Predictive Control Scheme of the PMSM with a Modulator of Variable Switching Frequency

Von der Naturwissenschaftlich-Technischen Fakultät

der Universität Siegen

Zur Erlangung des akademischen Grades

Doktor der Ingenieurswissenschaften

(Dr.-Ing.)

genehmigte Dissertation

von

M. Sc. Fernando David Ramirez Figueroa

aus Tlalnepantla de Baz, Mexiko

1. Gutachter: Prof. Dr.-Ing. Mario Pacas

2. Gutachter: Prof. Dr.-Ing. Ralph Kennel

Tag der mündlichen Prüfung: 07.03.2017

Acknowledgements

The present work was realized during my research activities at the Institute of Power Electronics and Electrical Drives of the University of Siegen. It was carried out thanks to the financial support of the German Academic Exchange Service (Deutscher Akademischer Austauschdienst - DAAD) and the Mexican National Council on Science and Technology (Consejo Nacional de Ciencia y Tecnología – CONACYT) under the scholarship "CONACYT-DAAD 2012."

First, I would like to express my deepest gratitude to my supervisor and head of the institute Prof. Dr.-Ing. Mario Pacas, for allowing me to be part of his working group and his unconditional support during my doctoral studies. I would also like to thank Prof. Dr.-Ing. Ralph Kennel of the Technische Universität München for his interest in my and his work as co-examiner. Moreover, I would like to thank my colleagues Dr.-Ing. Simon Feuersänger and Dr.-Ing. Manuel Steinbring for their support and vital help.

Furthermore, I would like to express my appreciation and gratitude to the personnel of the Institute for always providing a friendly and supportive working environment. I would like to mention Dipl.-Ing. Erhard Oerter and Mrs. Sabine Leukel for the great technical and administrative assistance. The time I spent working at the institute was an invaluable learning experience as well as the origin some great memories.

Special thanks to Ismael and Victor, for their great friendship and support has always that made a great difference. To my sister Carolina, for bringing me close to the family and home during the moments that I most need it, without her even knowing it. To Anna Maria for all her support on the final phase of my doctoral studies. Finally, I would like to thank my parents, Virginia and Fernando for all their unconditional support and love ever since their brought me to life.

Fernando David Ramirez Figueroa

Contents

Abstract

In this work, a novel predictive control scheme for the permanent magnet synchronous machine by using a pulse-width modulation technique of variable switching frequency and variable sampling time is proposed. The control scheme is based on the model-based predictive control theory and makes use of a flexible pulse width modulation stage that can be reconfigured during the operation of the system without stopping the execution.

The PI-controllers conventionally used in field-oriented control schemes are modified so that they can be used during the prediction stage of the control with different sampling times, without saturating the integral part of the controller. A finite set of switching frequencies is used for the optimization of a cost function, which aims to minimize not only the error in the torque and flux producing components of the currents of the space phasor of the PMSM. In addition, the cost function includes the reduction of the switching losses by directly reducing the switching frequency and maintaining a minimum level of desired performance mainly regulated by the maximum allowed torque ripple. The oversampling of the reference signals allows the adaption of the sampling frequency in order to avoid delays in the reaction of the system in case of transients.

The proposed control scheme demands a high degree of flexibility in the implementation and a fast signal processing. Therefore, a FPGA implementation was mandatory that allows the realization of most of the control algorithm in silicon and a parallel execution. The acquisition of analogue control signals is carried out by means of Delta-Sigma ADCs with a very short conversion time that permits an oversampling and a signal processing with a high time resolution. As a result, the control features not only an excellent dynamic behavior with high bandwidth allowing to consider the harmonics of the current i.e. the switching ripple.

The proposed predictive control approach, successfully overcomes the majority of the drawbacks of conventional field-oriented control and eliminates some of the drawbacks of conventional model predictive control, delivers good dynamics in the torque behavior and exploits the parallel processing capabilities and high computational power provided by an FPGA implementation. A sensorless implementation complements the proposed predictive control strategy. The faster processing capabilities

of the FPGA allow the sensorless operation of the drive at very low speeds without the need of additional signal injection.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit der prädiktiven Regelung einer permanenterregten Synchronmaschine mit Pulsweitenmodulation und variabler Schaltfrequenz. Das vorgestellte Regelkonzept nutzt eine Modell basierte prädiktive Regelung wobei eine Pulsweitenmodulation verwendet wird, die während des Betriebs rekonfiguriert werden kann. Konventionelle, digitale PI-Regler, die auch in der feldorientieren Regelung Verwendung finden, werden während der Prädiktionsphase mit verschiedenen Abtastzeiten so modifiziert, dass es zu keiner Sättigung der Regler kommt.

Mittels einer begrenzten Anzahl von Schaltfrequenzen kann die Kostenfunktion optimiert werden. Die zu minimierende Kostenfunktion beinhaltet unter anderem das Drehmoment und die flussbildende Komponente des Raumzeigers der permanenterregten Synchronmaschine. Während der Minimierung können die Schaltverluste reduziert werden, wobei eine minimal geforderte Dynamik erhalten bleibt, welche in der Regel durch die maximal erlaubte Drehmomentwelligkeit begrenzt wird. Die Überabtastung der Sollwerte erlaubt die Adaptierung der Schaltfrequenz während transienter Vorgänge um die Verzögerung in der Reaktion des Systems zu reduzieren.

Das vorgeschlagene Regelungsverfahren erforscht eine hohe Flexibilität und eine schnelle Signalverarbeitung, wodurch die Implementierung erfolgt darum mittels eines FPGAs. Die meisten notwendigen Regelungsalgorithmen werden in Silizium ausgeführt und parallel verarbeitet. Die Messdatenerfassung der analogen Signale wurde mittels Delta-Sigma ADCs durchgeführt, die eine sehr kurze Umsetzungszeit haben und damit die Überabtastung der Signale mit höherer zeitlicher Auflösung ermöglichen. Als Ergebnis hat das vorgeschlagene Regelungsverfahren ein hervorragendes dynamisches Verhalten mit hoher Bandbreite, wodurch die Berücksichtigung von Harmonischen möglich wird.

Der vorgeschlagene Ansatz der prädiktiven Regelung eliminiert einen Großteil der Nachteile der herkömmlichen feldorientierten Regelung sowie einige der Nachteile der gewohnten modellprädiktiven Regelung. Insbesondere wird eine gute Dynamik im Drehmomentverhalten erreicht, wobei die parallele Verarbeitungsmöglichkeit und hohe Rechenleistung durch die FPGA-Implementierung ausgenutzt wird. Eine sensorlose Implementierung vervollständigt das vorgestellte prädiktive Regelungsverfahren. Die

hohe Rechenleistung des FPGAs erlaubt den sensorlosen Betrieb des Antriebs bei niedrigen Geschwindigkeiten ohne zusätzliche Signaleinprägung.

Nomenclatures

Symbols

º	Degrees, which unless otherwise noticed, mean electrical degrees
$\|x\|$	Absolute value of x
C	Capacitor, Constant
clk_{DSP}	DSP execution clock
clk_{FPGA}	FPGA execution clock
D	Diode, down-sampling or decimation filter, duty-cycle
d,q	Rotor-flux oriented reference frame
e	Error
f	Frequency
f_B	Bandwidth frequency
f_D	Decimation frequency
f_{pen}	Penalizing function
f_N	Nyquist-rate sampling frequency
f_S	Switching frequency, sampling frequency
g	Cost function
i	Current
Im	Imaginary part
j	Constraint
J	Moment of inertia
K	CORDIC scaling factor
K_f	Sum of the CORDIC scaling factor during different iterations
K_I	Integral controller gain
K_P	Proportional controller gain
L	Inductance
M	Torque, decimation ratio
M_e	Electromagnetic Torque
M_L	Mechanical Load Torque
N	N steps ahead prediction horizon
p	Pair of Poles
R	Resistance
Re	Real part
S	Ideal switch (power semiconductor)
t	Time
T	Transistor
T_D	Dead-time
T_E	Execution time
T_P	Positive-phase switch
T_N	Integration time constant, negative-phase switch
T_S	Sampling time, switching period, modulation period

u, U	Voltage, phase U
U_{DC}	DC-link voltage
UVW	Three-phase AC system
V	Phase V
W	Phase W
w	Weight factor
α,β	Stator-fixed coordinate system
Δ	Difference operator, switching ripple, M_e ripple, quantization step
γ	Angle between the α-axis and the rotor flux-oriented reference frame
λ	Duty-cycle or power-on time within a period
μC	Microcontroller
ψ	Flux linkage
ω	Angular speed
Ω	Mechanical speed

Subscripts

x_1	Stator quantity
x_2	Rotor quantity
x_d	Real part of a space phasor in a *d,q*-reference frame
x_{Lim}	Limit
x_{max}	Maximum value
x_{mech}	Mechanical
x_N	Nominal value, normalized value
x_q	Imaginary part of a space phasor in a *d,q*-reference frame
x_{THLD}	Threshold limit value
x_U	Phase U
x_V	Phase V
x_W	Phase W
x_α	Real part of a space phasor in a stator-fixed reference frame
x_β	Imaginary part of a space phasor in a stator-fixed reference frame

Superscripts

x^*	Reference value
$\dot{x}$	Time derivative
$\hat{x}$	Estimated, predicted value
x^{dist}	Disturbance value
x^n	*n*-element of the finite set of possible solutions (FS-MPC)
x^N	Base value
x^{Opt}	Optimal value
x^P	Predicted, estimated value

Acronyms

ΔΣ	Delta-Sigma (ADC)
AC	Alternating Current
A/D	Analog-to-Digital
AV	Active Voltage space phasor
ADC	Analog-to-Digital Conversion
AlNiCo	Aluminum (Al), Nickel (Ni) and Cobalt (Co)
ASIC	Application Specific Integrated Circuit
CAD	Computer-Aided Design
CIC	Cascaded Integrator Comb Filter
CORDIC	Coordinate Rotational Digital Computer
CS-MPC	Continuous Set Model Predictive Control
DAC	Digital-to-Analog Converter
DC	Direct Current
DF	Decimation Filter
DSP	Digital Signal Processing, Digital Signal Processor
DTC	Direct Torque Control
DMPC	Direct Model-based Predictive Control
EESM	Electrically Excited Synchronous Machine
EMI	Electromagnetic Interference
FOC	Field Oriented Control
FS-MPC	Finite Set Model-based Predictive Control
FPGA	Field Programmable Gate Array
FIR	Finite Impulse Response
GaN	Gallium Nitrate
GPIO	General Purpose Input Output (pin)
GPC	Generalized Predictive Control
HDL	Hardware Description Language
THLD	Threshold
IGBT	Insulated Gate Bipolar Transistor
I/O	Input / Output
INFORM	Indirect Flux Detection by Online Reactance Measurement
LPF	Low Pass Filter
LSB	Least Significant Bit
LUT	Look-Up Table
MBC	Model-Based Control
MBD	Model-Based Design
MPC	Model-based Predictive Control
MSB	Most Significant Bit
MRAS	Model Reference Adaptive System
NTF	Noise Transfer Function
OpenCL	Open Computing Language
OSR	Oversampling Rate

PI	Proportional Integral (controller)
PLL	Phase Locked Loop
PM	Permanent Magnet
PMSM	Permanent Magnets Synchronous Machine
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read Only Memory
RPWM	Random Pulse Width Modulation
SAR	Successive-Approximation
SiC	Silicon Carbide
SoC	System-on-Chip
SmCo	Samarium-Cobalt
SPM	Space Phasor Modulation
SNR	Signal-to-Noise Ratio
STF	Signal Transfer Function
THD	Total Harmonic Distortion
VSI	Voltage Source Inverter
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
ZV	Zero Voltage space phasor

1 Introduction

Energy is the foundation of any technological development and as such the energy consumption in highly industrialized countries reflects how advanced is their technological development. Energy can be found in many forms, yet from the point of generation to the final point of utilization, the most effective way to transport it is converting it to electrical energy.

Another important form of energy is the mechanical one, which is needed in a wide spectrum of power ratings wherever manufacturing or production processes, transportation of persons or goods (among many others) take place, ultimately, electrical drives are well-suited electromechanical devices for the final conversion at the point of usage [1].

The research on the control of AC electrical drives was first limited by the available processing power, as microelectronics evolved and became more powerful, the implementation of more sophisticated and complex methods also became possible. Today, one of the most active research topics is the development of speed sensorless or encoderless AC drives featuring a dynamic behavior comparable to drives with a mechanical sensor on the shaft [2], [3]. Other very active research active topics deal with the enhancement of the control techniques like MPC [4] or artificial intelligence [5].

1.1 Motivation

The motivation of this work is the enormous and continuously increasing processing capabilities made available by modern microelectronics. A classic example consists of the current control of pulse-width modulated inverters. As shown in Figure 1.1, by using modern digital electronics the processing time of the control algorithm has been dramatically reduced. In this figure, T_{ADC} is the analog-to-digital conversion time, $T_{Control}$ the time it takes to execute the control algorithm and T_S is the sampling period usually equal to the switching period of the power converter, or half of it.

When a general-purpose microcontroller is used (Figure 1.1a) the main limitation factor is its computing power. The sampling period has to be adapted to this limit leading to a reduced bandwidth of the closed-loop system. Figure 1.1b depicts the case of a DSP used for the implementation. These components allow the execution of the control

algorithm in a few dozens of microseconds. Thus, the speed of the controller is no longer a restriction of the closed-loop system. The limitations of the sampling time are based on the switching losses of the power converter.

Finally Figure 1.1c, corresponds to an FPGA-based controller, due to their potential to translate all the potential parallelism of the control algorithm into the hardware architecture, FPGAs only take a fraction of the switching period to execute the control algorithm. Such behavior makes FPGA-based controller comparable to their analog counterparts [6].

FPGA-based controllers preserve the advantages of analog controllers, such as high bandwidth and almost no processing delay, without including their limitations: parameter drifting and poor level of integration. Furthermore, a closer look to Figure 1.1c shows that a lot of time is left unused in each sampling period, leaving the capabilities of FPGAs unexploited.

Therefore, the remaining processing time can be used to improve the performance of the controller by making use of *oversampling* techniques, *predictive* control and combine it with the possibility to redesign peripherals to a very precise level of detail.

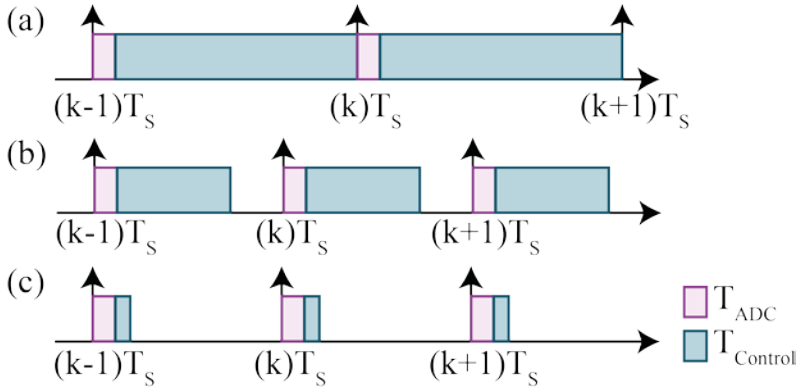

Figure 1.1: Processing time distribution: (a) General-purpose microcontroller, (b) DSP, (c) FPGA.

1.2 Objectives

The objective of this work is to develop a control technique for power converters that takes advantage of the powerful processing capabilities of FPGAs, which also brings the possibility to modify conventional peripherals to a very precise level.

A novel predictive variable switching frequency modulator-based control scheme for the PMSM is to be developed and validated. The advantages of field-oriented control and model-based predictive control are to be combined to create a novel control strategy.

Furthermore, the fast sampling times and highly accuracy of the Sigma-Delta ADCs can be used to improve the signal identification process of sensorless techniques.

1.3 State of the Art

A description regarding the state of the art of variable switching frequency modulator-based as well as model-based predictive control in the area of power electronics is presented. Because this project was implemented using an FPGA, a revision on state of the art regarding power electronic applications is also included.

