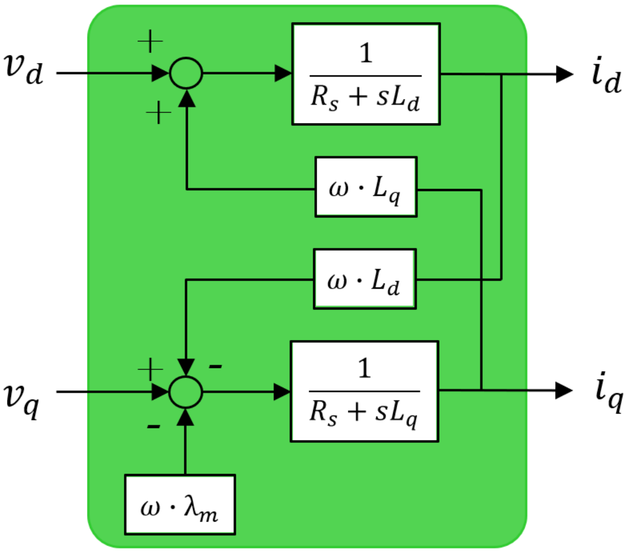


Figure 2.39: IPMSM model with SPMSM convention

where the other equations terms as to be intended as additive disturbances that shall be feedforward compensated in the PI current regulator.

For the PI tuning, the complete current loops shall be considered, including the PI transfer function and the converter transfer function. For the latter, assuming that the sampling strategy (that will be explained in chapter 5) *1 Sample, 1 Refresh* or simply *1S1R* will be used, it can be demonstrated ([23]) that the inverter can be modelled as follows:

$$\frac{V(s)}{V^{*}(s)} = \frac{1}{1 + s\tau_{inverter,delay}} \tag{2.147}$$

where the delay includes transport and sampling delays, and for the *1S1R* strategy it is computed as follows:

$$\tau_{inverter,delay} = T_{sw} + \frac{T_{s}w}{2} = 1.5 \cdot T_{sw} = \frac{1.5}{f_{sw}} \tag{2.148}$$

This can be justified knowing that the inverter output voltages are a pulse train with different amplitudes, but only after half-period the average value is the desired one.

Therefore, from the PI transfer function of equation (2.140), the current loops are:

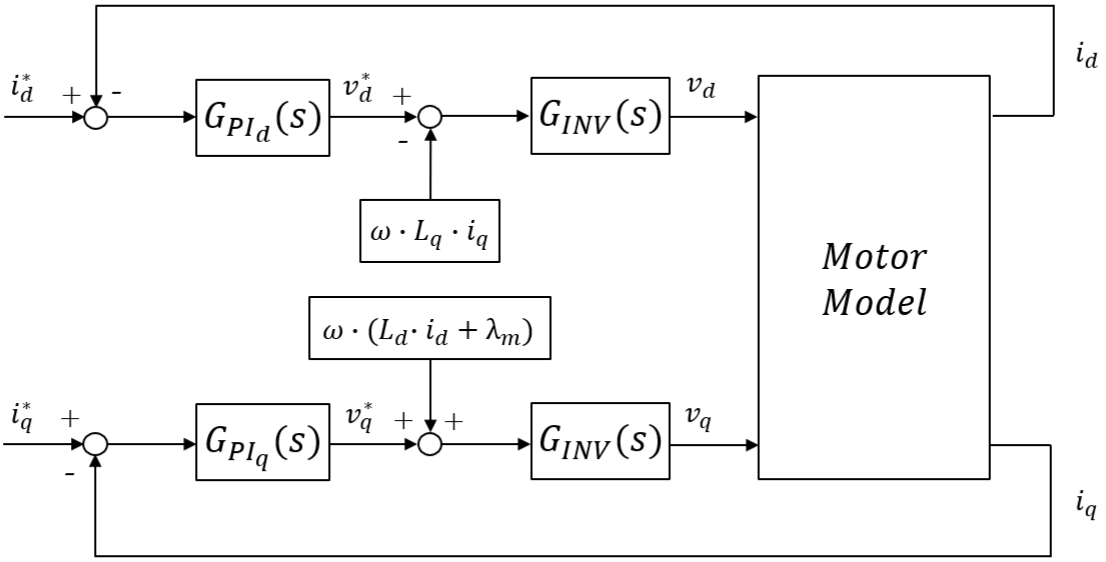


Figure 2.40: Current loops with feedforward

For sake of simplicity, but without loss of generality, the procedure is described for both current loops, since they have same open-loop transfer function, considering a generic inductance average value L and neglecting feedforwards:

$$\begin{aligned}G_{current,OL}(s) &= G_{PI}(s) \cdot G_{INV}(s) \cdot G_{IPM}(s) = \\ &= \frac{K_p}{s} \cdot \left(s + \frac{K_i}{K_p}\right) \cdot \frac{1}{1 + s\tau_{inverter,delay}} \cdot \frac{1}{L} \cdot \frac{1}{s + \frac{R_s}{L}}\end{aligned} \tag{2.149}$$

The open-loop transfer function presents:

- A zero from the PI current regulator in: $z_{PI} = -\frac{K_i}{K_p}$
- A pole from the PI current regulator integral action in the origin
- An electrical pole from the IPM dynamic model in $p_{IPM} = -\frac{R_s}{L}$
- A pole from the inverter dynamic model in $p_{INV} = -\frac{1}{\tau_{inverter,delay}}$

It is known from the control system theory, that the system frequency response depends on the pole with the smallest module, the dominant pole, limiting the system bandwidth. To increase it the PI current regulator parameters shall be tuned to compensate the dominant pole, choosing the PI zero as follows:

$$\frac{K_i}{K_p} = \frac{R_s}{L} \tag{2.150}$$

When evaluating the frequency response of a transfer function, it shall be considered the cut-off frequency, corresponding to the frequency where the transfer function has unitary module:

$$|G_{current,OL}(j\omega)|\Big|_{\omega=\omega_c} = 1 \tag{2.151}$$

The cut-off frequency can be approximated to the closed-loop bandwidth, thus the objective becomes to maximize the system bandwidth to obtain a faster response. In any case the phase margin, guaranteeing the system stability, shall be respected: this is evaluated from the following formula [18]:

$$\gamma = 180° + \angle G_{current,OL}(j\omega_c) \tag{2.152}$$

The typical value of the phase margin that allows a stable and not oscillating system response is $\gamma \geq 60°$.

Since the only remaining pole, apart from the one in the origin, is the inverter pole, the PI regulator shall be tuned to respect the previous condition, thus $\omega_{b,current} \leq |p_{inverter}|$.

Indeed, typically ω_b is chosen equal to $2\pi \cdot \frac{f_{sw}}{20}$ and eventually increased basing on experimental results.

From the open-loop transfer function module:

$$|G_{current,OL}(j\omega)|\Bigg|_{\omega=\omega_c} = \frac{K_p}{|j\omega_c|L} \cdot \frac{1}{|1 + j\omega_c \cdot \tau_{inverter,delay}|} = 1 \tag{2.153}$$

and since:

- $|j\omega_{\mathrm{c}}| = \omega_{\mathrm{c}}$
- $|1{+}j\omega_{\mathrm{c}} \cdot \tau_{\mathrm{inverter,delay}}| = |1{+}\frac{j\omega_{\mathrm{c}}}{p_{\mathrm{inverter}}}| \simeq 1$ because $\omega_c \leq |p_{inverter}|$

Therefore, the open-loop transfer function module becomes:

$$G_{current,OL}(j\omega)\Bigg|_{\omega=\omega_c} = \frac{K_p}{\omega_c \cdot L} = 1 \tag{2.154}$$

From the previous assumption, for which $\omega_c \simeq \omega_{b,current}$, the proportional constant can be determined as:

$$K_p = \omega_{b,current} \cdot L \tag{2.155}$$

and from condition in equation 2.150, the integral constant is determined as:

$$K_i = \frac{R_s}{L} \cdot K_p \tag{2.156}$$

As stated before, this procedure is used for the PI current regulator of both i_d and i_q current loops. It is worth mentioning that the inductances values used for tuning the controllers are the average constant values of the *d*-axis and *q*-axis inductances.

2.7.3 Field-Oriented Control

The Field-Oriented Control strategy employs two separated inner current loops working in the (d,q) reference frame. It is important to point out that the aforementioned reference frame is not the same for induction motors and synchronous motors: for the former, the orientation is defined by the rotor flux position, while for the latter by the rotor electrical position $\theta_e = p\theta_m$. The outer loop of the FOC is typically a speed loop.

Considering an IM and writing equation (2.120) as

$$\begin{cases} \lambda_{s,d} = k_r\lambda_r + \sigma L_s i_{s,d} = L_s i_{s,d} \\ \lambda_{s,q} = \sigma L_s i_{s,q} \end{cases} \tag{2.157}$$

where the dependence from the rotor current has been eliminated, it is possible to obtain the rotor flux amplitude as:

$$\lambda_r = \frac{L_s i_{s,d} - \sigma L_s i_{s,d}}{k_r} = L_m i_{s,d} \tag{2.158}$$

Referring to the rotor voltage equation expressed in the stator flux coordinates:

$$\mathbf{0} = R_r \mathbf{i}_r + \frac{d\lambda_r}{dt} - jp\omega_r\lambda_r \tag{2.159}$$

and applying Laplace transform,

$$\mathbf{0} = R_r \mathbf{i}_r + (s - jp\omega_r)\lambda_r \tag{2.160}$$

from equation (2.115) and knowing that $\tau = \frac{L_r}{R_r}$:

$$\lambda_r = \frac{L_m \mathbf{i_s}}{1 + s\tau_r - j\tau_r p\omega_r} \tag{2.161}$$

In steady-state form, when $s = j\omega$, it becomes:

$$\lambda_r = \frac{L_m \mathbf{i_s}}{1 + j(\omega - p\omega_r)\tau_r} = \frac{L_m \mathbf{i}_s}{1 + js\omega\tau_r} \tag{2.162}$$

where s represents now the slip [22].

As previously explained in Section 2.5.3 and referring to equation (2.135), the torque equation can be written putting in evidence the rotor flux vector:

$$T = \frac{3}{2} \cdot p \cdot k_r(\lambda_{r,d} i_{s,q} - \lambda_{r,q} i_{s,d}) \tag{2.163}$$

and considering the FOC reference frame, where

$$\lambda_{dqr} = \begin{bmatrix} \lambda_r \\ 0 \end{bmatrix} \tag{2.164}$$

the torque equation becomes:

$$T = \frac{3}{2} \cdot p \cdot k_r \lambda_{r,d} i_{s,q} \tag{2.165}$$

From all the previous considerations, it can be said the FOC technique for IM uses the d-axis current component to regulate the level of rotor excitation (see equation (2.162)) and the current on the q-axis to regulate the torque according to equation (2.165).

Since for an induction motor it not possible to measure the rotor flux orientation, the angle θ shall be estimated to define FOC (d,q) reference frame. From this, two method for implementing FOC for IM can be distinguished, as described in [22]. In both cases, the reference rotor flux value is obtained as the nominal amplitude value $\lambda_{r,nom}$, while the reference torque value derives from the torque factor at nominal flux: $k_{t,nom} = \frac{3}{2} \cdot p \cdot k_r \cdot \lambda_{r,nom}$.

- Indirect FOC

  In the I-FOC, the rotor flux position is estimated by integrating

$$p\omega_{slip} = \frac{L_m i_q}{\tau_r \lambda_r} \simeq \frac{i_q}{\tau_r i_d} \tag{2.166}$$

  and summing it to the rotor position:

$$\theta = p\theta_r + \int (p\omega_{slip})dt \tag{2.167}$$

  This implementation of FOC does not require a flux observer, making the control algorithm very simple, but on the other side, the mechanical rotor position is needed, so an encoder is needed. It also depends on τ_r which is subjected to variations due to temperature.

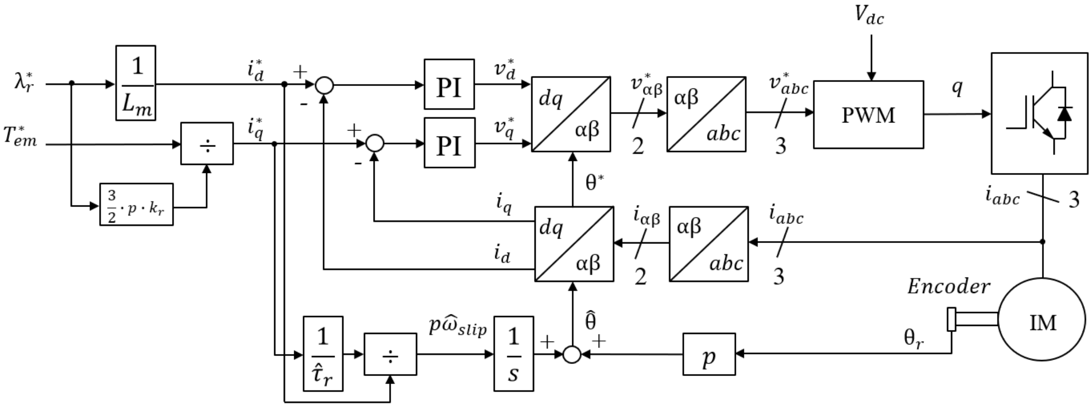


Figure 2.41: Indirect Field Oriented Control scheme for an induction machine

- Direct FOC

  This version of the FOC uses estimator or observers to obtain the rotor flux position. This type of solution require measured quantities like the rotor mechanical position and an accurate motor model making the implementation much complicated with respect to the I-FOC.

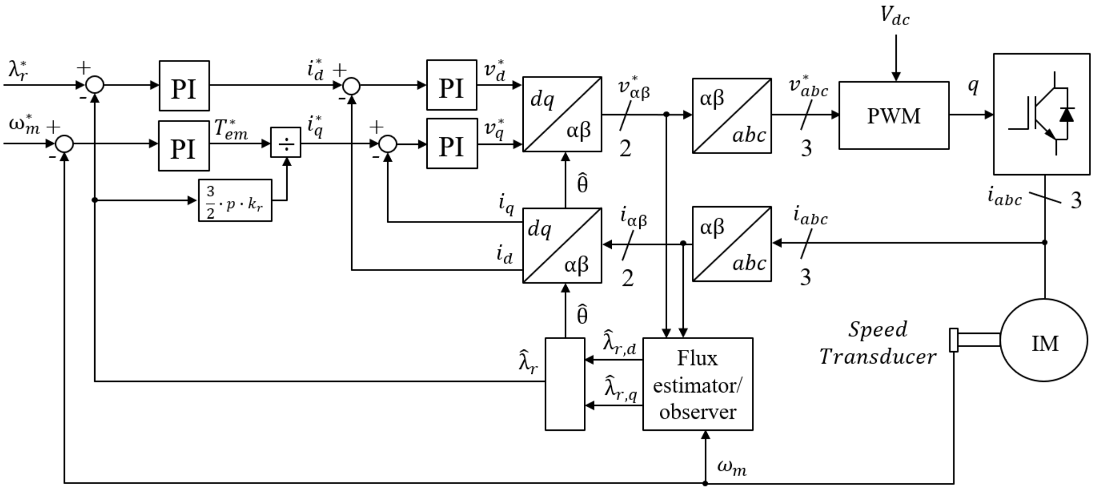


Figure 2.42: Direct Field Oriented Control scheme for an induction machine

When considering a synchronous machine, the main difference in the FOC algorithm is the definition of the (d,q) reference frame, whose coordinates are determined from the rotor electrical position θ_e obtained from measurements of the mechanical position or, adopting sensorless techniques, estimated from the electrical quantities.

According to the different type of synchronous machine, the definition of the *d*-axis changes (see 2.5.3), but the aim of the FOC remains to directly control the currents in a reference frame synchronous with the rotor.

Before entering in details, it is worth mentioning that for SPM machines, according to equation (2.62), the reference torque can be easily determined by rescaling the value T^* by the torque factor $\frac{1}{k_t}$ obtaining the reference value for the current ${i^*}_q$, while ${i^*}_d = 0$.

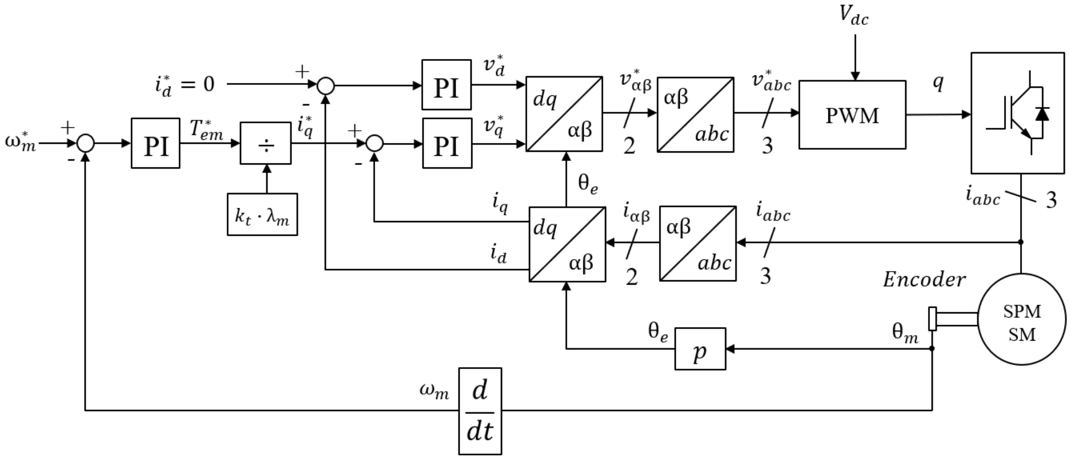


Figure 2.43: Field Oriented Control scheme for a surface-mounted permanent magnet synchronous machine

While for all the other synchronous machines, this technique allows the application of MTPA (Maximum Torque Per Ampere) locus, whose aim is to maximise the torque per phase current.

Therefore, given a reference torque from the outer speed loop, the reference current values i^*_{dq} can be obtained from [22]:

- An approximated MTPA, defined from a constant MTPA angle γ and constant torque factor k_t;

- MTPA look-up tables (LUT), derived from a numerical manipulation of the current-to-flux relationship, also known as *flux maps*; the definition of MTPA LUTs requires an accurate magnetic model in order to obtain torque accuracy in the control.

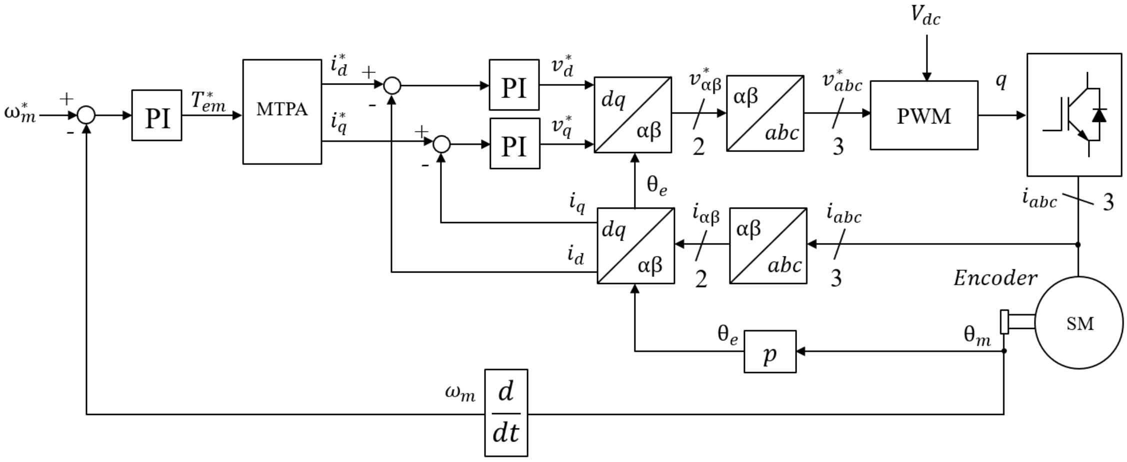


Figure 2.44: Field Oriented Control scheme for a synchronous machine with MTPA

When working with synchronous machines, as well as with induction motors, operating voltage and current limits derived by the converter shall be considered. First of all a torque limit, and thus a current limit (since $T_{max} = f(i_{max})$), corresponding to the maximum inverter current is imposed:

$$\begin{cases} |i_{d,max}| \leq I_{max} \\ |i_{q,max}| \leq \sqrt{I_{max}^2 - i_d^2} \end{cases} \tag{2.168}$$

The voltage limits are instead imposed to the current regulators which shall provide the correct voltage references to the converter:

$$\begin{cases} |v_{d,max}| \leq V_{max} = \frac{V_{dc}}{\sqrt{3}} \\ |v_{q,max}| \leq \sqrt{V_{max}^2 - v_d^2} \end{cases} \tag{2.169}$$

The limit imposed for the voltage on the d-axis comes from the assumption of a DC-bus voltage V_{dc} and a converter using PWM-BEM or SVPWM, thus the amplitude of the space vector voltage signal is restricted to $\frac{V_{dc}}{\sqrt{3}}$ to remain in the linear modulation range [24].

In any case, even if the FOC is an efficient control strategy, during flux weakening it becomes inaccurate and difficult to be manage.

Many flux-weakening strategies have been proposed in the literature [25], [26], but one of the easier and more robust solution is the feedback-based flux weakening proposed in [27]; this method, suitable for any type of synchronous machine and using a voltage feedback regulator, it is shown below:

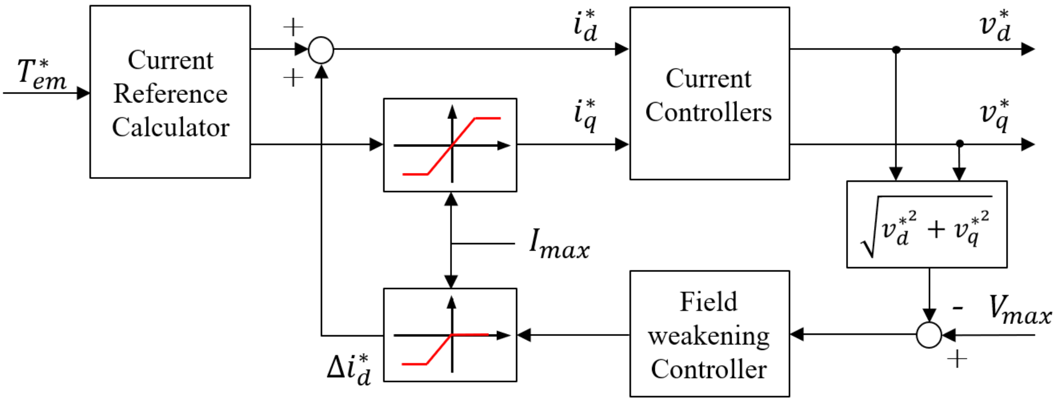


Figure 2.45: Flux weakening strategy based on voltage feedback

Considering a torque reference, the current references i^*_d and i^*_q are obtained. The objective is to maintain the voltage output of the current regulator inside the limit previously described and corresponding to $\frac{V_{dc}}{\sqrt{3}}$. This is performed by providing as input to the flux weakening regulator the difference between the voltage limit and the feedback voltage computed as shown in the scheme.

The output of the flux weakening regulator is a negative value of the d-axis current summed to the the reference d-axis current value. Therefore the overall i_d current is decreased allowing the speed to increase without overcoming the imposed voltage limit.

It is worth pointing out that the flux-weakening controller bandwidth should be lower than the one of the current loops to avoid interferences, but this means that, if the speed changes abruptly, the flux-weakening control could show oscillations in the response [28].

PI Speed Controller Tuning

In section 2.7.2 PI current controllers tuning procedure has been described, but the speed loop PI controller needs to be tuned as well.

The output of the PI speed controller is the reference torque to be provided to the current reference calculator.

To calibrate the speed loop, the first assumption is done on the torque, imposed equal to the ideal one, such that ${T_{em}}^* = T_{em}$.

Another assumption is that the speed loop shall be slower than the current loops, for this reason, the speed loop bandwidth is imposed to be very low with respect to the current loops one. In this way the motor electrical poles do not influence the closed loop system dynamic, since the cut-off frequency of the open-loop transfer function of the speed loop is moved at lower frequency with respect to the ones of the motor electrical poles.

Typically

$$\omega_{b,speed} \leq \frac{\omega_{b,current}}{10} \tag{2.170}$$

Starting from the motor mechanical equation in the time domain (the time dependence has been neglected for simplicity),

$$J\frac{d\omega_m}{dt} + \beta\omega_m + T_L = T_{em} \tag{2.171}$$

where J is the rotor moment of inertia, ω_m is the rotor mechanical rotational speed, β is the frictional coefficient, T_L is the load torque and T_{em} is the electromagnetic torque.

Assuming that the frictional coefficient is very small, thus negligible, and applying the Laplace transform, equation 2.171 becomes:

$$sJ\omega_m + T_L = T_{em} \tag{2.172}$$

and writing in the input/output form, where the electromagnetic torque is the input and the mechanical speed is the output,

$$\frac{\omega_m}{T_L - T_{em}} = \frac{1}{sJ} \tag{2.173}$$

With the assumption in equation 2.170, the motor electrical transfer function can be approximated to a unitary gain, therefore the speed loop is:

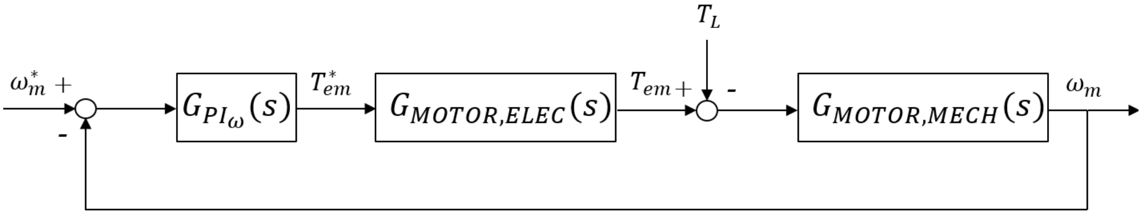


Figure 2.46: Speed loop

The open-loop transfer function of the speed loop is (neglecting the load torque which will be feedforward compensated):

$$\begin{gathered} G_{speed,OL}(s) = G_{PI_\omega}(s) \cdot G_{MOTOR,ELEC}(s) \cdot G_{MOTOR,MECH}(s) = \\ = \frac{K_p}{s} \cdot \left(s + \frac{K_i}{K_p}\right) \cdot 1 \cdot \frac{1}{sJ} \end{gathered} \tag{2.174}$$

The open-loop transfer function presents:

- A zero from the PI current regulator in: z_{PI} = -$\frac{K_i}{K_p}$

- A pole from the PI current regulator integral action in the origin
- A mechanical pole from the motor mechanical model in the origin

Even in this case, the cut-off frequency is evaluated as:

$$|G_{speed,OL}(j\omega)|\Big|_{\omega=\omega_c} = 1 \tag{2.175}$$

thus

$$|G_{speed,OL}(j\omega)|\Big|_{\omega=\omega_c} = \frac{K_p}{|j\omega_c|L} \cdot \left(|j\omega_c| + \frac{K_i}{K_p}\right) \cdot \frac{1}{|j\omega_c| \cdot J} = 1 \tag{2.176}$$

Since:

- $|j\omega_\mathrm{c}| = \omega_\mathrm{c}$
- No pole needs to be compensated from the PI regulator zero, it is chosen such that the condition in equation 2.170 is respected. Hence, the following approximation can be considered: $\omega_\mathrm{c} \gg \frac{K_i}{K_p}$

Therefore, the open-loop transfer function module becomes:

$$G_{speed,OL}(j\omega)\Big|_{\omega=\omega_c} = \frac{K_p}{\omega_c} \cdot \omega_c \cdot \frac{1}{\omega_c \cdot J} = 1 \tag{2.177}$$

Thus, assuming that $\omega_\mathrm{c} \simeq \omega_\mathrm{b,speed}$, the proportional constant can be determined as:

$$K_p = \omega_{b,speed} \cdot J \tag{2.178}$$

while the integral constant is:

$$\frac{K_i}{K_p} < \omega_{b,speed} \Rightarrow K_i < \omega_{b,speed} \cdot K_p \tag{2.179}$$

2.8 Unified Direct-Flux Vector Control

Another control technique that will be addressed in this treatise is the Unified Direct-Flux Vector Control (UDFVC). In this section, the control scheme developed in [29] will be reviewed and analysed since it represents one of the best solution for traction applications.

The main aim of the UDFVC is to implement a general control strategy used to drive sinusoidal AC motors (IM, SPM, IPM, Syr). What differentiates this technique from the ones mentioned in the previous sections is the adopted reference frame, which is the stator flux reference frame.

With this approach, the controlled quantities are the stator flux amplitude, regulated by the d-axis stator voltage vector component and the quadrature current component, which regulates the torque.

The main advantage of the UDFVC is the improvement of the performance of control techniques in the flux weakening region where the FOC is no more accurate. It is worth mentioning that also sensorless implementations of this proposed control scheme are possible.

Another advantage, related to the one previous one, is that the knowledge of the motor model is not required in the flux weakening range, as highlighted in [30].

For sake of generality, a different reference frame, with respect to the one found in the literature, is adopted for the SyR machine, where the d-axis is chosen along the minimum permeance direction [29].

The stator flux frame (d_s,q_s) is thus defined as a reference frame shifted of an angle δ (the load angle) with respect to the reference frame (d,q) defined by the rotor flux, for IM, and with respect to the d-axis of the reference frames (d_m,q_m) defined for synchronous machines and related to the rotor mechanical angular position.

An overview of the reference frames for the previously mentioned AC machine is presented in figure 2.47.

The unconventional choice of the SyR motor reference frame allows to write

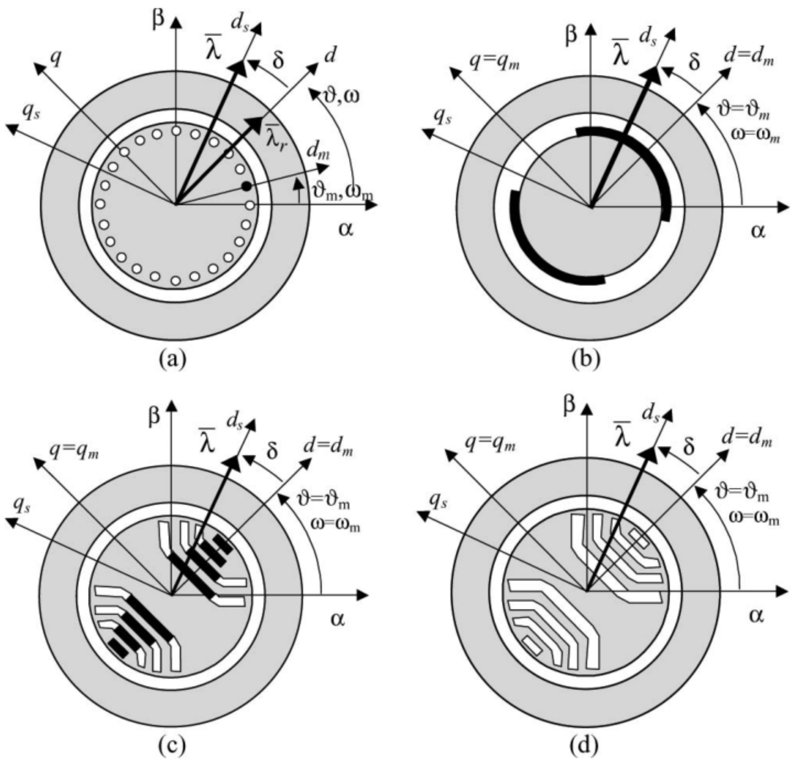


Figure 2.47: I-f control scheme (Source: [29]).

a general magnetic model for synchronous machines in rotor reference frame (*d,q*) which results to be different from the ones defined from equation (2.107) to equation (2.110), especially for SyR machine.

$$\lambda_{dq} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \cdot \mathbf{i}_{dq} + \begin{bmatrix} \lambda_m \\ 0 \end{bmatrix} \tag{2.180}$$

If considering a SPM, $L_d = L_q = L_s$, while for SyR motor, $\lambda_m = 0$.

Instead, for IM, the magnetic model is the one described in equation (2.120).