1.3.1 Variable Switching Frequency

It is well known that commutation losses are directly related to the switching frequency of the power converter. A trade-off between the commutation losses and the introduced harmonic distortion (THD), or the dynamic performance of the control e.g. the settling time or the torque ripple among other criteria, is always hard to find. Conventionally, the switching frequency is usually fixed when the control technique works with a modulation stage, like in the field-oriented control (FOC) [7], or is of variable nature as in direct torque control (DTC) [8].

Random pulse-width modulation (RPWM) is a widely used implementation of variable switching frequency modulation that has many years of existence [9]. This method maintains the switching frequency constant; however, the acoustic noise is randomly spread over a wide spectrum making it more pleasing to the human ear. RPWM is in constant development [10], [11], later it has been used to reduce the harmonic distortion introduced by the modulation technique [12].

In the case of single-phase inverters, the choice of the switching frequency has to be adjusted between the THD and the switching losses. Based on a detailed time-domain analysis of the current distribution Mao X. et al [13] in 2009 propose a variable switching method to reduce the switching losses and regulate the current ripple. Experimental results are presented, consisting of the full-bridge inverter with an LC filter and a resistive load.

One of the more detailed works on variable switching frequency modulator-based is that of Jiang D. et al. [14] in 2013, where two algorithms are proposed to change the switching frequency based on a current ripple criterion. The current ripple is calculated for a given fixed switching period and then proportionally updated; one method is aimed

to control the maximum current ripple peak while the other regulates the RMS value. Experiments are carried out on a three-phase RLC load, successfully regulating the current ripple mainly in steady-state. Wang Z. et al propose a chaotic space phasor modulation technique to be used in conjunction with FOC and DTC schemes in [15], to reduce the harmonics introduced by fixed switching frequencies; experimental results are presented.

1.3.2 Sensorless Control

Sensorless AC drives i.e. AC drives without angular encoder for the measurement of the angular position are much desired because of their advantages. By discarding the mechanical sensor, the hardware complexity can be reduced the noise immunity and reliability can be improved, along with fewer maintenance requirements [2]. Therefore, sensorless control schemes have been extensively researched in the past decades. These advanced control schemes have been enabled because of the constant increase in computational power of modern microelectronics [3].

There is a wide diversity of methods for different operating conditions as presented in [2] by Holtz. Roughly, these sensorless control methods can be classified in *fundamental wave* models and *signal injection* techniques. The *fundamental wave* models are based on the voltage model of the machine. Given that in voltage source inverter fed drives the stator voltage is a forced known quantity. Consequently, the estimate of the stator voltage can be integrated yielding the stator flux linkage, which can be used to obtain the stator field angle.

The main disadvantages of the *fundamental wave* models are DC-offset and drift effects derived from analog signal measurement, causing an accumulation of DC components at the integrator output. By introducing a negative low-gain feedback, the integrator can be stabilized. Another problem is the inaccuracies introduced by estimating the stator voltages, making the real and control values differ mainly because of the nonlinear dead-time delay. Finally, despite all the efforts to improve the performance of these models they become unobservable at zero frequency [2], [3].

The *signal injection* techniques make use of the anisotropic properties of the machine that are not included in the fundamental wave models. A test signal is used to excite the machine, it usually much higher that the fundamental frequency of the stator. These high-frequency signals do not affect the mutual flux linkage with the fundamental wave; therefore the high-frequency identification loop can be considered independent

of the fundamental wave control. As opposed to the fundamental wave models, these methods can be used at zero frequency.

In spite of all its disadvantages, the *fundamental wave* models are the most accepted ones in the industry. Perhaps first introduced by Frus and Kuo in 1976 in [16] since then, a great deal of progress has been achieved, like Barinberg et. al enhances the voltage model with an observer in [17]. Acarnley and Watson have a good review of the advancements in [18]. The most recent research on this topic has been done on the low-speed operation, some of the first works done by Jansson et al in [19] achieving a stable speed reversal, with work being continuously done [20]–[22].

In the case of the *signal injection* methods, Schroedl presented some of the first developments in [23], called the INFORM (indirect flux detection by online reactance measurement.) Some years later came to the alternating carrier injection by Linke and Kennel [24]. There is also some steady progress done on the research pertinent to these methods, most of them allowing a full range of the sensorless drive operation [25]–[27]. Perhaps some of the most recent and relevant research on this topic is being carried for medium voltage applications on the electrically excited synchronous machine (EESM) [22], [28], [29].

1.3.3 Model-based Predictive Control

Since the introduction of FOC and DTC more than 30 years ago, these control techniques have become the standard for variable speed controlled drives; nonetheless, these methods have their limitations and disadvantages. Model-based Predictive Control (MPC) is the only one among many other advanced control techniques (techniques which are more advanced than standard PID control) that has been of great influence on the research and development directions of industrial control systems, with many successful practical applications [30].

Although Predictive Control has yield a wide range of different schemes, they all have a main characteristic: the use of the model of the system for the prediction of the feature behavior of the controlled variables. Model-based Predictive Control (MPC) also known as *receding horizon control* has slowly matured with decades of continuous development, some of the first ideas first ideas on MPC date back the 1960s [31], since then it has slowly matured with time mainly used in the process industry [32].

With the constant increase in digital signal processing capabilities of microprocessor units, its application was possible in the field of power electronics [33],

[34]. The idea of MPC comes from the deadbeat predictive control technique, in this approach a model of the controlled system is used to calculate the required voltage reference so that the reference value is reached in the next sampling instant; this process is done once every sampling period. The voltage is then applied using a pulse-width modulation technique.

The main characteristic of MPC is the use of a model of the controlled system for the prediction of the possible future actions of the controlled variables; the controller uses this information in such a way that the optimal control action can be obtained, given an already established optimization criterion; which is expressed as a quality function to be minimized. MPC for power electronics has been mainly classified as Continuous Control Set (CS), which uses a modulator to generate the voltage outputs and Finite Control Set (FS) or Direct MPC, which considers only the natural space voltage phasors that the voltage source inverter can generate and thus, work without a modulator.

In CS-FCS, the optimization problem is analytically solved, the most developed method is the generalized predictive control (GPC), which has been proven to be fast enough for drive control [33]–[36]. Nevertheless, it becomes difficult to include system constraints in the algorithm. The switching frequency is usually fixed; the harmonic distortion introduced by the PWM is low. However, due to the long sampling times derived from low switching frequencies, the dynamic performance is usually worse than in linear controllers and the computational requirements are very high.

In [37] a multi-sampled method is proposed suitable for low switching frequencies, which solution is computed offline because of its heavy computational requirements. Another application of CS Nonlinear MPC aimed to control the flux and torque of an induction machine is presented in [38], it is implemented on a low-cost FPGA to demonstrate the feasibility of a cost-effective industrial drive system and tackle the high computational burden. Some of the newest contributions in the field of CS-MPC applied to power converters are presented in [39], [40].

The implementation of FS-MPC is much simpler and faster because the optimization solution does not have to be analytically found. The modulator stage can also be avoided, reducing the set of solutions to a finite set. Finally, if the prediction horizon is reduced to one, the optimization process becomes very trivial [30]. Nonetheless, to ensure a good performance, a high sampling rate is required, which also increases the computational effort. Moreover, because the switching frequency is variable, the harmonic distortion is hard to be determined and therefore controlled.

Another difficult task is to select a new voltage space phasor that will keep the desired torque and flux references while reducing the switching frequency.

Several applications of FS-MPC to power converters can be found in the literature [30], [41]–[47], all presenting better dynamic response than conventional PI-Control used in field-oriented control optimizations. Although most of the implementations are of variable switching frequency, in [44] it is shown that lower switching frequencies in conjunction with a modulation technique, can also deliver the high dynamic performance of conventional FS-MPC.

The possibility to include constraints on the cost function had made this type of MPC very popular in power electronics. The steady-state error remains an open issue for MPC [4], [48], to solve this problem the use of an integral term in the cost has been proposed in [49] and [50]. The calculation of the integral part is different to conventional linear PI-control because it has to be explicitly calculated along each sampling period based on a continuous model.

1.3.4 FPGA-based Control Schemes for Power Electronics and Drives

Novel industrial control systems must have a high performance, along with high reliability and flexibility. At the same time, the cost is also an important issue, time-to-market hast to be reduced as much as possible, the price of the controller has to be as low as possible as well as its energy consumption and of the controlled system. On top of that, the cost reduction gets even more challenging because the complexity of sophisticated control algorithms demands a lot of computing resources and reduced execution times [5].

Traditionally, there are two different approaches that designers can follow to implement digital control systems for industrial applications. The first is the pure software platform, associated to microcontroller or Digital Signal Processor controllers (DSP) [51]. These devices integrate a microcontroller unit along with several peripherals necessary to control the targeted system in real-time and communicate with external devices. The difference between microcontroller and DSP is the ratio between processing power and integrated peripherals. While the microcontroller integrates a general processing unit with many control and communications devices, the DSP integrate a high-performance processor unit and few peripherals.

Nevertheless, the boundaries between these two concepts are fading because microcontrollers are becoming more and more powerful while DSPs get more

peripherals integrated. The main advantages of this approach are the reliability of these technologies and their reduced costs; whereas the main drawback is the performance limitation due to the sequential execution related to software platforms, thus the potential parallelism that control algorithms can achieve remains unexploited.

The alternative approach is the use of Field Programmable Gate Arrays (FPGAs), which are devices that consist of elementary cells and interconnections that are fully programmable by the end user, enabling them to build specific hardware architectures that match the desired application. FPGAs can be seen as programmable microcontrollers with dedicated peripherals and hardware accelerators that allow the implementation of parallel algorithms.

FPGAs were originally used to integrate logic circuits usually designed with TTL basic logic circuits; on which simple Computer-Aided Design (CAD) tools were used to develop these applications. These days, FPGAs are being used to implement more and more complex tasks. This progress can be attributed to the steady evolution of tools that were originally reserved to ASICs; these tools are mainly based on Hardware Description Languages (HDLs), being VHDL and Verilog the most prominently used.

HDLs are mainly used to describe hardware and are essentially used to describe concurrent tasks i.e. actions that occur at the time; this is a great difference when compared to high-level computer languages, which are used to describe algorithms that are sequentially executed. However, a common mistake made by someone new to HDLs is to attempt to program them as high-level software languages [52].

The existence of IEEE Standards for these HDLs [53]–[55] has allowed the creation of high-performance CAD tools in the field microelectronics [6]. Conventionally, the designers take advantage of these tools to build digital circuits following a hierarchical and modular approach, in which ideas are defined at different levels of abstraction. Then the design has to be simulated by creating test-benches and later validate it with synthesized HDL code.

However, the design of FPGA-based controllers is rather intuitive and require that the designer master several areas e.g. microelectronics, control, electrical machine theory. All of these skill requirements naturally ends leading many control engineers to choose standard, already proved platforms, such as DSPs [56]. In the recent years, a plethora of methodologies to reduce the complexity of the design approach [6], [57]–

[59] have been proposed. All of them are based on friendly development procedures considering that the control engineer is not proficient in microelectronics.

Moreover, FPGA vendors are redesigning their development tools work with high-level programming languages such as C/C++ and more recently OpenCL [60], [61], allowing the designers to describe hardware components using these high-level languages. The latest generations of FPGAs integrate powerful embedded processor units making them full system-on-chip (SoC) platforms; thus, their applications are no longer reduced to hardware acceleration. To take full advantage of these SoC solutions, the use of software/hardware co-design methodologies is crucial [58].

A recent popular approach is Model-Based Design (MBD) for FPGAs, which allows the designers to simulate and later translate their design within MATLAB-Simulink friendly environments. The results of these simulations are later accurately translated to the hardware implementation, allowing rapid prototyping schemes [62]. Nevertheless, in the case of complex algorithms, this solution still leads most of the time, to un-optimized architectures regarding consumed resources [6], [56].

In all the cases, the most restraining factor for the control of a power converter is the constraining of the switching frequency due to the commutation losses. Based on this idea, two groups can be identified: high demanding applications and constrained switching frequency applications [5]. In the first case, the timing constraints are so rigorous that the digital controller represents the main limitation of the whole control loop [63]. One example of this type of application, are power converters based on wide bandgap semiconductors devices like Silicon Carbide (SiC) and Gallium Nitrate (GaN) that are allowing power converters work at several hundreds of kilohertz and even the megahertz switching frequencies [64]–[67].

On the second case where the switching frequency is constrained, using FPGA is of great interest because the control processing time can be drastically reduced, up to a fraction of the sampling period. Thus the behavior of the digital controller is very close to its analog counterpart [68], [69]. With the extra remaining time, enhancing capabilities can be added to the control algorithm, such as predictive schemes [70] and control of very complex systems [70], [71], real-time fault detection [72], among many others.

1.4 Outline of the Chapters

The structure of this work is organized as follows: Chapter 1 contains the motivations, objectives, and a brief review of the state of the art of the theories and technologies in which this work lays on. Chapter 2 summarize the theoretical fundamentals used for the development of this work like the space phasor theory, the modelling of the PMSM, field-oriented control, and model-based predictive control among others.

Following, the chapter 3 describes the fundamentals of FPGAs and the necessary concepts to understand the analog-to-digital conversion process made by sigma-delta converters. In chapter 4 the concepts and ideas around the proposed predictive variable switching frequency control are presented. Finally, in chapter 5 experimental results used to validate the proposed control are presented and discussed and chapter 6 the conclusions of this work are discussed as well as the relevance of the obtained results.

2 Theoretical Fundamentals

Electrical machine users are interested in the way their mechanical quantities (torque, speed, power) behave but in order to understand them, the electrical quantities (currents, voltages) have to be described and as a result, the complete behavior of the machine can be studied. Therefore, this chapter presents some fundamental concepts that were used for the development of this work.

2.1 The Space Phasor

There are many methods to describe electrical machines for instance, the matrix calculation, which yields to a more difficult and less clear mathematical description. The space phasor is a simple and powerful method; additionally, it allows the final user to see the physical background of various phenomena [73]. Kovacs and Racz [74] introduced the space phasor theory in 1959, it was further developed by Stepina J. [75] and Serrano-Iribanegaray [76]–[78]. The study of electrical machines require the knowledge of these internal and external quantities, which are spatially distributed and space phasors happen to be very appropriate to describe them [78]. In general, the instantaneous values of the three-phase quantity *r* can be described in the complex space phasor $\underline{r}(t)$ form as follows:

$$\underline{r}(t) = \frac{2}{3}\left(r_U(t) + \underline{a} \cdot r_V(t) + \underline{a}^2 \cdot r_W(t)\right) \cdot e^{j\gamma_i} \tag{2.1}$$

$$\underline{a} = e^{j\frac{2\pi}{3}} = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \tag{2.2}$$

where: $r_U(t)$, $r_V(t)$ and $r_W(t)$ are the electric or magnetic instantaneous values of a three-phase system of the respective *U*, *V* and *W* phases and the spatial operator is given by (2.2). For stator quantities the angle $\gamma_i = \gamma_1$ and for rotor quantities $\gamma_i = \gamma_2$. Consequently, the stator current space phasor can be defined as:

$$\underline{i}_1 = \frac{2}{3}\left(i_U(t) + \underline{a} \cdot i_V(t) + \underline{a}^2 \cdot i_W(t)\right) \cdot e^{j\gamma_1} \tag{2.3}$$

The angle γ_1 defines the position of the axis to the first stator winding U with respect an arbitrarily defined reference frame; which is often selected as $\gamma_1 = 0$. The $2/3$ factor in (2.3), scales the length of the space phasor to the correspondent amplitude of the phase current in a symmetric sinusoidal system. Usually the windings of a three-phase alternating current machine are either connected in delta or star with isolated neutral point:

$$i_U + i_V + i_W = 0 \tag{2.4}$$

Therefore, it is possible to calculate the instantaneous values of the phase currents from the space phasor and the other way around, without losing information. Space phasors can also be represented in different coordinate systems; in the following, the required systems for this work will be described.

2.1.1 Fixed α,β-Coordinate System

The space phasor is a complex function with real and imaginary time-dependent parts and is referred to a given complex plane, whose orientation in space has to be defined. As previously mentioned, one of the most used coordinates systems is defined by pointing the real axis of the space phasor in the direction of the first stator winding, usually called U.

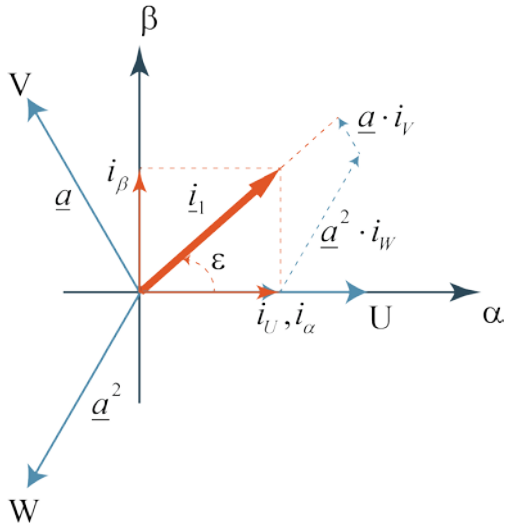

Figure 2.1: Graphical representation of the current space phasor in the stator fixed coordinate system.

This system, depicted in Figure 2.1 is known as stator-fixed coordinate system, the real and imaginary axes are named as α- and β-axis respectively:

$$\underline{i}_1 = \frac{2}{3}\left(i_U(t) + \underline{a}\cdot i_V(t) + \underline{a}^2\cdot i_W(t)\right)\cdot e^{j\gamma_1} = i_{1\alpha} + j\cdot i_{1\beta} = i_1(t)\cdot e^{j\varepsilon(t)};\ \text{with}\ \gamma_1 = 0 \quad (2.5)$$

$$i_{1\alpha} = \mathrm{Re}\{\underline{i}_1\} \quad (2.6)$$

$$i_{1\beta} = \mathrm{Im}\{\underline{i}_1\} \quad (2.7)$$

From (2.4) it is known that the sum of the phase values is zero, therefore it is possible to obtain the current components along the real and imaginary axes as:

$$i_{1\alpha} = i_U \quad (2.8)$$

$$i_{1\beta} = \frac{1}{\sqrt{3}}\left(i_V - i_W\right) \quad (2.9)$$

Similarly, the instantaneous values of the phase quantities related to the stator current space phasor can be obtained from the spatial projection of $\underline{i}_1$ on the three-phase axes:

$$i_{1U} = \mathrm{Re}\{\underline{i}_1\} = i_{1\alpha} \quad (2.10)$$

$$i_{1V} = \mathrm{Re}\left\{\underline{i}_1\cdot\underline{a}^{-1}\right\} = -\frac{1}{2}\cdot i_{1\alpha} + \frac{\sqrt{3}}{2}\cdot i_{1\beta} \quad (2.11)$$

$$i_{1W} = \mathrm{Re}\left\{\underline{i}_1\cdot\underline{a}^{-2}\right\} = -\frac{1}{2}\cdot i_{1\alpha} - \frac{\sqrt{3}}{2}\cdot i_{1\beta} \quad (2.12)$$

$i_{1\alpha}$ and $i_{1\beta}$ can be seen as the instantaneous currents in a similar two orthogonal phase windings that create the same resultant mmf-wave as the three-phase windings; from this perspective equations (2.10) to (2.12) denotes a three-phase to two-phase variable transformation, being the α-phase on the equivalent two-phase machine coincident with the *U*-phase of the three-phase machine. From this point on, the indices α and β will be used to design stator-related quantities based on the stator-fixed reference frame. Equations (2.5) to (2.12) are also valid for other electrical or magnetic quantities such as: voltage $\underline{u}$, flux linkages $\underline{\psi}$, magnetic flux density $\underline{B}$, many others.

2.1.2 Rotating *d,q*-Coordinate System

The rotating *d,q*-coordinate system becomes its name from the *direct* and *quadrature* axes. As shown in Figure 2.2, the current space phasor (2.3) referred to the stator-fixed α,β-coordinate system, can be transformed into a rotating *d,q*-coordinate system, where γ is the angle between the real α-axis of the fixed reference frame and the real d-axis of the rotating reference frame, which can vary in time as $\gamma(t)$.

In the case of the PMSM, which is used for the development of this work, the rotating *d,q*-coordinate system is aligned to permanent rotor flux. Considering the rotation of the system, the relations on equations (2.13) and (2.14), so that the rotation from the α,β- to the *d,q*-reference frames can be performed with (2.15) and inversely with (2.16).

$$i_{1\alpha} + j \cdot i_{1\beta} = \underline{i}_1 \cdot e^{j\varepsilon} \tag{2.13}$$

$$i_{1d} + j \cdot i_{1q} = \underline{i}_1 \cdot e^{j(\varepsilon-\gamma)} \tag{2.14}$$

$$\left(i_{1\alpha} + j \cdot i_{1\beta}\right) = \left(i_{1d} + j \cdot i_{1q}\right) \cdot e^{j\gamma} \tag{2.15}$$

$$\left(i_{1d} + j \cdot i_{1q}\right) = \left(i_{1\alpha} + j \cdot i_{1\beta}\right) \cdot e^{-j\gamma} \tag{2.16}$$

From now on, the indices *d* and *q* will be used to name stator-related quantities based on the rotational reference frame.

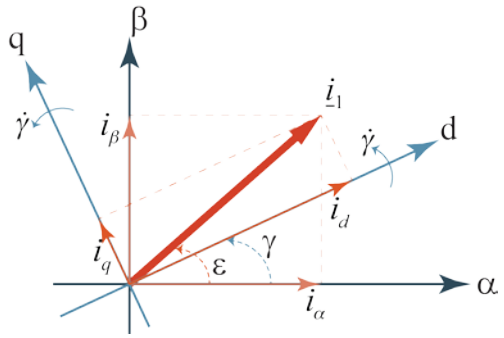

Figure 2.2 :Transformation of fixed α,β-coordinate system to the rotating *d,q*-coordinate system.

2.1.3 Coordinate Transformations

With the introduction of the current space phasor and the different reference systems, some transformations can be derived. As presented in Figure 2.1 and from

equations (2.5), (2.8) and (2.9) the transformation of a three-phase system to an orthogonal two-phase coordinate system can be described by the following expression:

$$\begin{bmatrix} r_\alpha \\ r_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 2 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \end{bmatrix} \cdot \begin{bmatrix} r_U \\ r_V \\ r_W \end{bmatrix} \tag{2.17}$$

Conversely, the inverse transformation can be derived from equations (2.10) - (2.12), as:

$$\begin{bmatrix} r_U \\ r_V \\ r_W \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} r_\alpha \\ r_\beta \end{bmatrix} \tag{2.18}$$

The transformations for the fixed to rational reference frames can be also derived, if the real and imaginary terms from (2.15) and (2.16) are compared the following matrix expressions are obtained:

$$\begin{bmatrix} r_d \\ r_q \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \cdot \begin{bmatrix} r_\alpha \\ r_\beta \end{bmatrix} \tag{2.19}$$

Being the inverse transformation:

$$\begin{bmatrix} r_\alpha \\ r_\beta \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \cdot \begin{bmatrix} r_d \\ r_q \end{bmatrix} \tag{2.20}$$

These coordinate transformations are also known as vector-rotations.

2.2 Permanent Magnets Synchronous Machine

The first permanent magnet (PM) excitation systems applied to electrical machines date back to the 19th century, however, the use of very poor magnetic materials discouraged their use, in favor of electromagnetic excitation systems. The discovery of the AlNiCo (an alloy of iron with aluminum, nickel, and cobalt) magnet in 1932 brought back to attention the PM excitation system. Nevertheless, at that time the performance

of the magnetic material was not enough and the development of power electronics was yet to come, so the applications of PMSM where limited [79].

The Permanent Magnets Synchronous Machine (PMSM) has gained a lot of interest over the years; mainly due to several advantages including high power density, high efficiency, and high reliability. They have a wide range of applications: like small disk drive units, high-performance motors for hybrid electric vehicles, to some megawatts machines used for oil and gas applications as well as wind turbines, and electric ship propulsion [79]–[81].

Like every rotating electrical machine, the PMSM is an energy converter as it can operate either as a generator or as a motor; it is composed of two parts: the stator and the rotor [82]. The stator is comprised of three identical windings symmetrically distributed in space (120º electric degrees between them). When the stator windings are current fed by a balanced three-phase alternating current (AC), a revolving field is generated along the air gap.

The rotation speed of the revolving field is proportional to the number of poles of the machine and the frequency of the stator currents. The rotor consists of permanent magnets, which replace the conventional electromagnetic field poles, thus eliminating the extra hardware like slip rings and brushes needed to excite the field windings [83]. For the development of this work a PMSM was considered, the construction and mathematical description will be further described in the following sections.

2.2.1 Permanent Magnets for Electrical Machines

During the decade of the 1950's materials that retain magnetism were introduced in electrical machine research. Materials that retain magnetism are known as *hard materials*. The ability to retain permanent magnetism is present in cobalt, iron, and nickel; which are also known as ferromagnetic materials. Several materials such as *AlNiCo*, ferrites (ceramic) such as barium ($BaO6Fe_2O_3$) and strontium ($SrP6Fe_2O_3$) ferrites, and rare-earth materials such as samarium-cobalt (*SmCo*) and neodymium-iron-boron (*NdFeB*) are available as permanent magnets for use in electrical machines [79], [83], [84].

Of these materials, *AlNiCo* has a high remanence flux yet small coercive field strength and can be quickly demagnetized. Therefore, this material is used since the end of the 1970's and has substituted the ferrite materials. The advantage of the ferrite

materials is their cost, as well as a large electrical resistance value, which reduces the development of Eddy-currents. Ferrite materials are mostly used in DC-actuators, the automobile industry and in fan motors.

Nevertheless, this material generates low torque density motors. *SmCo* has better magnetic characteristics; therefore machines with this material are smaller. Rotors with *SmCo* materials were used many years ago; they were later replaced by *NdFeB* magnets which are lower in cost but are sensitive to the temperature and prone to corrosion [49], [79], [85].

2.2.2 Permanent Magnets Rotor Types

The permanent magnets of the PMSM can be classified by the direction of the field flux. In a *radial field* PM the flux directions flows along the radius of the machine, this is the most prominent configuration. In an *axial field* PM, the direction of the flux is parallel to the rotor shaft, these machines are getting more use because of their high power density and maximum possible acceleration capabilities [83].

The magnets can be placed in many forms on the rotor; regardless of the way the permanent magnets are mounted the principle of operation of the machine is the same. However, it will impact on the values of the direct and quadrature axes inductances. In general, the permanent magnet rotors can be classified as *surface* and *interior* mounted, some examples can be found in Figure 2.3.

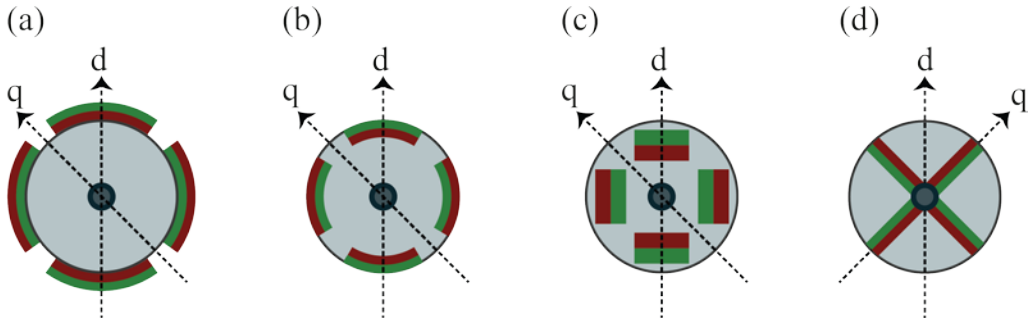

Figure 2.3: Permanent magnets rotor types: a) Surface. (SPM) b) Surface-inset (SIPM). c) Interior with radial orientation (IPM). d) Interior with circumferential orientation. All these rotors have two pairs of poles.

2.2.2.1 Surface mounted permanent magnets

This arrangement (shown in Figure 2.3a) provides the highest air-gap flux density as the air-gap is directly without the intermission of any other part of the rotor. Because of this, the structural integrity is not so robust and therefore this kind of machines is normally not driven faster than 3000 min^{-1}. The reluctance between the *d,q*-axes is fairly

small, (less than 10%) which allows the simplification of the model for this kind of machines.

2.2.2.2 Interior permanent magnets

This kind of rotors can be further subdivided, depending on how deep are the permanent magnets buried in the rotor laminations. The interior-inset variant (as in Figure 2.3b) has the permanent magnets placed in the grooves of the outer periphery of the rotor laminations, providing a uniformly cylindrical surface on the rotor; this arrangement is much more mechanically robust than the surface mounted variant, which is usually called inset rotor.

Finally as shown in Figure 2.3c-d, the permanent magnets are placed in the middle of the rotor, they are called interior PMSM. This type of machines is the most mechanically robust and therefore suitable for high-speed applications. Their construction process is much more complex and the ratio between the direct and quadrature inductances is also the largest.

2.2.3 Modeling of the PMSM

The behavior of the PMSM can be derived from the equations that describe the electrically excited synchronous machine (EESM). However, in the former, the excitation is provided by PMs on the rotor as opposed to the later, where a winding has to be excited with a DC current and a damping cage also part of the system making the mathematical model to be more complex. With the introduction of the space phasor, the current, voltage and flux phasor of the machine can be defined.

Therefore, the equations of the PMSM mathematical model can be derived from the EESM in the *d,q*-reference frame [22], [49], [85], by assuming that there is no damping cage and the permanent flux in the d-axis of the rotor is constant ψ_{d0}. The simplified structure of the windings is shown in Figure 2.4 for a two-pole model. In addition, the machine is considered to be magnetically symmetric, thus $L_d = L_q = L_1$:

$$\psi_d = L_1 \cdot i_d + \psi_{d0} \tag{2.21}$$

$$\psi_q = L_1 \cdot i_q \tag{2.22}$$

$$u_d = i_d \cdot R_1 + \frac{d\psi_d}{dt} - \omega\psi_q \tag{2.23}$$

$$u_q = i_q \cdot R_1 + \frac{d\psi_q}{dt} + \omega\psi_d \tag{2.24}$$

By replacing the flux equations in the voltage equations of the PMSM follows:

$$u_d = i_d \cdot R_1 + L_1 \cdot \frac{di_d}{dt} - \omega \cdot L_1 \cdot i_q \tag{2.25}$$

$$u_q = i_q \cdot R_1 + L_1 \cdot \frac{di_q}{dt} - \omega \cdot L_1 \cdot i_d + \omega \cdot \psi_{d0} \tag{2.26}$$

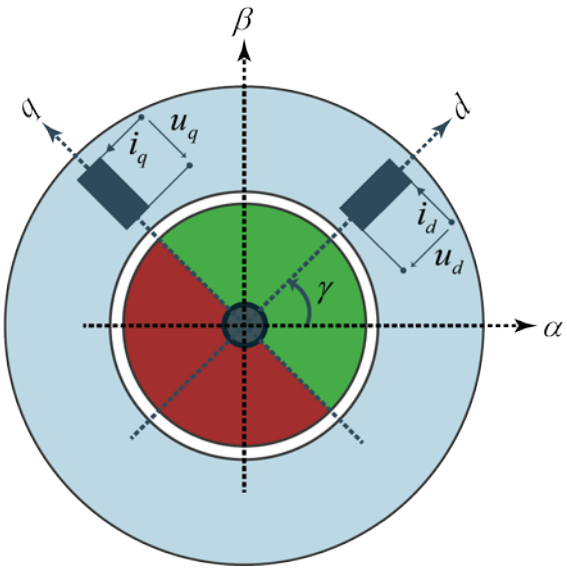

Figure 2.4: Model of a PMSM with two poles.