The electrical equations, equation (2.77) and equation (2.83) have to be modified according to the new stator flux-oriented reference frame (d_s,q_s) [30]:

$$\mathbf{v}_{dqs} = R_s \cdot \mathbf{i}_{dqs} + \frac{d}{dt} \cdot \begin{bmatrix} \lambda \\ 0 \end{bmatrix} + \lambda \cdot \begin{bmatrix} 0 \\ \omega + \frac{d\delta}{dt} \end{bmatrix} \tag{2.181}$$

Also the torque equation is modified in:

$$T = \frac{3}{2} \cdot \frac{p}{L_d} \left(-\frac{L_q - L_d}{L_q} \cdot \lambda^2 \cdot \frac{sin2\delta}{2} + \lambda_m \lambda sin\delta \right) \tag{2.182}$$

obtained by combining equation (2.62) and

$$\lambda_{dq} = \lambda \cdot \begin{bmatrix} cos\delta \\ sin\delta \end{bmatrix} \tag{2.183}$$

for synchronous machines.

For IM, since

$$\lambda_{dq} = \begin{bmatrix} L_s & 0 \\ 0 & \sigma L_s \end{bmatrix} \cdot \mathbf{i}_{dq} \tag{2.184}$$

the torque equation becomes:

$$T = \frac{3}{2} \cdot p \cdot \frac{1 - \sigma}{\sigma L_s} \cdot \lambda^2 \cdot \frac{sin2\delta}{2} \tag{2.185}$$

The obtained torque equations are expressed as a function of λ and δ, resulting in a difficult control of the torque itself since the two terms are not decoupled. To provide an easier and decoupled approach to torque control, the torque-current component i_{qs} is introduced [30]. This modifies the voltage equation from equation (2.181) as function of λ and i_{qs}:

$$\frac{d}{dt}\begin{bmatrix} \lambda \\ i_{qs} \end{bmatrix} \simeq \begin{bmatrix} 1 & 0 \\ \frac{k}{L_d} & \frac{b}{L_d} \end{bmatrix} \cdot \begin{bmatrix} v_{ds} \\ v_{qs} - \lambda\omega \end{bmatrix} \tag{2.186}$$

where the voltage drops across the resistors have been neglected and the variables k and b are:

$$k = k(\delta) = -\frac{1}{2}\left(1 - \frac{L_d}{L_q}\right) sin2\delta \tag{2.187}$$

$$b = b(\lambda, \delta) = \frac{L_d}{\lambda^2}\left(\frac{dT}{d\delta}\right)_{\lambda=const.} = -\left(1 - \frac{L_d}{L_q}\right) \cdot cos2\delta + \frac{\lambda_m}{\lambda} cos\delta \tag{2.188}$$

These two defined terms are clearly machine magnetic state dependent, so the torque control and thus the i_{qs} control will depend on the actual flux size and position [30].

Therefore, from equation (2.186) and considering again the resistive drops, the current equation for the q_s-axis current component is, for an IPM machine, as demonstrated in [30]:

$$L_d \frac{di_{qs}}{dt} = -R_s i_{qs} + k \cdot (v_{ds} - R_s \cdot i_{ds}) + b \cdot (v_{qs} - \lambda\omega) \tag{2.189}$$

the factors k and b are defined in equation (2.187) and equation (2.188) for an IPM, while for a SPM machine

$$\begin{cases} k_{SPM} = 0 \\ b_{SPM}(\lambda, \delta) = \frac{\lambda_m}{\lambda} \cdot cos\delta \end{cases} \tag{2.190}$$

and for a SyR machine,

$$\begin{cases} k_{SyR} = -\frac{1}{2}\left(1 - \frac{L_d}{L_q}\right) sin2\delta \\ b_{SyR}(\lambda, \delta) = -\left(1 - \frac{L_d}{L_q}\right) \cdot cos2\delta \end{cases} \tag{2.191}$$

For what concerns an IM, the current equation of the q_s-axis current component is, as described in [31]:

$$\sigma L_s \frac{di_{qs}}{dt} = -R_{eq} i_{qs} - \omega_{slip} \cdot \sigma L_s \cdot i_{ds} + (v_{qs} - \lambda \cdot \omega_m) \tag{2.192}$$

where $R_{eq} = R_s + \frac{k_r}{k_s} \cdot R_r$ and ω_{slip} is the slip speed defined as the difference ω - ω_m.

From the equations previously presented, the following key features of the UDFVC can be highlighted [29]:

- One of the main features of the UDFVC is that no particular model accuracy is required unless the control has to be performed at low speed; as mentioned before, this allows the flux PI controller closed loop bandwidth not to be influenced by magnetic saturation phenomena. This regulator directly controls the stator flux through the d_s-axis voltage which is completely decoupled from the q_s-axis as obtained from the previous equations;

- The torque is instead regulated by the q_s-axis current component by means of a PI controller whose closed-loop bandwidth is dependent (as explained in the previous Section) on the proportional gain and the machine inductance. According to the different type of motor, the inductance is σL_s for IM, L_s for SPM and L_q for IPM and SyR;

- From equation (2.189), it is clear that the torque response is dependent on the b factor as analysed in [30]. The control stability dependence on this term will be discussed later.

As already mentioned, one of the main advantages of UDFVC is the algorithm performances in flux-weakening operation where, typically, the MTPV operation occurs. Indeed, when flux-weakening is performed, the machine working region moves in a zone where the MTPA curve may cross the MTPV trajectory; this happens if the required speed overcomes a certain limit which is analytically determined for IM, while must be empirically determined for synchronous machines.

In MTPV operation the objective is to obtain the maximum torque with the minimum flux; in this working region, many limits are present and must be respected: first of all the voltage and current limits imposed by the inverter and then, the load angle, for which a maximum value is defined.

In order to maintain the control stable, the load angle δ must always respect the following condition: $\delta < \delta_{max}$. The value of the maximum load angle is machine dependent.

The torque corresponding to the maximum load angle δ_{max} is called pull-out torque [29]. From equation (2.182), it is possible to determine the value of this angle by imposing the partial torque derivate with respect to the load angle δ equal to zero [29].

Therefore, from

$$\frac{\partial T}{\partial \delta} = \frac{3}{2} \cdot p \cdot \frac{1}{L_d} \cdot \left(-\frac{L_q - L_d}{L_q} \cdot \lambda^2 \cdot cos2\delta + \lambda_m \lambda cos\delta \right) = 0 \tag{2.193}$$

it follows

$$\frac{L_q - L_d}{L_q} \cdot \lambda^2 \cdot cos2\delta = \lambda_m \lambda cos\delta \tag{2.194}$$

For SyR machine, where $\lambda_m = 0$, the maximum load angle is:

$$\delta_{max,Syr} = 135° \tag{2.195}$$

This value, which is different from what can be found in the literature, derives from the unconventional choice of the reference frame. In fact, the typical value for the maximum load angle is $\delta_{\mathrm{max,Syr}} = 45\mathrm{deg}$, according to the standard adopted reference frame.

For SPM motor, since $L_d = L_q$,

$$\delta_{max,SPM} = 90^\circ \tag{2.196}$$

For IPM machine, the value is dependent on the anisotropy factor, thus the solution varies between:

$$90^\circ < \delta_{max,IPM} < 135^\circ \tag{2.197}$$

If the motor saliency is high, the limit value is close to 135∘, otherwise becomes close to 90∘ for low-saliency machines.

For IM, from equation (2.185), the derivative as in equation (2.193) is computed and imposed equal to 0 obtaining the maximum load angle value:

$$\delta_{max,IM} = 45^\circ \tag{2.198}$$

Supposing that the target of the UDFVC is a speed-controlled AC drive, the control scheme is shown in figures 2.48 and 2.49.

The q_s-axis current reference is computed from the torque equation:

$$T = \frac{3}{2} \cdot p \cdot \lambda \cdot i_{qs} \tag{2.199}$$

Typically, the flux reference is obtained from the MTPA law, but a simpler

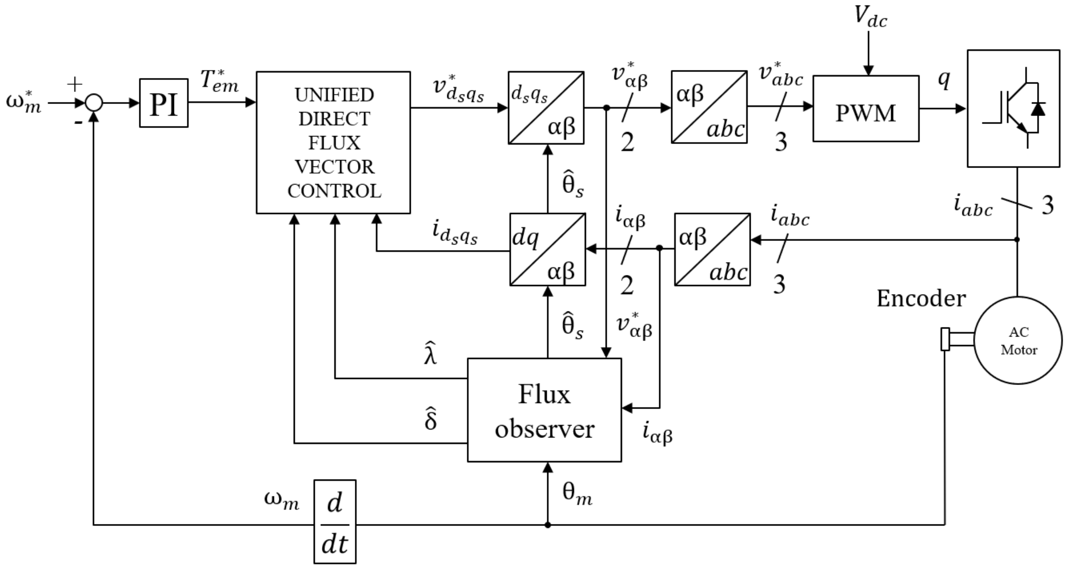


Figure 2.48: Unified Direct Flux Vector Control scheme

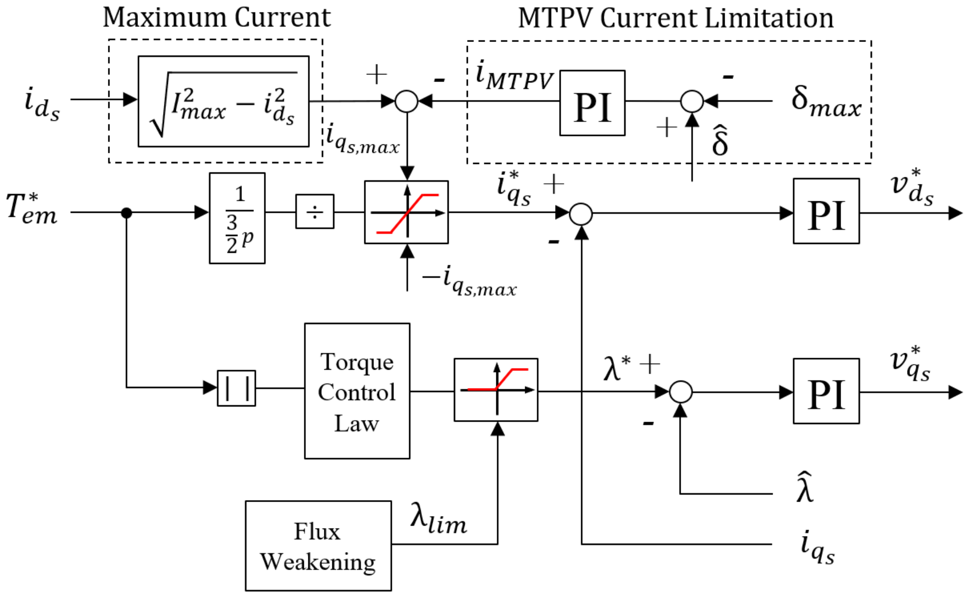


Figure 2.49: Unified Direct Flux Vector Control scheme: inner control loops

control law can be applied, as proposed in [32]:

$$\lambda^* = \lambda_m + \frac{|T^*|}{T_{nom}} \cdot (\lambda_{nom} - \lambda_m) \tag{2.200}$$

This control law requires less computational effort with respect to the MTPA, but it is worth mentioning that the terms λ_{nom} and T_{nom} are machine dependent. Focusing on IPM machines, two conditions must be respected to obtain the best performances:

- The value of the reference flux when the torque is null shall be exactly equal to the permanent magnet flux λ_m;
- The value of the reference flux when the torque value is equal to the rated torque is derived from the MTPA law, thus the rated torque is maximum for a given current.

As described at the beginning of the section, the reference frame in which the UDFVC works is the stator flux reference frame (d_s,q_s), thus a stator flux observer is required. It is based on the current-to-flux model at low speed and on back-electromotive force integration at high speed, as described in [33] and [34].

The employed observer is a reduced-order $VI\theta$ closed-loop observer, using the machine magnetic model defined in the rotor reference frame (d_m,q_m) and described in equation (2.180) and equation (2.120), where the crossover frequency between the models used for low-speed and high-speed corresponds to the observer gain g, measured in radians per second [29].

The output variables employed in the control algorithm are: the estimated flux amplitude $\hat{\lambda}$, the sine and cosine of the estimated flux phase angle $\theta_s = \theta+\delta$ with respect to the reference frame (α,β) and the sine and cosine of the estimated load angle δ. These last two terms are obtained by rotating the sine and cosine of the flux phase angle of an angle θ, corresponding to the mechanical angle for synchronous machines; for IM, it represents the rotor flux vector phase angle obtained from the rotor flux observer.

The stator flux observer is shown in figure 2.50.

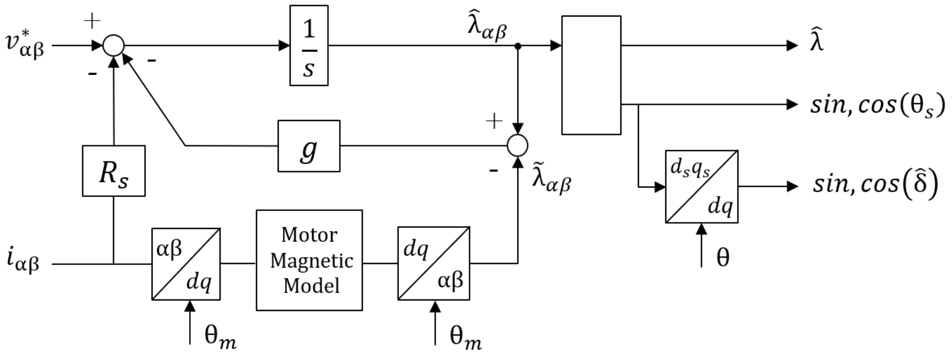


Figure 2.50: Unified $VI\theta$ flux observer

For what concerns the regulators employed in the control scheme, two proportional-integral controllers are used:

- The flux regulator is characterised by a very fast closed-loop response which is influenced only by the dynamics of the flux observer whose bandwidth is on the order of kilohertz [30]; the flux amplitude is thus regulated by controlling the v_{ds} which is decoupled from the voltage d_s-axis component.

  In the flux-weakening region, the flux reference shall be limited referring to the inverter maximum voltage $V_{max} = \frac{V_{\mathrm{dc}}}{\sqrt{3}}$. Considering the resistive drops, the flux reference is:

$$\lambda^*_{lim} \leq \frac{V_{max} - R_s \cdot i_{qs} \cdot sign(\omega)}{|\omega|} \tag{2.201}$$

- The i_{qs} current regulator instead is regulated by controlling the voltage q_s-axis component. The main difference with the flux regulation is that the current regulation interacts with the flux loop according to equation (2.186). The presence of the d_s-axis voltage component is compensated by the integral action in steady-state, while the back EMF component is compensated in feedforward [30].

  The current reference is limited according to the maximum inverter current

I_{max} as follows:

$$i_{qs}^{*} \leq \sqrt{I_{max}^{2} - i_{ds}^{2}} \tag{2.202}$$

Another limitation is present in the current reference for the MTPV operation with the aim of limiting the load angle to the maximum load angle δ_{max} (corresponding to the condition $b = 0$).

The MTPV current limitation strategy adopted is developed in [35] and uses a PI regulator providing as output a negative i_{MTPV} current component which corrects the maximum level of i_{qs} current to maintain the condition: $\delta \leq \delta_{max}$. The δ_{max} control scheme is shown in the following figure:

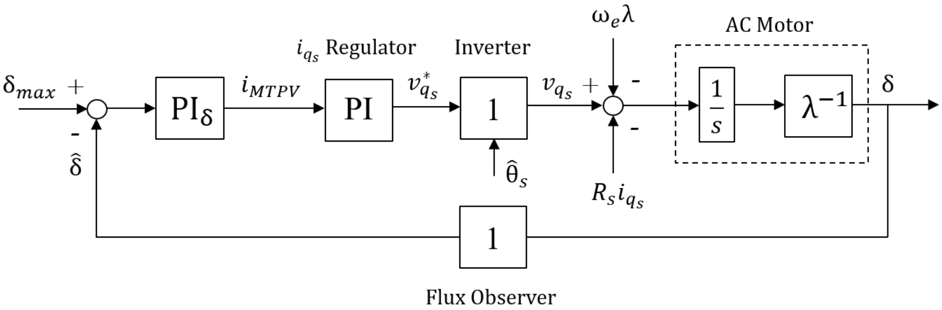


Figure 2.51: δ_{max} control scheme.

As described in [35], the closed-loop bandwidth of the δ PI regulator is:

$$\omega_{bw,\delta} = \frac{k_{p,\delta} \cdot k_{p,i_{qs}}}{\lambda} \tag{2.203}$$

where $K_{p,\delta}$ is the proportional δ regulator gain, while $K_{p,i_{qs}}$ is the proportional gain of the i_{qs} current regulator.

It can be observed that the closed-loop bandwidth varies according to the flux amplitude λ, that shall be set to the minimum value corresponding to the maximum speed. This control loop guarantees that the factor b is always greater than zero, guaranteeing the control stability as demonstrated in [30].

The efficiency of this control strategy is proven by experimental results as reported in [29], [30], [32], [35], [36] and [37].

Chapter 3

Technologies Used - Software Description

This chapter is dedicated to the description of the baseline on which the AURIX/Arduino-like Board is based on.

The main elements characterising the baseline are:

- ERIKA OS
- iLLD low-level drivers provided by Infineon
- Complex Device Drivers from I&M
- AURIX/Arduino-like Board Support Package (BSP)

The supported compilers are:

- HighTec (based on GCC)
- Tasking

In the following sections a focus on the relevant components of the workflow will be proposed together with the main software used during the thesis work.

3.1 ERIKA Enterprise RTOS v3

ERIKA Enterprise v3 is an open-source RTOS implementation of the OSEK/VDX API designed for small microcontrollers and supporting last multi- and many-core chips and officially certified OSEK/VDX for Infineon AURIX™ TriCore.

OSEK/VDX is a standard for software architectures for electronic control units (ECUs) in the automotive field. It has been created from the merge of two projects [38]:

- OSEK: Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug (Open systems and the corresponding interfaces for automotive electronics) founded in 1993 by a German automotive company consortium and the University of Karlsruhe.
- VDX: Vehicle Distributed eXecutive, a similar project founded by French cars manufacturers in 1994.

The reason why OSEK/VDX has been created, is the incompatibility issues of the control units produced by different automotive companies caused by different protocols and interfaces.

These problems have been overcome introducing specifications independent from the HW, scalability between different requirements, portability and reusability of the application software, specification of abstract interfaces for RTOS and network management, savings in costs and development time.

The OSEK/VDX consortium provides the OIL language (OSEK Implementation Language) as a standard configuration language for static definition of RTOS objects instantiated and used by the application.

The main features of ERIKA Enterprise are presented in the following [39]:

- Real-time kernel, priority based, with stack sharing for RAM optimization.
- Minimal multithreading RTOS interface. A minimal configuration of ERIKA Enterprise is typically in the order of 1 to 6 KB of Flash memory.
- RTOS API for: Tasks, Events, Alarms, Resources, Application modes, Semaphores, Error handling.
- Support for conformance classes (BCC1, BCC2, ECC1, ECC2) to match different application requirements.
- Support for preemptive and non-preemptive multitasking.
- Support for fixed priority scheduling and Preemption Thresholds.
- Support for Earliest Deadline First (EDF) scheduling.
- Support for stack sharing techniques, and one-shot task model to reduce the overall stack usage.
- Support for shared resources.
- Support for periodic activations using Alarms.
- Support for centralized Error Handling.
- Support for hook functions before and after each context switch.
- Dual license: GPLv2+Linking Exception (for ERIKA Enterprise v2.x and v3.x), and Commercial (for ERIKA Enterprise v3.x)

3.2 HighTec Development Platform

The HighTec Development Platform is based on Eclipse, an integrated development environment (IDE) developed by Eclipse Foundation and basically written in C and Java. It is a free and open-source software using plug-ins to implement functionalities within and on top of the run-time system.

The HighTec Development Platform supports market leading microcontrollers for automotive and industrial applications, in particular TriCore, AURIX and ARM.

The main features of the toolset are listed below [40]:

- Advanced Multi-core support
- Optimized for Auto-Coding
- Commercial standard and math libraries (no open source, no viral GPL implication)
- Based on industry-standard compiler technologies for highest code portability
- Toolsets contain C compiler, C++ compiler, assembler, linker, IDE and various utilities
- AUTOSAR MCAL driver support (including sales and on-site training)
- Unique Qkit ISO26262 validation suite for ASIL D certification
- Used by leading Tier1 and OEM, in automotive and industrial markets

3.3 AURIX/Arduino-like Board Support Package

Considering the baseline previously illustrated, the main element employed in this thesis work is the Board Support Package developed for the AURIX/Arduino-like Board.

The aim of a Board Support Package in embedded systems is to provide a high-level interface between hardware and the operating system and an interface to device drivers allowing the communication with hardware's peripherals such as memory, communication busses, ADC, Timers, etc.

The BSP developed for the hardware used in this thesis has been focused on the code generation for the Infineon AURIX™ and allows a quick deploy of application models in Simulink.

Thanks to the Simulink library, it is possible to create instances of the blocks composing the library in a Simulink model.

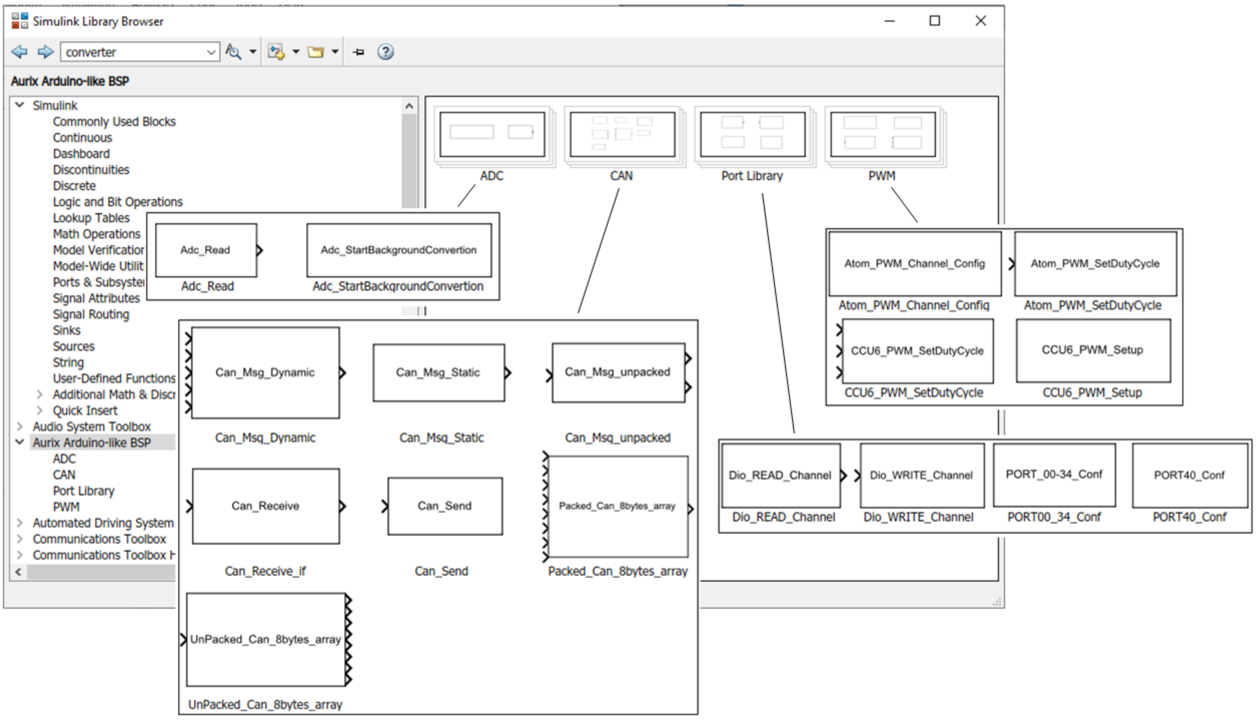


Figure 3.1: AURIX/Arduino-like Library

A description of the main blocks will be set out below [41]:

- GPIO

  The TC277 implements digital General Purpose Input/Output port lines and the related blocks allow to select the desired port pins, determining the port functionality and depending on this it is possible to select hysteresis function (if configured as input) or speed grade (if configured as output). Blocks dedicated to the read and write functions on the port pin are also present.

- ADC

  Only 5 ADC channels have been configured: 2 channels of ADC2 and 3 channels of ADC7 are available with a fixed ADC resolution of 12 bits. The blocks related to the ADC allow the start of the background conversion and the result read from a specific group and channel between the available ones.

- CAN

  To use the CAN interface present on the Board, dedicated blocks have been developed to compose Static CAN messages with a structure which cannot be modified on run-time, Dynamic CAN messages whose structure can instead be modified at run-time, to unpack received CAN messages returning the number of useful bytes and a pointer to the array of data (CAN message payload) and to send and receive CAN messages.

- PWM

  Two main blocks are present allowing the complete configuration of the Timer T12 to support three-phase PWM generation: CCU6_PWM_Setup allows configuration of the operating mode of the Timer T12, PWM period and dead-time period and CCU6_PWM_SetDutyCycle allows to set run-time the duty cycle of the 3 PWM active channels.

3.4 MATLAB/Simulink

MATLAB (MATrix LABoratory) is a software designed for numerical computing and it uses a programming language developed by MathWorks®. It supports

Model-Based Design co-operating with Simulink©, a software designed for modelling, simulations and analysis of dynamic systems used also for code auto generation, verification and validation of embedded systems.

3.5 Plexim PLECS®

PLECS® by Plexim is a modelling and simulation software for power electronics systems covering not only the electrical domain but also the magnetic, thermal, mechanical and control aspects of such a system.

The main features have been set out below divided by domain [42]:

- Electrical domain
  Linear and non-linear models of passive, active and electromechanical components and models of electrical machines (DC machines, AC induction machine, AC synchronous machines, switched reluctance machines).

- Control domain
  Possibility to realise analog and digital control systems using signal processing blocks (both continuous and discrete time), C–script blocks for custom functionalities in ANSI C, DLL blocks for compiled object code in the models and Finite State Machine (FSM) blocks.

- Thermal domain
  PLECS can consider the semiconductor's conditions (forward current, junction temperature, blocking voltage, etc.) before and after the switching instead of determining the switching losses from the current and the voltage (using Look-Up Table).

- Magnetic domain
  PLECS magnetic models are based on the permeability – capacitance analogy

and include saturation, hysteresis and eddy currents phenomena. It can also model frequency – dependent losses through magnetic resistances.

- Mechanical domain

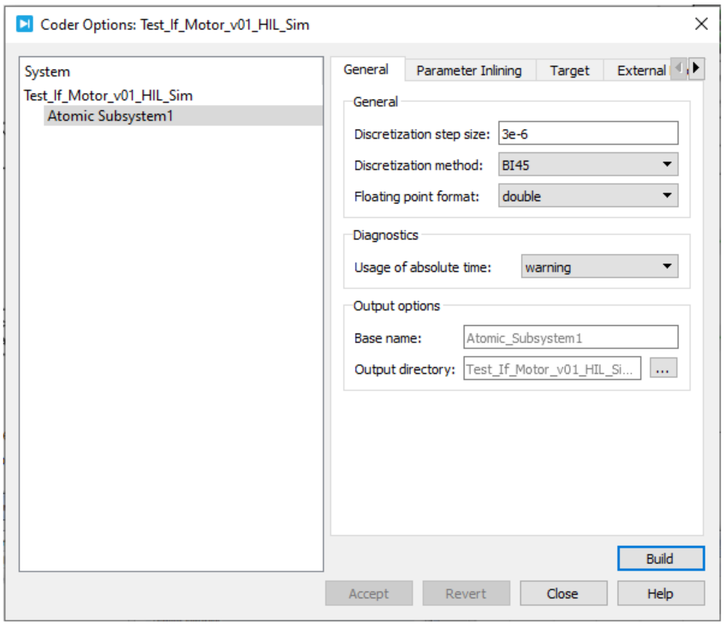
  PLECS allows to model translational and/or rotational motions through non-linear blocks and, combining them with all the previously described domains, it is possible to obtain a complete system model, turning to be very useful for servomechanisms and ePowertrain applications.

- PIL

  Using a dedicated package, PLECS can perform a Processor-in-the-Loop simulations.

Another important feature of PLECS is the Coder: it supports the generation of ANSI-C code from a PLECS Blockset or PLECS Standalone model [43].

In PLECS environment, the Coder window allows to specify the discretization step size for the real-time simulation, the simulation method and the floating point format as shown in figure 3.2.


Coder Options: Test_If_Motor_v01_HIL_Sim

System
Test_If_Motor_v01_HIL_Sim
Atomic Subsystem1

General

Discretization step size: 3e-6
Discretization method: BI45
Floating point format: double

Diagnostics

Usage of absolute time: warning

Output options

Base name: Atomic_Subsystem1
Output directory: Test_If_Motor_v01_HIL_Si...

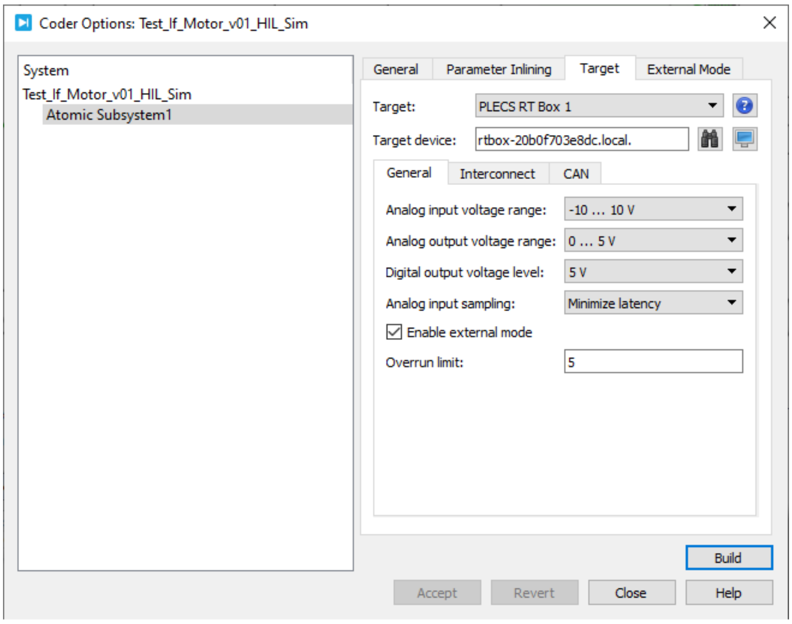
Figure 3.2: PLECS Coder window: general real-time simulation settings

This is typically employed in Hardware-in-the-Loop simulations, where the system consists of the controller implemented in a real hardware, while the plant is commonly simulated on a real-time hardware as the RT Box, described in section 4.2.

For this purpose, an RT Box Target support package has been developed by PLECS to interface with the RT Box itself.

From the Coder window, it is possible to select the desired target from the *Target* menu (refer to figure 3.3), defining general settings as the analog input and output signals voltage ranges, as well as the digital output voltage level to better interface with the hardware controller peripherals. External mode shall be necessarily enabled for interfacing with the RT Box.