In addition to the electrical and magnetic equations, some mechanical quantities have to be mathematically described i.e. the rotor position and the angular speed are described so the system can be fully described. As presented in Figure 2.4, the angle between the α-axis and the d-axis is denoted by γ, (this coordinate transformation is explained in 2.1.3). For a machine with p-pair of poles, the mechanical angle is defined by:

$$\gamma = p \cdot \gamma_{mech} \tag{2.27}$$

The electrical and mechanical angular velocities are described by:

$$\frac{d\gamma}{dt} = \dot{\gamma} = \omega \tag{2.28}$$

$$\frac{d\gamma_{mech}}{dt} = \dot{\gamma}_{mech} = \omega_{mech} \tag{2.29}$$

$$\dot{\gamma}_{mech} = \frac{\dot{\gamma}}{p} \text{ respectively } \omega_{mech} = \frac{\omega}{p} \tag{2.30}$$

The electromagnetic torque can be expressed as in (2.31) [73], which in *d,q*-components is described as in (2.32):

$$M_e = \frac{3p}{2} \cdot \mathrm{Im}\left\{\underline{\psi}_1^* \cdot \underline{i}_1\right\} \tag{2.31}$$

$$M_e = \frac{3p}{2} \cdot \mathrm{Im}\left\{\psi_d \cdot i_q - \psi_q \cdot i_d\right\} \tag{2.32}$$

If the flux equations are introduced in the electromagnetic (M_e) torque equation ψ_{d0} is considered to be constant then M_e is proportional to i_q:

$$M_e = \frac{3p}{2} \cdot \left(L_1 \cdot i_d \cdot i_q + \psi_{d0} \cdot i_q - L_1 \cdot i_d \cdot i_q\right) = \frac{3p}{2} \cdot \psi_{d0} \cdot i_q \tag{2.33}$$

The mechanical equation (2.33) completes the mathematical modeling of the PMSM; where M_L defines the load torque and J the total moment of inertia of the driven masses as reflected to the shaft of the motor:

$$M_e - M_L = J \cdot \frac{d\omega_{mech}}{dt} = J \cdot \ddot{\gamma}_{mech} \tag{2.34}$$

2.3 Voltage Source Inverter

A power converter in its most simple expression is an array of on-off power electronic semiconductor switches that is able to convert and control electrical power. Of course, power converters are not ideal and these devices will introduce nonlinearities i.e. harmonics and electromagnetic interference (EMI) problems. A DC-to-AC power converter is known as an *inverter*. Thus, the function of an inverter is to convert a DC input voltage to a symmetric AC output voltage with a desired magnitude and frequency. If an inverter has a constant input voltage it is called Voltage Source Inverter (VSI) [86], [87].

The basic configuration of the power converter used to feed the PMSM is presented in Figure 2.5; it comprises a three-phase diode rectifier, a capacitive filter in DC-link and a three-phase VSI. The inverter is made of six Insulated Gate Bipolar Transistors

(IGBTs) (T_1 – T_6) and six free-wheeling diodes (D_1 – D_6). With this kind of VSI topology, it is possible to get eight different states of the switches in the inverter i.e. eight different space voltage phasors.

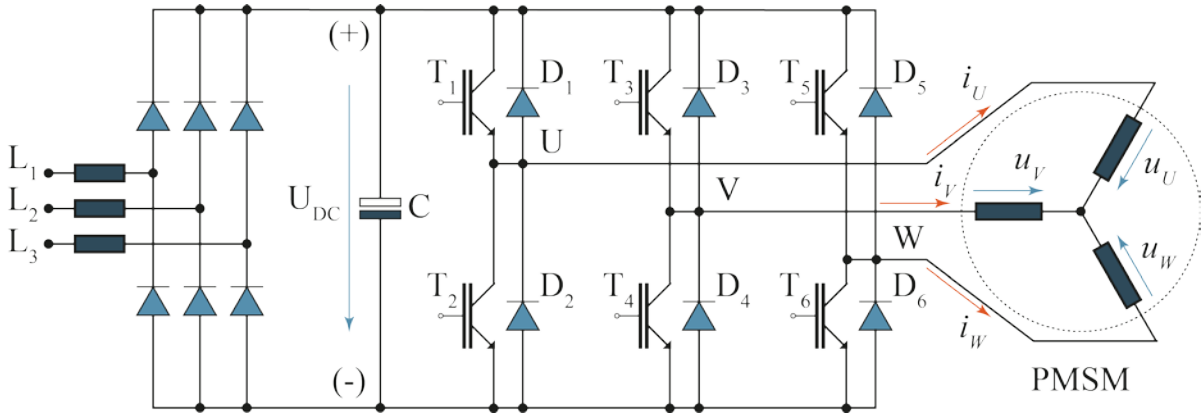

Figure 2.5: Three-phase power converter connected to the stator of the PMSM.

2.3.1 Generation of the Voltage Space Phasors

The switches on the VSI are defined as (S_U, S_V, S_W) where the value "1" defines the switch set to the positive voltage and "0" when the switch is set to the negative voltage (the simplified diagram can be seen in Figure 2.6a; the eight different switching configurations generate a the voltage space phasors: $\underline{u}_0, \underline{u}_1, \underline{u}_2, \underline{u}_3, \underline{u}_4, \underline{u}_5, \underline{u}_6, \underline{u}_7$. The two space phasors $\underline{u}_0, \underline{u}_7$ are the zero voltage space phasors (ZV) with the all the phases connected to the negative potential or all connected to the positive potential, respectively; the rest $\underline{u}_1 .. \underline{u}_6$ are called active voltage space phasors (AV), as can be seen in Figure 2.6b.

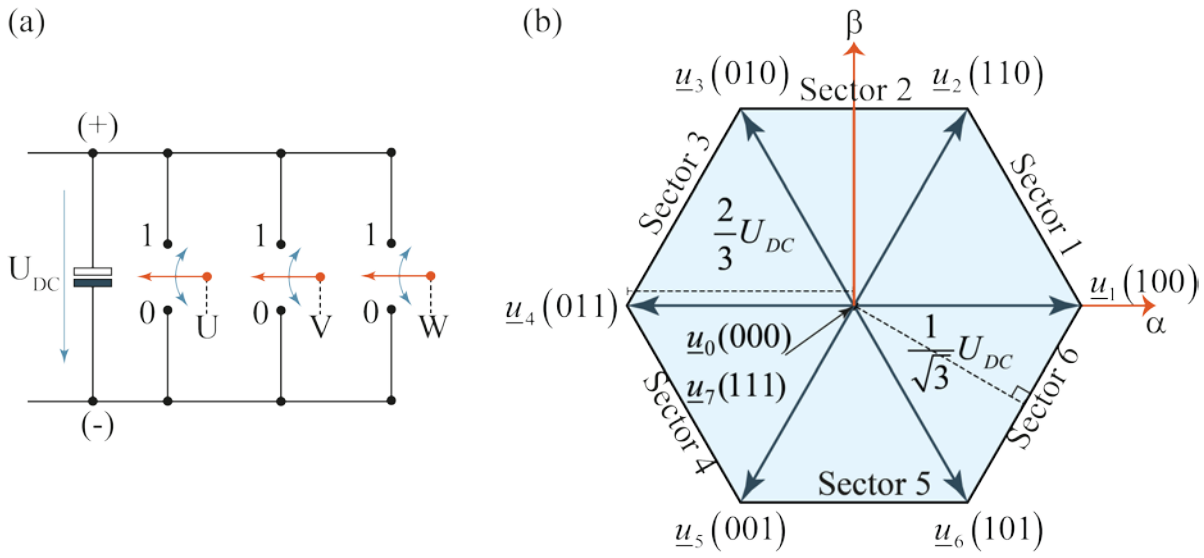

Figure 2.6: a) Simplified switching scheme of a two-level VSI. b) Voltage space phasors derived from the switching states of a two-level VSI.

The voltage space phasors divide the plane into six sectors every 60º or $\pi/3$ radians, as presented in in Figure 2.6b, which can be mathematically described as in equation (2.35). Finally, the components in the α,β-reference frame of the eight available space voltage phasors are presented in Table 2.1.

$$\underline{u}_1 = \underline{u}_\upsilon = \begin{cases} \frac{2}{3} \cdot U_{DC} \cdot e^{j(\upsilon-1)\frac{\pi}{3}} & \upsilon = 1,\ldots 6 \\ 0 & \upsilon = 0,7 \end{cases} \tag{2.35}$$

Table 2.1: Voltage space phasors in the α,β-reference frame

Voltage space phasor	$\underline{u}_0$	$\underline{u}_1$	$\underline{u}_2$	$\underline{u}_3$	$\underline{u}_4$	$\underline{u}_5$	$\underline{u}_6$	$\underline{u}_7$
Switching state	000	100	110	010	011	001	101	111
$\frac{\underline{u}_\alpha}{U_{DC}}$	0	$\frac{2}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	0
$\frac{\underline{u}_\beta}{U_{DC}}$	0	0	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	0	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	0

2.4 Space Phasor Modulation

This technique arises from the idea that inverters are only capable of generating a finite number of voltage space phasors, as presented in section 2.3, the two-level VSI is capable of generating eight. Nevertheless, if the stator voltage space phasor to be applied differs from the natural voltage space phasors that the inverter can generate, then different combinations of the available voltage space phasors have to be applied in order to create the desired output; this process is also known as modulation.

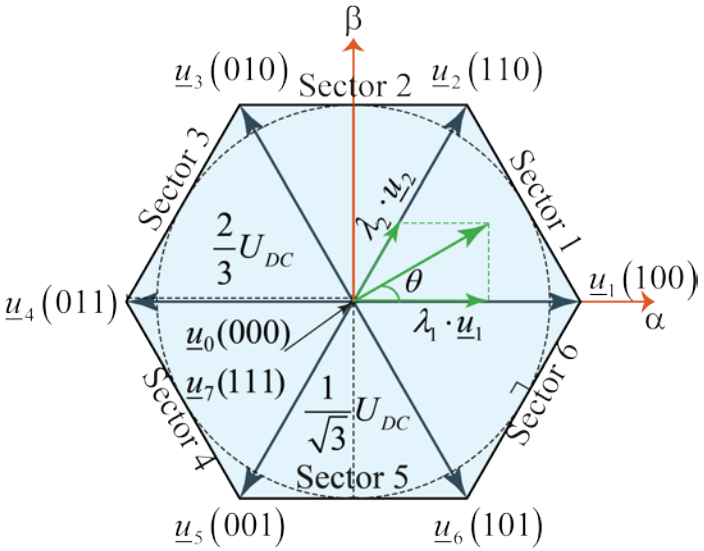

Figure 2.7: Example of the generation of the voltage space phasor $\underline{u}_S$ by using SPM.

One of the most well-known modulation techniques is the space phasor modulation (SPM) [88] technique (also referred as space vector modulation SVM), which relies on the space phasor theory. The SPM technique generates the voltage space phasor $\underline{u}_S$, which can be located inside one of the sectors of the hexagon voltage-limits of the VSI (Figure 2.6b), using the three nearest voltage space phasors (two active and one zero voltage space phasors).

Figure 2.7 shows as an example the voltage space phasor $\underline{u}_S$ located in the first sector, in order to create the desired voltage space phasor. During one modulation period the active voltage space phasors $\underline{u}_1$ and $\underline{u}_2$ should be applied the respective λ_1 and λ_2 intervals of time, in which the space phasors are active, as well as the zero voltage space phasors $\underline{u}_0$ and $\underline{u}_7$ for the rest of the modulation period.

λ_1 and λ_2 are the duty cycles of the active space phasors and correspond to the length of the projections of the desired space phasors on the direction of the active space phasors. Thus, the voltage space phasor $\underline{u}_S$ can be calculated by the following expressions:

$$\underline{u}_S = u_S \cdot e^{j\theta} = \lambda_1 \cdot \underline{u}_1 + \lambda_2 \cdot \underline{u}_2 \tag{2.36}$$

$$\lambda_k = \frac{T_k}{T_S} \quad k = 0..7 \tag{2.37}$$

Where T_S is the duration of one modulation period and λ_k is the power-on time within one T_S or the duty cycle of the respective voltage space phasor i.e. switching cycle *i*. The active duty-cycles λ_k for the first sector can be calculated as follows:

$$\lambda_1 = \frac{T_1}{T_S} = \frac{2}{\sqrt{3}} \cdot \frac{u_S}{u_{\max}} \cdot \sin\left(\frac{\pi}{3} - \theta\right) \tag{2.38}$$

$$\lambda_2 = \frac{T_2}{T_S} = \frac{2}{\sqrt{3}} \cdot \frac{u_S}{u_{\max}} \cdot \sin(\theta) \tag{2.39}$$

$$\lambda_{0,7} = 1 - \lambda_1 - \lambda_2 \tag{2.40}$$

Where: $0 \leq \theta \leq \pi/3$.

During the rest of the modulation period $T_{0,7} = \lambda_{0,7} \cdot T_S$ one of the two zero space phasors $\underline{u}_0$ and $\underline{u}_7$ is applied. From Figure 2.7 it can be seen that the maximum amplitude

of the voltage space phasors is $u_{max} = 2U_{DC}/3$; nonetheless, for some practical cases e.g. reduction of harmonics in the output voltage, only the inner circle in the hexagon limit is used which corresponds to $u_{max} = U_{DC}/\sqrt{3}$ [89]. Following this, the sequence in which the voltage space phasors are applied has to be determined.

From the switching losses point of view, it can be noticed that the most favorable sequence is to change the state of every pair of switches only once every switching period. Figure 2.8 presents the switching patterns of the voltage space phasors in the first sector, a conventional implementation of SPM applies a zero space phasor at the beginning and end of the switching period. This will reduce the current ripple, reducing the torque ripple and magnetic losses associated with ripple currents in the machine [83]. In this figure, it can also be noted that the switching period is the double of the modulation period T_S, a similar procedure can be followed for the remaining sectors.

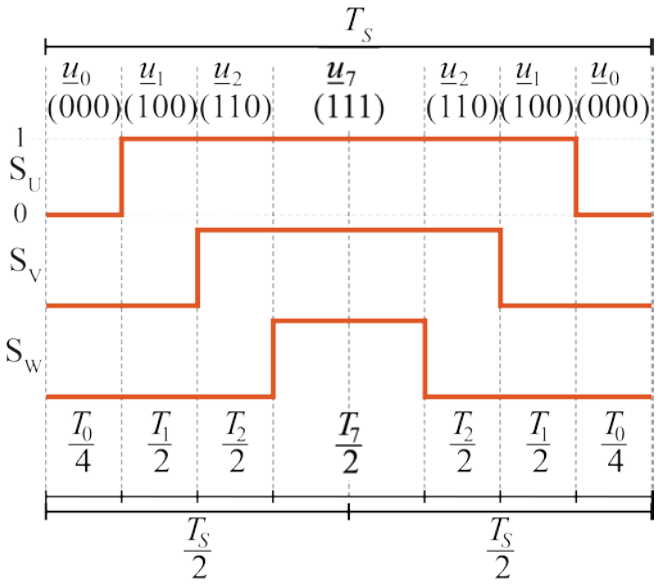

Figure 2.8: Switching patterns of the voltage space phasors in the first sector.

2.5 Field Oriented Control Principle

Field Oriented Control (FOC) was developed and first presented by Hasse and Blaschke [7], [90], [91], the idea behind this control technique is inspired by the DC control schemes, where the flux and torque are decoupled and therefore the mathematical modeling is simplified. Thus, by orienting or aligning the reference frame to a rotating *d,q*-coordinate system, the decoupling of the flux and torque forming components of the variables in the AC machine is achieved (see Figure 2.9) [89].