Once the target has been selected, the *External Mode* menu allows first to build the developed model, then to connect the Coder with the RT Box and to control triggers for the virtual oscilloscope that can be used in the simulation environment to observe the system behaviour (refer to figure 3.4).



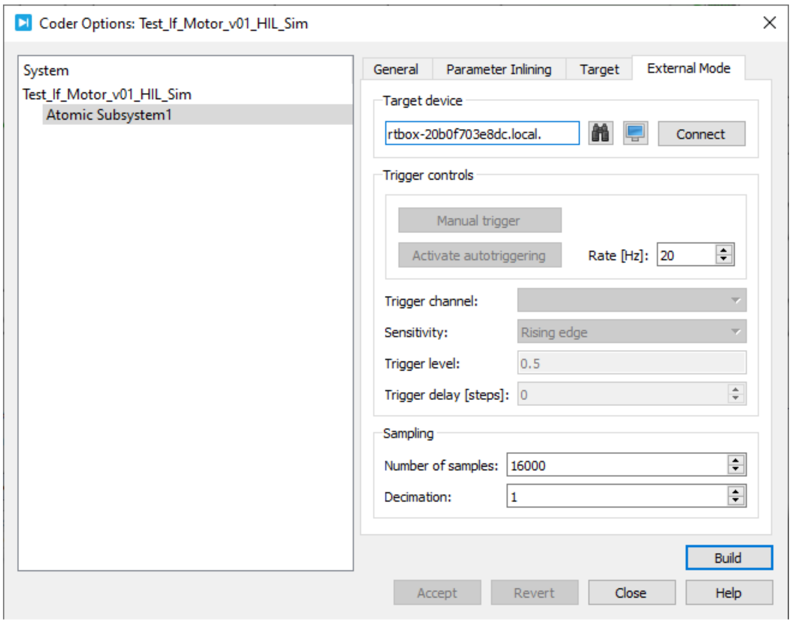
Coder Options: Test_If_Motor_v01_HIL_Sim

System
Test_If_Motor_v01_HIL_Sim
  Atomic Subsystem1

Target: PLECS RT Box 1
Target device: rtbox-20b0f703e8dc.local.

Analog input voltage range: -10 ... 10 V
Analog output voltage range: 0 ... 5 V
Digital output voltage level: 5 V
Analog input sampling: Minimize latency
☑ Enable external mode
Overrun limit: 5

Figure 3.3: PLECS Coder window: Target settings



Target device: rtbox-20b0f703e8dc.local.

Trigger controls

Rate [Hz]: 20

Trigger channel:

Sensitivity: Rising edge

Trigger level: 0.5

Trigger delay [steps]: 0

Sampling

Number of samples: 16000

Decimation: 1

Figure 3.4: PLECS Coder window: External Mode settings

3.5.1 RT Box Target Support Library

The support package previously mentioned is composed of different component allowing the interconnection and signal exchange between the controller peripherals and the real-time simulation hardware. In the following the main blocks will be presented:

- Analog In

  It allows to receive the output measured voltage in a selectable analog RT Box input channel. It is possible to scale and introduce an offset on the input signal according to the formula [44]:

$$signal_{out} = signal_{input} \cdot Scale + Offset \tag{3.1}$$

Figure 3.5: RT Box Target Support Library: Analog In block (Source: [44])

- Analog Out

  It outputs the signal from the simulation environment to a selectable analog RT Box output channel. The output signal can be modified according to formula 3.1 and the minimum and maximum output voltage can be set to be correctly interfaced with the input voltage range of the external hardware (i.e., the controller ADC input voltage range).

Figure 3.6: RT Box Target Support Library: Analog Out block (Source: [44])

- Digital In

  It allows to read digital input from a selectable digital RT Box input channel, defining also the input characteristic (i.e., if an internal pull-up or pull-down resistor is connected to the digital input).

Figure 3.7: RT Box Target Support Library: Digital In block (Source: [44])

- Digital Out

  It sets a digital output to a selectable digital RT Box output channel.

Figure 3.8: RT Box Target Support Library: Digital Out block (Source: [44])

- PWM Capture

  It averages a digital input over the period of a model step. The block output is the percentage of time during which a digital input signal was active over the last model step period [44].

Figure 3.9: RT Box Target Support Library: PWM Capture block (Source: [44])

Chapter 4

Technologies Used - Hardware Description

This chapter is dedicated to the hardware resources employed in the thesis work starting with a focus on the custom hardware used as controller to the real-time machine used for the Hardware–in–the–Loop tests.

4.1 AURIX/Arduino-like Board

The hardware implementing the controller has been developed by Ideas & Motion S.r.l., a private company located in Cherasco, nearby Turin.

The Board has been developed for the HYPER_SDF project with the objective to propose an open automotive development platform combining high-performance multi-core processors and state-of-the-art safety architecture. The computational platform aims to address future ADAS applications in the Active Safety field performing sensor fusion and providing interconnections with the E/E system of a vehicle using communication buses such as CAN, LIN, Ethernet.

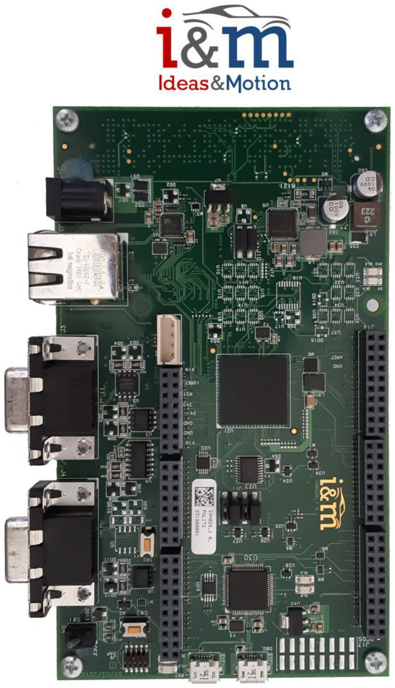


Figure 4.1: The AURIX/Arduino-like Board

4.1.1 Infineon AURIX Microcontroller

The core element of the Board is the Infineon AURIX TC277. A 32-Bit Single-Chip Microcontroller based on the Infineon TriCore Architecture, developed for the powertrain domain and able to support ASIL D applications.

It is a high-performance microcontroller with three CPU cores with two 32-bit super-scalar TriCore CPUs each having the following features [45]:

- Superior real-time performance
- Fully integrated DSP capabilities
- Fully pipelined Floating-point unit (FPU)
- up to 200 MHz operation at full temperature range

- up to 120 Kbyte Data Scratch-Pad RAM (DSPR)
- up to 32 Kbyte Instruction Scratch-Pad RAM (PSPR)
- 16 Kbyte Instruction Cache (ICACHE)
- 8 Kbyte Data Cache (DCACHE)

It also implements [45]:

- 64-Channel DMA Controller with safe data transfer
- Multiple on-chip memories:
  - All embedded NVM and SRAM are ECC protected
  - up to 4 Mbyte Program Flash Memory (PFLASH)
  - up to 384 Kbyte Data Flash Memory (DFLASH) usable for EEPROM emulation
  - 32 Kbyte Memory (LMU)
  - BootROM (BROM)
- Hardware I/O Monitor (IOM) for checking of digital I/O
- One Generic Timer Module (GTM) providing a powerful set of digital signal filtering and timer functionality to realize autonomous and complex Input/Output management
- One Capture / Compare 6 module (Two kernels CCU60 and CCU61)
- One General Purpose 12 Timer Unit (GPT120)
- Versatile Successive Approximation ADC (VADC)
  - Cluster of 8 independent ADC kernels
  - Input voltage range from 0 V to 5.5V (ADC supply)

- Digital programmable I/O ports
- On-chip debug support for OCDS Level 1 (CPUs, DMA, On Chip Buses)
- multi-core debugging, real time tracing, and calibration
- four/five wire JTAG (IEEE 1149.1) or DAP (Device Access Port) interface
- Power Management System and on-chip regulators
- Clock Generation Unit with System PLL and Flexray PLL
- Embedded Voltage Regulator

A focus on the employed peripherals will be set out below:

VADC

The microcontroller provides a series of analog input channels all connected to a cluster of A/D converters based on the Successive Approximation Register (SAR). The Versatile ADC structure is shown in figure (4.2).

The conversion speed and sampling time are adjustable and can be adapted to sensors. The VADC is capable of a conversion time below 1 μs (depending on result width and sample time) and it allows to select result width between 8/10/12 bits, Fast Compare Mode and independent result registers.

For what concerns the conversions result data, they can be preprocess to a certain extent before being stored for retrieval by the CPU or a DMA channel.

Results of a certain group can be stored in one of the 16 associated group result registers or in the common global result register (this is what happen when using the background source). The group result register returns the result value and the channel number while the global result register returns the result value, the channel number and the group number.

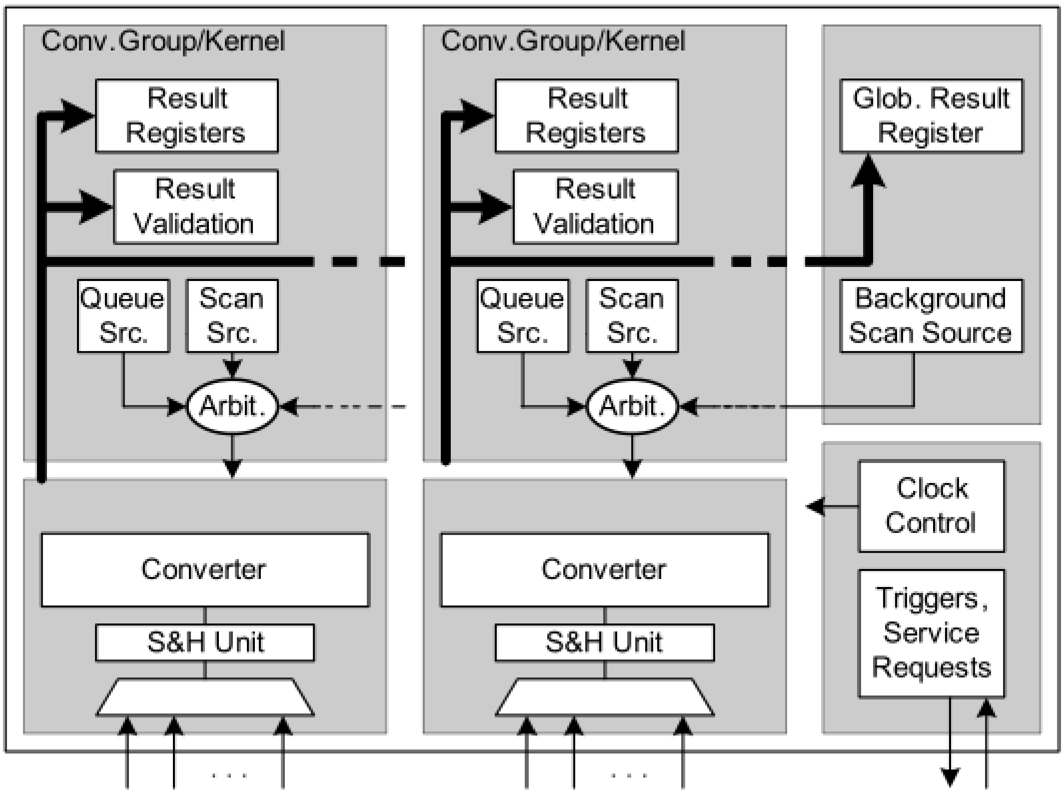


Figure 4.2: ADC Structure Overview (Source: [45])

Individual SAR converters have a dedicated S&H unit and the number of input channels for the chosen microcontroller is 64 divided in 8 groups.

Each converter of the ADC cluster can operate independently from the others, can be controlled by a dedicated set of registers and triggered by a dedicated group request source.

The VADC provides two types of request scan sources [45]:

- Queued Request Source

  A queued request source supports short conversion sequences (up to 8) of arbitrary channels (contrary to a scan request source with a fixed conversion order for the enabled channels). The programmed sequence is stored in a queue buffer (based on a FIFO mechanism). The requested channel numbers are entered via the queue input, while queue stage 0 defines the channel to be converted next. The trigger and gating unit generates trigger events from the selected external (outside the ADC) trigger and gating signals. For example, a timer unit can issue a request signal to synchronize conversions to PWM events. Trigger events start a queued sequence and can be generated either via software or via the selected hardware triggers.

- Channel Scan Request Source
  The VADC provides two types of channel scan sources:
  - Source 1: Group scan source
    This scan source can request all channels of the corresponding group.
  - Source 2: Background scan source
    This scan source can request all channels of all groups. When performing background scan source, once the conversion is triggered, the A/D converters start to convert continuously and store the result in the global result register, but this result is maintained only till the new result is available.

Another important feature provided by the VADC is the possibility to schedule self-timed conversion as well as triggered by external signals to perform synchronized conversion for parallel sampling (within a synchronization group) to optimize control of electrical drives and to perform equidistant sampling in a fixed raster with minimum jitter to optimize filter algorithms or audio applications.

CCU6

The Capture/Compare Unit 6 (CCU6) is a high-resolution 16-bit capture and compare unit mainly designed for AC drive control.

It is made up of a Timer T12 Block with three capture/compare channels and a Timer T13 Block with one compare channel. The T12 channels can independently generate PWM signals or accept capture triggers, or they can jointly generate control signal patterns to drive AC-motors or inverters. The CCU6 structure is shown in figure 4.3.

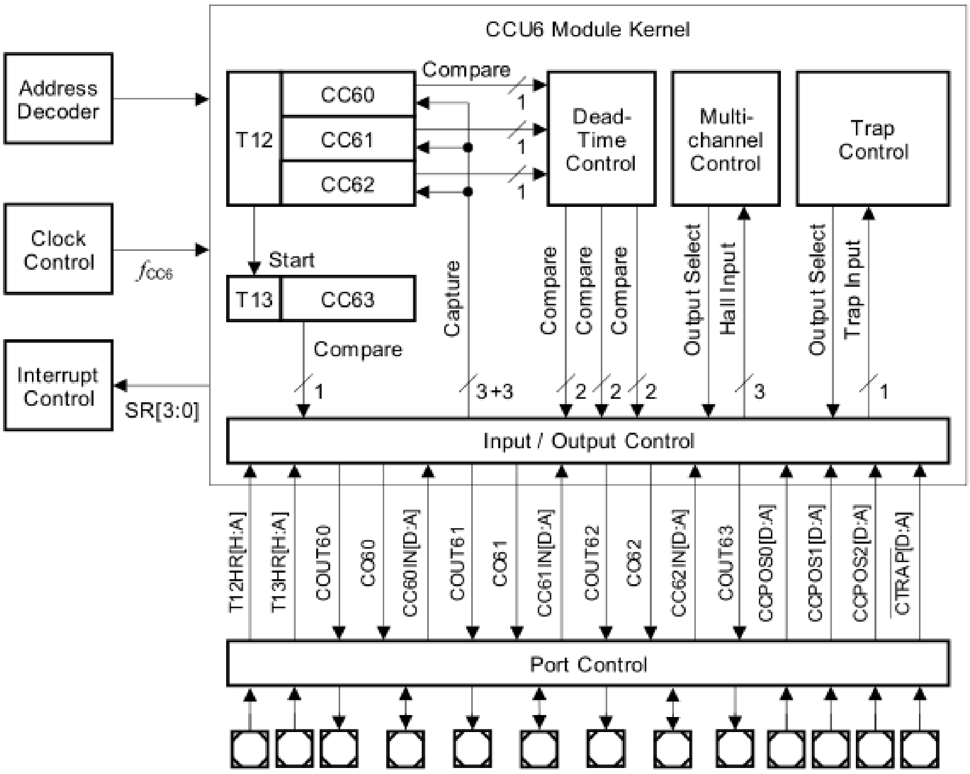


Figure 4.3: CCU6 Block Diagram (Source: [45])

In this thesis only the Timer T12 has been used. The main features of this Block are described below [45]:

- Three capture/compare channels, each channel can be used either as capture or as compare channel
- Generation of a three-phase PWM supported (six outputs, individual signals for high-side and low-side switches)
- 16-bit resolution, maximum count frequency = peripheral clock
- Dead-time control for each channel to avoid short-circuits in the power stage

- Concurrent update of T12 registers
- Center-aligned and edge-aligned PWM can be generated
- Single-shot mode supported
- Start can be controlled by external events
- Capability of counting external events
- Many interrupt requests sources

The Timer T12 also allows fast emergency stop without CPU load via external signal, noise filter for position input signals, integrated error handling and selection of the output levels to be adapted to the power stage.

Timer T12 is the main unit to generate the 3-phase PWM signals. A 16-bit counter is connected to 3 channel registers via comparators, that generate a signal when the counter contents match one of the channel registers contents (a more detailed description of how it works will be provided in the following).

The Timer T12 receives the input clock from the module clock (received from the TBU (Time Base Unit)) via a programmable pre scaler and an optional 1/256 divider or from an input signal. These options are controlled via bit fields in the dedicated block registers.

The T12 can count up or down or up/down depending on the chosen operation mode and the counting direction is indicated by a direction flag.

The T12 counter register is related to a Period Register defining the maximum count value for the Timer.

Two operating mode can be selected: the Center-Aligned mode and the Edge-Aligned mode.

In Edge-Aligned mode, the counter direction is always upwards.

Referring to the Center-Aligned mode, the count direction of the Timer is set from up to down (and from down to up) after it has reached the period value. Once the counter has reached that value (it is important to underline that the counter exceeds that value by one before counting down causing jitter) a *Period Match* signal is generated, and the counting direction is inverted.

The Period Register receives a new period value from its Shadow Period Register to be loaded on the Period Register only when the Shadow Transfer control signal is enabled.

The resulting timer period, for Center-Aligned mode, is [45]:

$$T12_{PER} = (Period\text{-}value + 1) * 2 \quad inT12clocks(f_{T12}) \tag{4.1}$$

The use of shadow registers not for the period only but for other values related to the PWM generation allows a concurrent update by software for all relevant parameters.

Two further signals indicate whether the counter contents are equal to 0000_{H} (T12_ZM = zero match) or 0001_{H} (T12_OM = one match). These signals control the counting and switching behaviour of T12.

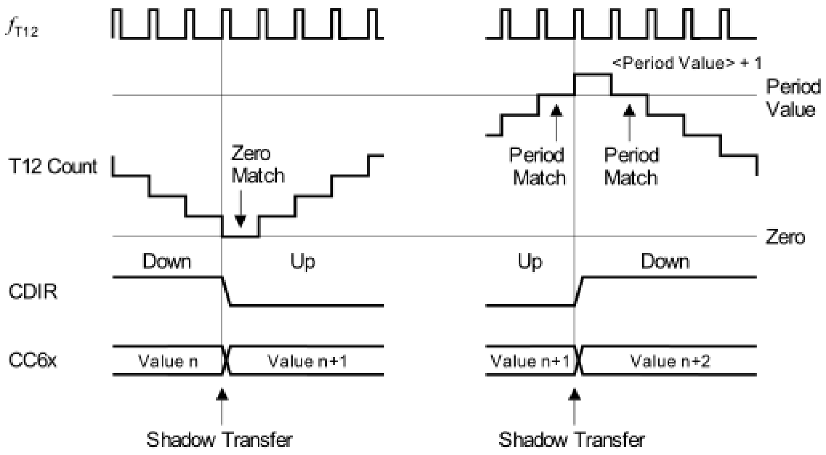


Figure 4.4: T12 Operation in Center-Aligned Mode (Source: [45])

The Timer T12 is operated in Compare Mode where three individual compare channels can generate a three-phase PWM pattern.

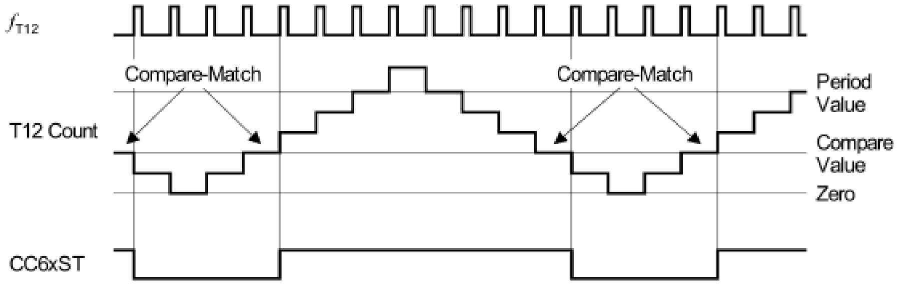


Figure 4.5: Compare Operation in Center-Aligned Mode (Source: [45])

What is necessary to distinguish between simulation and real application of AC drives control, focusing on a three-phase inverter, is that the time instant, in which a commutation between high-side and low-side switches of one power inverter stage occurs, is not immediate, but a delay is present depending on the switching time of the devices from which the inverter is made.

Indeed, in a not ideal switch, turn on and turn off times are not strictly identical and if during the commutation of high-side and low-side switches these time intervals overlap, it causes a short-circuit in the power inverter phase.

For this reason, it is recommended to insert a certain amount of time delay also called "dead time" into the inverter control scheme. In this way, the switch will be always turned off first and then the other will be turned on after the dead time is expired so that short-circuits in the power stage can be avoided.

One of the main features of the CCU6 is the Dead-Time Generation block, shown in figure (4.6), used to solve the previously explained problem by hardware, resulting in a more reliable solution and delaying the commutation by a programmable time.

The length of the dead time is programmed by a dedicated bit field in the block registers. This value is the same for all the three channels.

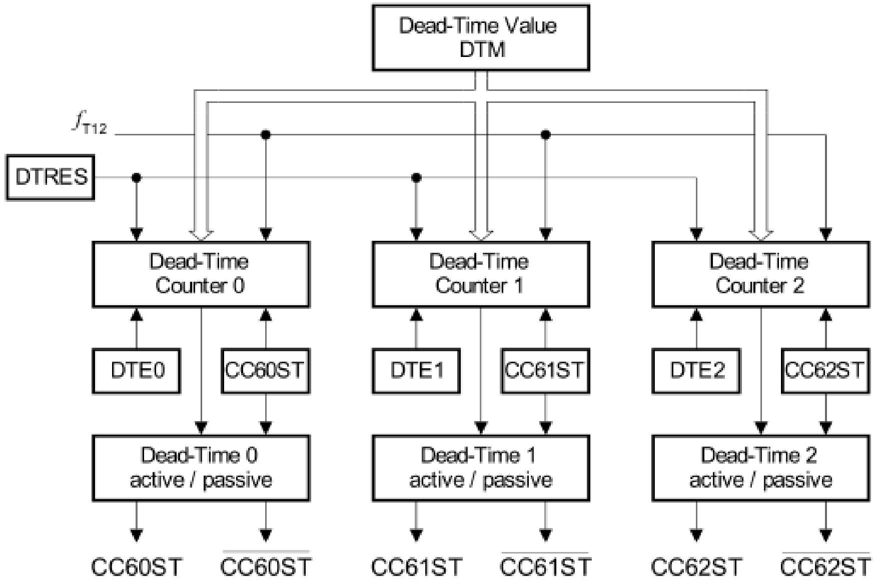


Figure 4.6: Dead-Time Generation Block Diagram (Source: [45])

4.2 Plexim RT Box

PLECS RT Box is a real – time simulator designed for Hardware-in-the-Loop testing and emulates power stages of a power electronic system.

The device to be tested, typically a control hardware, is connected to the RT Box allowing to perform tests without a real plant.

The RT Box uses PLECS environment to interact with the models developed for simulation purposes, models that can also be employed for HIL testing by using dedicated blocks to create an interface with the external world.

These blocks allow the definition of the Analog and Digital Input and Output port of the RT Box.

To simulate a real plant for verification and validation purposes, analysing the system behaviour during transients and faults and also to avoid conditions which could be destructive, a simulation time in the order of μs is necessary for testing of motor drives where PWM frequencies are typically between 5 kHz and 20 kHz.

RT Box can be also used for rapid control prototyping performing as a controller for the power stage.

Figure 4.7: Plexim RT Box (Source: [46])

Many interfaces are available for a fast and easy connection of external hardware; they are presented below [46]:

- The Analog Breakout Board makes the analog inputs and outputs individually accessible via BNC sockets.

- The Digital Breakout Board makes all digital inputs and outputs accessible via terminal blocks and pin headers.

- The LaunchPad Interface facilitates a simple connection of the RT Box with the LaunchPad and LaunchPad XL development kits from Texas Instruments.

- The ControlCard Interface allows to connect various ControlCards from Texas Instruments with the RT Box.

To facilitate the interface between the RT Box and the Board used during this thesis work, the LaunchPad Interface Board has been employed.

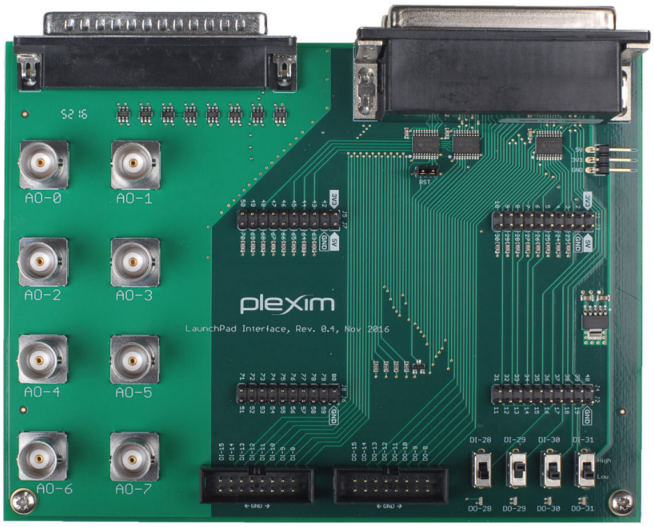


Figure 4.8: LaunchPad Interface Board (Source: [47])

For what concerns the RT Box internal architecture, it is equipped with two ARM cores: one is used to manage all the communications to and from PLECS and also through the web interface; the second ARM core is used for the real-time applications.

The block diagram in Fig. 4.9 clarifies the internal RT Box system architecture.

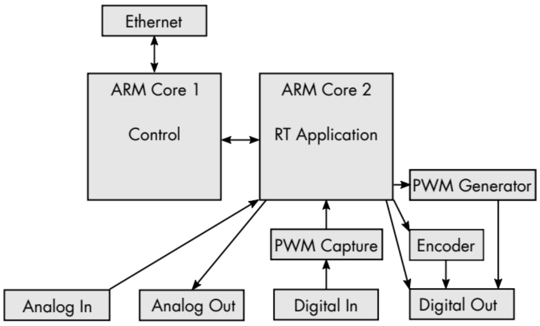


Figure 4.9: RT Box internal architecture block diagram (Source: [44])

Simulations, inside RT Box, are executed with a fixed discretization step, t_{Cycle} and this is the periodic interval at which the step function is executed.

To perform a real-time simulation, the step function must be completely executed inside the periodic interval defined by the discretization step; if this condition is not respected, the overrun error is shown in the RT Box diagnostic panel and by one of the four external LED.

Fig. 4.10 shown the input/output timing of the RT Box.

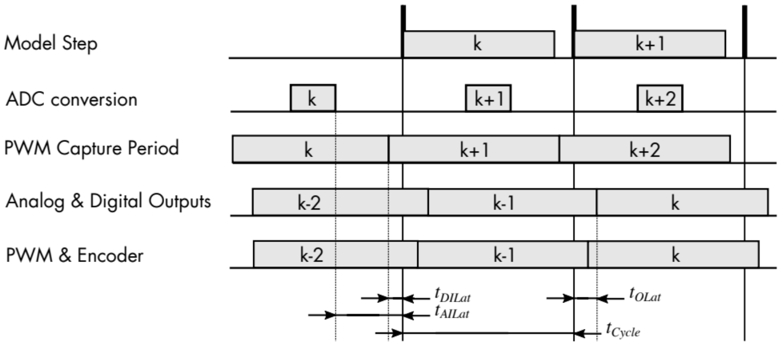


Figure 4.10: RT Box I/O timing (Source: [44])

Technical specifications of the RT Box 1 are presented in Appendix

Chapter 5

Implementation and Results

5.1 Implementation

In section 4.1 has been introduced the Board implementing the controller and its peripherals with all the possible available resources. The current section is dedicated to how these peripherals have been used, describing the selected solutions, from a hardware point of view; for what concerns the software, the step-by-step implementation of the control algorithm will be described in the test sections.

Before entering in details, it is worth defining the adopted project assumptions from which the models have been derived.

The assumptions made for the inverter are presented in the following table:

Inverter Specifications		
Parameter	Symbol	Value
DC-link Voltage	V_{dc}	400 V
Rated current	I_{max}	200 A
Switching frequency	f_{sw}	10 kHz
Dead-time	Δt_{dt}	3 µs

Table 5.1: Inverter assumptions

5.1.1 A/D Converter

For what concerns the analog-to-digital converter, the initial adopted source has been the background scan source, however, from the tests (described in details in the following sections), it turned out to be not as performing and reliable as expected, as well as for the group scan source.

The final choice has been the queue scan source, where selectable ADC channels are queued and read in the defined order.

In the application shown in this thesis, the ideal configuration for the sampling of the input quantities would have been the parallel acquisition of all of the desired quantities through a precise number of independent ADCs, according to the number of signal to be acquired (one for each ADC group).

However, due to hardware limitations, it has not been possible to apply this solution; therefore only 5 of the 64 channels have been used:

- 3 channels dedicated to the sampling of the three-phase currents (ADC Group 7; channels 2, 3 and 4);
- 1 channel dedicated to the sampling of the DC-link voltage (ADC Group 2; channel 0);

- 1 channel dedicated to the sampling of the signal from the ideal encoder measuring the rotor mechanical angle (ADC Group 2; channel 1).

The VADC has been set up in order to obtain the best possible performances, thus the maximum available frequency for the VADC module has been exploited (corresponding to 100 MHz) to reach small conversion times.

It is worth pointing out that, to obtain good control performances, the sampling of the controlled quantities shall be perfectly synchronized with the execution of the PWM commands.

5.1.2 Capture/Compare Unit

The PWM generation with dead-time has been entrusted to the CCU6 Timer T12, used in Center-Aligned Mode.

This choice has been based on the necessity to perform the sampling of the three-phase currents in a time instant where no commutations of the power stage occur, to obtain a currents feedback as clean as possible and avoiding sparks on the acquired signals.

This is required when dealing with AC drives for traction applications where the currents have relevant values. By using an Edge-Aligned Mode, this result could not be achieved.

From the converter switching frequency f_{sw}, the switching period $T_{sw} = 100$ µs is obtained; it defines the PWM period to be set for the Timer T12.

Since the counter of the Timer T12 is an up-down counter and since, as previously mentioned, the sampling synchronization is required, the CCU6 triggers the ADC conversion in correspondence of the upper vertex of the triangular carrier of the modulator, starting the control routine with the acquisition of the required input quantities.

These quantities are sampled once in a sampling period T_s, imposed to be equal to the switching period T_{sw}.

If the conversion is correctly performed (exactly in correspondence of one of the triangular carrier vertex, assuming 1 sample 1 refresh (*1S1R*) strategy), for instance, the current sample is equal to the current average value in that period (T_{sw}), thus the ripple does not affect the result.

It is worth mentioning that, when using DSP, delays are introduced (the faster is the DSP, the smaller is the delay), thus if the quantities are sampled at the time instant *t*, the actuation of the computed commands will occur at time *t+1*.

A double sampling technique could be used to reduce this delay since the quantities are sampled twice in a period T_s.

From the provided basic software, the CCU6 code has been improved with some features in order to adapt the Board to this thesis application. This is related to the TRAP functionality, which allows the CCU6 module to provide or not PWM signals to the output.

In this type of application, it is required that the commutation signals provided to the inverter, during the start-up procedure, correspond to the open position of the power switches not to supply current to the load.

Thus two functions, *CCU6_EnablePWM* and *CCU6_DisablePWM* have been created to be interfaced with the GPIO pins of the Board.

5.1.3 GPIO

As described in the previous section, the GPIO have been used to interface the Board with the RT Box LaunchPad switches, more details will provided in the test section.

This allows to provide external commands when the corresponding switch is manually activated.

Other GPIOs have been used for test purposes.