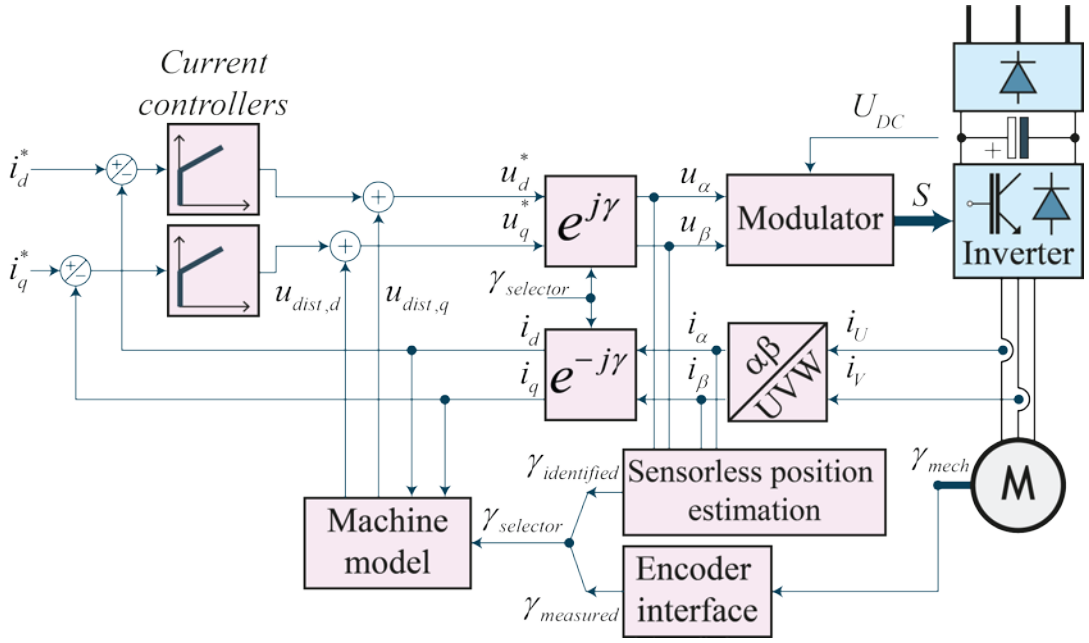

Figure 2.9: Fundamental blocks of the field-oriented control scheme.

If the equation for the i_d and i_q currents were absolutely decoupled, like in the case of the DC machine for the armature and field current, this idea would perfectly work with two separate PI current controllers. However if the voltage equations of the machine are carefully analyzed, in particular for the PMSM presented in (2.25) and (2.26),it can be noticed that a coupling between these equations exists. The stator voltage in each axis is coupled through the rotational voltage of its orthogonal axis, which can be better understood if the equations are rewritten as follows:

$$u_d = i_d \cdot R_1 + L_1 \cdot \frac{di_d}{dt} - \omega \cdot L_1 \cdot i_q = i_d \cdot R_1 + L_1 \cdot \frac{di_d}{dt} + u_q^{dist} \tag{2.41}$$

$$u_q = i_q \cdot R_1 + L_1 \cdot \frac{di_q}{dt} - \omega \cdot L_1 \cdot i_d + \omega \cdot \psi_{d0} = i_q \cdot R_1 + L_1 \cdot \frac{di_q}{dt} + u_d^{dist} \tag{2.42}$$

Therefore, these u_d^{dist} and u_q^{dist} can be considered as disturbances quantities by the controllers, which are calculated using a model of the machine and added to the output of the current controllers. The reference voltages u_d^* and u_q^* are then transformed to the stator-fixed α,β-reference frame, which are in turn used by the modulator to generate the desired switching signals for the inverter. The modulator implements the SPM technique presented in 2.4 or some other kind of pulse-width modulation technique.

The reference i_q^* can be obtained from a super-imposed speed controller or a direct desired torque reference. The i_q^* reference value is generally set to zero for the PMSM

therefore no additional flux controller is needed; a different situation could arise if the machine shall operate in the field-weakening region. In order to achieve this, a negative current should be fed in the *d*-axis, depending on the desired speed; mainly because the modern PM are nearly impossible to be demagnetized [89].

It can be concluded that a good dynamic performance of the current controllers will derive in a good dynamic response of the electromagnetic torque as well as the magnetic flux being kept around the desired reference value.

2.6 Model–based Predictive Control Principle

A wide range of predictive control methods have found applications in power electronics. All these methods share a same idea, the use of a model of the controlled system for the prediction of the future actions of the controlled variables. With this information and a predefined optimization criterion, the controller will choose the optimal action, in the case of model-based predictive control (MPC) the optimization criterions is expressed as a cost function to be minimized [30].

For power electronics the variations of MPC are classified depending on the use of a modulation stage or not. In the first case, a continuous set of solutions (CS-MPC) [35] resulting in a fixed switching frequency. The second case, generates the switching signals for the power converter directly coming from the natural voltage space phasors that the inverter is able to generate, which delivers a finite set of solutions (FS-MPC) [45] and presents a variable switching frequency. The MPC control strategy can be summarized as follows:

- Define a model of the system to be controlled.
- Define a *quality or cost function*.

Conventionally MPC are used in discrete-time, with a fixed sampling interval T_S, where the inputs to the system are limited to change only at the discrete sampling instants, i.e. $t = kT_S$ where $k\epsilon\{0,1,\dots\}$ indicates the sampling instants. Then $\underline{u}(k)$ can be defined as the output of the controller, $\underline{y}(k)$ as the output of the controlled system and $\underline{w}(k)$ as the desired or reference value.

The *cost function* is a crucial part of MPC, in this function the predicted system behavior obtained from the model can be penalized e. g. difference between current reference and predicted values. The cost function is minimized at every instant *k*, over

a finite horizon of length N and the controller calculates a *future* output sequence following this optimization criterion. Thus, the output of the control contains N elements that can be used to control the system.

However, in conventional implementations such as FS-MPC, only the first element is used, i.e. only the $k+1$ solution is applied. At the next sampling step, i.e. $k+1$ the whole process is repeated using the latest measured information. The horizon is shifted one step and the optimization process is carried out again. This is called the *receding horizon*, compensating for future disturbances or modeling errors and is presented in Figure 2.10.

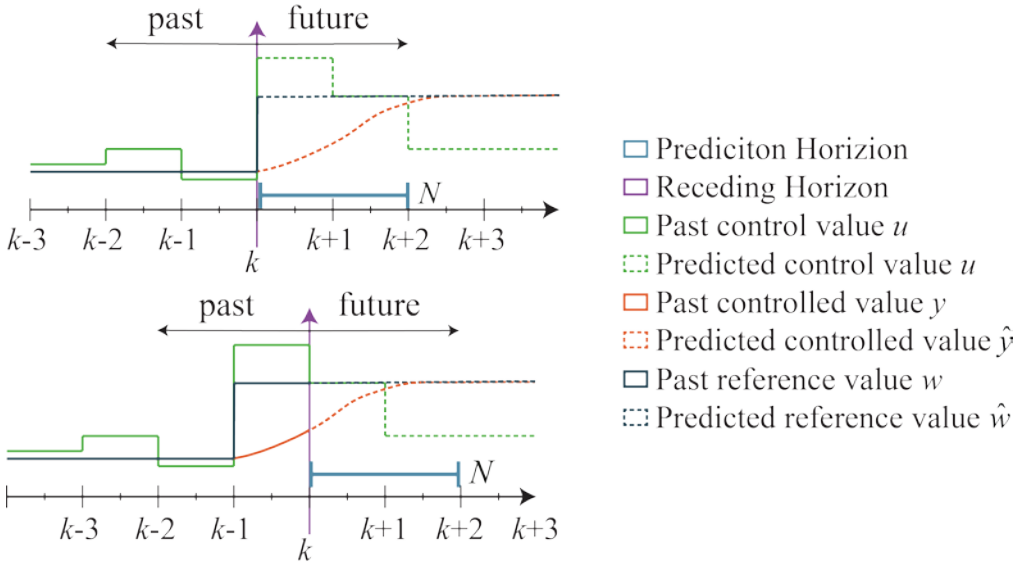

Figure 2.10: Receding prediction of horizon, $N = 2$.

Given that the predicted system output $\hat{y}(k+1)$ calculated at the instant time t is not equal to the real system output $y(k)$ measured at the instant time $t+1$ because of these nonlinearities and modeling errors, it is better to start the new prediction from a the measured system output rather than from the predicted process output at the previous sample time. In this sense, MPC introduces a feedback mechanism propagating forward in time the past states of the controlled system [92].

2.6.1 Defining the Cost Function

The cost function definition is a crucial stage in the design of a model-based predictive controller, since it allows the selection of the control objectives of the application as well as the required constraints. For power electronic applications the cost function g can be described [43] as follows:

$$g = \sum_{i=1}^{n} w_i \cdot \left\| e_i \right\| \tag{2.43}$$

$$\left\| e_i \right\| = \left\| x_i^* - x_i^P \right\| \tag{2.44}$$

Here e_i is the error between the reference (denoted by *) and the predicted (denoted by the superscript ^{P}) value of the variable x_i, n is the number of variables considered in the cost function and the operator $\|\cdot\|$ indicates the operation applied to the error, being the most usual absolute or square value. w_i is a weight that defines the relevance of the variable x_i inside the cost function, its range of values i. e. $w_i \in [0,1]$ where values close to 1 denote that the variable x_i has a higher influence and values close to 0 that it has a low importance in the cost function.

Given that the cost function combines variables with different physical meanings their ranges can be different therefore, it is important to normalize these ranges so all the x_i variables are a non-dimensional per-unit quantities:

$$e_{iN} = \frac{x_i^* - x_i^P}{x_i^N} = \frac{e_i}{x_i^N} \tag{2.45}$$

Thus, the error is redefined as a per-unit error, where each error e_i is divided by the base value x_i^N. If the whole prediction of the variable x_i during the sampling time T_S would like to be considered and not only the final value at t_{k+1} an integral cost function could be defined as:

$$g = \frac{1}{T_S} \int_{}^{T_S} w_i \cdot e_{iN} dt \tag{2.46}$$

This could result in a more accurate tracking of the reference, although MPC implementations usually have fast sampling periods, the two cost functions (2.43) and (2.45) have no noticeable difference on their performances [43]. Until now, most of the cost functions for power electronic applications have a prediction of horizon of one step. This is mainly because of the *receding horizon* principle, which states that the predictive control is adjusted, corrected and controlled on each instant; another reason is that a larger prediction increases the computational requirements.

As previously stated, not only the control variables can be considered in the cost function; other constraints can also be included. In the case of power electronics these

constraints could be the physical limits of the controlled system e. g. maximum current and voltage values of the power converter and the electrical machine. Furthermore, this allows the designer to set a compromise between the control effort and how well the reference is followed. A general form of a cost function, which includes constrains is:

$$g = \sum_{i=1}^{n} w_i \cdot \left\| e_{iN} \right\| + \sum_{j=1}^{n_C} f_C\left(j^P\right) \tag{2.47}$$

Where: j is any constraint, n_C is the number of constraints and the function f_{pen} that penalizes this constrains is defined by:

$$f_{pen}\left(j^P\right) = \begin{cases} 0, & \left|j^P\right| < j_{\max} \\ \infty, & \left|j^P\right| \geq j_{\max} \end{cases} \tag{2.48}$$

Consequently, in the case of power electronics a very important constraint is the current limit, so that (2.47) and (2.48) can be rewritten in terms of the predicted stator current i_1^P as:

$$g = \sum_{i=1}^{n} w_i \cdot \left| e_{iN} \right| + f_{pen}\left(i_1^P\right) \tag{2.49}$$

$$f_{pen}\left(i_1^P\right) = \begin{cases} 0, & \left|i_1^P\right| < i_{\max} \\ \infty, & \left|i_1^P\right| \geq i_{\max} \end{cases} \tag{2.50}$$

2.6.2 Optimization of the Controller Output

In model-based predictive control schemes a cost function over a finite horizon length *N* for given states of the controlled system, which can be measured or estimated is minimized. For this purpose, the values of the predicted outputs are calculated, as function of past values of inputs and outputs and of future control signals by evaluating the chosen model and replacing these results in the cost function. The result is an expression whose minimization leads to the optimal output controller values [36]. Thus, the optimal control signal under the constrained conditions of the cost function *g*:

$$\underline{u}_{optimal} = \min_{\underline{u}} g \tag{2.51}$$

Hence, the optimal voltage space phasor will be the one with the minimum *g* value, in this case the minimum error between the desired and the predicted currents.

2.6.3 Finite-Set MPC Current Controller

Figure 2.11 shows the basic diagram of a Finite-Set MPC current control scheme, the prediction of horizon is of length one (FS-MPC is also known as Direct MPC DMPC). If the system variables are defined as: $\underline{w}(k) = \underline{i}^*(k)$, $\underline{y}(k) = \underline{i}(k)$ and $\underline{\hat{y}}(k) = \underline{i}^P(k)$. The cost function has still to be defined; it is here where the strength and simplicity of MPC relies, because different control criteria can be combined with disturbances on the model among other aspects that will improve the performance of the controller. In this case, the current error for the next period can be expressed in *d,q*-coordinates as follows:

$$g = \left|i_d^*(k+1) - i_d^P(k+1)\right| + \left|i_q^*(k+1) - i_q^P(k+1)\right| \tag{2.52}$$

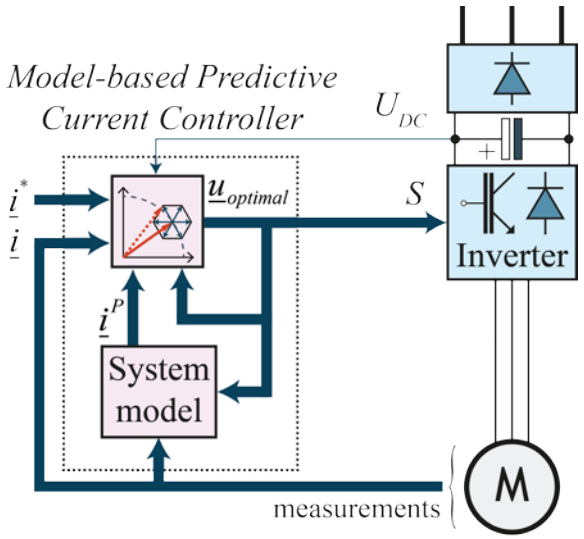

Figure 2.11: Model-based predictive current control scheme.

Here $i_d^P(k+1)$ and $i_q^P(k+1)$ are the real and imaginary parts of the predicted current space phasor $\underline{i}_1(k+1)$ and $i_d^*(k+1)$ and $i_q^*(k+1)$ are the respective desired parts of the current space phasor reference $\underline{i}_1^*(k+1)$. As the future value of the reference has to be known, one possible solution is to estimate the future reference value by extrapolation as in [45].

However, for simplification purposes, it can is assumed that the reference does not significantly change within one sampling time, hence $i_1^*(k+1) \approx i_1^*(k)$. If the derivative di/dt is replaced by a forward Euler approximation as in (2.53) and in turn substituted in the voltage equations of the PMSM (2.25) and (2.26), we can obtain the following equations to predict the currents:

$$\frac{di}{dt} \approx \frac{\Delta i}{\Delta t} \approx \frac{i(k+1)-i(k)}{T_S} \tag{2.53}$$

$$i_d^p(k+1) = i_d(k) + \frac{T_S}{L_1}\left[-i_d(k)\cdot R_1 + \omega \cdot L_1 \cdot i_q(k) + u_d^{\lambda}(k+1)\right] \tag{2.54}$$

$$i_q^p(k+1) = i_q(k) + \frac{T_S}{L_1}\left[-i_q(k)\cdot R_1 - \omega \cdot \left(i_d(k)\cdot L_1 + \psi_{d0}\right) + u_q^{\lambda}(k+1)\right] \tag{2.55}$$

Where u_d^{λ} and u_q^{λ} are the eight possible configurations of the two-level VSI (as presented in table 2.1) and transformed to the *d,q*-reference frame with (2.19) and the measured or estimated rotor position γ. These voltage space phasors are evaluated in (2.54) and (2.55), so different values can be assigned to the cost function *g*. Once this is achieved the minimization process can be performed, being the smallest value the optimal control output.