5.1.4 Debug

Since the Board does not present an adequate number of Digital-to-Analog converter (DAC) outputs, an alternative solution has been proposed for debugging purposes: in the code, different arrays have been defined with a number of samples that allow to observe the variables of interest in a sufficient wide time range. According to the execution time, the samples are collected every 100 µs.

Therefore, an array of 2500 elements can represent an interval of time of 0.25 s. It is not a huge interval, but it is big enough to observe the behaviour of the system during transients.

From the Universal Debug Engine (UDE), provided by the HighTec Development Platform, a *.xml* file can be exported.

A MATLAB script has been written, reading the file and plotting data as shown in the tests section.

5.2 Validation

As described in section 2.1, after the controller synthesis, the following step in the Model-Based Design flow is characterised by offline and real-time simulations, to test and compare the expected results from the former to the real ones from the latter.

Section 3.5, describes the software environment used for offline simulations, where the code to be implemented in the Board is employed to perform SIL tests.

For real-time simulations, the RT Box 1, mentioned in section 4.2 has been used to emulate the behaviour of the eDrive, with the Board connected, and working as a controller to test not only the final code, but all the related peripherals.

The HIL setup scheme is shown in figure 5.1, while the corresponding real HIL setup is presented in figure 5.2.

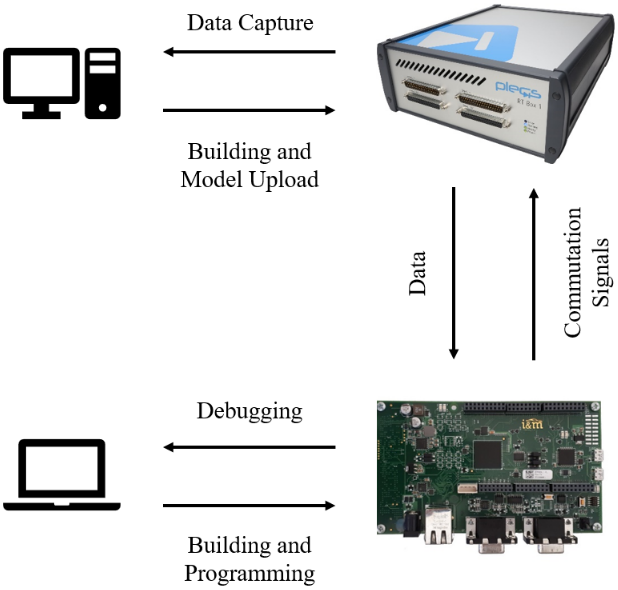
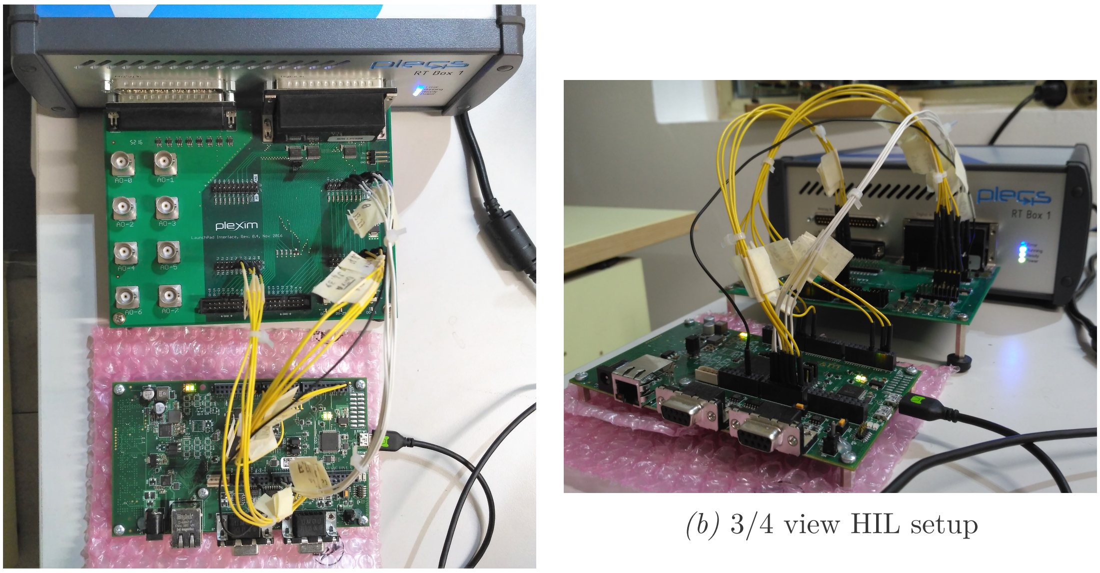


Figure 5.1: HIL setup scheme



(a) Top view HIL setup

(b) 3/4 view HIL setup

Figure 5.2: HIL setup

In the following, the step-by-step procedure from the peripherals test to the motor control algorithm will be described.

5.3 Test ADC

In section 3.5.1, the main blocks used in PLECS environment, to interface the RT Box with external elements, have been described.

The aim of this test has been to verify and calibrate the ADC from both RT Box and Board side.

The schematic in PLECS environment used to perform this test is shown in figure 5.3

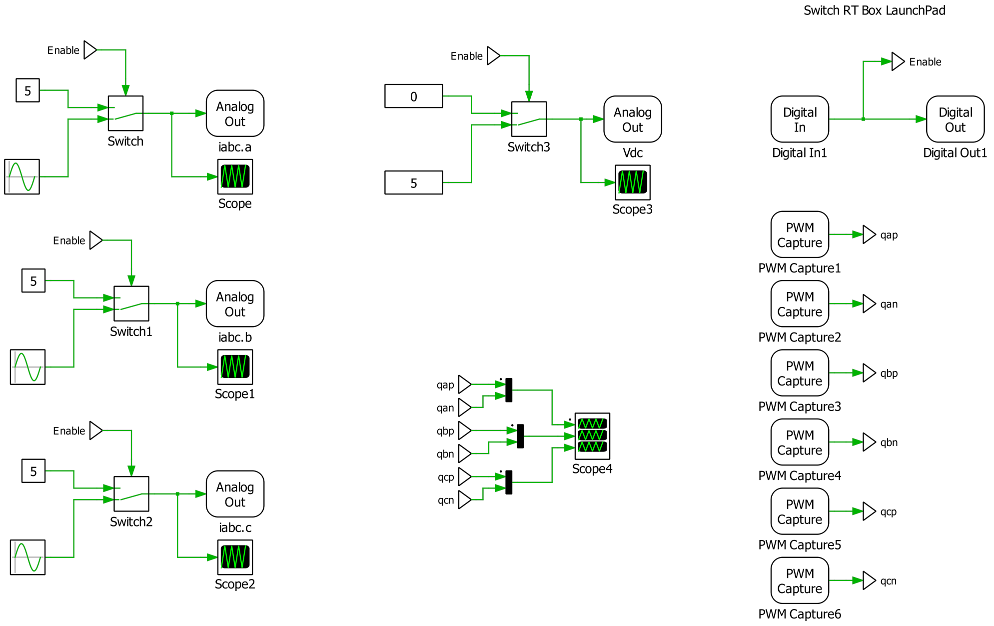


Figure 5.3: PLECS schematic for ADC test

The blocks previously mentioned have been used to adapt the voltage range of the analog output RT Box signals to the input voltage range of the Board A/D converters, which is 0 V - 5 V, and scale the analog quantities to be adapted to the values defined in the assumptions in table 5.1.

For what concerns the currents, in the blocks *Analog Output*, the minimum output voltage and the maximum output voltage have been set to 0 V and 5 V, respectively.

Since the current minimum and maximum values have been defined to -200 A and 200 A, respectively, according to the formula 3.1, the *Scale* and *Offset* parameters have been set to obtain:

$$signal_{out} = signal_{input} \cdot \frac{5}{400} + 2.5 \tag{5.1}$$

being 5 the output voltage range, 400 the current values range and 2.5 the offset to be introduced to obtain the waveform between the desired range.

From the Board side, the signal scaling has been performed with the following code, considering the resolution of the A/D converter is 12-bit.

```
// Three-phase currents sampling and scaling
Adc_StartQueueConv( 1, TRUE );
Adc_ReadChRes(4, &iabc.a);
iabc.a = ((iabc.a*400)/4095 - 200);
Adc_ReadChRes(3, &iabc.b);
iabc.b = ((iabc.b*400)/4095 - 200);
Adc_ReadChRes(2, &iabc.c);
iabc.c = ((iabc.c*400)/4095 - 200);
```

The ratio 400 A/2^{12} = 400 A/4095 defines the resolution of the measure.

For the DC-link voltage the procedure is similar, but the full-scale is different: the minimum and the maximum value have been defined to 0 V and 550 V, thus

$$signal_{out} = signal_{input} \cdot \frac{5}{550} \tag{5.2}$$

```
// DC-link voltage sampling and scaling
Adc_StartQueueConv( 0, TRUE );
Adc_ReadChRes(0, &Vdc);
Vdc = ((Vdc*550)/4095);
```

Once the signal scaling procedure has been completed, with the help of the Universal Debug Engine (UDE) window, the precision of the sampled values has been verified, based on the values imposed in the constant block and in the sinusoidal wave block, switching between them.

The frequency of the sinusoidal wave has been imposed to be equal to the modulation carrier frequency ($f_{sw} = 10$ kHz) to verify that in correspondence of the sampling instant, the acquired value is exactly the average current value.

This test has been carried out also to verify if the commutation signals generated by the capture/compare unit have been correctly acquired by the *PWM Capture* blocks. A duty-cycle constant value, tunable from the UDE interface, has been set to test the function *CCU6_PWM_SetDutyCycle(duty_a, duty_b, duty_c)* and the correct dead-time insertion. The result is shown in figure 5.4.

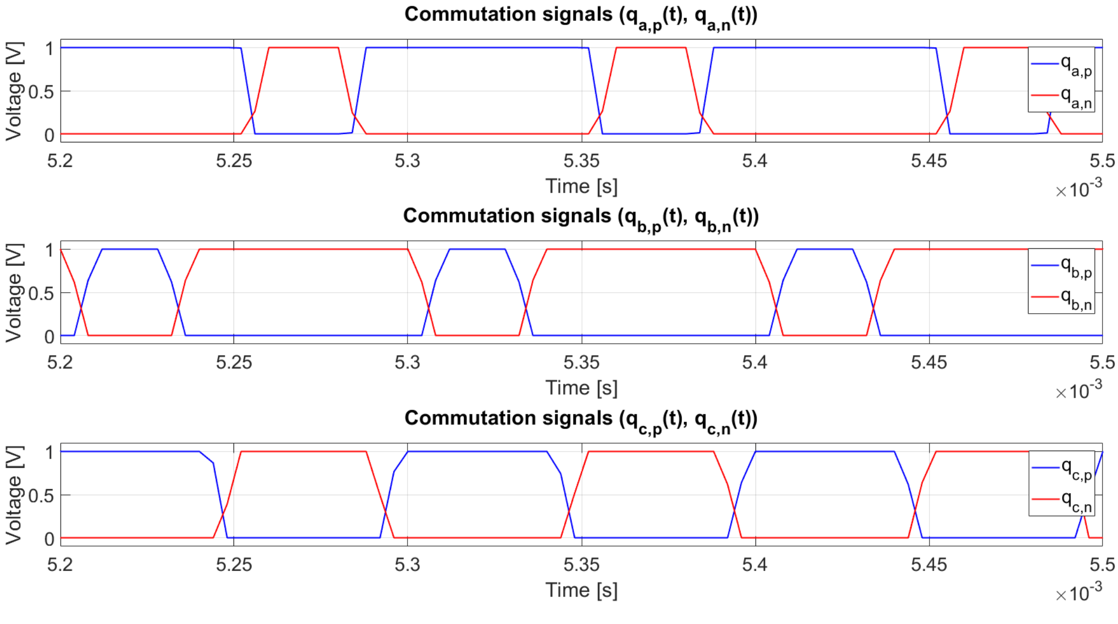


Figure 5.4: Commutation signals acquired by the *PWM Capture* block

The function allows to set the three main channel duty-cycles; the complementary channels are obtained accordingly.

The test has given satisfying results, showing that the VADC and CCU6 units work properly and the sampled values are very close to the real ones. A ±2 units accuracy has been observed.

5.4 Test Single Inverter Leg with RL Load

After the correct behaviour of the CCU6 and the *PWM Capture* blocks has been verified from the previous test, the one described in the following is aimed to implement a preliminary inverter configuration, using a single inverter leg with an RL load (refer to figure 5.5), with fixed input and output voltage to evaluate the sampled inductor current.

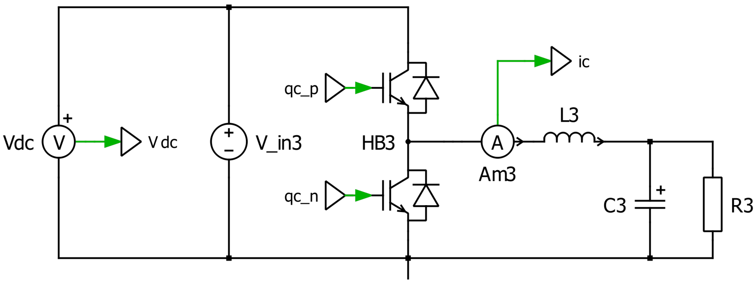


Figure 5.5: Single inverter leg with RL load

The commutation signals have been obtained once again from the function *CCU6_PWM_SetDutyCycle(duty_a, duty_b, duty_c)*, where constant duty-cycle values are imposed, tunable from the UDE interface.

The initial simulation parameters used for this test are shown in the following table:

Simulation		
System Parameters		
Parameter	Symbol	Value
Phase inductance	L	200 µH
Phase resistance	R	1 Ω
Output capacitance	C	1 mF
Input voltage	V_{in}	100 V
Capacitor initial condition voltage	V_{out}	50 V
Duty-cycle	d	50 %

Table 5.2: Single inverter leg with RL load test: default simulation parameters

The capacitance value has been chosen high enough to be approximated as a constant voltage source.

The PLECS simulation environment employed for this test is shown in appendix 6, in figure 4.

The inductor current has been measured using an ammeter, whose signal is then addressed to the *Analog Output* block and thus read by the Board A/D converter.

The test has been performed by changing the parameters defined in table 5.2 and verifying if the expected current average values (from PLECS virtual oscilloscope) were equal to the measured ones.

This has been repeated for all the three legs, obtaining almost identical results. The comparison between expected and measured current average values is shown in the burn down charts presented in appendix 6 with details on the adopted simulation parameters and obtained measures.

From the burn down charts, it can be observed that the measured values are close to the expected ones, except for particular simulation conditions, where very low inductance and high duty-cycle values have been assumed. To better investigate this situation, from the PLECS virtual oscilloscope it has been observed that the sampling is performed 6 μs before the correct time instant, and thus giving wrong results.

Anyway, the previously mentioned situation represents a limit case, which is not reached, typically, in motor control applications, also because traction motors inductance values are higher than the one considered for this test (typically in the order of mH).

In any case, this may represents a limiting condition if high performances control has to be implemented because high current controller bandwidths cannot be achieved.

A possible solution to this problem could have been to use the second timer present in the CCU6 unit, Timer T13, to add a programmable delay to the routine triggered by the *Period Match* from Timer T12. In this way, when the interrupt (corresponding to the *Period Match* condition) is activated, the Timer T13 starts to count till the set value (equal to the programmable delay that want to be imposed) is reached. Once a second interrupt, triggered by the counter, has been enabled, the Timer T13 trigger an A/D conversion.

5.5 Test V/f Control

As explained in section 2.7.1, the V/f control is used for preliminary tests of the inverter to check that modulation performs in the expected way, for calibrations and to check if the controller is able to generate a voltage with the reference frequency and amplitude defined in (d,q) reference frame.

Offline Simulation

For this type of test, first an offline simulation has been realised trying to reproduce as accurate as possible the real system where the controller would be implemented and tested.

A relevant modification has been realised in this simulation that will be also reported in the one which will interact with the RT Box and the controller Board (the real-time simulation or HIL simulation): the electrical angle θ_e is computed starting from the desired output frequency (the emulation of the encoder has not been implemented) as shown in appendix, in figure 5.

Anyway the angle range has been chosen in the range between 0 and 2π, to reproduce the range of the angular position of a rotor, since the final objective is to drive an AC motor, in particular a synchronous machine.

This is used to create the three-phase voltage system being part of the L-R-E load, according to the following formulation:

$$\begin{cases} e_a = \hat{E}cos(\theta_e) \\ e_b = \hat{E}cos(\theta_e - \frac{2\pi}{3}) \\ e_c = \hat{E}cos(\theta_e + \frac{2\pi}{3}) \end{cases} \tag{5.3}$$

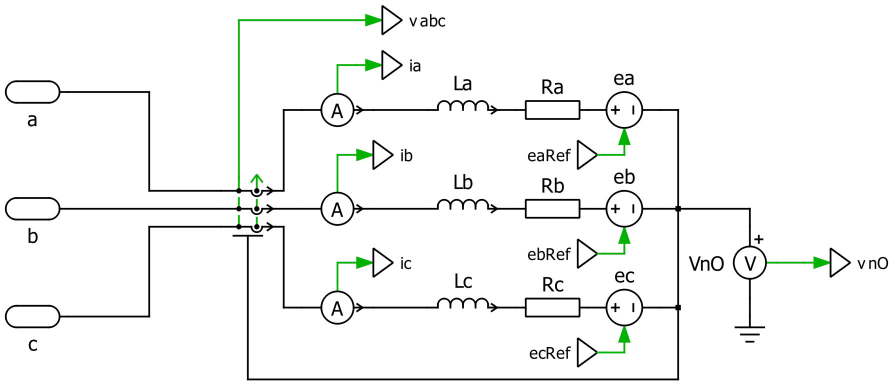


Figure 5.6: Balanced L-R-E load

Concerning the offline simulation, the simple open-loop control algorithm has been implemented in the block subsystem *CONTROL*, using a C-script block.

The voltage references are set in the code, receiving the computed angle θ_e, used for the direct and inverse rotational transformations.

The voltage references, after the inverse Park and Clarke transformations have been applied, are in the three-phase reference frame. They are provided as input to the PWM function, which computes the duty-cycle values to be fed to the modulation block *PWM*. The obtained commutation signals are then provided to the inverter. It is worth mentioning that in this simulation, dead-times have been implemented.

The schematic developed in PLECS environment is shown in appendix, in figure 5, where the inverter has been implemented as shown in figure 5.7, using the proposed PLECS configuration *Sub-cycle Average* for the IGBT Half Bridges:

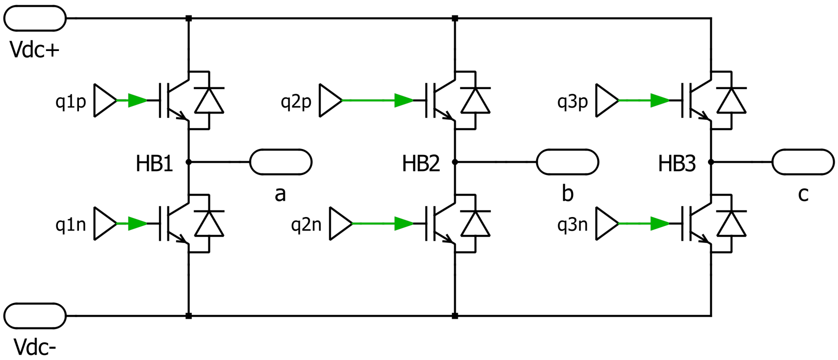


Figure 5.7: Inverter implemented in PLECS for both offline and Hardware-in-the-Loop simulation

HIL Simulation

In the HIL simulation (refer to appendix, figure 6), few changes have been applied to interface the real-time machine to the controller. The schematic modifications concerned how the commutation signals are provided to the inverter, in fact the *PWM Capture* blocks have been used (as in the previous tests).

The discretization step size has been chosen to be equal to 3 µs, according to the maximum RT Box capability for this simulation (the diagnostic tool allows to understand which is the maximum step size that can be chosen, reporting an error if overrun occurs).

5.5.1 Test V/f Control - Simulation 1

A first simulation has been implemented using the system parameters defined in the following table.

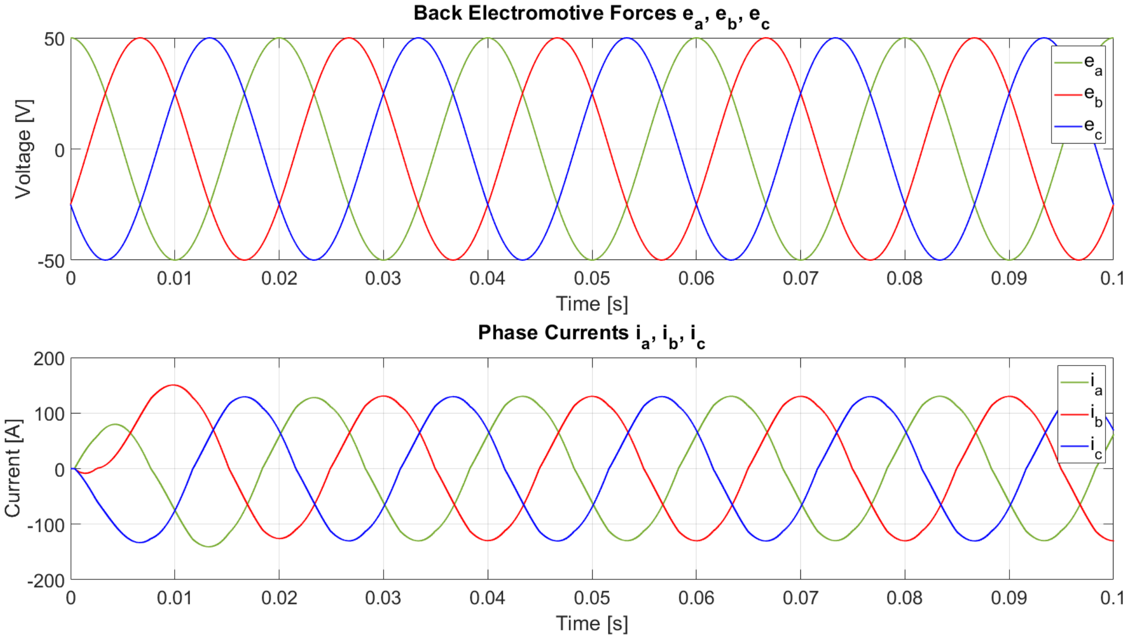
Simulation 1		
System Parameters		
Parameter	Symbol	Value
Phase inductance	L	1 mH
Phase resistance	R_s	100 mΩ
Back EMF peak value	$\hat{E}$	50 V
d-axis voltage reference	$V_{d,Ref}$	100 V
Desired output frequency	f_o	50 Hz

Table 5.3: V/f control test: simulation 1 parameters

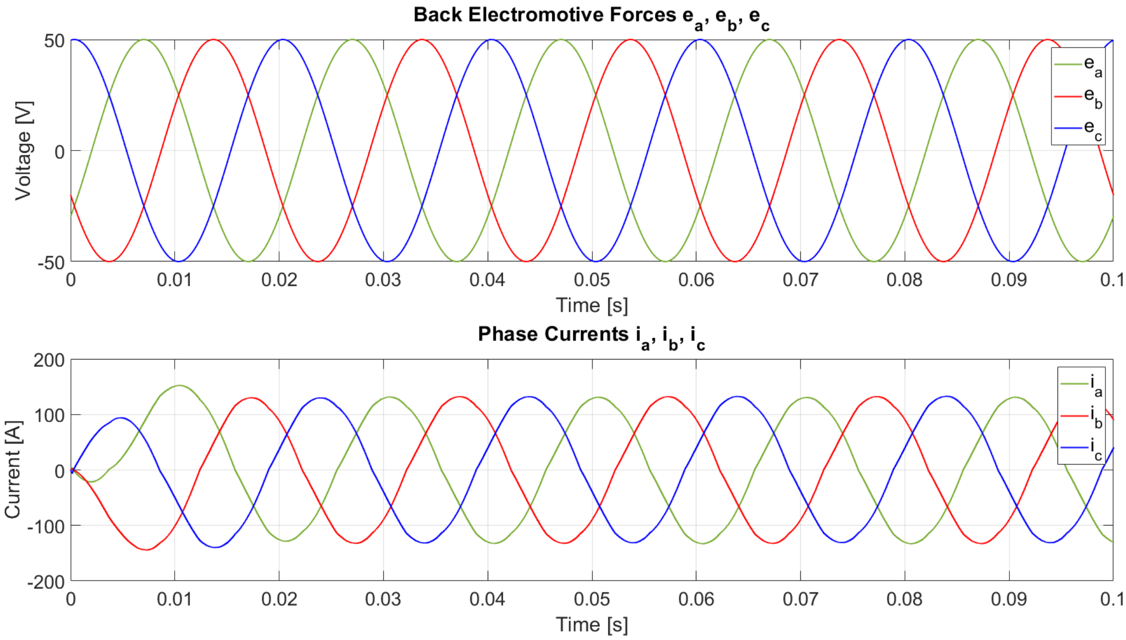
The obtained results are shown in figure 5.8 and figure 5.9.

What can be observed is that the phase currents present a phase shift with respect to the E component of the load, this is due to the high inductances value, which makes the voltage drop across the inductor higher (since the inductor voltage is proportional to the inductance value) and, considering the phasor diagram, this contributes to the phase shift between the two vector (inductor voltage vector and voltage generator vector).

In particular, in figure 5.9, it can be observed how the phase voltages are distorted; the main reason of this distortion is related to the presence of the dead-times. With the chosen high inductance value it is not so evident in the phase currents, but it still present.

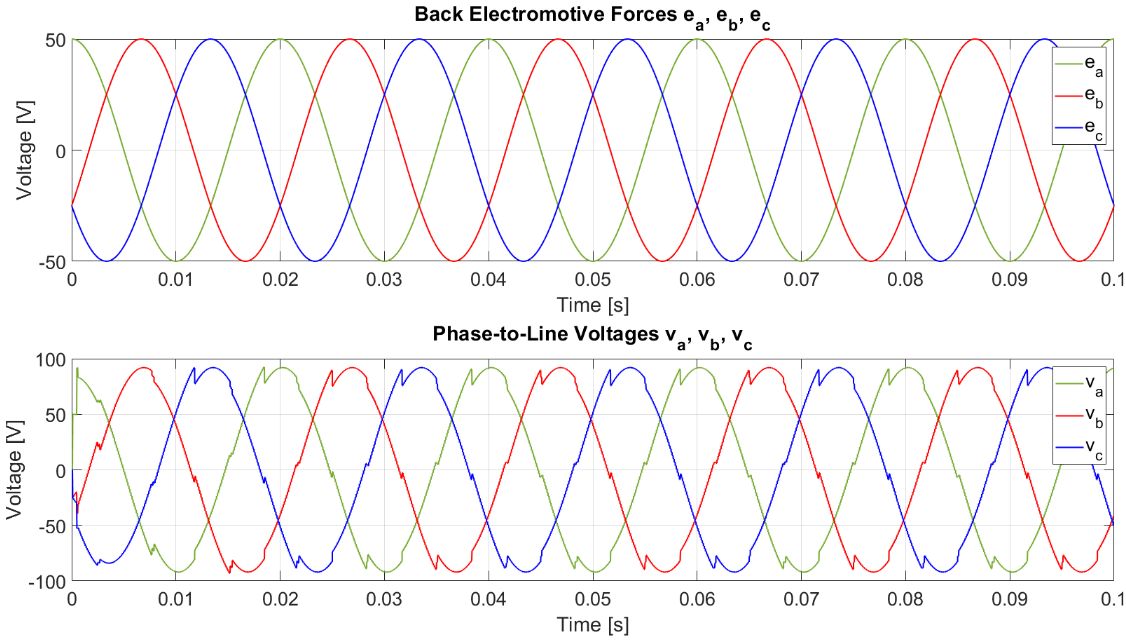


(a) Test V/f control offline simulation 1: output phase currents

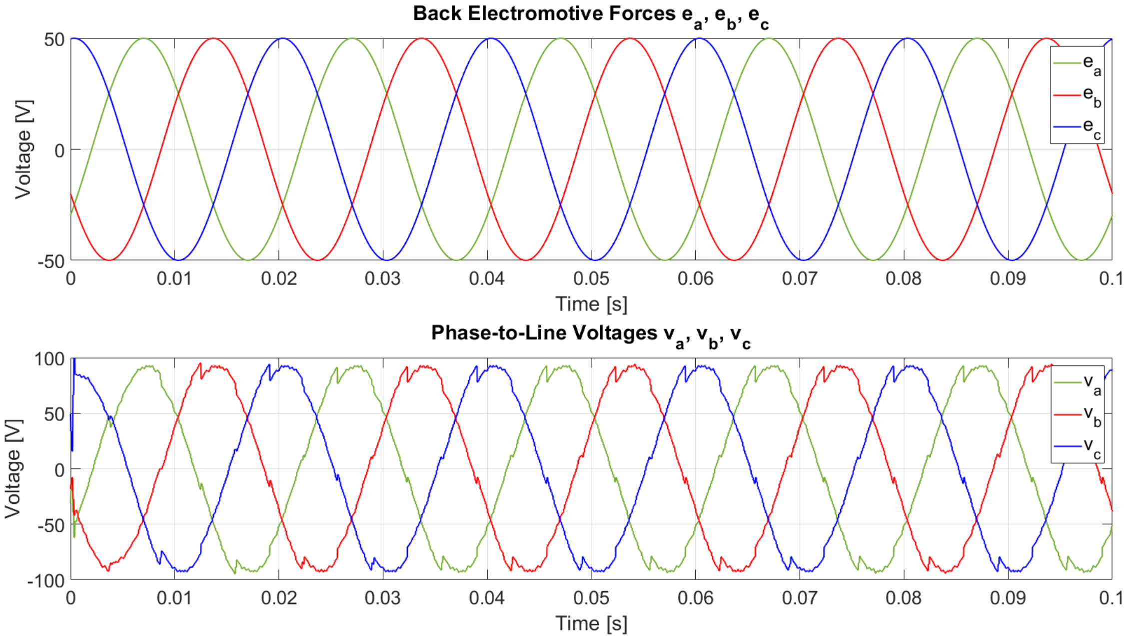


(b) Test V/f control HIL simulation 1: output phase currents

Figure 5.8: Test V/f control simulation 1 results



(a) Test V/f control offline simulation 1: output phase voltages



(b) Test V/f control HIL simulation 1: output phase voltages

Figure 5.9: Test V/f control simulation 1 results

5.5.2 Test V/f Control - Simulation 2

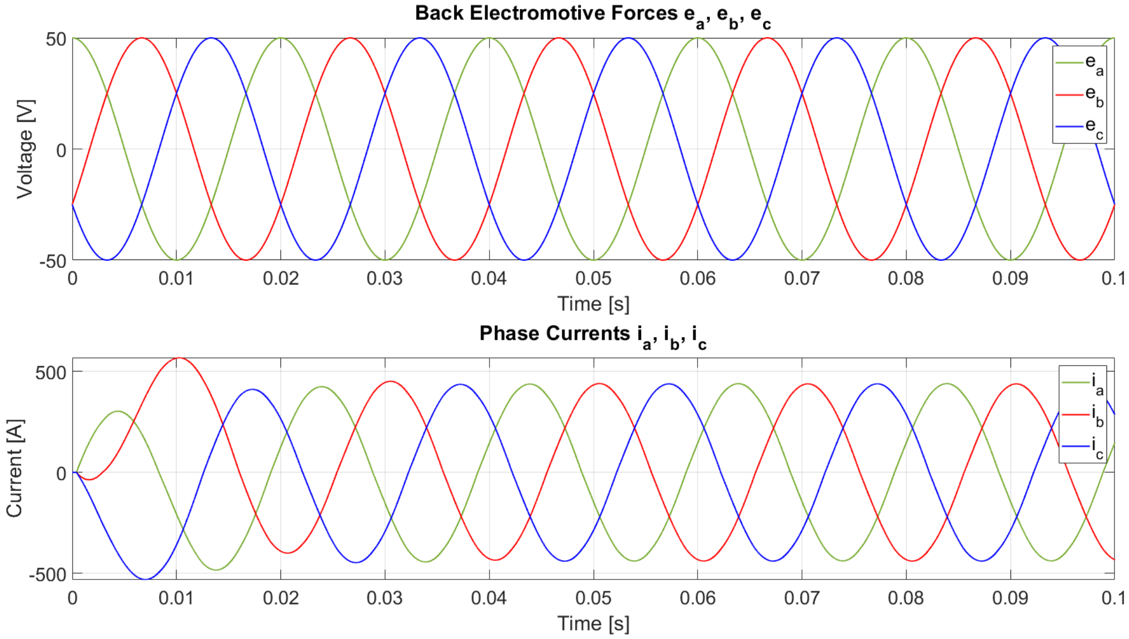
A second simulation has been performed using the system parameters defined in the following table.