2.7 Summary of the Chapter

This chapter introduces the space phasor concept, which is necessary to understand and model the inner and outer quantities of three-phase AC machines. Then the fixed and rotating reference frames are explained, along with the necessary equations to transform between these coordinate systems. Some concepts about the PMSM and the permanent magnets used to build it are presented, followed by the types of rotors that can be found.

The mathematical model of the PMSM in a rotating reference frame is also introduced. Subsequent, the concept of voltage source inverter and the idea on how to generate a desired voltage space phasor using the space phasor modulation technique are presented. Later, the basics around field-oriented control for PMSM are presented. Finally, the concepts of model-based predictive control are reviewed.

3 FPGA-Based Controller Design for PMSM Drives

Field Programmable Gate Arrays (FPGAs) have been historically used to create custom hardware logic circuits (mostly I/O functions) that cannot be implemented in conventional microcontrollers or digital signal processors. In the recent years, FPGA have greatly improved due to the steady progress in the design and manufacturing of electronic components. More recently, many types of FPGAs are built whit soft/hard microprocessors cores.

As a result, modern FPGAs can be considered as true System-on-Chip (SoC) digital platforms, where designers have the possibility to integrate one or several processors with dedicated peripherals and hardware accelerators. This chapter presents the basics and digital signal processing of FPGAs, as well as FPGA-based design of controllers in conjunction with Analog-to-Digital (A/D) Sigma-Delta conversion.

3.1 FPGA General Structure

FPGAs are part of a wide family of programmable logic components, actually they belong to the so called semi-custom Application Specific Integrated Circuits (ASICs) [6]. FPGAs consist of predefined elementary cells and interconnection networks that can be reprogrammed by the end-user, thus allowing a rapid-prototyping process and making the design process more flexible and affordable [5]. FPGAs where first introduced to the market in 1985 by Xilinx [93] and since the technology has sustained a steady development process to the today commercially available systems.

The FPGA-technology is now being considered by a growing number of designers in numerous fields of application such as telecommunications, image and signal processing, medical equipment, robotics, automotive and embedded control systems among many others. The common denominator is the always increasing demand toward data processing capabilities and data throughput that can be achieved thanks to the features and capabilities of FPGA chips. More recently, industrial electrical control systems are also getting more attention in this matter because of the ever increasing level of expected performance while at the same time the cost of the control system has to be reduced [6].

The fundamental structure of an FPGA is depicted in Figure 3.1; it consists of logic blocks, configurable I/O blocks, memory blocks and an interconnection network. Logic blocks are the basic cell in the FPGA, they can be configured to perform combinatorial or sequential operations and for that purposes they include sets of Lookup Tables (LUTs) and sets of D-Flip-Flops. The internal structure of the logical cell differs from an FPGA family to another.

The interconnection network can also be programmed by the end-user, to join as many logical blocks as needed. The configurable I/O blocks are the interface between the elements inside the FPGA and the external devices. Finally, the memory blocks (different kinds of RAM and ROM) are also integrated, so local storage can be performed and dedicated logic circuits can be built.

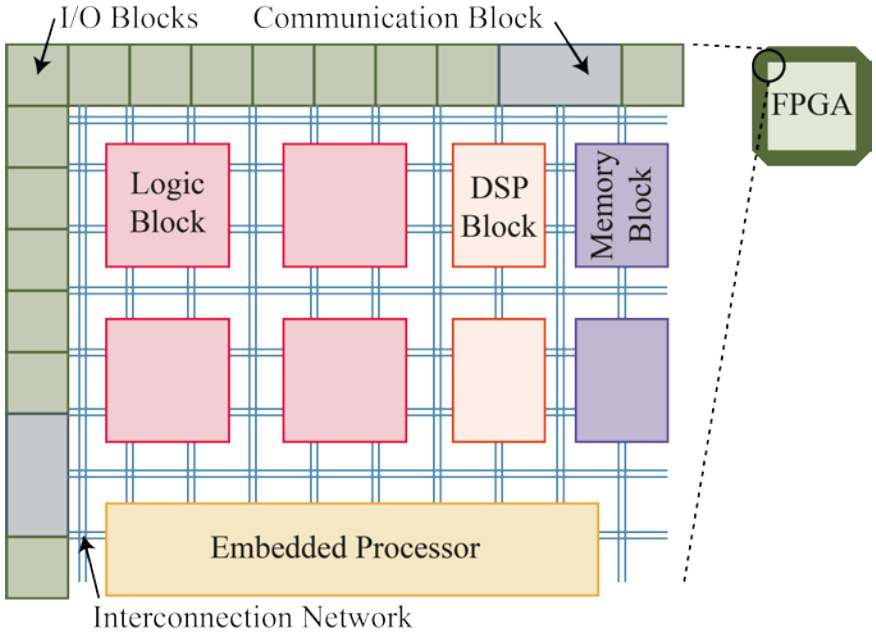

Figure 3.1: FPGA general structure.

Because of the increasing level of integration, these days recent FPGAs include hardwired digital signal processing (DSP), clock manager and communication blocks. The DSP blocks usually contain adders, multipliers and accumulators which can realize complex computations without needing them to be implemented with logic cells. The clock managers are usually Phase-Locked-Loops (PLLs) that can fractionally multiply or divide the source frequency, correct phase shifts and compensate propagation delays. The communication blocks, normally consist of transmission and reception buffers for several communication protocols such as Ethernet, USB, PCI, SPI and I2C protocols.

Among the most popular configuration technologies are SRAM and Flash based technologies. The most recent FPGAs can integrate one or several hard processors or

have the capacity to implement several soft processor cores, as well as several analog interfaces. The hard processor or non-synthesizable cores have a custom layout in dedicated silicon, which is integrated with the FPGA; they are generally faster but less flexible.

The soft or synthesizable core processors use logic cells to implement the processor. They are flexible to configure and several of them can be implemented; however, they have slower clock rates. For all of these, FPGAs are considered as true System-on-Chip (SoC) or System-on-Programmable-Chips (SoPCs) digital platforms [5].

3.2 Digital Signal Processing using FPGAs

Signal processing has been used to transform and shape analog and digital signals. Digital signal processing (DSP) has found many engineering applications such as communications, industrial instrumentations, biomedical processing and robotics. DSP has become a mature technology, replacing analog signal processing because of its insensitivity to change in temperature, aging, and hardware upgrades [94].

FPGAs feature the possibility to distribute algorithms and process signals in parallel, making them the perfect platform to implement signal processing algorithms. Understanding how digital signal processing using FPGAs is done, is crucial for the development of a high-performance controller. In the following, several concepts regarding digital signal processing will be explained.

3.2.1 Advantages of using FPGAs

Some of the reasons why FPGAs have outperformed ASICs seems to be related to the fact that FPGAs include the advantages of ASICs without including their disadvantages. A summary of the features can be found in Table 3.1 [94]. As already stated, one of the main advantages of FPGAs over conventional digital signal processors (DSPs) in digital signal processing is their ability to exploit parallelism, i.e. similar hardware functions can be concurrently executed in different parts of the chip [95].

Table 3.1 Advantages of FPGAs against ASICs

Advantages	Disadvantages
Size, weight and power dissipation reduction	Larger development times
Higher throughput	Non-reprogrammable
Better copy protection	
Reduced test board cost	
Reduced device and inventory costs	
Rapid prototyping	
Reprogrammable	

As an example, the implementation of a Finite Impulse Response (FIR) filter is presented in Figure 3.2. While the DSP requires *n* cycles to calculate one output sample, the FPGA can compute the data in parallel and output a new sample in only one clock cycle. In a low cost FPGA the maximum clock frequency, usually ranges the 100 MHz, meanwhile a multicore DSP run at frequency up to 2 GHz, thus the clock difference factor usually lays around 10 ~ 20, as presented in (3.1).

$$ClockDifferenceFactor = \frac{clk_{DSP}}{clk_{FPGA}} = \frac{2GHz}{100MHz} = 20 \tag{3.1}$$

Nevertheless, the parallel processing capabilities of FPGAs outperform the high clock speeds of DSPs, as it has been shown in several studies like in [96], where the FPGA can provide a greater than 10 times speedup performance with a 2 times smaller power consumption than of the DSP. Thus, DSPs are no match for FPGAs in digital signal processing.

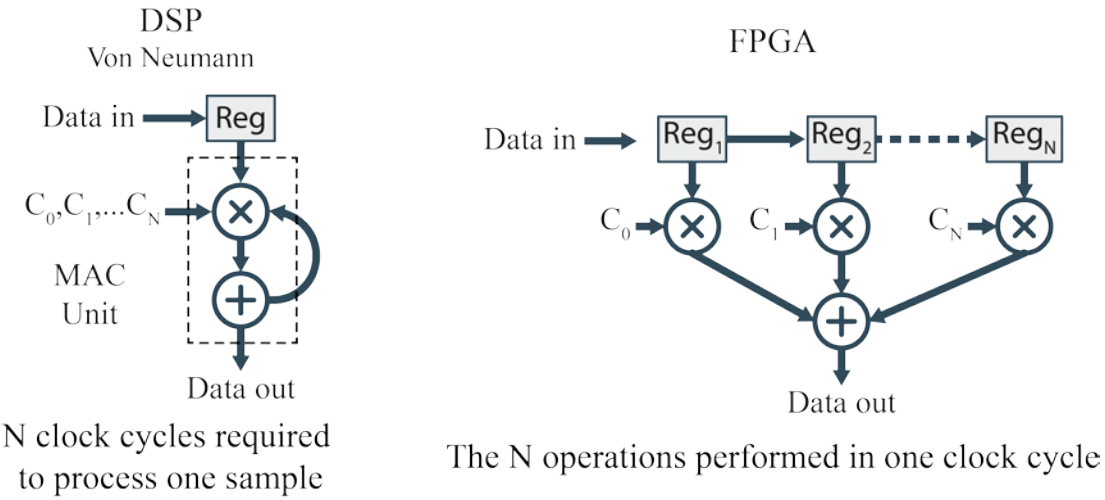

Figure 3.2: FIR Filter implementations: DSP vs FPGA.

Another advantage of FPGAs is the flexibility to find a trade-off between silicon-area i.e. the number of utilized cells and speed during the implementation process. Either, resources can be shared to implement an area efficient design that will require

more clock cycles to be processed or, if speed is the design target and the hardware resources of the FPGA are enough, a full parallel design can be accomplished. To illustrate this concept, the equation that characterizes the response of the FIR filter for k = 3 is considered. The computation of (3.2), involves four multiplications and three additions.

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3] \tag{3.2}$$

Figure 3.3 graphically describes a parallel, serial-parallel and full-serial implementation. Although the full parallel implementation requires more hardware, it will deliver the output in fewer clock cycles that the other two type of implementations, which will share the resources in time in order to complete the calculations. Depending on the available resources and the timing requirements, the designer can choose the optimum solution that meets the requirements.

One of the less exploited yet important advantages of FPGAs is the ability to re-design conventional peripherals in a very detailed level. This advantage is presented in the following chapter, where some conventional peripherals such as the PWM unit are redesigned allowing new possibilities for control techniques.

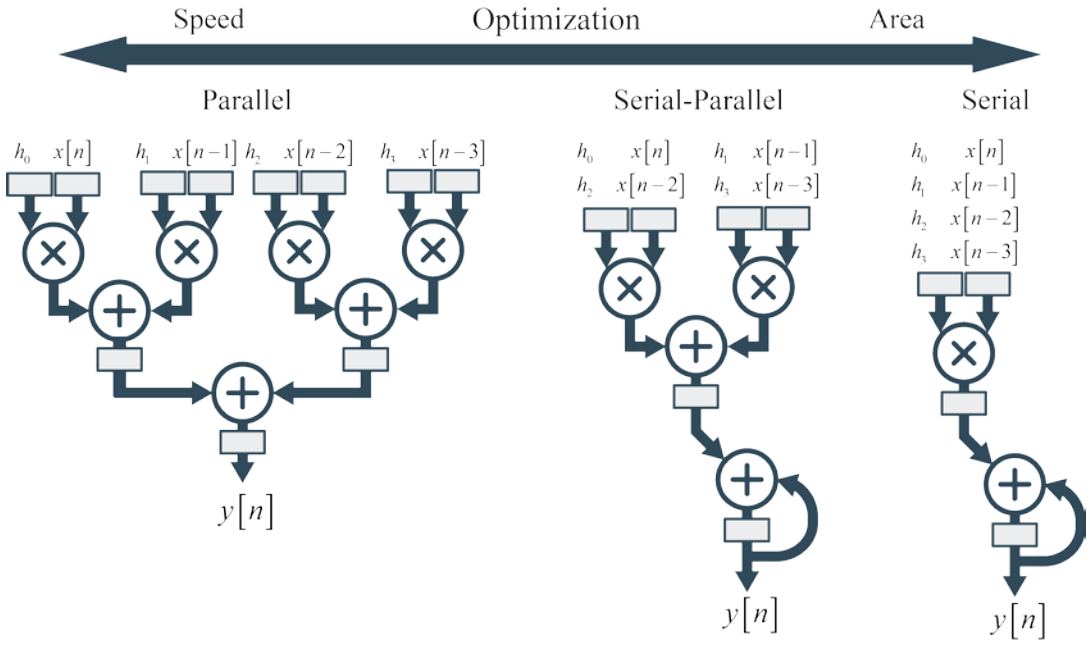

Figure 3.3: Speed/Area optimization trade-off in FPGAs implementations.

3.2.2 Speed vs Area Optimization Methods

There are several speed and area optimization methods for FPGA design, two of the most widely used are resource sharing for area optimization and pipeline used for speed optimization. The resource sharing optimization process is very similar to the process explained in Figure 3.3, therefore only the pipeline technique will be explained. When the pipelining technique is used to optimize a design, a whole process is broken into smaller stages and each stage outputs data for the next stage, inserting a register between stages, with all the stages operating concurrently.

When this optimization process is applied, the latency i.e. the time for a new input to become a new output will not be reduced; it could even be increased by the register delay. However, the throughput i.e. the rate at which new data can be input will be reduced. Furthermore, by reducing a big process into smaller stages, higher clock speed can be achieved. Figure 3.4 (a) presents an example with total delay time T_a and Figure 3.4 (b) has an extra register in the middle dividing the T_a into balanced T_b and T_c delay times, where $T_b \simeq T_c$ and $T_a = T_b + T_c$.

In Figure 3.4a the latency and throughput time would be equal to T_a. In contrast, in Figure 3.4b the latency of the design would also be T_a but the maximum throughput would be restricted to the maximum of T_b and T_c. Supposing that T_b is the maximum and that T_b is approximately half of T_a, then the throughput would be nearly as twice of the original design. Additionally, given that the combinatorial logic is also reduced and simplified, the frequency of the clock can also be increased, reducing the latency and throughput times of the optimized design.

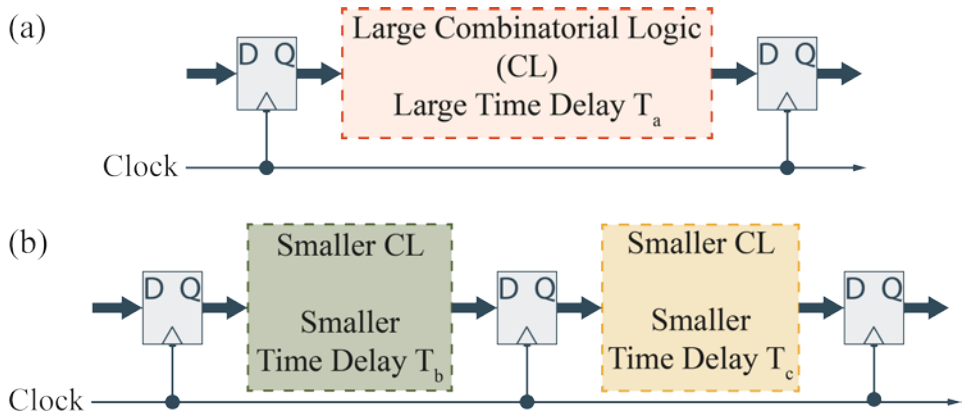

Figure 3.4: Pipeline optimization process.