Simulation 1		
System Parameters		
Parameter	Symbol	Value
Phase inductance	L	1 mH
Phase resistance	R_s	100 mΩ
Back EMF peak value	$\hat{E}$	50 V
d-axis voltage reference	$V_{d,Ref}$	200 V
Desired output frequency	f_o	50 Hz

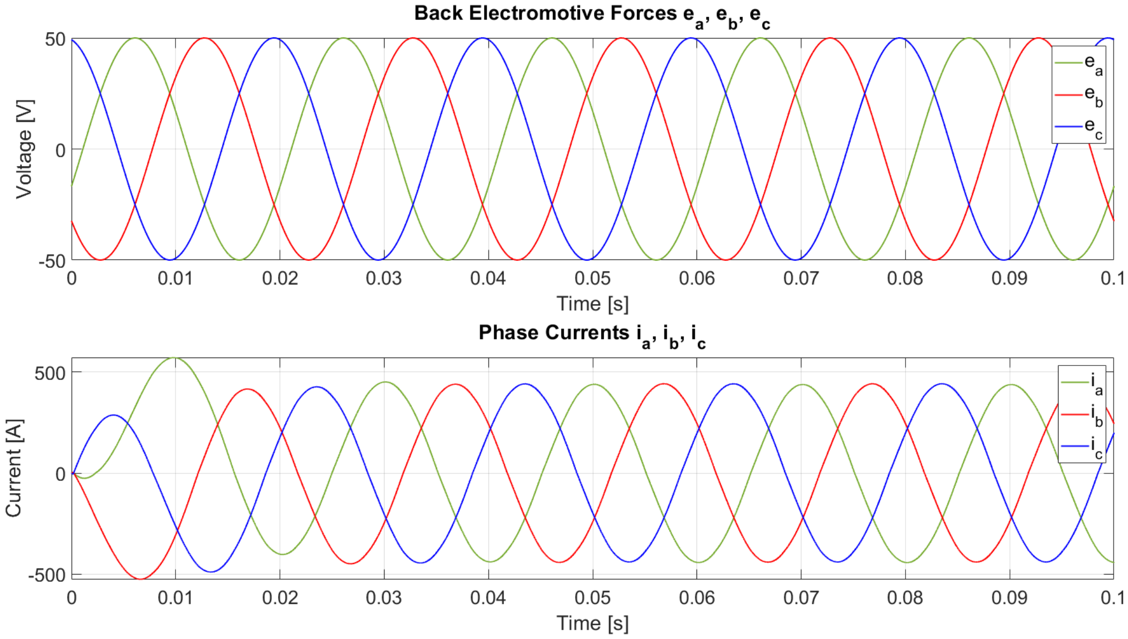
Table 5.4: V/f control test: simulation 2 parameters

The obtained results are shown in figure 5.10 and figure 5.11.

What it is worth highlighting in this simulation is the peak value of the phase currents: it is clear that in real applications the currents are limited by the inverter and current protection solutions should be implemented. Anyway, what is interesting is that if in Simulation 1, the phase currents peak value is close to the peak value of the phase voltages, this does not happen in Simulation 2.

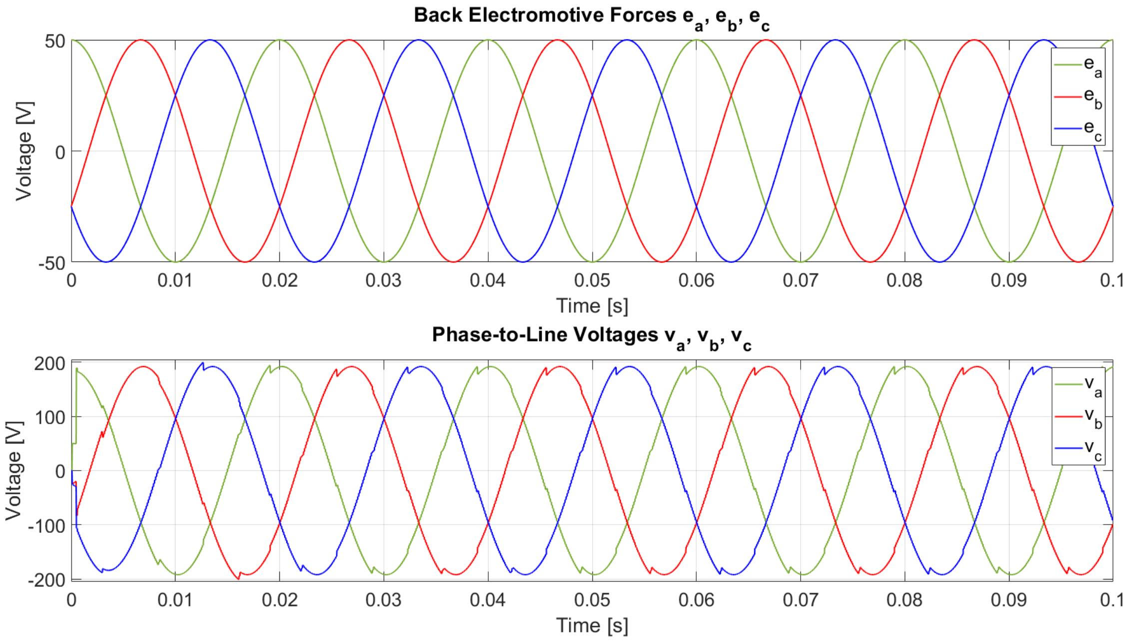


(a) Test V/f control offline simulation 2: output phase currents

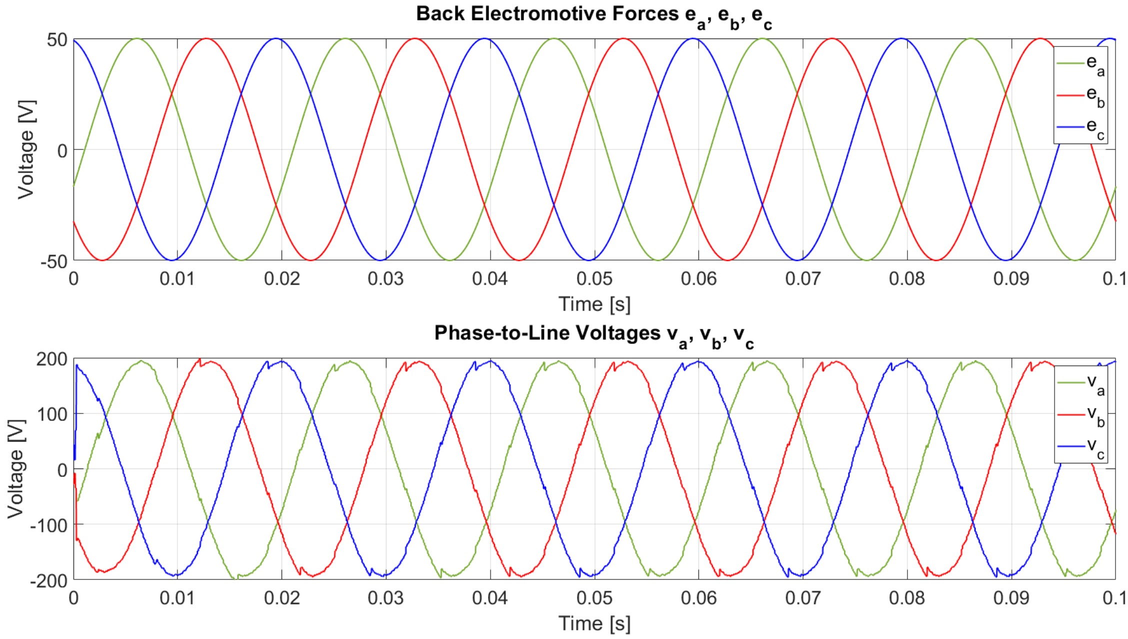


(b) Test V/f control HIL simulation 2: output phase currents

Figure 5.10: Test V/f control simulation 2 results



(a) Test V/f control offline simulation 2: output phase voltages



(b) Test V/f control HIL simulation 2: output phase voltages

Figure 5.11: Test V/f control simulation 2 results

5.6 Test I-f Control with L-R-E Load

The aim of the I-f control, as explained in section 2.7.2 is to verify the correct inverter operations and tune PI current regulators parameters. The adopted control scheme is reported in figure 2.38.

Offline Simulation

Even for this type of test, first an offline simulation has been realised trying to reproduce as accurate as possible the real system where the controller would be implemented and tested.

For this purpose an *Analog Conditioning* subsystem has been introduced taking into account for the signal conditioning circuit, considering gain, offset and a low-pass filter:

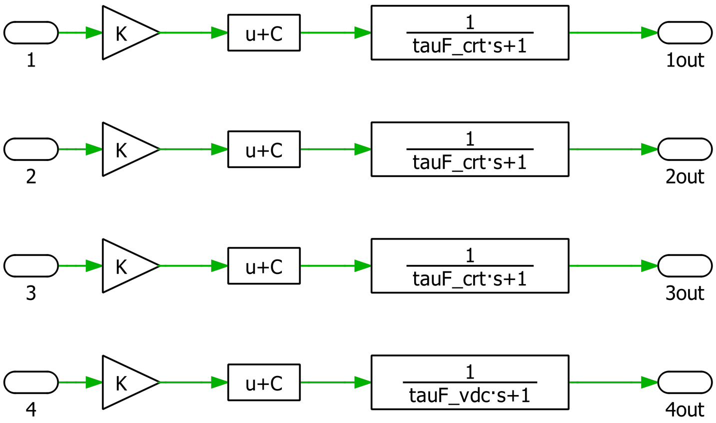


Figure 5.12: Analog conditioning circuit

The control is performed using the C-script block provided by PLECS, where the algorithm has been implemented. It receives as input the three-phase currents, the DC-link voltage, the electrical angle θ_e, an enable signal and the reference value for the current in the *d*-axis.

The outputs are the duty-cycle values, going as input to the *PWM* subsystem, performing the modulation, and providing as output the commutation signals for the legs of the three-phase inverter.

The sampling of the three-phase currents and the DC-link voltage has been also simulated.

The modification introduced in section 5.5 for the computation of electrical angle θ_e from the desired output frequency is also applied here. Even in this case, it is assumed for all the following tests that the desired output frequency is $f_o = 50$ Hz.

The schematic developed in PLECS environment is shown in appendix, in figure 7.

The next step has been the definition of load values, assuming that the DC-link voltage is always fixed at $V_{dc} = 400$ V, they have been chosen similar to the ones of a real synchronous machine.

HIL Simulation

The simulation developed in PLECS environment for the HIL test is shown in appendix, in figure 8; differences between the model used for simulation in PLECS and for simulation in the RT Box have been highlighted in the previous section and are still valid.

The discretization step size has been chosen to be equal to 3 μs.

5.6.1 Test I-f Control with L-R-E Load - Simulation 1

For a first simulation, a step response has been implemented and the following values have been used:

Simulation 1		
System Parameters		
Parameter	Symbol	Value
Phase inductance	L	100 µH
Phase resistance	Rs	30 mΩ
Back EMF peak value	$\hat{E}$	50 V
d-axis current reference	$I_{d,Ref}$	50 A

Simulation 1		
Control Parameters		
Parameter	Symbol	Value
PI current controller bandwidth	$\omega_{b,current}$	$2\pi\frac{f_{sw}}{15}$
PI current controller zero frequency	$f_{zero,current}$	$f_{b,current}\cdot\frac{1}{5}$
PI current controller proportional constant	K_p	$2\pi\cdot f_{b,current}\cdot L$
PI current controller integral constant	K_i	$2\pi\cdot f_{zero,current}\cdot K_{pi}$

Table 5.5: I-f control with L-R-E load test: simulation 1 parameters

Starting from the following parameters, the simulation has been performed.

The same parameters have been used in the HIL simulation carried out with the interaction between RT Box and Board.

The obtained results from the offline simulation have been compared with the HIL simulation. The comparison is shown in figure 5.14.

What can be observed from the results is that the currents obtained from the HIL simulation present a higher distortion with respect to the output currents of the offline simulation. This could be caused by the difference type of simulation (since the offline simulation is performed with a variable step solver, while the HIL simulation is necessarily performed with a fixed step solver); it is also important to evidence that delays may have been introduced by the internal hardware of the RT Box and the Board.

It is also worth mentioning that from the currents behaviour it is easy to understand how the presence of the dead-time contributes to the distortion of the currents themselves. Performing a Fourier analysis in PLECS, it has been observed that a sixth harmonic distortion is present in the q-axis current. This is caused by the fact that zero-crossing is performed six times in a period.

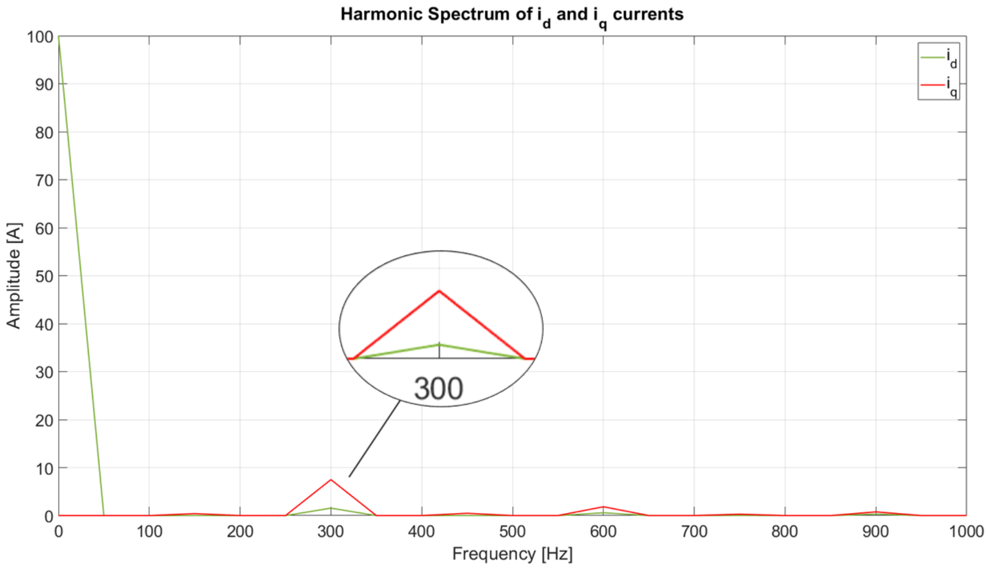
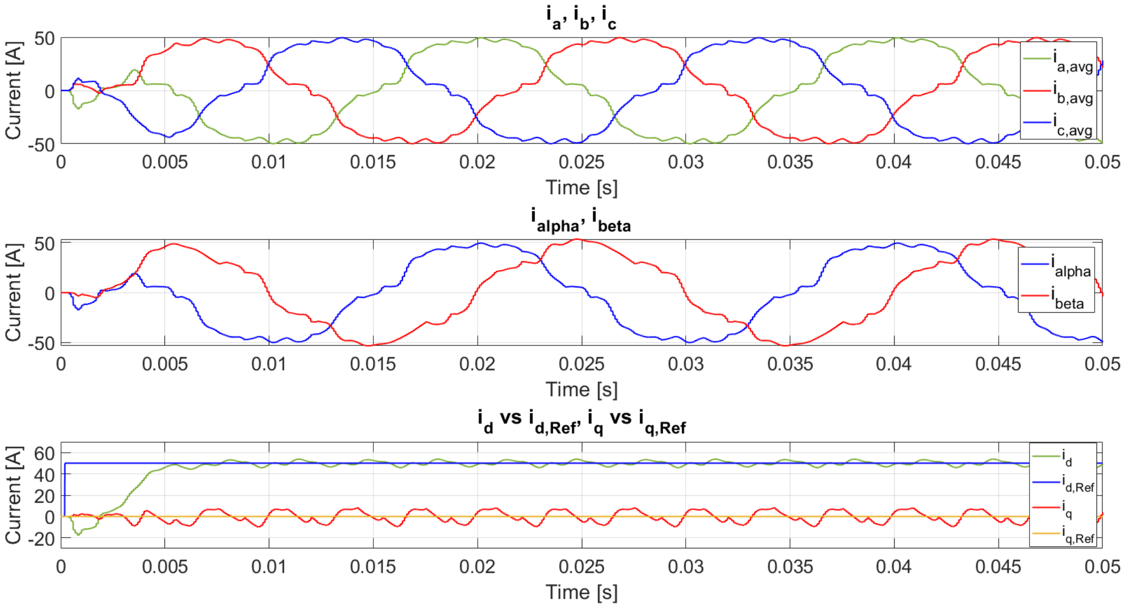
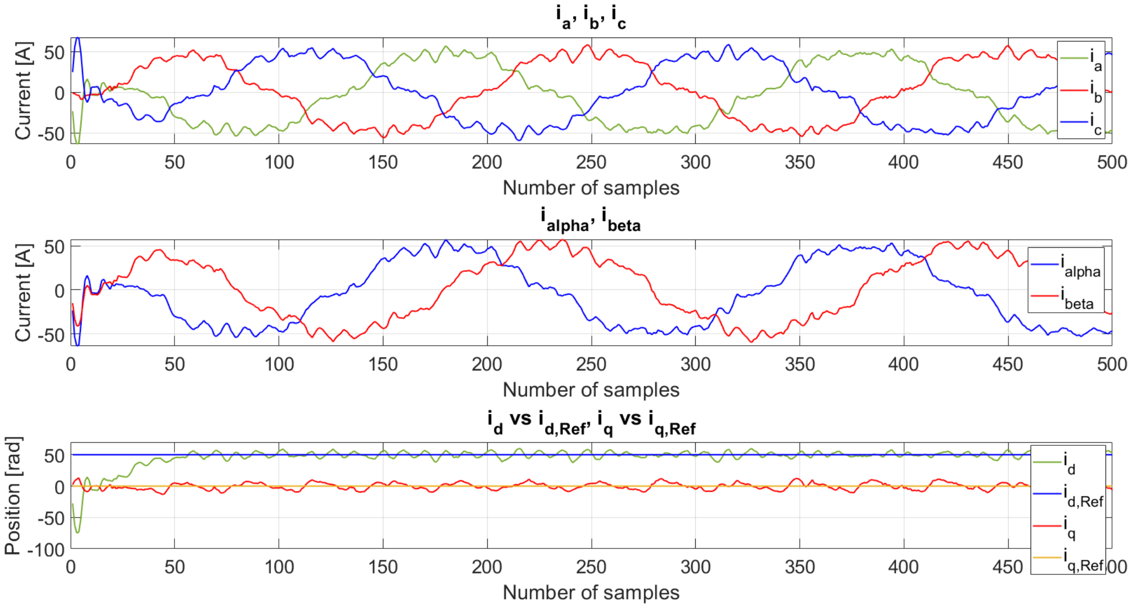


Figure 5.13: Harmonic spectrum of the output currents in the (d,q) reference frame

It is also interesting to see how the two PI current regulators behave during the HIL simulation. In figure 5.15 and in figure 5.16 it is clearly observed how the integral action tries to compensate for the presence of the E component of the L-R-E load, since it is seen by the system as an additive disturbance.



(a) Offline simulation 1: output currents



(b) HIL simulation 1: output currents

Figure 5.14: Simulation 1 results.

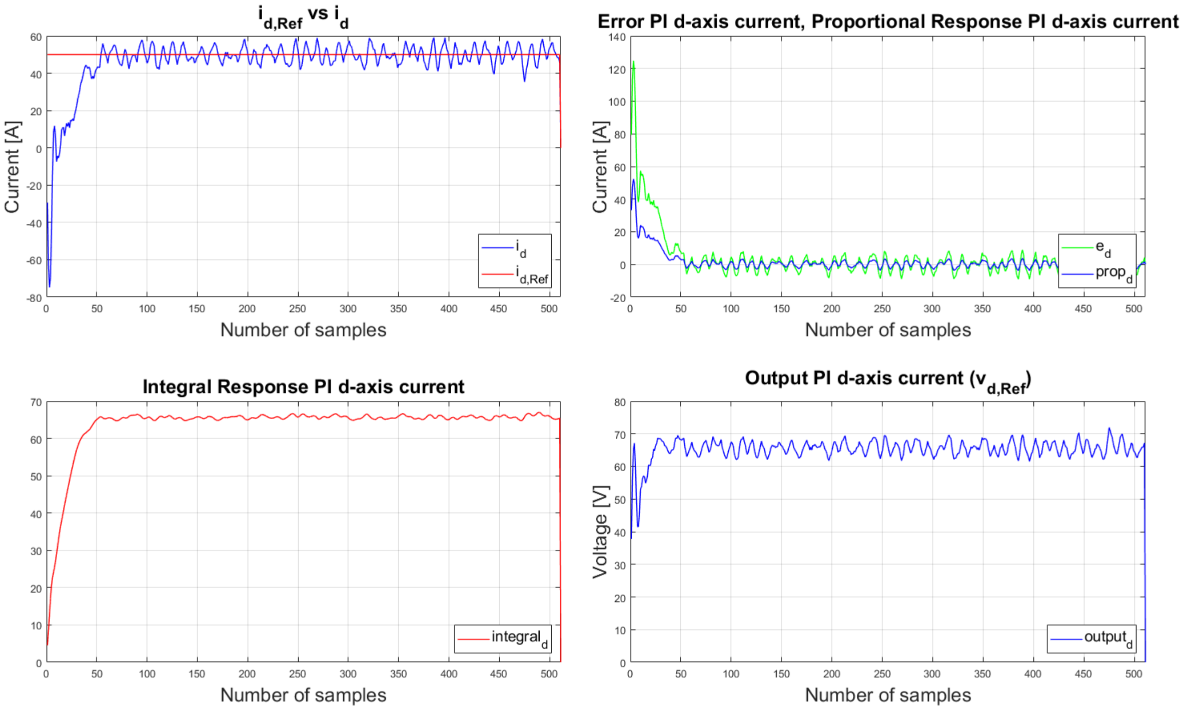


Figure 5.15: Simulation 1: behaviour of the PI d-axis current regulator

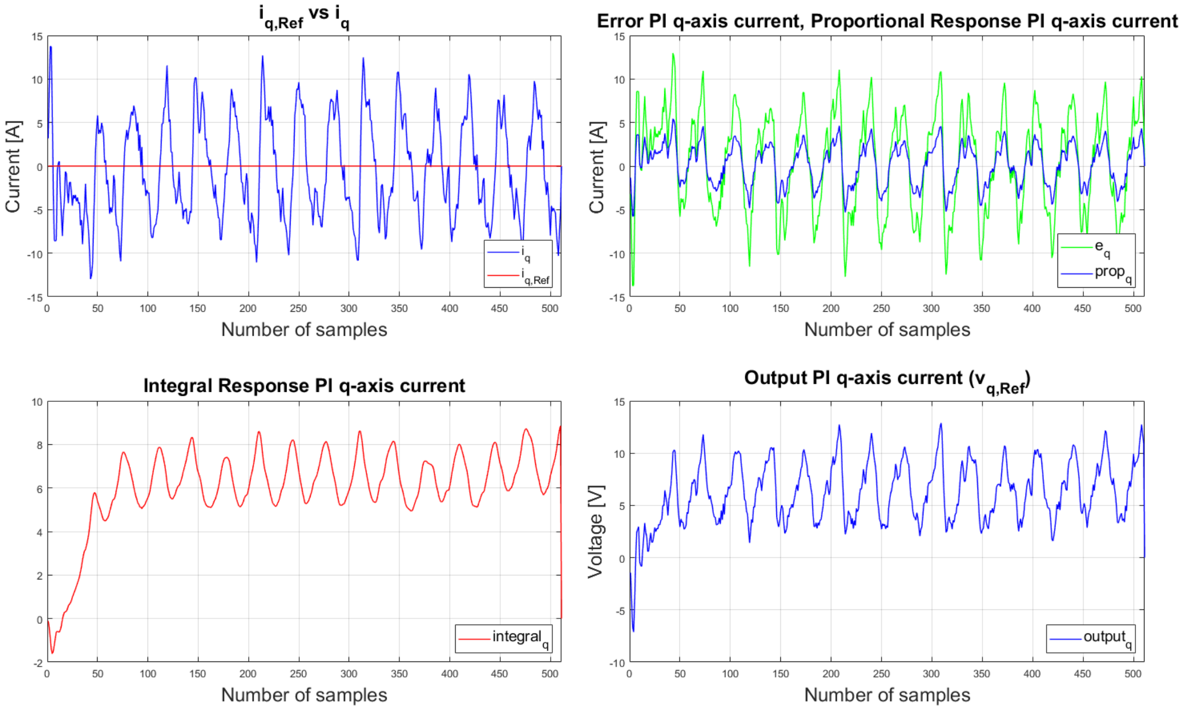


Figure 5.16: Simulation 1: behaviour of the PI q-axis current regulator

5.6.2 Test I-f Control with L-R-E Load - Simulation 2

A second test has been performed by changing the resistance value and the reference current value for the d-axis, as shown in the following table (even in this case a step response has been analysed):

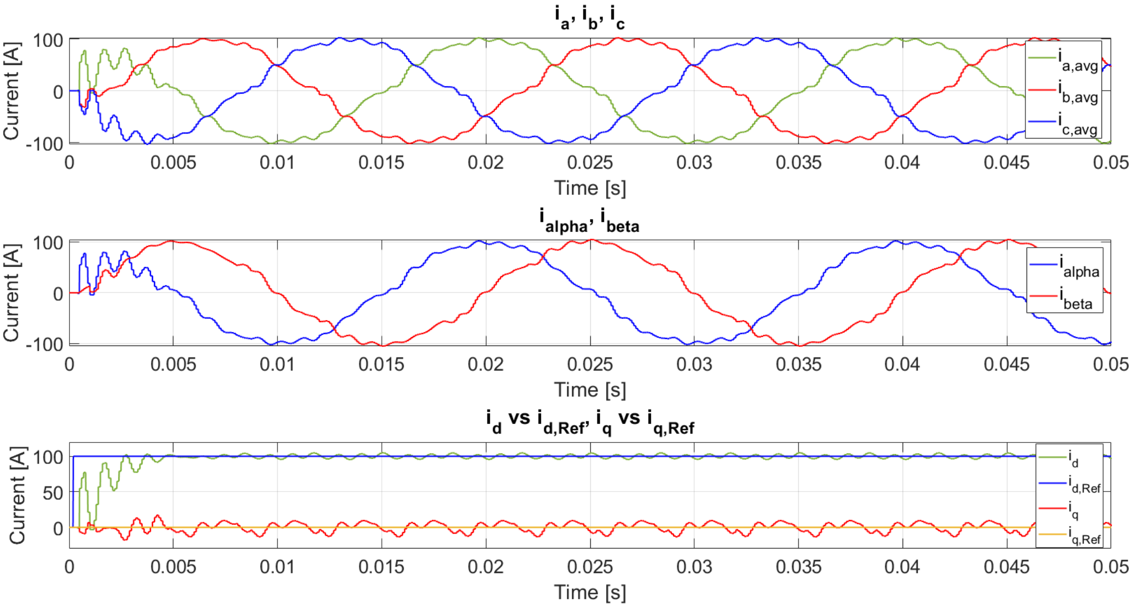
Simulation 2		
System Parameters		
Parameter	Symbol	Value
Phase inductance	L	100 µH
Phase resistance	R_s	100 mΩ
Back EMF peak value	$\hat{E}$	50 V
d-axis current reference	$I_{d,Ref}$	100 A

Simulation 2		
Control Parameters		
Parameter	Symbol	Value
PI current controller bandwidth	$\omega_{b,current}$	$2\pi\frac{f_{sw}}{12}$
PI current controller zero frequency	$f_{zero,current}$	$f_{b,current}\cdot\frac{1}{5}$
PI current controller proportional constant	K_p	$2\pi\cdot f_{b,current}\cdot L$
PI current controller integral constant	K_i	$2\pi\cdot f_{zero,current}\cdot K_{pi}$

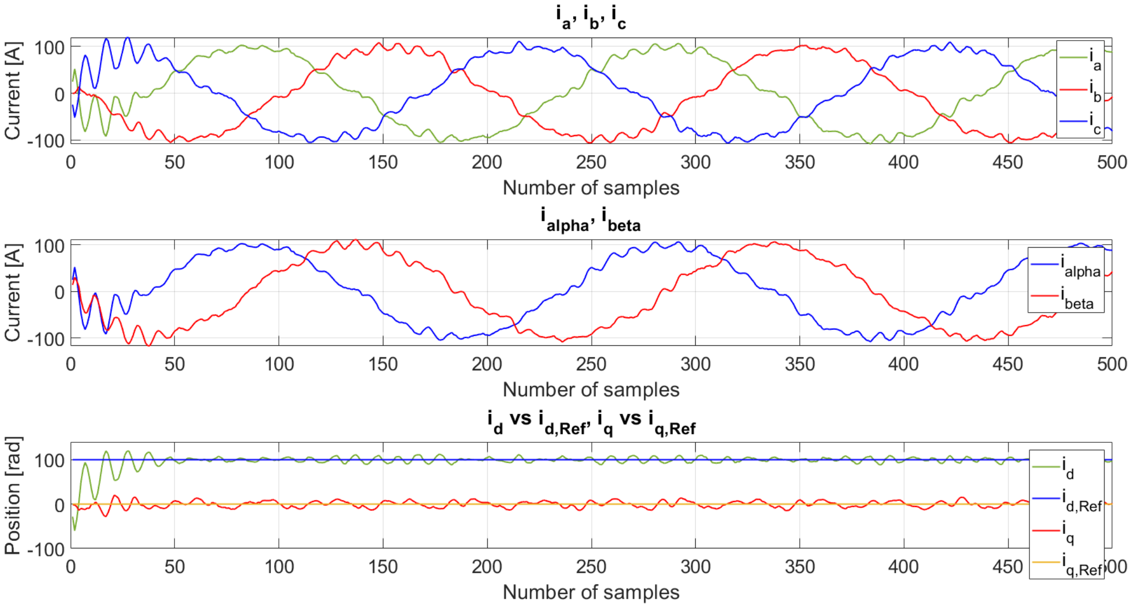
Table 5.6: I-f control test with L-R-E load: simulation 2 parameters

The aim of this simulation has been to observe the performances of the control, in a more demanding situation in terms of current reference and current controller bandwidth. Currents present a lower distortion, this is also related to the different resistance value.

On the other side, oscillations are observed at the start up: this means that, with the current reference required in this simulation, the control is stressed. The results are shown in figures 5.17, 5.18 and 5.19.



(a) Offline simulation 2: output currents



(b) HIL simulation 2: output currents

Figure 5.17: Simulation 2 results

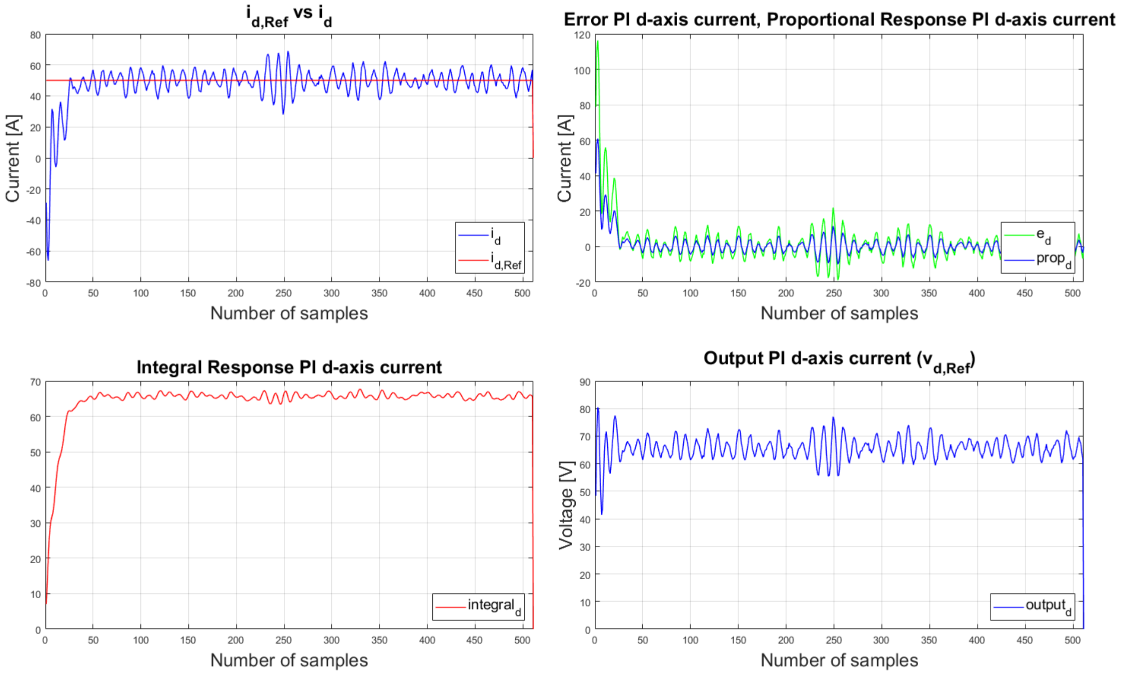


Figure 5.18: Simulation 2: behaviour of the PI d-axis current regulator

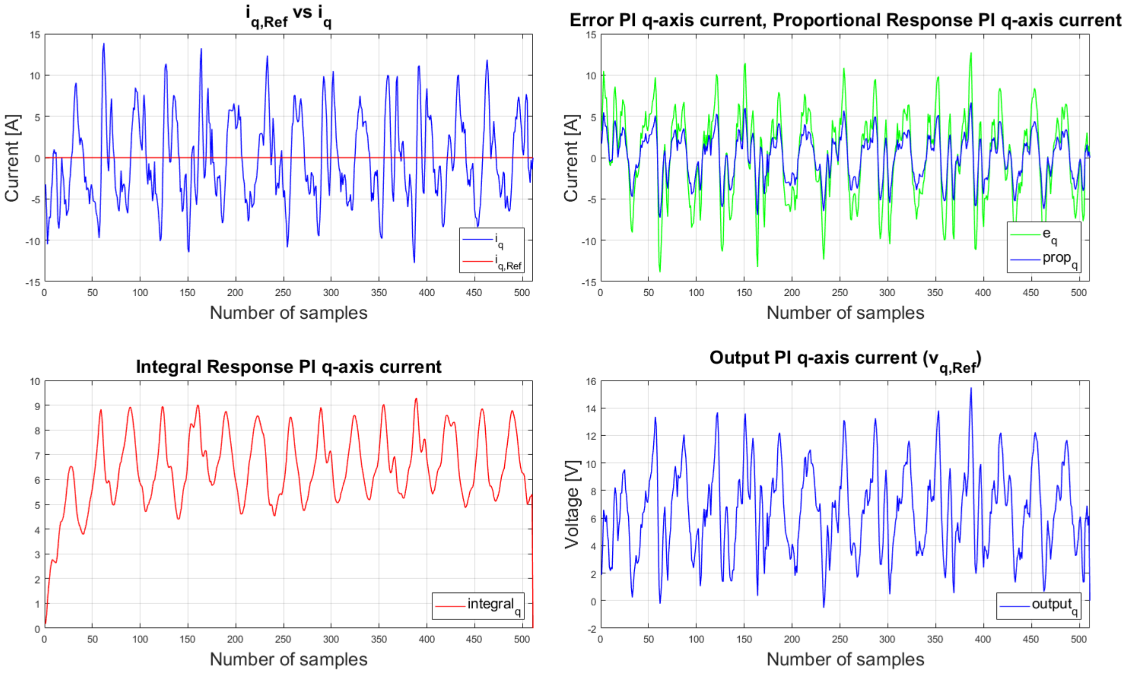


Figure 5.19: Simulation 2: behaviour of the PI q-axis current regulator

5.6.3 Test I-f Control with L-R-E Load - Simulation 3

The last performed test is aimed to analyse the system behaviour in response to a ramp input, the simulation data are shown in the following tables:

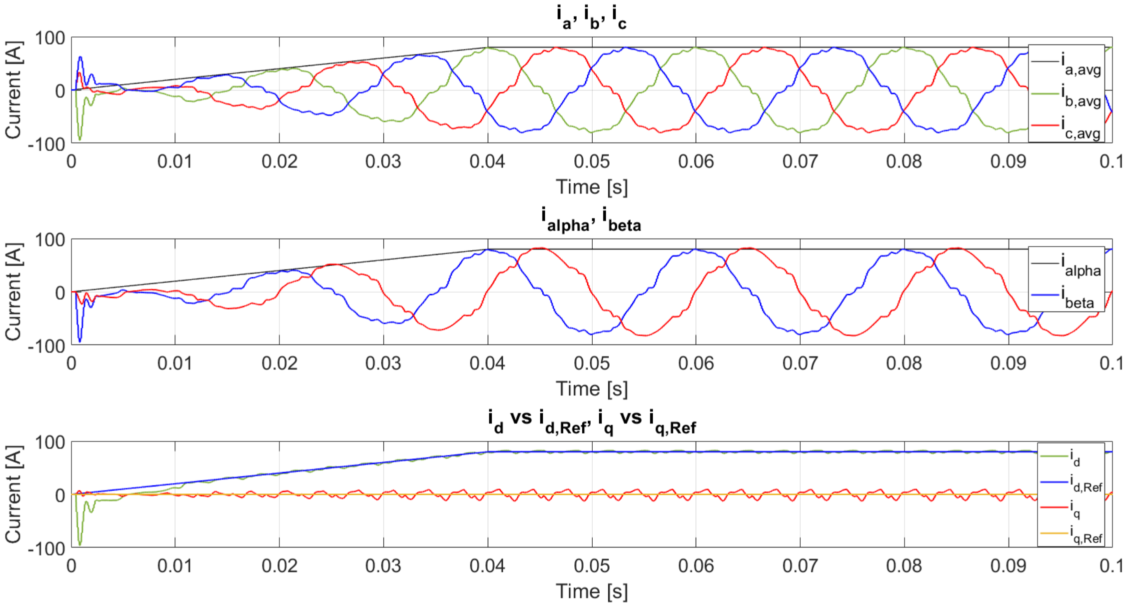
Simulation 3		
System Parameters		
Parameter	Symbol	Value
Phase inductance	L	100 µH
Phase resistance	R_s	100 mΩ
Back EMF peak value	$\hat{E}$	50 V
d-axis current reference (ramp from 0 A)	$I_{d,Ref}$	80 A
Ramp slope		2000 A/s

Simulation 3		
Control Parameters		
Parameter	Symbol	Value
PI current controller bandwidth	$\omega_{b,current}$	$2\pi\frac{f_{sw}}{15}$
PI current controller zero frequency	$f_{zero,current}$	$f_{b,current}\cdot\frac{1}{5}$
PI current controller proportional constant	K_p	$2\pi\cdot f_{b,current}\cdot L$
PI current controller integral constant	K_i	$2\pi\cdot f_{zero,current}\cdot K_{pi}$

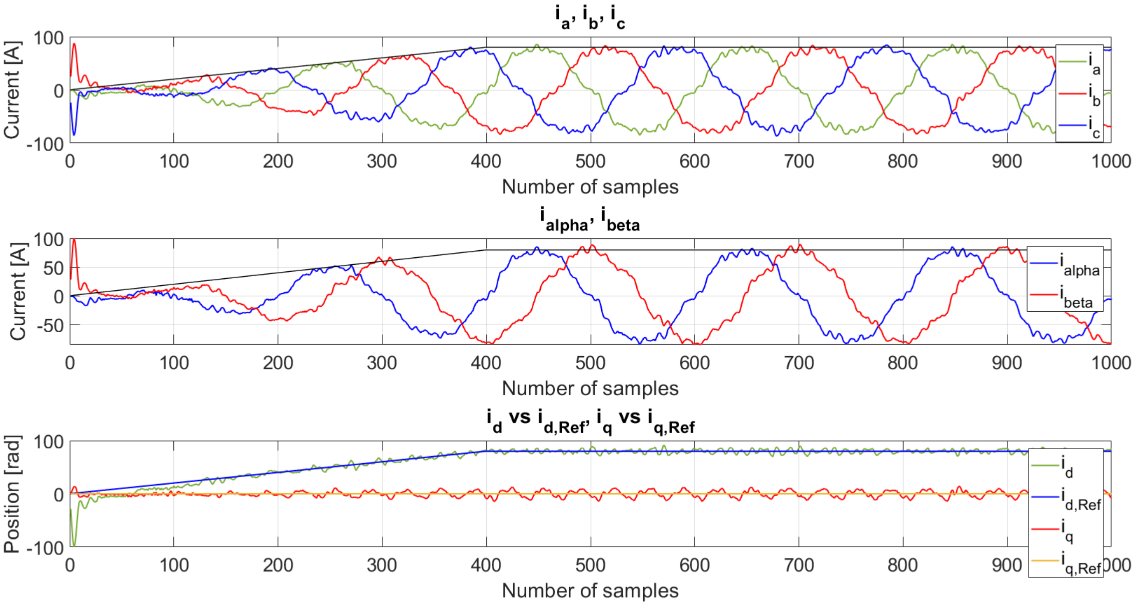
Table 5.7: I-f control test with L-R-E load: simulation 3 parameters

For this simulation, a higher number of samples has been required from the debugger to better analyse the system response. The aim of this simulation is to observe how the current behaves when the input does not changes abruptly, as in the previous step response simulations, but increases linearly in time with a constant slope (computed relatively to the switching period). This could be a typical situation for a real motor when speed increases linearly.

The results are shown in figure 5.20, 5.21 and 5.22



(a) Offline simulation 3: output currents



(b) HIL simulation 3: output currents

Figure 5.20: Simulation 3 results

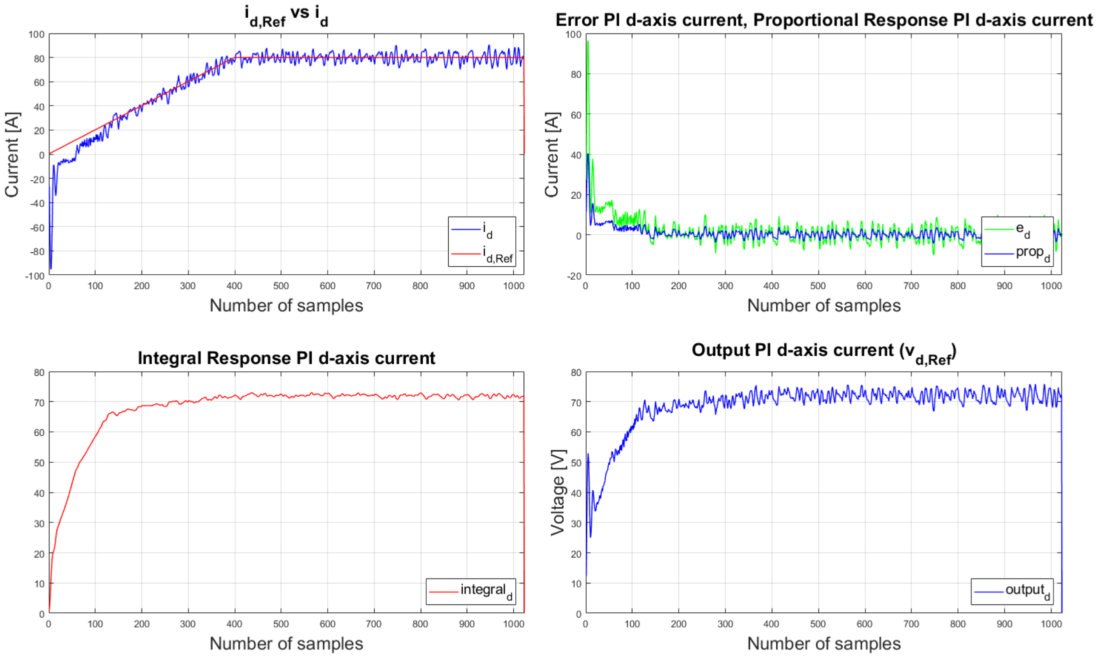


Figure 5.21: Simulation 3: behaviour of the PI d-axis current regulator

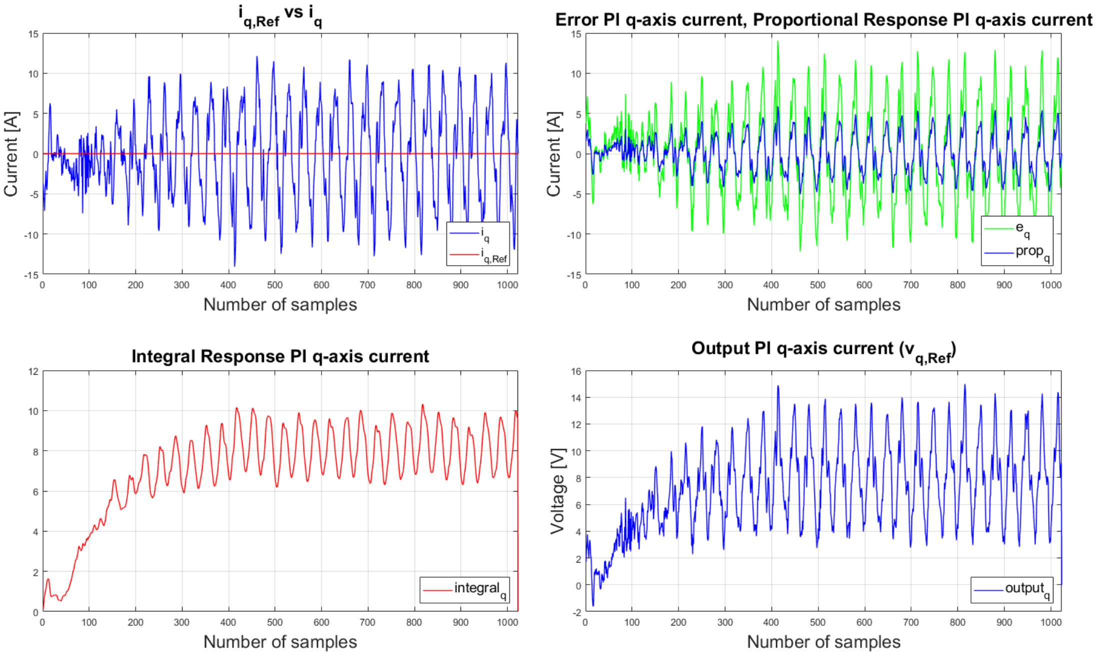


Figure 5.22: Simulation 3: behaviour of the PI q-axis current regulator

5.7 Test I-f Control with IPM Motor

Once a basic I-f control has been implemented and tested with a static load, a preliminary motor control has been realised, using a motor model, provided by PLECS, with a real motor parameters shown in the following table:

Motor Parameters		
Parameter	Symbol	Value
Phase resistance (@ 20 °C)	R_s	77 mΩ
Direct load inductance (150 A, MTPA)	L_d	1.07 mH
Quadrature load inductance (150 A, MTPA)	L_q	0.79 mH
Leakage inductance	L_σ	10 μH
Torque (@ 150 A, MTPA)	T	150 Nm
Characteristic current	I_o	150 A
Magnetic flux	λ_m	0.14 Vs
DC-link voltage	V_{dc}	400 V
Pole pairs	p	4

Table 5.8: I-f control test with IPM motor: model parameters

The adopted convention for the IPM machine is the SPMSM one, with the *d*-axis defined according to the direction of the magnets north pole N.

This has been chosen since the motor model employed, in both offline and HIL simulations, is based on a PMSM with sinusoidal back EMF, where constant parameters can be used.

The speed is imposed to the motor by a *Rotational Speed (Controlled)* block, where the rotor mechanical angle and rotor mechanical rotational speed are measured by means of the *Angle Sensor* and the *Rotational Speed Sensor* blocks, respectively. The measured quantities are thus provided to the control system.

The motor model allows to obtain, by using the *Probe* block, the following motor quantities:

- Stator phase currents i_a, i_b, i_c in A
- Stator flux linkages λ_d, λ_q (in the (d,q) reference frame) in Vs
- Rotational speed ω_m in rad/s
- Rotor position θ_m in rad
- Electromagnetic torque T_{em} of the machine in Nm

The purpose of this simulation is to verify that the previously implemented I-f control algorithm is suitable for controlling a motor.

This motor model, as mentioned, provides the rotor mechanical angle, with a range (in rad) between $-\pi$ to π. For this reason the parameters in the *Analog Output* block had to be adapted following the procedure described in section 5.3.

A preliminary simulation has shown a particular periodical behaviour of the motor quantities, presenting spikes. Further investigations, highlighted that this problem is related to the sampling of the mechanical angle.

Actually, the Board A/D converter has not been able to correctly acquire the angular position because of a low-pass filter from the hardware side. This filter presents a cut-off frequency of $f_c = 159.15$ kHz and in particular a too high capacitance of 10 nF. This does not allow the A/D converter to reach the 0 when reading a value, causing an anomalous behaviour in the presence of the discontinuity when the angle changes from π to $-\pi$. This behaviour is shown in figure 5.23. A suitable value for the low-pass filter cut-off frequency would have been in the order of few MHz.

Since this has represented a blocking problem, difficult to overcome from the hardware side, the proposed solution has been to directly obtain from the simulation

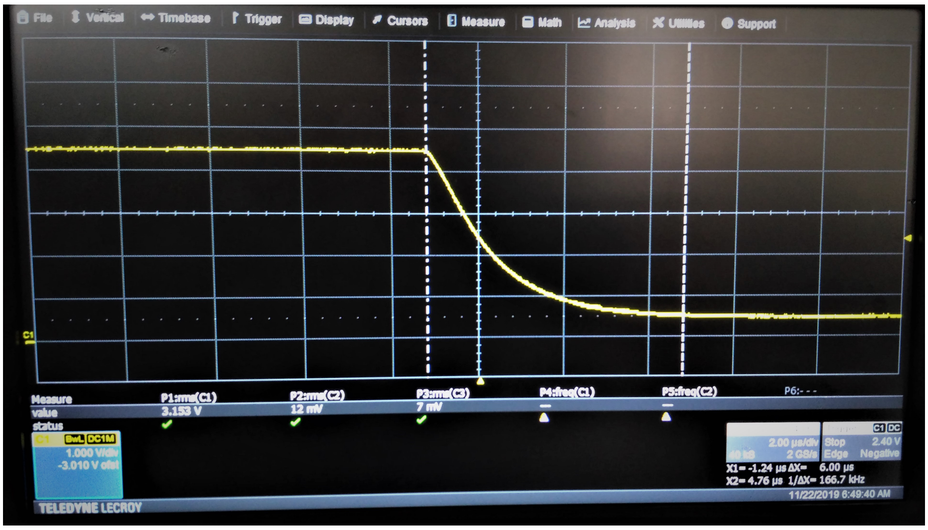


Figure 5.23: Rotor mechanical angle signal behaviour after the low-pass filter in correspondence of the discontinuity

environment (for both offline and HIL simulations) the sine and cosine of the rotor electrical angle.

This also required few modifications of the A/D converter usage: the DC-link voltage is no more sampled, but imposed as a constant in the control algorithm. In this way, the two signals corresponding to $sin\theta_e$ and $cos\theta_e$ are acquired and directly used for the rotational transformations.

Offline Simulation

The offline simulation environment is shown in appendix, in figure 9.

As can be observed, changes have been applied to the three-phase PWM modulator block to obtain a better simulation of dead-times.

Also the inverter model has been changed with an instantaneous model, where the components, and in particular the switches, are not simulated with a sub-cycle average model. This has been done to allow, for further implementations, the development of dead-time compensation techniques. The procedure for a robust implementation of dead-time compensation is described in [48].

The motor model used for offline simulation is based on the traditional *Rotor Reference Frame*, where the electrical model is defined transforming the three-phase quantities in (d,q) axis ones, resulting in constant parameters in the differential equations, to obtain an efficient model from the computational point of view.

HIL Simulation

In the HIL simulation environment, shown in appendix, in figure 10, since it is performed with a fixed discretization step, the inverter model, as well as the motor model employed in the offline simulation, cannot be used. In particular for the motor, the *Voltage-behind-reactance* model is employed, since the *Rotor Reference Frame* option presents variable inductances, capacitances and resistances, that do not make it suitable for real-time simulations with fixed step solvers.

For what concerns the inverter model, the one presented in section 5.5 has been employed.

5.7.1 Test I-f Control with IPM Motor - Simulation 1

The simulation has been performed with the parameters specified in the following table:

Simulation Parameters		
Parameter	Symbol	Value
d-axis current reference	$I_{d,Ref}$	0 A
q-axis current reference	$I_{q,Ref}$	0 A
Imposed speed (constant)	ω_m	100 rpm

Table 5.9: I-f control test with IPM motor model: simulation 1 parameters

The aim of this simulation is to verify that the vector control is synchronised with the rotor angular velocity.

Since both current references in d-axis and q-axis have been set to 0, the expected result is that, if the modulation is enabled, just small currents should be present, according to the imposed speed value. The obtained results have been compared and, apart from the difference caused by the simulation setting (variable step for the offline simulation and fixed step for the real-time simulation), as noise having the same period of the discretization step, the results are similar.

In particular the presence of this noise has been observed in correspondence of the condition in which the commutation signals are equal to zero: for the offline simulation, as expected, the output currents are null, since all the power switches are open. In the HIL simulation, the presence of a current has been observed. This is related to the adopted inverter model; further investigations have allowed to understand that the measured output current values are the same as the DC source current.

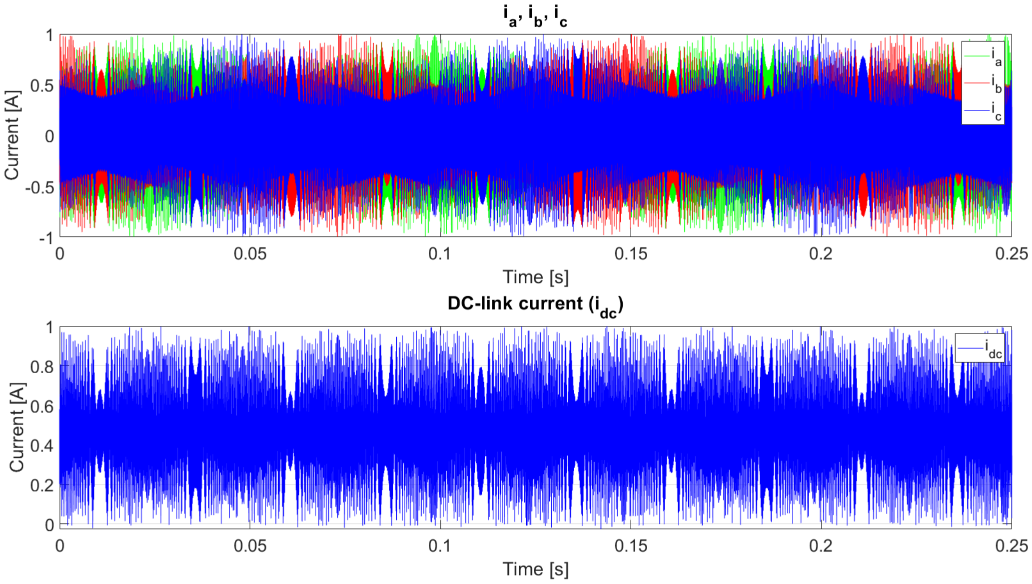


Figure 5.24: Load currents and DC-link currents when no modulation is performed

As can be observed from the figure above, the amplitude of the stator phase currents is exactly the same as the one of the DC-link current, proving that the previous hypothesis is correct.

5.7.2 Test I-f Control with IPM Motor - Simulation 2

The simulation has been performed with the parameters reported in the table 5.10, where a step response in analysed for the current in the q-axis. The imposed speed is 1000 rpm and a fast start-up has been simulated; at time t = 0.12 s the current step acts.

Simulation 2		
System Parameters		
Parameter	Symbol	Value
d-axis current reference	$I_{d,Ref}$	0 A
q-axis current reference	$I_{q,Ref}$	50 A
Imposed speed (ramp from 0 rpm)	ω_m	1000 rpm
Acceleration		10000 rpm/s

Simulation 2		
Control Parameters		
Parameter	Symbol	Value
PI d-axis current controller bandwidth	$\omega_{b,current,d}$	$2\pi\frac{f_{sw}}{20}$
PI q-axis current controller bandwidth	$\omega_{b,current,q}$	$2\pi\frac{f_{sw}}{20}$
PI d-axis current controller zero frequency	$\omega_{zero,current,d}$	$\frac{R_s}{L_d}$
PI q-axis current controller zero frequency	$\omega_{zero,current,d}$	$\frac{R_s}{L_q}$
PI d-axis current controller proportional constant	$K_{p,d}$	$\omega_{b,current,d} \cdot L_d$
PI d-axis current controller integral constant	$K_{i,d}$	$\omega_{zero,current,d} \cdot K_{p,d}$
PI q-axis current controller proportional constant	$K_{p,q}$	$\omega_{b,current,q} \cdot L_q$
PI q-axis current controller integral constant	$K_{i,q}$	$\omega_{zero,current,q} \cdot K_{p,q}$

Table 5.10: I-f control test with IPM motor model: simulation 2 parameters

In this simulation, since the current has been applied only on the q-axis, the resulting motor behaviour is similar to an SPM machine and the torque is clearly composed of the alignment torque component only.

This test is aimed to analyse the behaviour of the current in the q-axis when an abrupt change occurs and, as a consequence, how the motor electromagnetic torque react to this stimulus.

In this simulation the feedforward compensation on the PI regulators has been applied, as described in section 2.7.2.

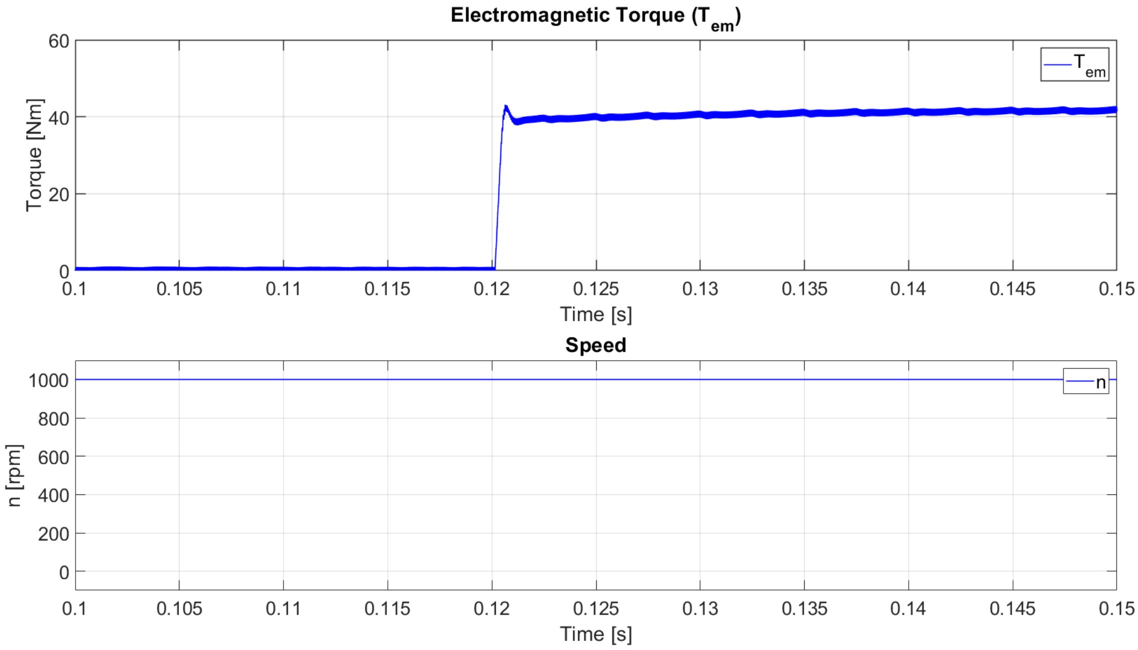
To do it, the computation of the speed from the rotor angle is necessary, thus the following formulation has been used:

$$\omega_e = [sin\theta_e(k) \cdot cos\theta_e(k-1) - cos\theta_e(k) \cdot sin\theta_e(k-1)] \cdot f_{sw} \tag{5.4}$$

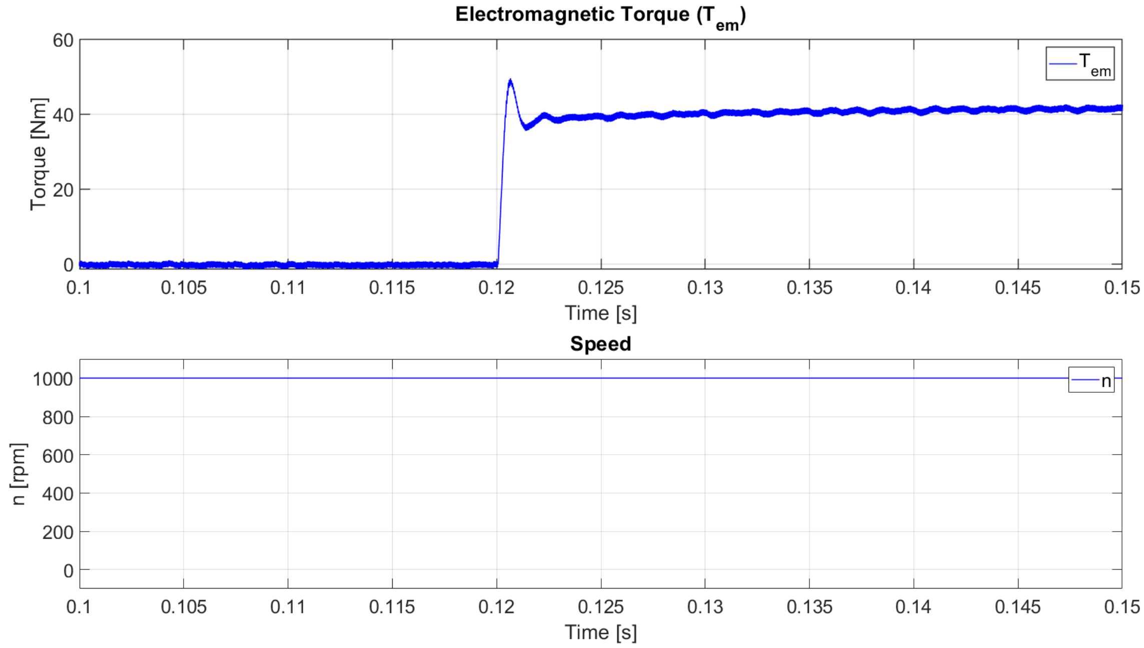
And the corresponding code,

```
SinCosRotPrev.sin = SinCosRot.sin;
SinCosRotPrev.cos = SinCosRot.cos;

SinCosRot.sin = sinTheta_e;
SinCosRot.cos = cosTheta_e;

//  Electrical rotational speed computation
w_e = ( SinCosRot.sin * SinCosRotPrev.cos - SinCosRot.cos *
    SinCosRotPrev.sin ) * fs;
```

The results from both offline and HIL simulations are shown in figures 5.25 and 5.26, respectively. The simulation time window of 0.25 s has been restricted to better observe the transient response of the system.