3.2.3 Fixed-Point Number Representation

FPGAs are programmed to a bit-level precision, and allow a wide variety of computer arithmetic implementations. The area and speed of the design can be greatly affected by the word length of the variables therefore, it should be carefully chosen. In general terms, it can be expected that fixed-point implementations are faster and consume far less area than floating-point implementations, which have a wider dynamic range and no scaling is required. Although some of the newest FPGA families like the Arria 10 and Stratix 10 families from Altera [97], are starting to include floating-point hardware operators, a full floating-point design can still not be implemented in of the commercially available FPGAs.

Therefore, for most of the applications in power electronics, a fixed-point representation with a proper scaling is sufficient. A fixed data type is characterized by a word length if n bits, by the position of the binary point and whether it is signed or unsigned. As shown in Figure 3.5: b_i is the ith binary digit, b_{n-1} is the Most Significant Bit (MSB) and when the number is signed it usually indicates the sign, finally b_0 is the position of the Least Significant Bit (LSB).

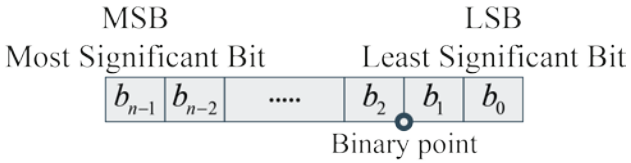

Figure 3.5: Fixed-Point Number Representation

The two compliment representation (2C) is today by far the most popular signed numbering system in DSPs [94]. Fixed-point arithmetic can produce numbers that are larger than the established bit-width, resulting in a loss of information thus, rounding or truncating operations have to be performed.

3.3 CORDIC Algorithm

Coordinate Rotational Digital Computer is abbreviated as CORDIC, the key concept of CORDIC arithmetic is based on the principle of two-dimensional geometry. The iterative formulation of the algorithm was first described by E. Volder in 1959 [98]. The algorithm was later improved by J. Walther [99] who showed that by making small variations to simple parameters, it could be used as a single algorithm to implementation of a wide range of functions which include: trigonometric, logarithmic, transcendental functions, complex number multiplications, linear system solutions among many others.

Although it may not be the fastest method to perform these operations, it has become attractive to designers because of its low hardware requirements, given that the same iterative algorithm can be used to perform all of these functions using the basic shift-add operations of the form $a \pm b \cdot 2^{-i}$ [100]. In the following, the basics of the algorithm will be explained.

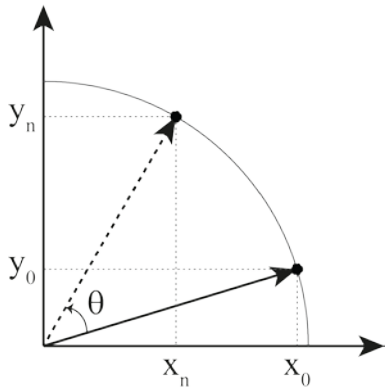

Figure 3.6: Rotation of a vector on a two-dimensional plane

As presented in Figure 3.6, the rotation of a two-dimensional vector $\vec{p} = [x_0 \quad y_0]$ through the angle θ to obtain the rotated vector $\vec{p}_n = [x_n \quad y_n]$ could be performed by the product $\vec{p}_n = \vec{R}\vec{p}_0$ (3.3). If the cosine term is factored from equation (3.3), it can be reinterpreted as the product of a scaling-factor $K = \cos\theta$ and a pseudo rotation matrix, as presented in equation (3.4):

$$\vec{p}_n = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{3.3}$$

$$\vec{\mathrm{R}} = \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \tag{3.4}$$

To achieve the simplicity during the hardware implementation, the central ideas of the CORDIC arithmetic are:

- Rotations are to be decomposed into a sequence of elementary rotations through predefined angles that could be implemented with minimum hardware cost.
- Avoid scaling, that might involve extra arithmetical operators (multiplications mostly). The latter is assumed by the fact the scale-factor *K* does not contain angle rotation information. If the set of small pre-defined angles are restricted to $\alpha_i = \tan^{-1}(2^{-i})$, consequently $tan(\alpha_i) = 2^{-i}$ can be effortlessly implemented by a shift operation. The algorithm performs the rotation over the angle θ by doing a certain number of micro-rotations through α_i. At the same time, the factor

$K = \cos\theta$ can be calculated in advance since it does not depend on the rotation direction; therefore, it becomes a constant.

The iterative pseudo-rotation, which is the core of the CORDIC algorithm, can be expressed as follows:

$$\begin{aligned} x_{i+1} &= x_i - d_i \cdot y_i \cdot 2^{-i} \\ y_{i+1} &= y_i + d_i \cdot y_i \cdot 2^{-i} \end{aligned} \tag{3.5}$$

$$z_{i+1} = z_i - d_i \cdot \alpha_i \tag{3.6}$$

where: $d_i = \pm 1$, is used to decide the rotation direction. The angle accumulator z_i introduced in (3.6) tracks the rotation angle in each iteration process. The magnitude gain that is generated in every iteration can be calculated with (3.7). This scaling-factor will accumulate and converge to $K \sim 1.6467605$ depending on the number of iterations it can be calculated with (3.8). Therefore, instead of scaling during each rotation, the outputs can be scaled by $\frac{1}{K_f}$.

$$K_i = \cos^{-1}\left(\tan^{-1}\left(2^{-i}\right)\right) = \sqrt{1+2^{-2i}} \tag{3.7}$$

$$K_f = \prod_{i=0}^{n} \sqrt{1+2^{-2i}} \tag{3.8}$$

The generalized form of the algorithm can be seen in (3.9), for $m = 1, 0, -1$ and $\alpha_i = \tan^{-1}(2^{-i})$, 2^{-i}, or $\tanh^{-1}(2^{-i})$, the algorithm works in circular, linear or hyperbolic coordinate systems. The algorithm can operate in *rotation* mode and *vectoring* mode. In *rotation* mode the components of a resultant vector due to the rotation of a vector through a given angle are derived, in *vectoring* mode given the component values, the magnitude and phase angle of a vector can be estimated. Finally, table 3.1 summarizes the operations that the algorithm can perform, when *m = 0* or in a linear coordinate system, which is the relevant implementation for this work.

$$\begin{aligned} x_{i+1} &= x_i - m d_i \cdot y_i \cdot 2^{-i} \\ y_{i+1} &= y_i + d_i \cdot y_i \cdot 2^{-i} \\ z_{i+1} &= z_i - d_i \cdot \alpha_i \\ \text{where:}\quad d_i &= \begin{cases} sign(z_i), & rotation\,mode \\ -sign(y_i), & vectoring\,mode \end{cases} \end{aligned} \tag{3.9}$$

Table 3.2 CORDIC Algorithm Operations

Rotation mode	Vectoring mode
$x_n = K_f^{-1}\left(x_0 \cos z_0 - y_0 \cos z_0\right)$	$x_n = K_f^{-1}\sqrt{x_0^2 + y_0^2}$
$y_n = K_f^{-1}\left(y_0 \cos z_0 - x_0 \cos z_0\right)$	$y_n = 0$
$z_n = 0$	$z_n = z_0 + \tan^{-1}\left(y_0 / x_0\right)$

3.4 Delta-Sigma Analog-to-Digital Conversion

The signals which characterize most of the phenomena in the real world are analog, i.e. continuous in amplitude and time. Analog-to-Digital converters (ADCs) translate analog signals, to digital signals that are discrete in amplitude and time. After the discretization process, the resolution and bandwidth of the signal becomes finite. Typically, ADCs can be classified as Nyquist-rate and Oversampling-rate. For Nyquist-rate the most popular technologies are Successive-Approximation (SAR) and Pipelined ADCs, they provide tradeoffs among signal bandwidth, output resolution and the complexity of the hardware [101]. For oversampling, only the Delta-Sigma (ΔΣ) ADCs are common [62], [102].

3.4.1 Basics of Analog-to-Digital Conversion

For the control hardware in power electronics and drives, the SAR type ADCs are currently the most utilized, because of their low cost, ease of use, resolutions and speeds as shown in Figure 3.7. However, ΔΣ ADCs, based on the combination of *oversampling* and quantization *error shaping* techniques, ΔΣ ADCs achieve a high level of insensitivity to analog circuit deficiencies, making them the best choice in many cases, to realize embedded analog-to-digital interfaces in modern SoC designs [103]. Although this technology has been gaining interest in recent years, it has existed for more than 50 years [104].

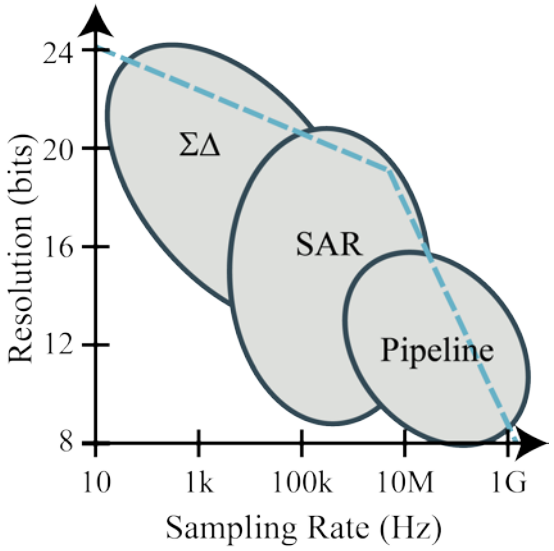

Figure 3.7: ADC architectures, resolution and sampling rates.

3.4.1.1 Nyquist Rate Analog-to-Digital Conversion

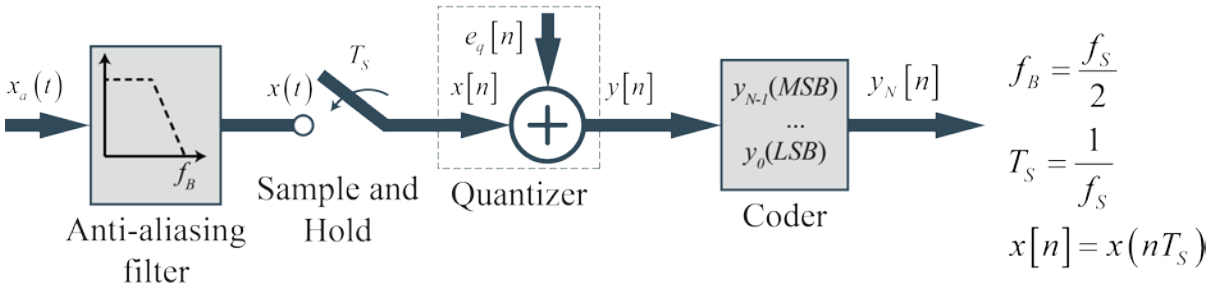

Figure 3.8: Conceptual scheme of a conventional Nyquist-rate ADC.

A conceptual scheme of a conventional Nyquist rate ADC system is shown in Figure 3.8; it includes an *anti-aliasing filter*, a *sample-and-hold* circuit, a *quantizer* and a *coder*. First, the analog input signal $x_a(t)$ is filtered through the anti-aliasing block, otherwise high frequency components of the input signal would interfere or being aliased into the signal bandwidth f_B.

The resulting band limited signal $x(t)$ is sampled at uniformly spaced time intervals T_S, thus yielding a discrete time signal $x[n] = x(nT_S)$. After the sample-and-hold process, the quantizer maps the continuous range of amplitude of $x[n]$ into a discrete set of levels. To conclude, the coder assigns an exclusive binary number to each level, providing the output digitalized data.

The quantization process introduces a fundamental limitation in the performance of the ADC, the continuous to discrete transformation generates an error, commonly referred to as *quantization* error [103]. To further simplify the analyses of the noise in the quantizer, the following assumptions, which are normally met in practice [101], [103] are made:

- The error sequence $e_q[n]$, is a sample sequence of a stationary random process.
- $e_q[n]$ is uncorrelated with $x[n]$,
- the probability density function of $e_q[n]$ is uniform over the range of $\pm\Delta/2$.
- the error sequence is a white noise process.

An ADC or quantizer with Q outputs levels is said to have N bits of resolution where $N = \log_2(Q)$. For an ADC with Q quantization levels, only input values separated by at least $\Delta = 2V/(Q-1)$ can be distinguished to different output levels [101]. By considering an N-bit ADC with $Q = 2^N$ quantization levels, i.e. the quantization step is $\Delta \equiv 2V/(Q-1) = 2V/(2^N-1)$ with $2V$ being the full-scale output range of the quantizer. The corresponding mean noise power is given by:

$$e_q^2 = \frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e_q^2 de_q = \frac{\Delta^2}{12} = \frac{\left(\frac{2V}{2^N-1}\right)^2}{12} \simeq \frac{\left(\frac{2V}{2^N}\right)^2}{12} \tag{3.10}$$

If the signal is treated as a zero mean random process and its power is e_x^2, then the signal-to-noise ratio (SNR) is:

$$SNR = 10\log\left(\frac{e_x^2}{e_q^2}\right) = 10\log\left(\frac{e_x^2}{V^2}\right) + 4.77 + 6.02\left[dB\right] \tag{3.11}$$

From equation (3.11) it can be seen that for each additional bit of resolution in the ADC, there is about a 6dB improvement in the SNR. Following, the dynamic range of the ADC, which is a direct measure of the range of input amplitudes for which the ADC produces a positive SNR. For a full-scale sinusoidal inputs such that $x(t) = V\sin(2\omega t)$ the power of the signal is $e_x^2 = e_q^2 = \Delta^2/2$. Finally, the dynamic range can be defined as:

$$\frac{\frac{V^2}{2}}{\frac{\Delta^2}{12}} \simeq \frac{\frac{V^2}{2}}{\frac{\left(\frac{2V}{2^N}\right)^2}{12}} \tag{3.12}$$

Which when reduced to a dynamic range value is:

$$SNR = 10\log\left(\frac{e_x^2}{e_q^2}\right) = 6.02N + 1.76[dB] \tag{3.13}$$

Note that as presented in (3.13), the ratio of $V^2/2$ to $\Delta^2/2$ is the peak SNR of the ADC for a sinusoidal input. Therefore, the dynamic range of the Nyquist-rate converters is the same as its peak SNR. Later, by using the dynamic range of Sigma-Delta converters in (3.13), and calculating the corresponding N, the resolution of a Nyquist-rate converter can be determined, such that it will produce the same dynamic range.

For Nyquist-rate ADCs, each signal sample is quantized at the full precision or full precision of the converter. The resolution of such converters is limited by the technology in which these chips are made. Furthermore, if the signal is sampled too close to the Nyquist-rate, the anti-aliasing filter must have a very sharp cutoff, a non-trivial design requirement for analog filters [101].

3.4.1.2 Oversampling Analog-to-Digital Conversion

The sampling process imposes a limit on the bandwidth of the signal f_B. The Nyquist theorem states that the sampling frequency f_S must be at least twice the bandwidth of the signal $f_S = 2f_B$ in order to be digitally reconstructed. If an ADC samples faster than the Nyquist-rate, it falls into the *oversampling* ADCs category. The oversampling ratio can be then defined as: $OSR > f_S/2f_B$. Consequently, one of the advantages of oversampling ADCs is that the cutoff frequency of the anti-aliasing filter no longer requires to be very sharp, as shown in Figure 3.9.