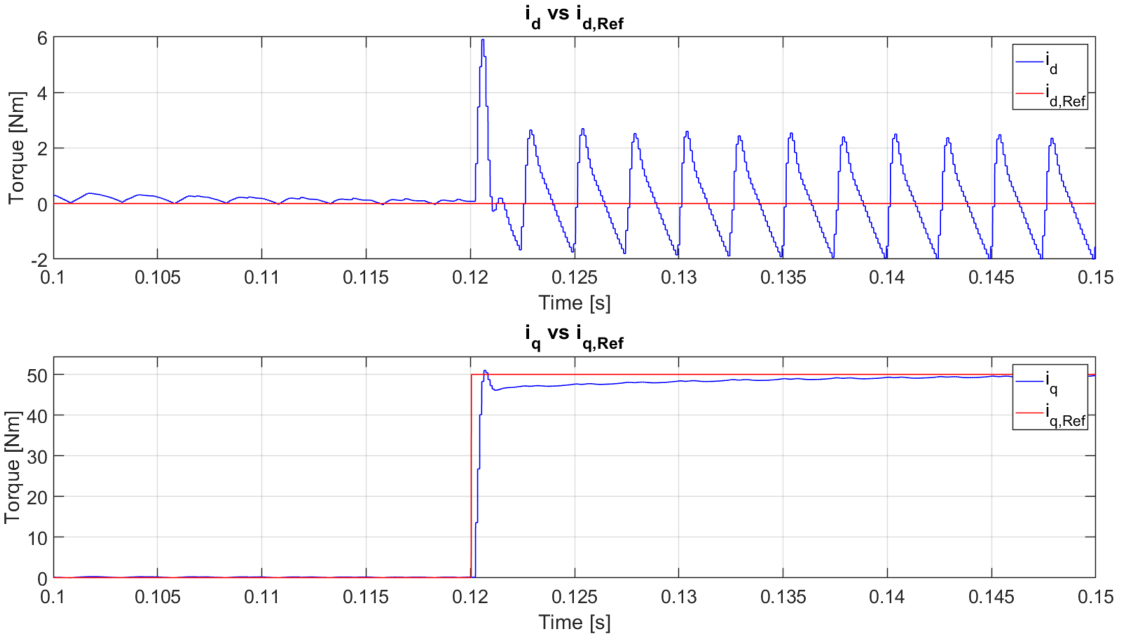


(a) Offline simulation 2: electromagnetic torque and speed

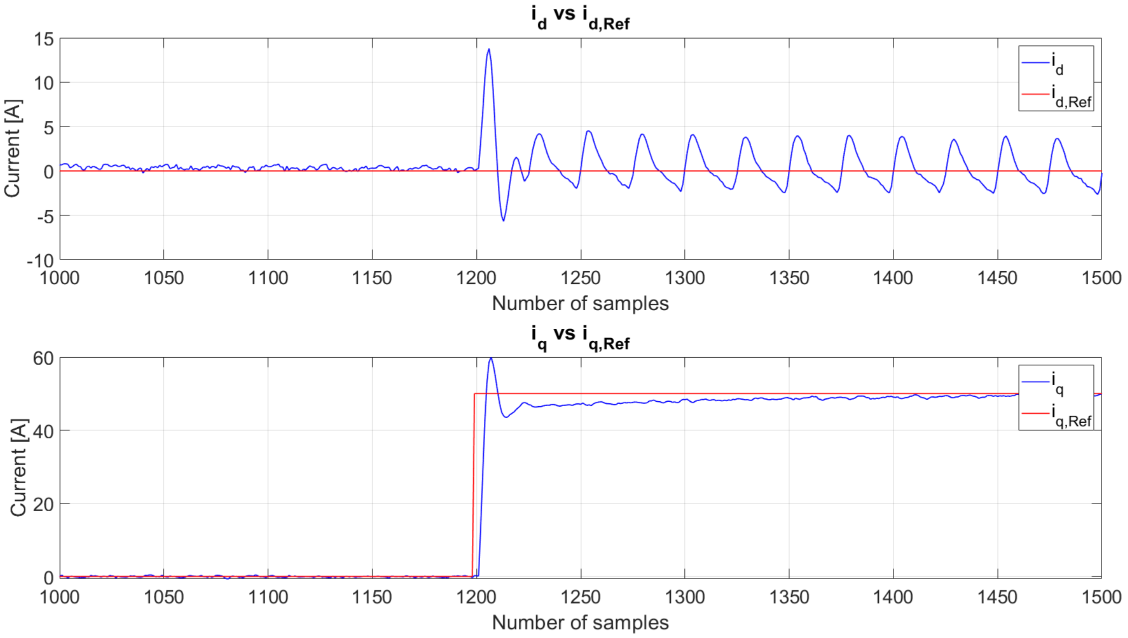


(b) HIL simulation 2: electromagnetic torque and speed

Figure 5.25: Simulation 2 results



(a) Offline simulation 2: (d,q) reference frame output currents



(b) HIL simulation 2: (d,q) reference frame output currents

Figure 5.26: Simulation 2 results

5.7.3 Test I-f Control with IPM Motor - Simulation 3

Considering the control technique applied in this simulation, two possible further implementations are possible.

The reference values for both d-axis and q-axis currents could be imposed starting from a torque reference and from the relation between them.

From this, an MTPA locus can be defined, laying the groundwork for the Field-Oriented Control technique.

The MTPA is a curve of the currents or flux values which minimises the stator current amplitude, given a reference torque and assuming the machine is working in the constant torque region. Since the current amplitude is minimised, also stator copper losses are reduced, therefore the MTPA region can be considered, at low speed, close to the maximum efficiency condition.

The alternative could be to implement a flux observer and an MTPA which provides both q-axis current and flux amplitude references. This approach lead to the Direct-Flux Vector Control technique.

In this simulation, the first solution has been first implemented, where the MTPA has been obtained from a MATLAB script, starting from the machine and inverter parameters, assuming magnetic linearity and constant values for motor inductances and resistance.

The obtained maps have been included in the control system in the form of Look-Up Tables, and read by a proper function in the control routine, given the torque reference.

Even in this case, a fast start-up situation is simulated, with an imposed speed of 1000 rpm with an acceleration of 10000 rpm/s. The adopted simulation parameters are shown in table 5.11.

Simulation 3		
System Parameters		
Parameter	Symbol	Value
Reference Torque	T_{Ref}	100 Nm
Imposed speed (ramp from 0 rpm)	ω_m	1000 rpm
Acceleration		10000 rpm/s

Simulation 3		
Control Parameters		
Parameter	Symbol	Value
PI *d*-axis current controller bandwidth	$\omega_{b,current,d}$	$2\pi\frac{fsw}{15}$
PI *q*-axis current controller bandwidth	$\omega_{b,current,q}$	$2\pi\frac{fsw}{15}$
PI *d*-axis current controller zero frequency	$\omega_{zero,current,d}$	$\frac{R_s}{L_d}$
PI *q*-axis current controller zero frequency	$\omega_{zero,current,d}$	$\frac{R_s}{L_q}$
PI *d*-axis current controller proportional constant	$K_{p,d}$	$\omega_{b,current,d} \cdot L_d$
PI *d*-axis current controller integral constant	$K_{i,d}$	$\omega_{zero,current,d} \cdot K_{p,d}$
PI *q*-axis current controller proportional constant	$K_{p,q}$	$\omega_{b,current,q} \cdot L_q$
PI *q*-axis current controller integral constant	$K_{i,q}$	$\omega_{zero,current,q} \cdot K_{p,q}$

Table 5.11: I-f control test with IPM motor model: simulation 3 parameters

The obtained MTPA loci are shown in figures 5.27 and 5.28.

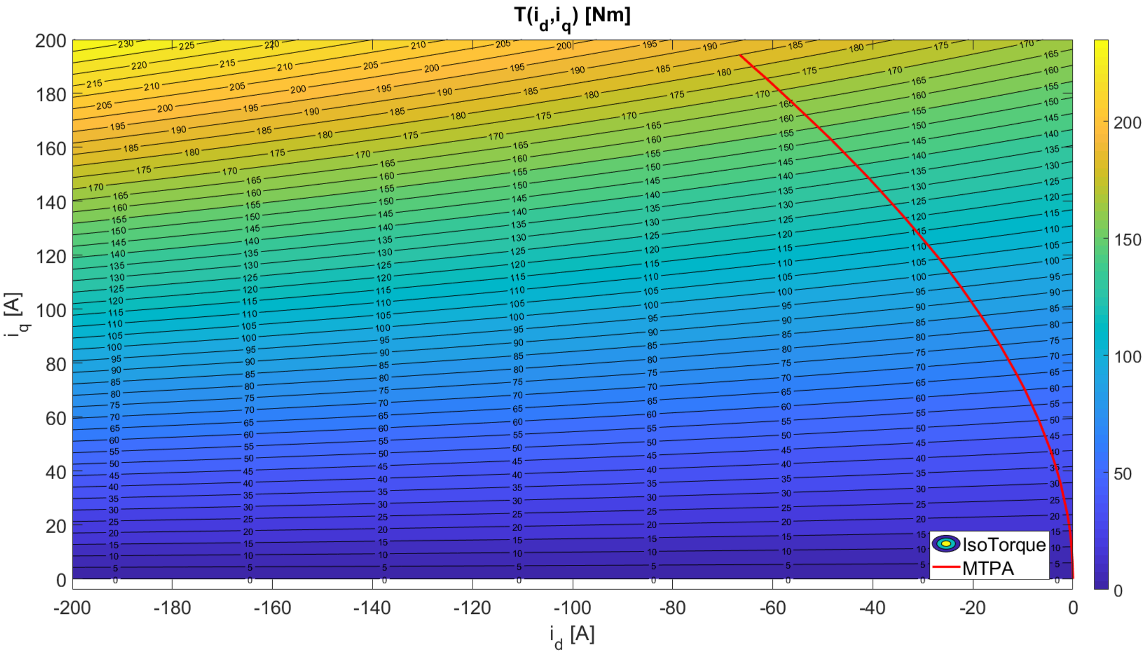


Figure 5.27: MTPA trajectory in the (i_d, i_q) plane

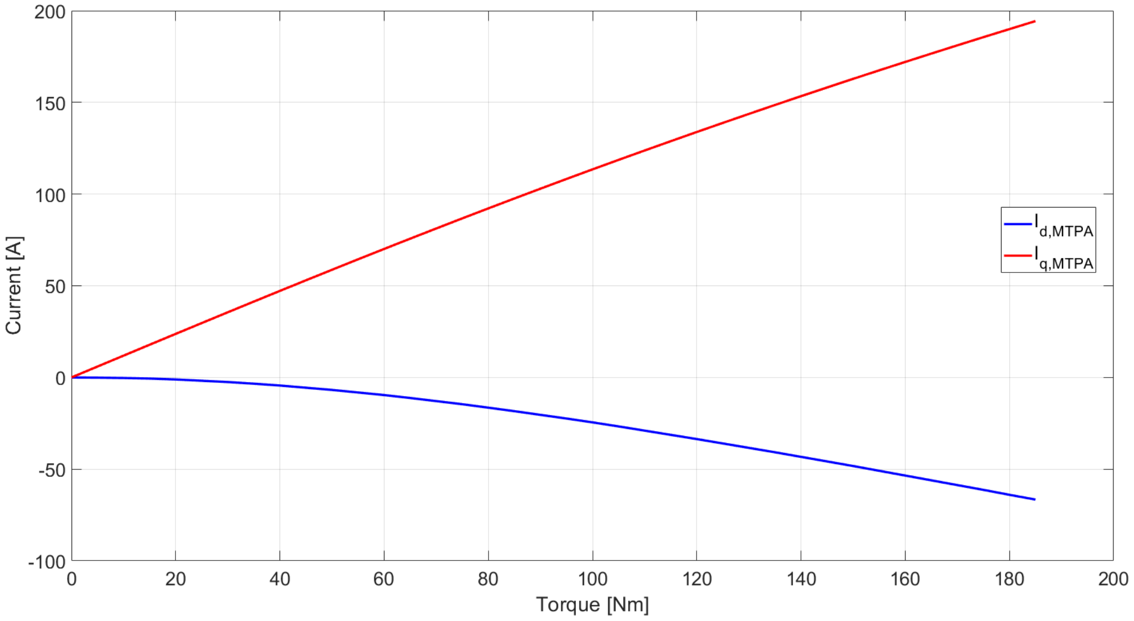
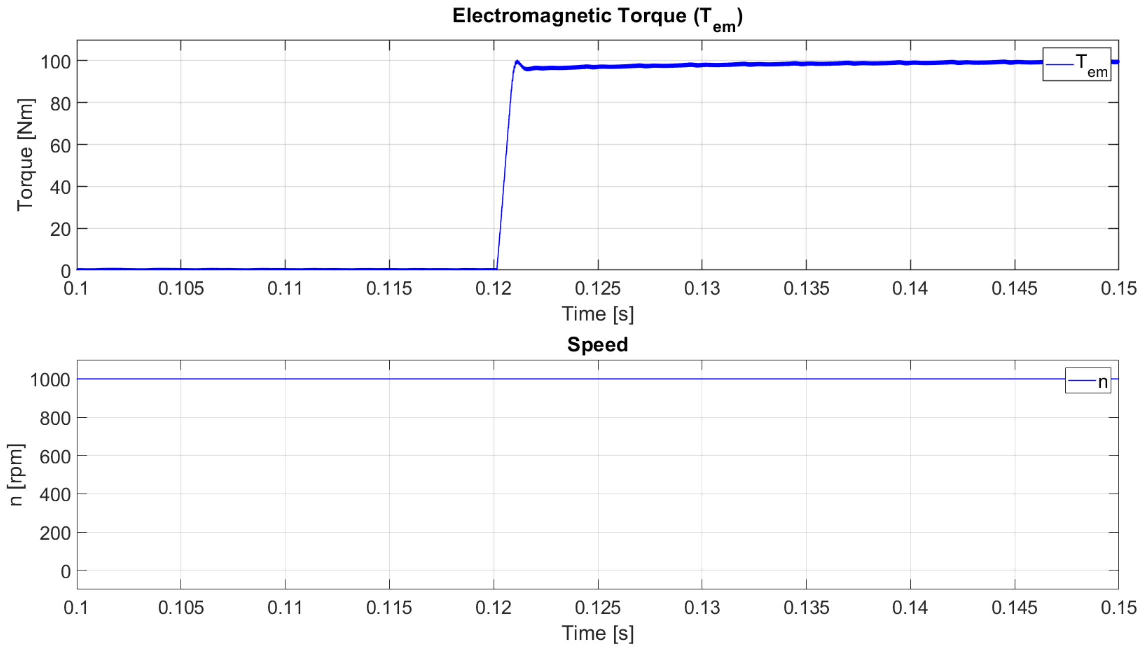
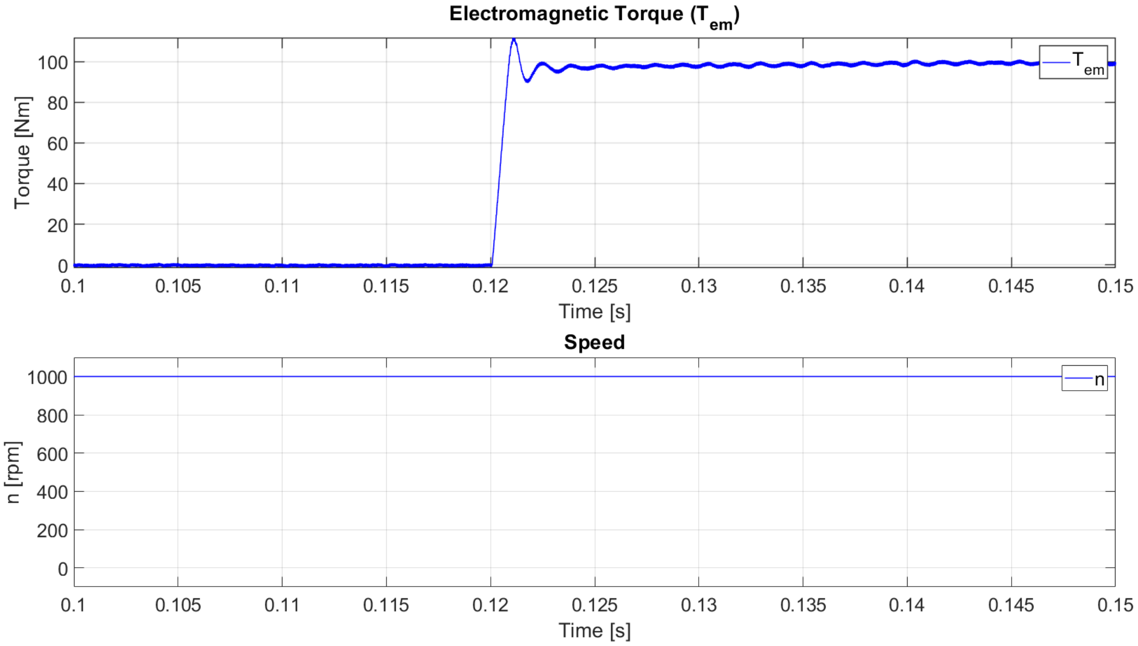


Figure 5.28: d-axis and q-axis current MTPA values

The simulation results are presented in figures 5.29 and 5.30, proving the MTPA provides the current references values correctly.

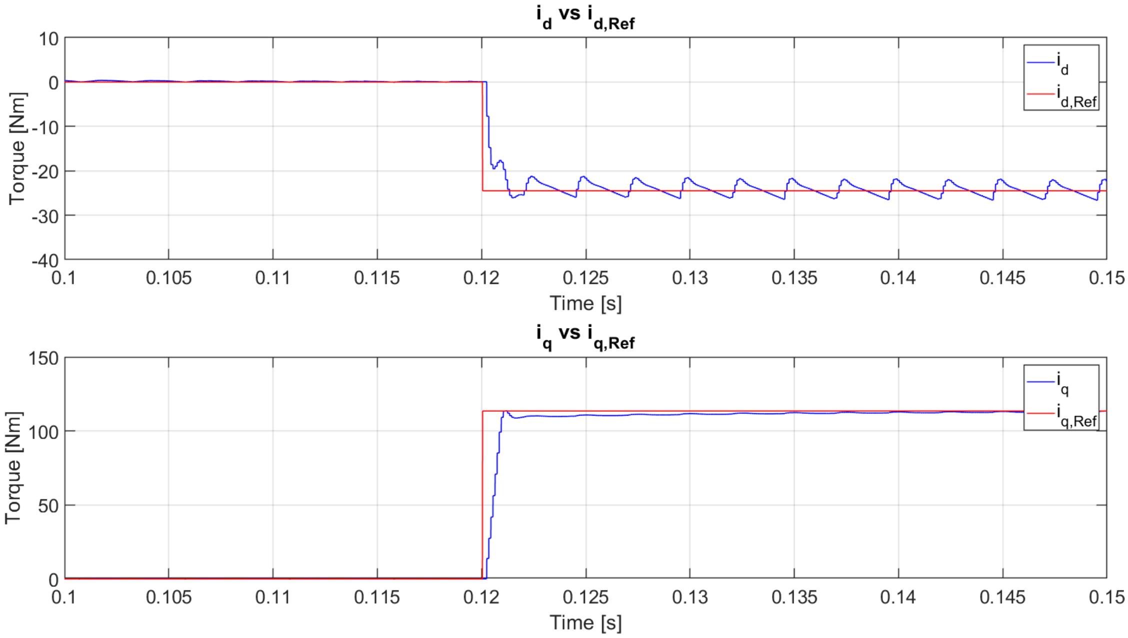


(a) Offline simulation 3: electromagnetic torque and speed

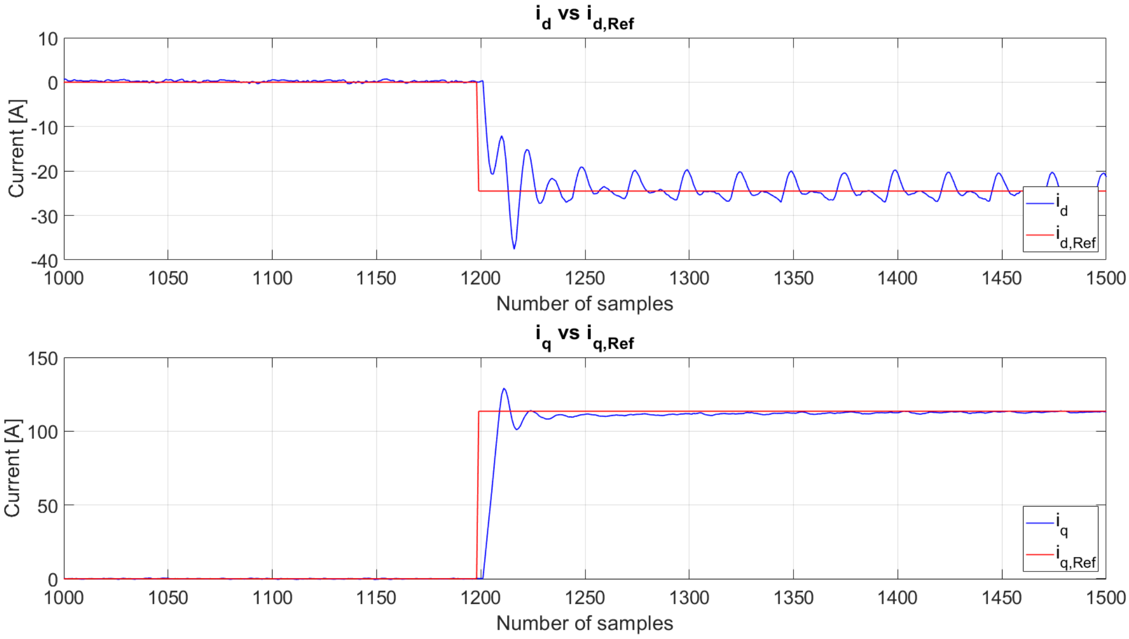


(b) HIL simulation 3: electromagnetic torque and speed

Figure 5.29: Simulation 3 results



(a) Offline simulation 3: (d,q) reference frame output currents



(b) HIL simulation 3: (d,q) reference frame output currents

Figure 5.30: Simulation 3 results

Chapter 6

Conclusions

During this thesis work, a review of the state-of-the-art for what concerns motor control techniques and the hardware and software solutions for its implementation in traction applications, has been proposed.

Supporting the theoretical aspects, different offline simulations have been first performed to evaluate the control code to be later implemented in the target hardware.

Before this, the employed Board peripherals have been tested using ad hoc developed tests with the real-time machine RT Box.

Later, a step-by-step procedure has been followed starting from a simple open-loop control to a closed-loop current control with a static load and then a closed-loop current control with an IPM motor model based on a real motor data. It has been also improved deriving MTPA trajectories for both stator currents and stator flux to be used for further control techniques implementations, as FOC and UDFVC.

All the previously mentioned steps have been validated with Hardware-in-the-Loop tests which allowed to verify the suitability of the Device Under Test for motor control applications.

The algorithm implemented in this thesis, before being experimentally tested with a real motor test bench, should be subjected to many enhancements; starting from proper dead-time compensation techniques, to the development of a robust control strategy as the Direct-Flux Vector Control, supporting also the operation in MPTV region and resulting to be more performing with respect to the Field-Oriented Control according to the reviewed literature.

Considering the initial thesis objectives, the implementation of the motor control algorithm has required more time than the expected and it has not been possible to provide an equivalent implementation based on the auto-generated code using the developed Board Support Package, since many issues from the Board software sides have been encountered.

This would have required improvement of the previously mentioned BSP and re-implementation of some blocks according to the new required specifications.

In view of the above, it has not been possible to compare the performances of the two different approaches and that the employed Board presents many limitations from both hardware and software sides, but with focused improvements aimed at exploiting Infineon AURIX MCU maximum computation capabilities together with the performances of the RTOS, it can be used for high performances control.

In conclusion, my personal contribution brought with thesis work has been to prove that the employed Board, basically do not developed for motor control applications, but mainly for functional safety and ADAS ones, can be also employed in electrical traction in the automotive field, according to the previously mentioned improvements.

Appendices

Appendix A

The test cases performed for the single inverter leg with RL load, addressed in section 5.4, are here presented in details, starting from the default parameters defined in table 5.2.

The following table shows the obtained results for the three inductor currents. The first column indicates which parameter has been changed with respect to the default ones.

		i_a	
Test Conditions	Test Case	Average Value (Expected)	Average Value (Measured)
Default	1	47	48
duty-cycle $= 60\%$	2	57	59
duty-cycle $= 70\%$	3	67	69
duty-cycle $= 80\%$	4	77	78
duty-cycle $= 90\%$	5	87	89
duty-cycle $= 99\%$	6	97.05	97
duty-cycle $= 10\%$	7	7	7
duty-cycle $= 5\%$	8	2	1
$L = 100$ µH	9	47	50
$V_{in} = 80$ V	10	37.7	38
$L = 100$ µH, *duty-cycle* $= 60\%$	11	57	61
$L = 100$ µH, *duty-cycle* $= 70\%$	12	67	71
$L = 100$ µH, *duty-cycle* $= 80\%$	13	77	82
$L = 100$ µH, *duty-cycle* $= 90\%$	14	87	90
$L = 100$ µH, *duty-cycle* $= 99\%$	15	97	97
$L = 100$ µH, *duty-cycle* $= 10\%$	16	7	7
$L = 100$ µH, *duty-cycle* $= 5\%$	17	2	1
$L = 100$ µH, $V_{in} = 80$ V	18	38	40
$L = 100$ µH, $V_{in} = 80$ V	19	23.5	25
$L = 100$ µH, $V_{in} = 80$ V, *duty-cycle* $= 90\%$	20	43.5	45

Table 1: Expected and measured current i_a values with test conditions

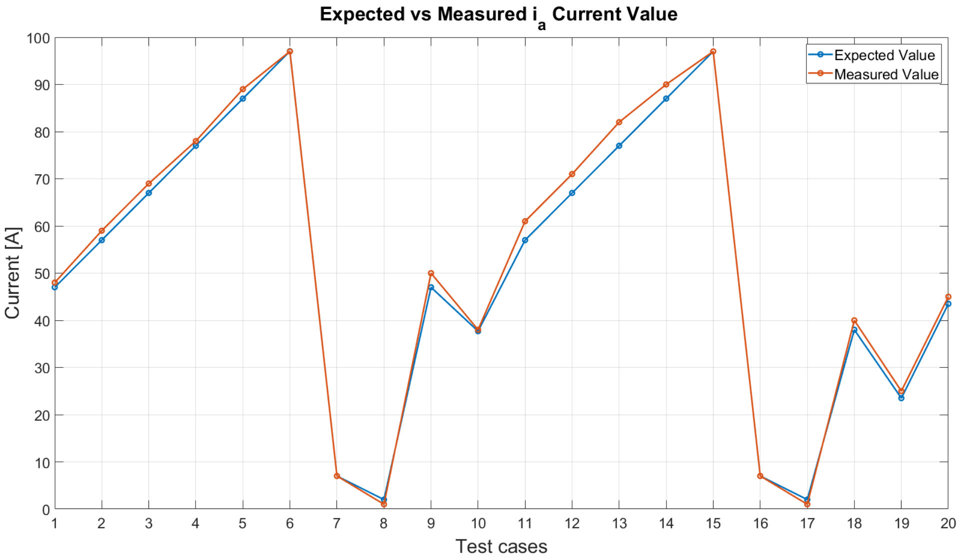


Figure 1: Burn down chart showing the expected and the measured i_a current

		i_b	
Test Conditions	Test Case	Average Value (Expected)	Average Value (Measured)
Default	1	47	48
duty-cycle = 60%	2	57	58
duty-cycle = 70%	3	67	69
duty-cycle = 80%	4	77	79
duty-cycle = 90%	5	87	89
duty-cycle = 99%	6	97.05	97
duty-cycle = 10%	7	7	6
duty-cycle = 5%	8	2	1
$L = 100$ µH	9	47	50
$V_{in} = 80$ V	10	37.7	39
$L = 100$ µH, *duty-cycle* = 60%	11	57	59
$L = 100$ µH, *duty-cycle* = 70%	12	67	71
$L = 100$ µH, *duty-cycle* = 80%	13	77	83
$L = 100$ µH, *duty-cycle* = 90%	14	87	91
$L = 100$ µH, *duty-cycle* = 99%	15	97	97
$L = 100$ µH, *duty-cycle* = 10%	16	7	7
$L = 100$ µH, *duty-cycle* = 5%	17	2	1
$L = 100$ µH, $V_{in} = 80$ V	18	38	39
$L = 100$ µH, $V_{in} = 80$ V	19	23.5	24
$L = 100$ µH, $V_{in} = 80$ V, *duty-cycle* = 90%	20	43.5	45

Table 2: Expected and measured current i_b values with test conditions

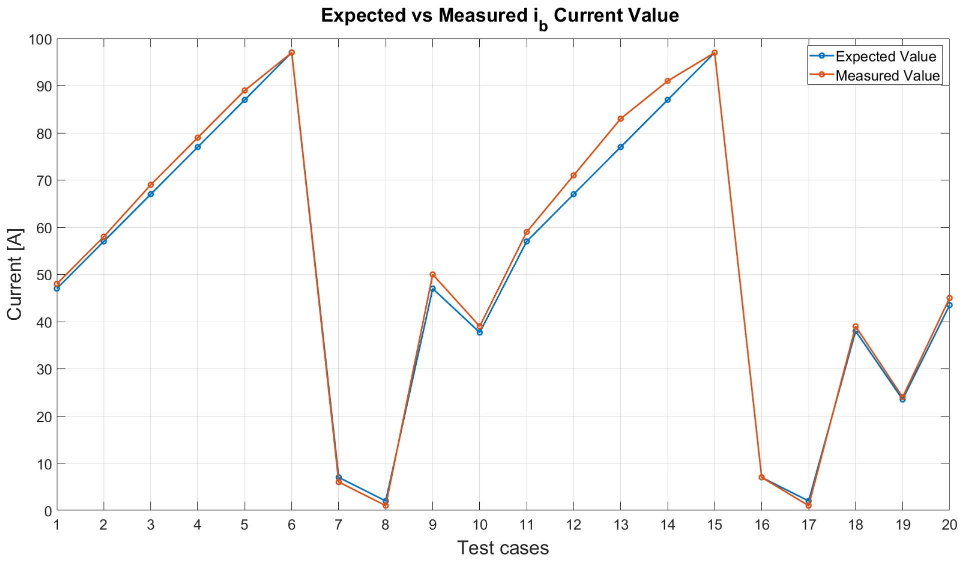


Figure 2: Burn down chart showing the expected and the measured i_b current

		i_c	
Test Conditions	Test Case	Average Value (Expected)	Average Value (Measured)
Default	1	47	47
duty-cycle $= 60\%$	2	57	58
duty-cycle $= 70\%$	3	67	69
duty-cycle $= 80\%$	4	77	79
duty-cycle $= 90\%$	5	87	89
duty-cycle $= 99\%$	6	97.05	97
duty-cycle $= 10\%$	7	7	6
duty-cycle $= 5\%$	8	2	1
$L = 100$ µH	9	47	49
$V_{in} = 80$ V	10	37.7	39
$L = 100$ µH, *duty-cycle* $= 60\%$	11	57	61
$L = 100$ µH, *duty-cycle* $= 70\%$	12	67	70
$L = 100$ µH, *duty-cycle* $= 80\%$	13	77	82
$L = 100$ µH, *duty-cycle* $= 90\%$	14	87	91
$L = 100$ µH, *duty-cycle* $= 99\%$	15	97	97
$L = 100$ µH, *duty-cycle* $= 10\%$	16	7	7
$L = 100$ µH, *duty-cycle* $= 5\%$	17	2	1
$L = 100$ µH, $V_{in} = 80$ V	18	38	40
$L = 100$ µH, $V_{in} = 80$ V	19	23.5	24
$L = 100$ µH, $V_{in} = 80$ V, *duty-cycle* $= 90\%$	20	43.5	44

Table 3: Expected and measured current i_c values with test conditions

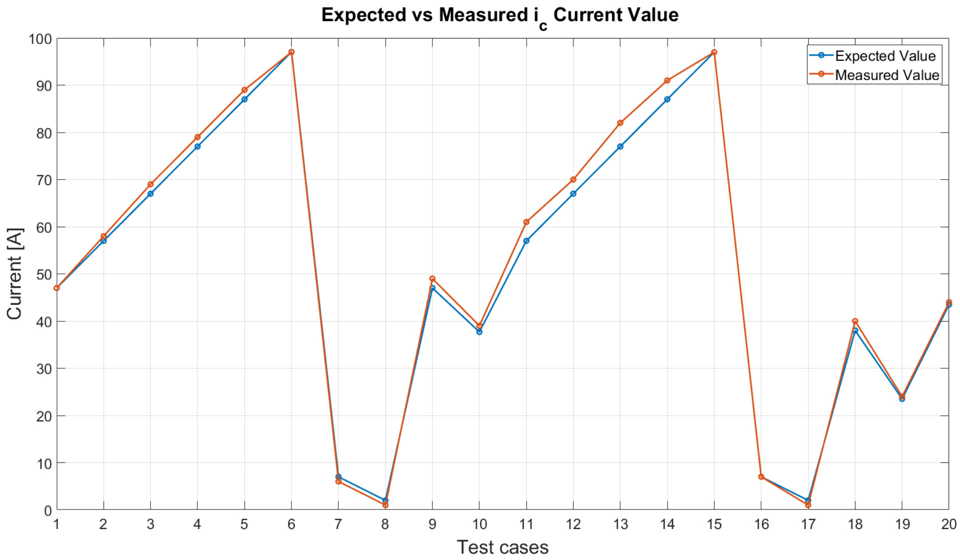


Figure 3: Burn down chart showing the expected and the measured i_c current

Appendix B

Test Single Inverter Leg with RL Load

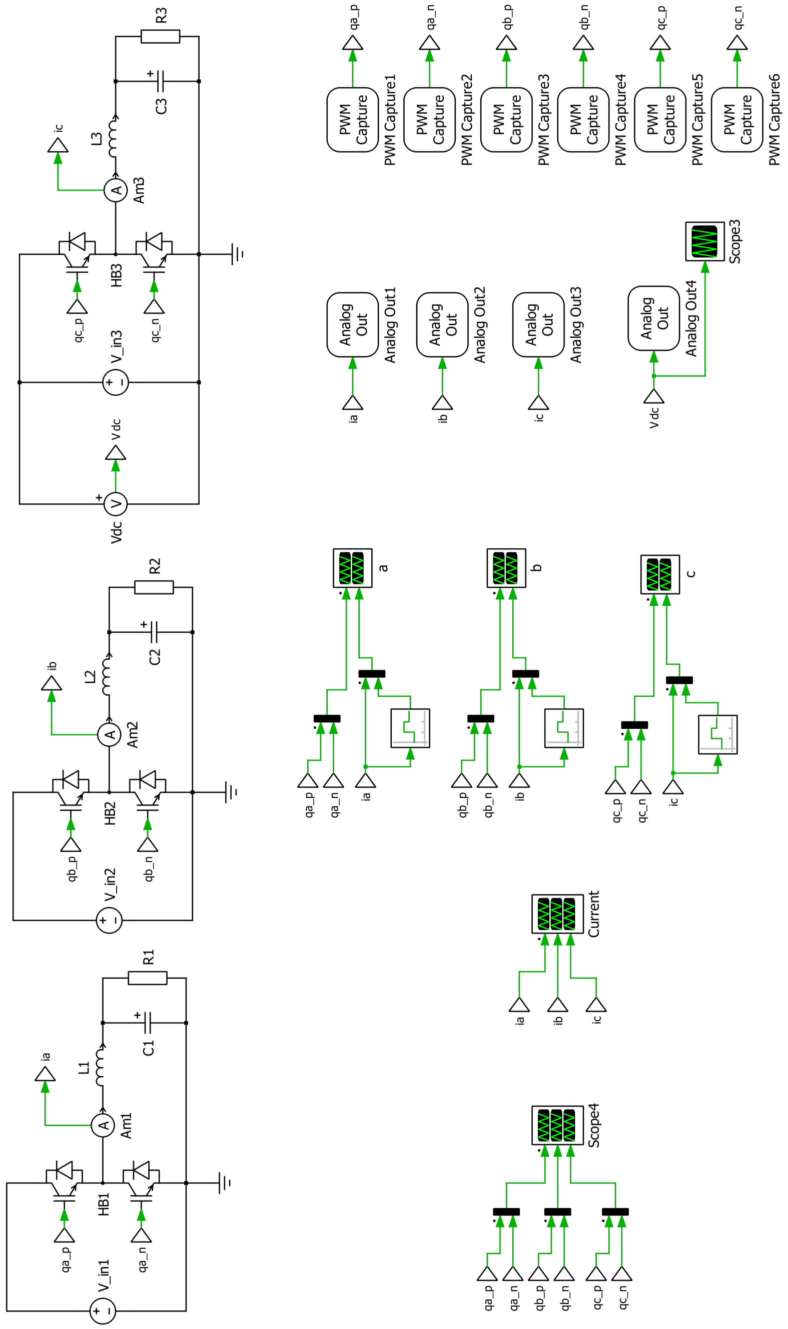


Figure 4: PLECS schematic for single inverter legs with RL load HIL test

Test V/f Control

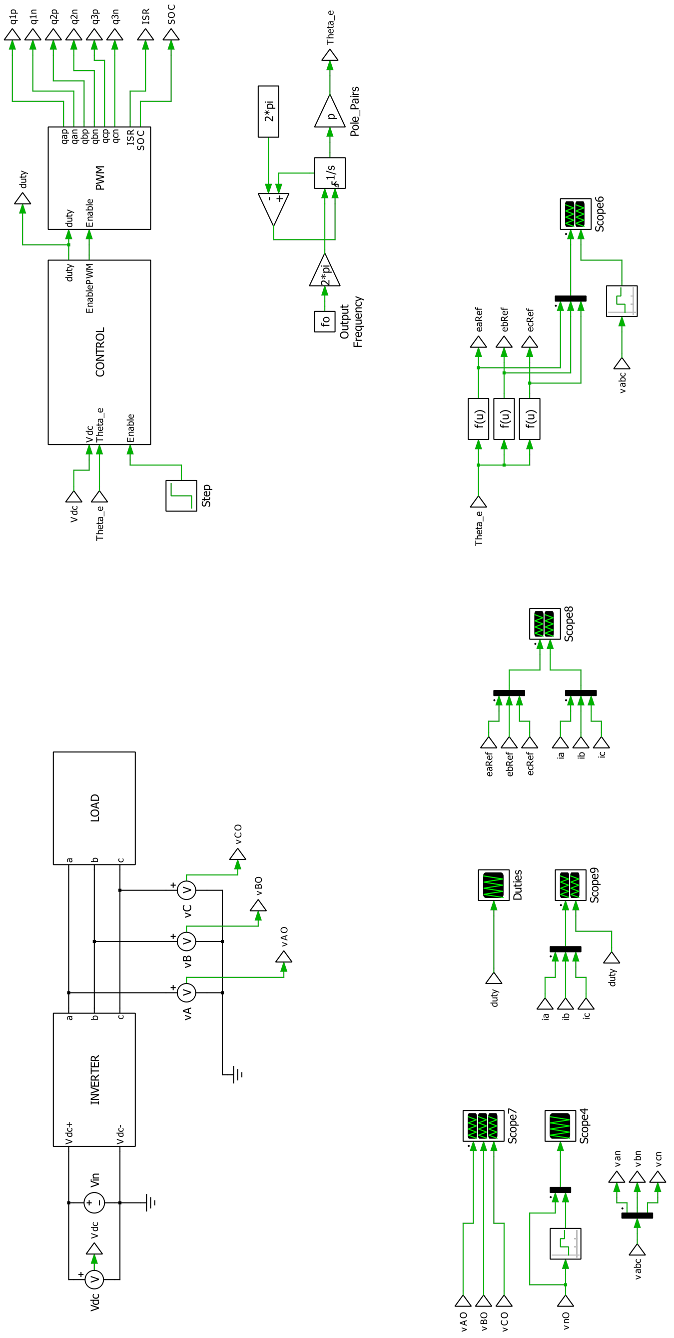


Figure 5: System overview of PLECS offline simulation for V/f control test

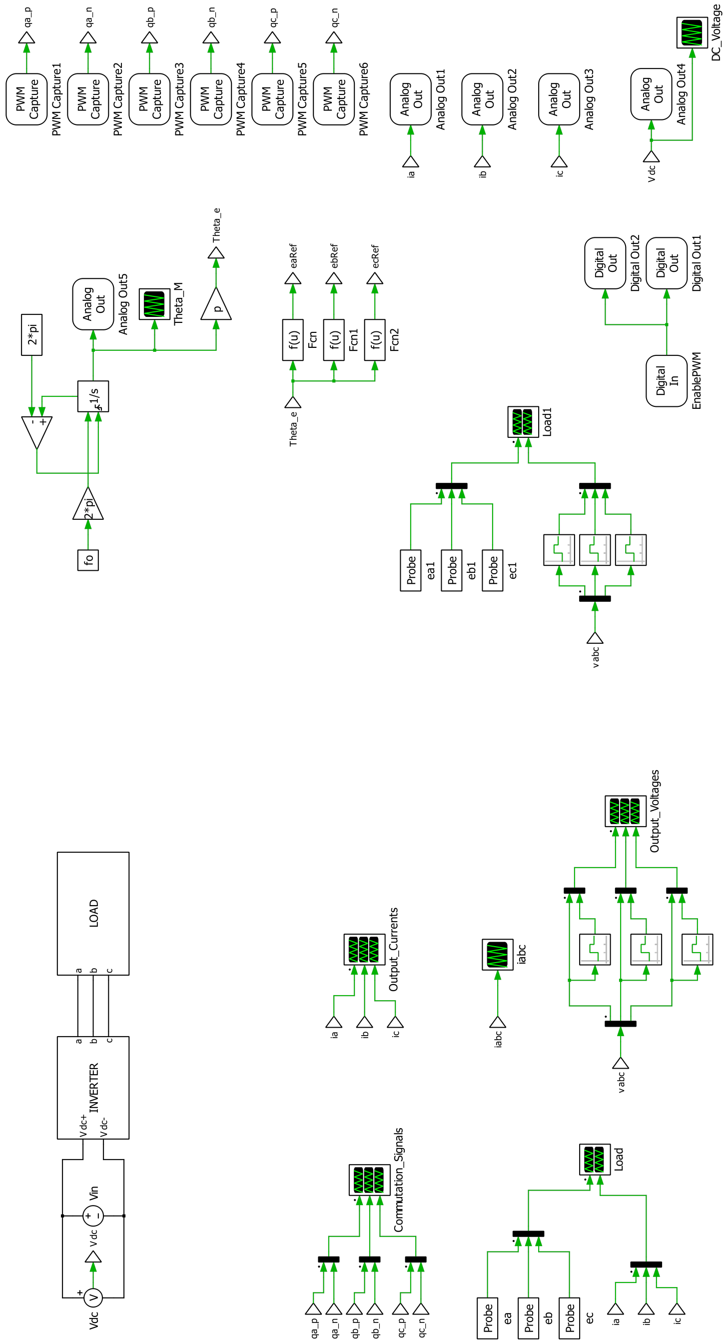


Figure 6: System overview of PLECS HIL simulation for V/f control test

Test I-f Control with L-R-E Load

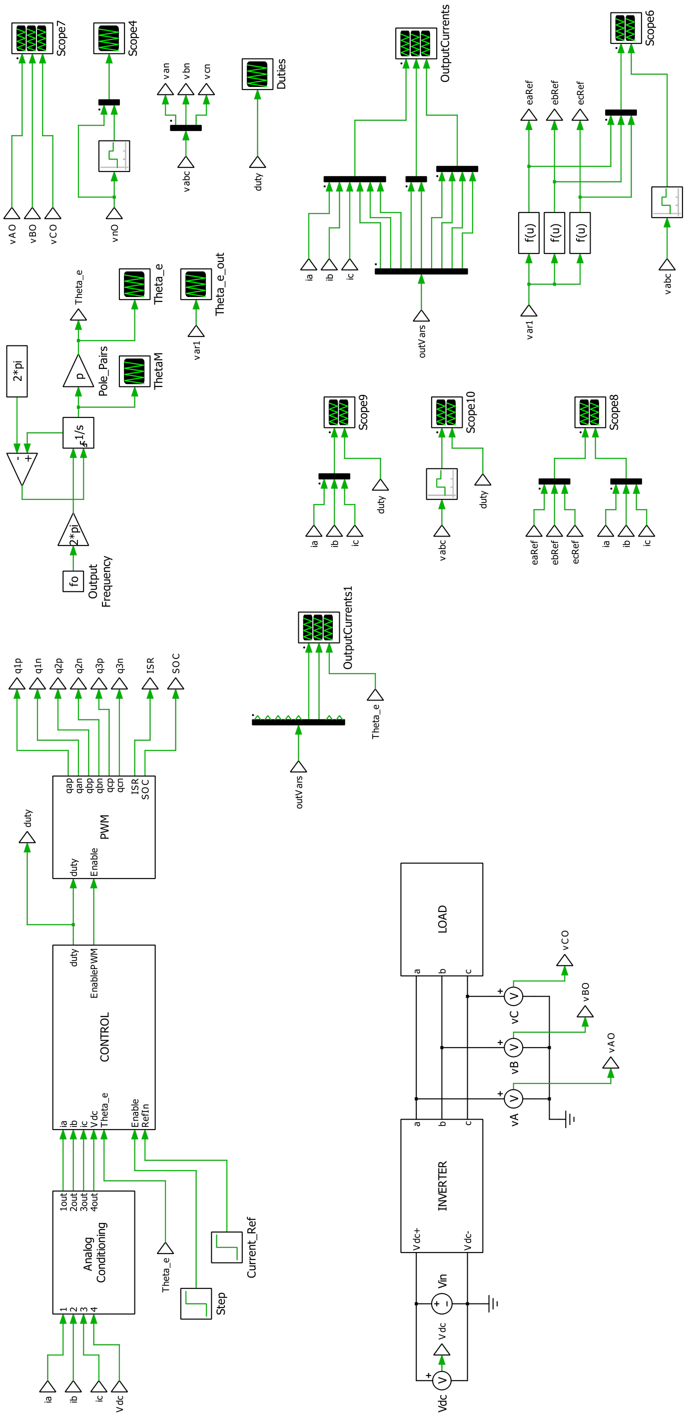


Figure 7: System overview of PLECS offline simulation for I-f control with L-R-E load test

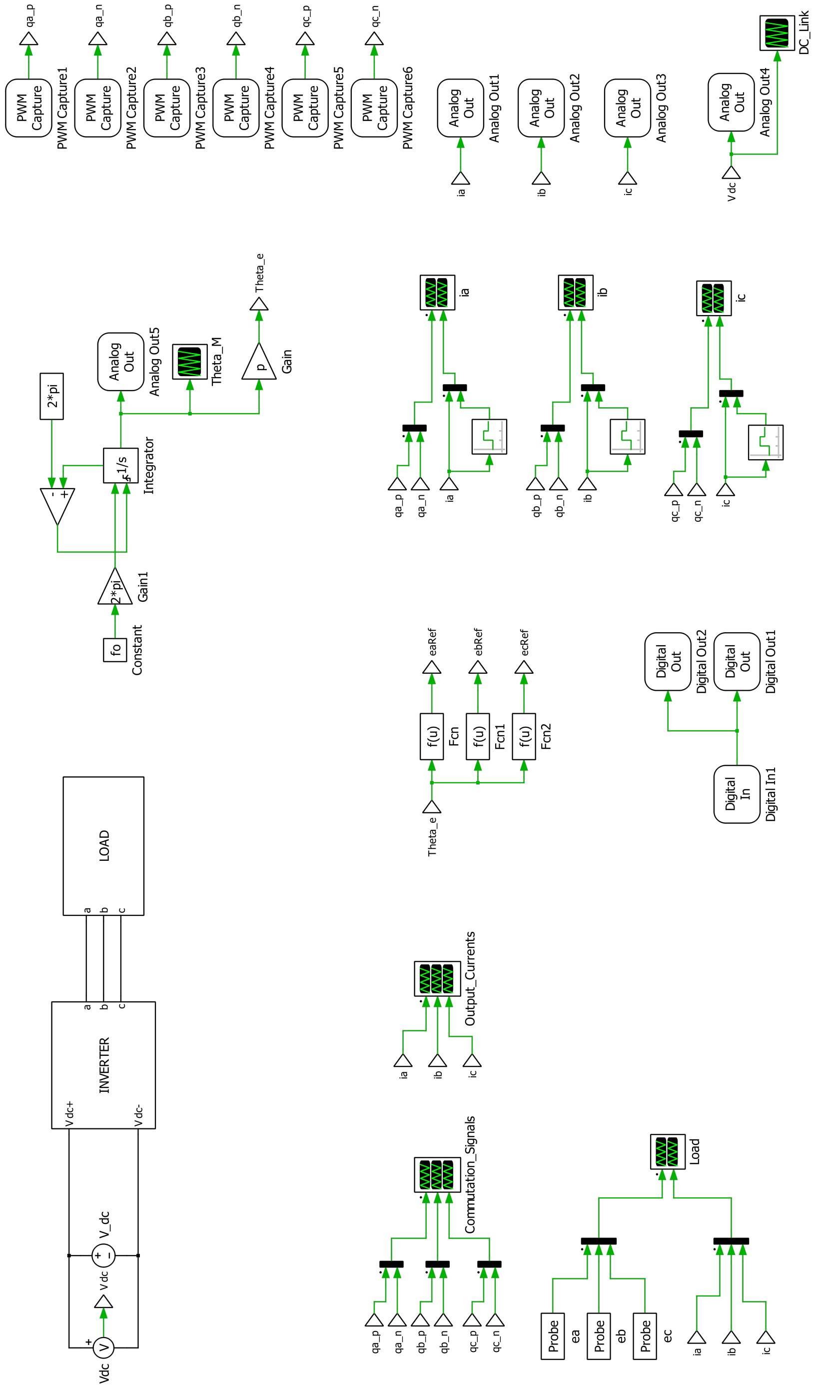


Figure 8: System overview of PLECS HIL simulation for I-f control with L-R-E load test

Test I-f Control with IPM Machine

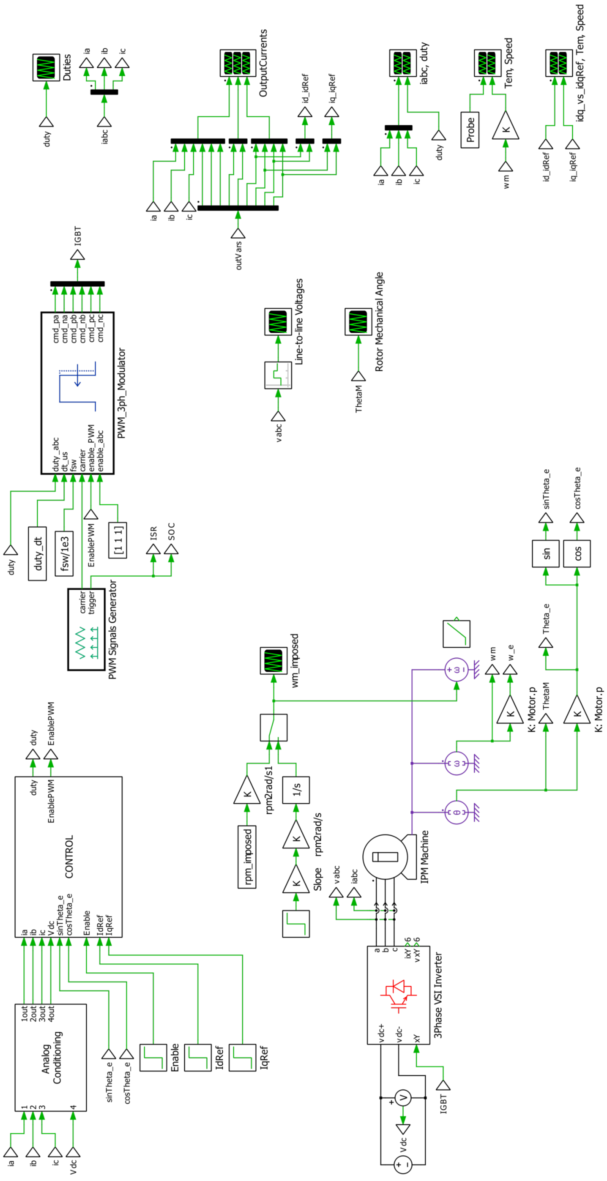


Figure 9: System overview of PLECS offline simulation for I-f control test with IPM motor

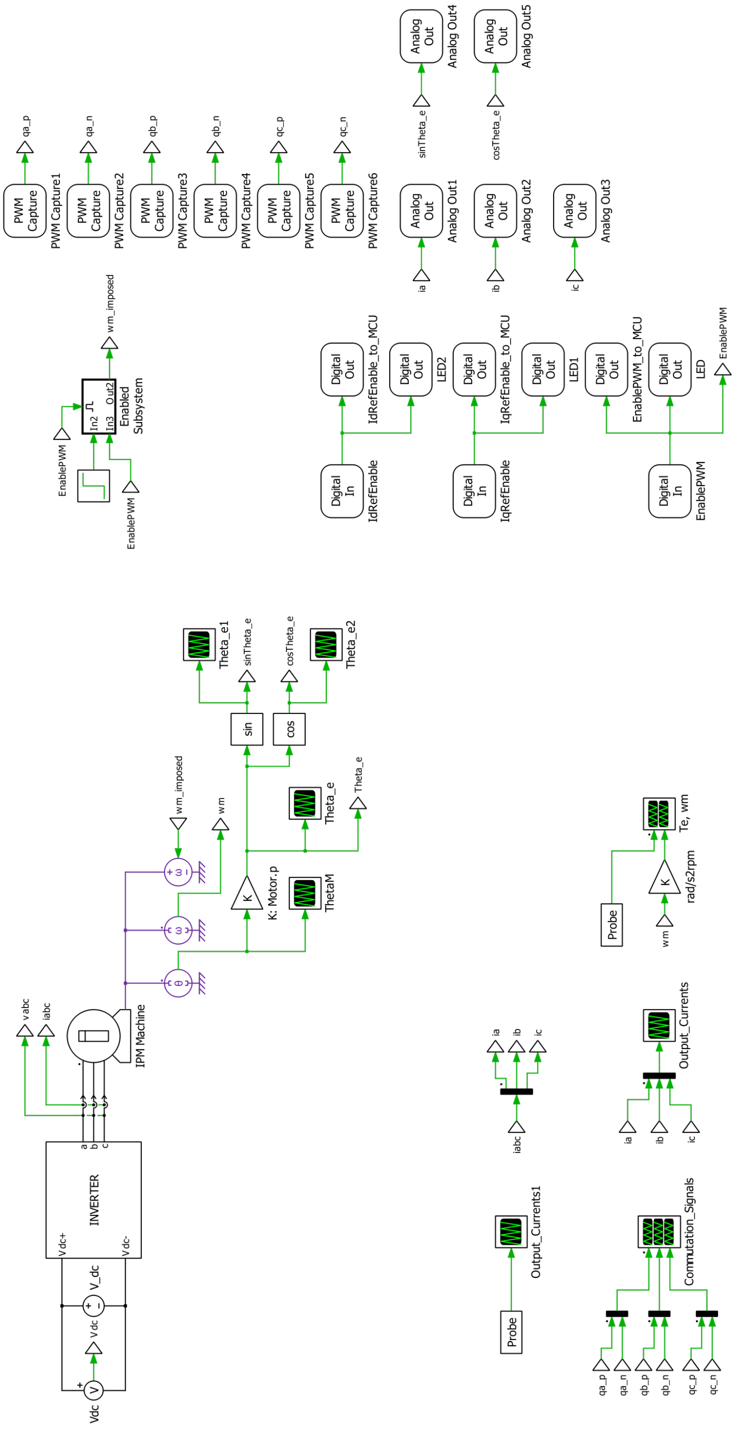


Figure 10: System overview of PLECS HIL simulation for I-f control test with IPM motor

Bibliography

[1] N. Mohan, *Electric Drives an Integrative Approach.* MNPERE Minneapolis, 2003.

[2] R. Aarenstrup, *Managing Model-Based Design.* The MathWorks, Inc, 2015.

[3] M. Violante, *Model-Based Software Design - Course Slides*, a.y. 2016-2017.

[4] TheMathWorks, *Target Language Compiler™ For Use with Real-Time Workshop® - Reference Guide Version 5.* The MathWorks, Inc., 2002.

[5] M. Violante, *Operating Systems for Embedded Systems – Course Slides*, a.y. 2017-2018.

[6] S. Bacha, I. Munteanu, and A. I. Bratcu, *Power Electronic Converters Modeling and Control with Case Studies.* Springer-Verlag London, 2014.

[7] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics, Second Edition.* KLUWER ACADEMIC PUBLISHERS, 2004.

[8] R. Bojoi, *Elettronica Industriale di Potenza - Course Slides*, a.y. 2017-2018.

[9] K. H. Nam, *AC Motor Control and Electric Vehicle Applications.* CRC Press, 2010.

[10] A. Karpati, *The Power Electronic Handbook - Chapter 5, Section 2.* CRC Press LLC, 2002.

[11] *Induction motors fed by PWM frequency inverters - Technical Guide*, WEG.

[12] *Three-phase asynchronous motors - Generalities and ABB proposals for the coordination of protective devices*, 1SDC007106G0201, ABB SACE, Jul. 2008.

[13] R. Bojoi, G. Pellegrino, P. Pescetto, and S. Rubino, *Control of AC eDrives: from theory to implementation*, IEMDC 2019.

[14] A. Nogueira, «PRACTICAL ISSUES ON THE MODELING OF PERMANENT-MAGNET MACHINES AND COGGING TORQUE CALCULATIONS IN TWO- DIMENSIONAL FINITE-ELEMENT ANALYSIS,» *International Journal of Research and Reviews in Applied Sciences ISSN 2076-734X*, vol. 15, p. 146, May 2013.

[15] R. Bojoi, *Hybrid and Electrical Propulsion System, Azionamenti con motori sincroni a riluttanza - Course Slides.*

[16] E. Armando, P. Guglielmi, G. Pellegrino, M. Pastorelli, and A. Vagati, «Accurate Modeling and Performance Analysis of IPM-PMASR Motors,» *IEEE Transactions on Industry Applications*, vol. 45, no. 1, pp. 123–130, 2009. DOI: `10.1109/TIA.2008.2009493`.

[17] R. Bojoi, G. Pellegrino, P. Pescetto, and S. Rubino, *Control of AC eDrives: from theory to implementation - Appendix*, IEMDC 2019.

[18] K. Ogata, *Modern Control Engineering, Fifth Edition.* Prentice Hall, 2010.

[19] K. J. Astrom, *Control System Design - Lecture Notes for ME 155A, Chapter 6. PID Control*, 2002.

[20] F. Blaabjerg, *Control of Power Electronic Converters and Systems - Volume 2.* Academic Press, 2018.

[21] S. Buso, *Digital control of switching mode power supplies - Slides*, 2011.

[22] G. Pellegrino, *Laboratory of Power Converters and Electrical Drives - Course Slides*, a.y. 2018-2019.

[23] D. G. Holmes, T. A. Lipo, B. P. McGrath, and W. Y. Kong, «Optimized Design of Stationary Frame Three Phase AC Current Regulators,» *IEEE Transactions on Power Electronics*, vol. 24, no. 11, pp. 2417–2426, 2009. DOI: `10.1109/TPEL.2009.2029548`.

[24] L. Wang, S. Chai, D. Yoo, L. Gan, and K. Ng, *PID and Predictive Control of Electrical Drives and Power Converters using MATLAB ® /SIMULINK ®*. IEEE Wiley, 2015.

[25] Jong-Hwan Song and Seung-Ki Sul, «Torque maximizing control of permanent magnet synchronous motor under voltage and current limitations of PWM inverter,» in *Proceedings of Applied Power Electronics Conference. APEC '96*, vol. 2, 1996, 758–763 vol.2. DOI: `10.1109/APEC.1996.500525`.

[26] Jang-Mok Kim, Kitae Park, Seog-Joo Kang, Seung-Ki Sul, and Jung-Lock Kwon, «Improved dynamic performance of interior permanent magnet synchronous motor drive in flux-weakening operation,» in *PESC Record. 27th Annual IEEE Power Electronics Specialists Conference*, vol. 2, 1996, 1562–1567 vol.2. DOI: `10.1109/PESC.1996.548789`.

[27] Jong-Hwan Song, Jang-Mok Kim, and Seung-Ki Sul, «A new robust SPMSM control to parameter variations in flux weakening region,» in *Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 2, 1996, 1193–1198 vol.2. DOI: `10.1109/IECON.1996.566049`.

[28] S. K. Sul, *Control of Electric Machine Drive Systems*. IEEE Wiley, 2011.

[29] G. Pellegrino, R. I. Bojoi, and P. Guglielmi, «Unified Direct-Flux Vector Control for AC Motor Drives,» *IEEE Transactions on Industry Applications*, vol. 47, no. 5, pp. 2093–2102, 2011. DOI: `10.1109/TIA.2011.2161532`.

[30] G. Pellegrino, E. Armando, and P. Guglielmi, «Direct Flux Field-Oriented Control of IPM Drives With Variable DC Link in the Field-Weakening Region,» *IEEE Transactions on Industry Applications*, vol. 45, no. 5, pp. 1619–1627, 2009. DOI: `10.1109/TIA.2009.2027167`.

[31] R. Bojoi, P. Guglielmi, and G. Pellegrino, «Sensorless Stator Field-Oriented Control for Low Cost Induction Motor Drives with Wide Field Weakening Range,» in *2008 IEEE Industry Applications Society Annual Meeting*, 2008, pp. 1–7. DOI: `10.1109/08IAS.2008.18`.

[32] G. Pellegrino, E. Armando, and P. Guglielmi, «Field oriented control of IPM drives for optimal constant power operation,» in *2007 European Conference on Power Electronics and Applications*, 2007, pp. 1–10. DOI: `10.1109/EPE.2007.4417644`.

[33] P. L. Jansen and R. D. Lorenz, «A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine drives,» *IEEE Transactions on Industry Applications*, vol. 30, no. 1, pp. 101–110, 1994. DOI: `10.1109/28.273627`.

[34] A. Vagati, M. Pastorelli, G. Franceschini, and V. Drogoreanu, «Digital observer-based control of synchronous reluctance motors,» in *IAS '97. Conference Record of the 1997 IEEE Industry Applications Conference Thirty-Second IAS Annual Meeting*, vol. 1, 1997, 629–636 vol.1. DOI: `10.1109/IAS.1997.643133`.

[35] G. Pellegrino, E. Armando, and P. Guglielmi, «Direct-Flux Vector Control of IPM Motor Drives in the Maximum Torque Per Voltage Speed Range,» *IEEE Transactions on Industrial Electronics*, vol. 59, no. 10, pp. 3780–3788, 2012. DOI: `10.1109/TIE.2011.2178212`.

[36] ——, «Direct-flux field-oriented control of IPM motor drives with robust exploitation of the maximum torque per voltage speed range,» in *2010 IEEE*

International Symposium on Industrial Electronics, 2010, pp. 1271–1277. DOI: `10.1109/ISIE.2010.5637031`.

[37] R. Bojoi, Z. Li, S. A. Odhano, G. Griva, and A. Tenconi, «Unified direct-flux vector control of induction motor drives with maximum torque per ampere operation,» in *2013 IEEE Energy Conversion Congress and Exposition*, 2013, pp. 3888–3895. DOI: `10.1109/ECCE.2013.6647216`.

[38] `www.osek-vdx.org`.

[39] `http://www.erika-enterprise.com/wiki/index.php?title=Main_Features_of_ERIKA_v3`.

[40] `https://hightec-rt.com/en/products/development-platform.html`.

[41] P. Scandale, «Model Based Design of Automotive Embedded System,» Politecnico di Torino, 2019.

[42] `https://www.plexim.com/plecs`.

[43] `https://www.plexim.com/products/plecs_coder`.

[44] *RT Box User Manual*, Plexim GmbH, Jul. 2019.

[45] *Infineon AURIX™ TC27x C-Step 32-Bit Single-Chip Microcontroller - User's Manual*, V2.2, Infineon Technologies AG, Dec. 2014.

[46] `https://www.plexim.com/products/rt_box`.

[47] *RT Box LaunchPad Interface*, Plexim GmbH, Sep. 2019.

[48] I. R. Bojoi, E. Armando, G. Pellegrino, and S. G. Rosu, «Self-commissioning of inverter nonlinear effects in AC drives,» in *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*, 2012, pp. 213–218. DOI: `10.1109/EnergyCon.2012.6347755`.