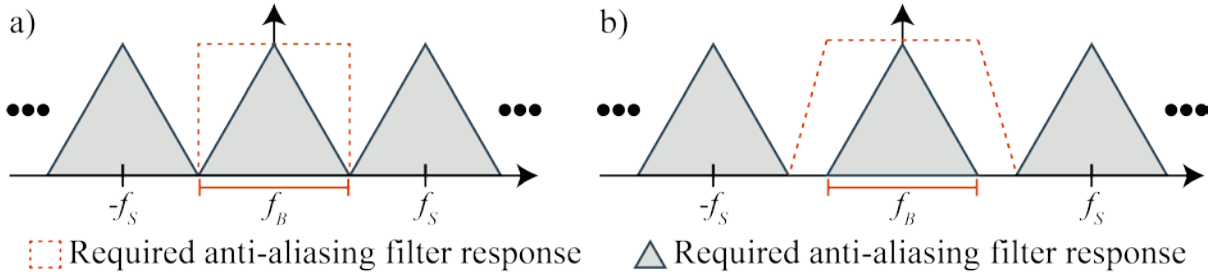

Figure 3.9: Anti-aliasing filter comparison: a) $f_S = 2f_B$ b) $f_S = 4f_B$.

In oversampling ADCs, the same noise power produced by a Nyquist-rate converter is expected, but its frequency distribution is different because of the higher sampling rate. However, the signal bandwidth is still $[-f_B, f_B]$, and only a small fraction of the total noise power falls in the signal spectrum band. Thus, the SNR is increased as shown in Figure 3.10.

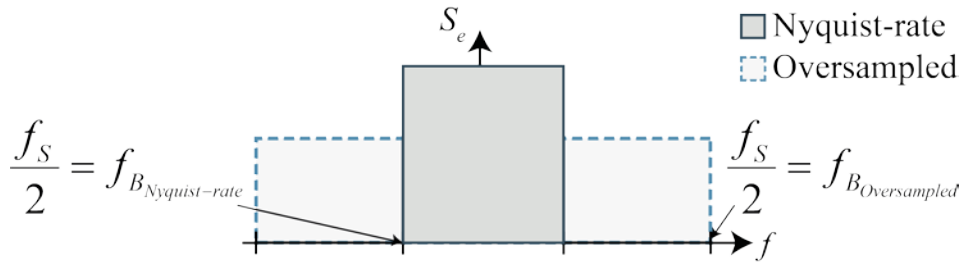

Figure 3.10: Noise power spectrum comparison.

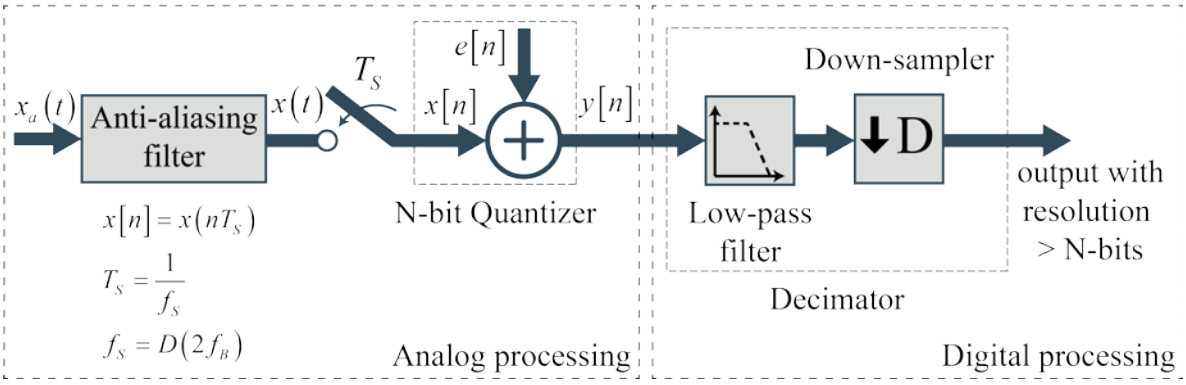

Figure 3.11: Oversampled ADC

A low pass filter following the ADC can then attenuate the noise power outside the signal band. Finally, the signal can be down-sampled to the Nyquist-rate without affecting the SNR; the combined operation of low-pass filter and down-sampling is called *decimation*. A block diagram of the oversampled ADC can be seen in Figure 3.11, where $e(n)$ is a sample sequence of a stationary random process, uncorrelated with $x(n)$, and the probability density function of the error process is uniform over the range of quantization [101].

3.4.2 Delta-Sigma Analog-to-Digital Converters

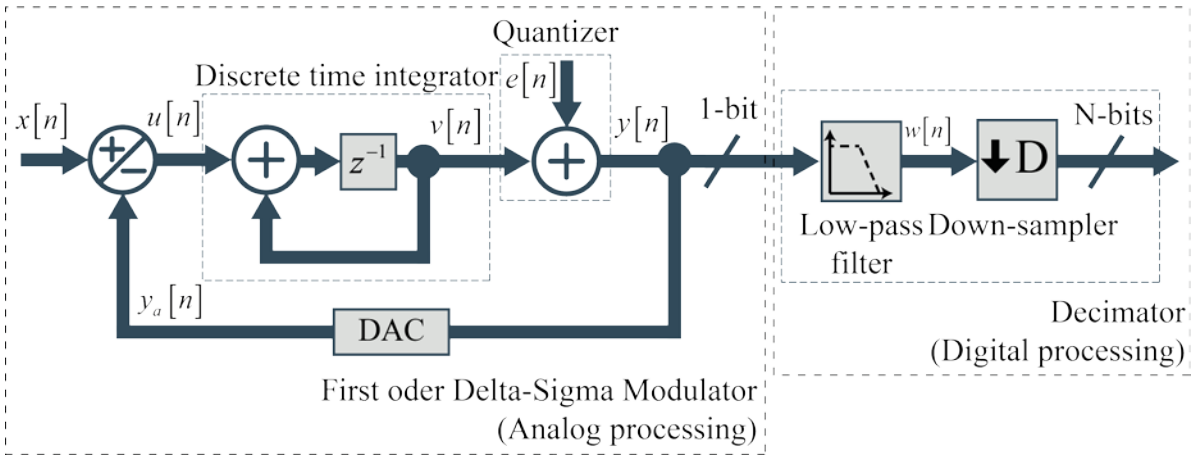

Figure 3.12: Block diagram of a first order delta-sigma analog-to-digital converter.

The basic diagram of a first order ΔΣ ADC is shown in Figure 3.12, it basically consists of an analog sigma-delta modulator followed by a digital decimator. The

modulator stage contains an integrator, a quantizer or analog-to-digital converter and a digital-to-analog (DAC) converter. The quantized signal is not the direct $x[n]$ input signal but a filtered difference of the input and an analog representation of the quantized output $y_a[n]$. This filter is a discrete time integrator with the transfer function $G(z) = \frac{z^{-1}}{1-z^{-1}}$ [101].

3.4.2.1 Delta-Sigma Modulator

By considering an ideal DAC, which can be replaced by a unity gain transfer function, and $H_x(z)$ and $H_e(z)$ be the transfer functions of the signal and the noise respectively, the ΔΣ modulator output $Y(z)$ is given by:

$$Y(z) = X(z)H_x(z) + E(x)H_e(z) \tag{3.14}$$

Hence, for the first order modulator follows : $Y(z) = X(z)z^{-1} + E(z)(1 - z^{-1})$ therefore, $H_x(z) = z^{-1}$ and $H_e(z) = (1 - z^{-1})$. It can be seen that the ΔΣ modulator processes the signal and quantization noise simultaneously. While the input signal is transferred to the output with a sample delay, the quantization noise is shaped by a first order z-domain differentiator, i.e. a high-pass filter.

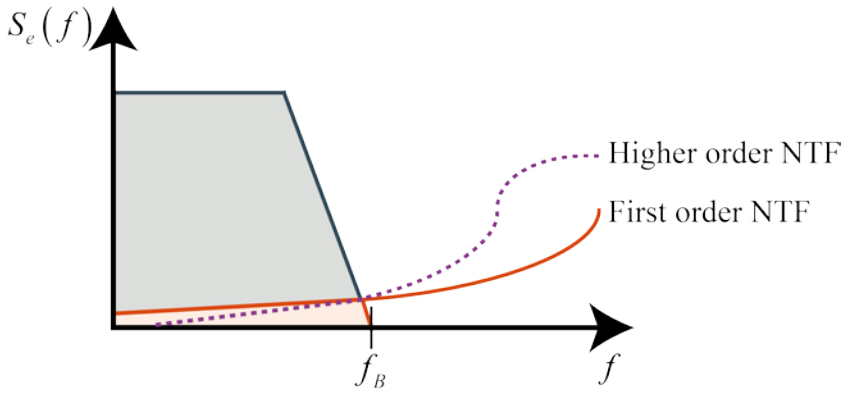

Figure 3.13: Noise shaping from delta-sigma modulator.

Figure 3.13 shows the noise distribution generated by the high-pass filter for different Noise Transfer Functions (NTFs), the noise within the bandwidth signal is severely attenuated. Meanwhile outside the bandwidth of the signal it is not attenuated and even actually amplified; nevertheless, this noise can be removed using a digital low-pass filter. The DAC is required to be linear in the whole conversion range. Since a 1-bit DAC is effortlessly linear, most of the ΔΣ ADCs contain 1-bit DACs and the corresponding 1-bit quantizers, which becomes a simple comparator.

To evaluate the performance of this type of converters, the total signal and noise power at the output of the converter have to be found. This can be done by evaluating the power spectral densities at the output of the converter based on the input power spectral densities of the signal and noise. If a stationary random process with power spectral density $P(f)$ is the input to a linear filter with transfer function $H(f)$, the power spectral density of the output random process is $P(f)|H(f)|^2$.

For the oversampled ΔΣ modulator, $|H_x(f)| = |H_e(f)| = 1$ and if the signal $e[n]$ is considered to be white noise, then $P_e(f) = e_e^2/f_S$ and therefore the power spectral density of the noise at the output is also $P_{ey}(f) = e_e^2/f_S$. As such the *in-band* noise power, e_{ey}^2 at the output of the converter is [101]:

$$e_{ey}^2 = \int_{-f_B}^{f_B} P_{ey}(f)\,df = 2\int_0^{f_B} P_{ey}(f)\,df = 2\int_0^{f_B} \frac{2e_{ey}^2}{f_S}\,df = e_e^2\left(\frac{2f_B}{f_S}\right) \tag{3.15}$$

It can be noticed that some of the noise power is now located outside of the signal band because of the oversampling, and so the *in-band* power e_{ey}^2 is less that what it would have been without oversampling e_e^2. Since the signal power is assumed to occur whithin the signal band only, it remains unmodified, thus the signal power at the output is the same as the signal power at the input i.e. $e_{xy}^2 = e_x^2$. Finally, the maximum SNR or, dynamic range can be obtained:

$$\begin{aligned} SNR &= 10\log\left(\frac{e_x^2}{e_{ey}^2}\right) = 10\log\left(e_x^2\right) - 10\log\left(e_y^2\right) + 10\log\left(\frac{f_S}{2f_B}\right) \\ &\text{if } \frac{f_S}{2f_B} = 2^{\kappa} \text{ then:} \\ SNR &= 10\log\left(\frac{e_x^2}{e_{ey}^2}\right) = 10\log\left(e_x^2\right) - 10\log\left(e_y^2\right) + 3.01\kappa \end{aligned} \tag{3.16}$$

From (3.16) it can be seen that every time that the oversampling ratio κ doubles, the SNR improves by 3 dB, which is equivalent to half bit of improvement in the resolution. A L^{th} order modulator can be directly extended from the first order ΔΣ modulator where the Signal Transfer Function (STF) is $STF \equiv H_x(z) = z^{-1}$ and the Noise Transfer Function (NTF) is $H_e(z) = (1 - z^{-1})^L$. Based on this knowledge and following the same process as in (3.15) (for a deeper explication see [103]), the total power of the quantization noise within the signal band is given by (3.17). Finally, the

ideal in-band signal-to-noise ratio of an L[th] order ΔΣ modulator can be calculated as follows [101], [103]:

$$P_e = \int_{-f_B}^{f_B} S_e(f)\,df \approx e_q^2 \frac{\pi^{2L}}{(2L+1)OSR^{2L+1}} \tag{3.17}$$

$$SNR_{\Delta\Sigma} = 10\log\left(\frac{e_x^2}{P_e^2}\right) = 6.02N + 1.76 - 10\log\left(\frac{\pi^{2L}}{2L+1}\right) + (2L+1)10\log(OSR)[dB] \tag{3.18}$$

Thus, the performance of a L^{th} order modulator can be improved by increasing the modulator order and the oversampling ratio. Note that if $L = 0$ and $OSR = 1$ then the expression falls back to that of a Nyquist-rate ADC, as presented in (3.13).

3.4.2.2 Decimation Filter

The output of the ΔΣ modulator delivers a high rate bit stream with the noise shaped such that it falls outside the bandwidth area as much as possible. Therefore, a decimator, i.e. the combination of a low-pass filter with a down-sampling process is needed, in order to eliminate as much noise as possible and to reduce the frequency in which the data is delivered. With the data-rate reduction also comes an increase in the bit-width of the output. Different filter architectures have been proposed for the decimation stage of ΔΣ ADCs, among the most popular are the Cascaded Integrator-Comb (CIC) filters [105] and the Sinc[N] FIR filters [106].

The Sinc[N] FIR requires many computational components, nevertheless a CIC Sinc[N] implementation is a popular approach with an area efficient design, because no digital multipliers are needed. As presented in figure 3.14, CIC Sinc[N] can be efficiently implemented by cascading K stages of accumulators operating at the high sampling frequency f_S, followed by K stages of cascaded differentiators operating at a lower sampling frequency $f_D = f_S/M$, where f_D is the decimation frequency and M is the decimation ratio [105].

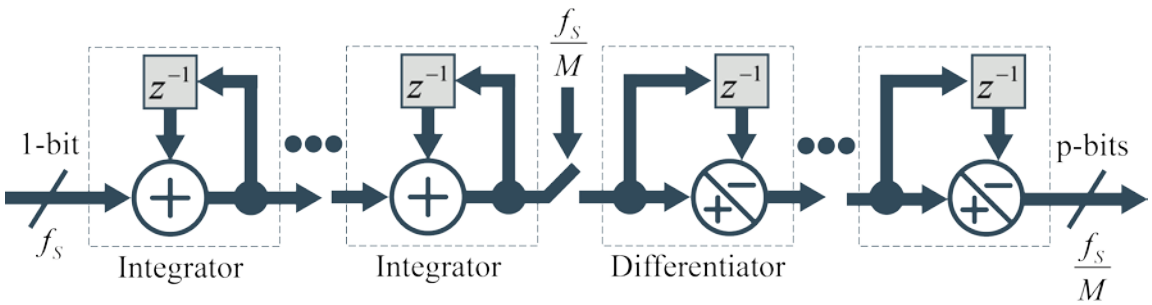

Figure 3.14: CIC realization of Sinc[K] decimation filter with K integrator stages and K decimation stages.

The transfer function of a K[th] order CIC Sinc[K] filter is shown in equation (3.19) [107] which presents the form of a *K*-stage cascaded *M*-tap moving average filter [108], [109]. The frequency response is then obtained by replacing $z = e^{j\omega}$, as presented in (3.20), where the digital angular frequency is determined by $\omega = 2\pi f/f_S$. A filter which frequency response is described by (3.20) is called a Sinc[K] filter, being the $sinc(x)$ function defined as $sin(x)/x$ [106].

$$H(z) = \left(\frac{1}{M}\sum_{m=0}^{M-1} z^{-m}\right)^K = \left(\frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}}\right)^K \tag{3.19}$$

$$\left|H\left(e^{j\omega}\right)\right| = \left|\frac{\sin\frac{M\omega}{2}}{M\sin\frac{\omega}{2}}\right|^K \tag{3.20}$$

In this work the ADS1204 from Texas Instruments [110] was used, it consists of four 2[nd] order ΔΣ ADCs. For a L^{th} order ΔΣ modulator a *K=L+1* order decimation filter is needed, thus, for the 2[nd] order modulator used in this work, at least a Sinc[K] filter with *K = 3* order should be used. The frequency response of a Sinc[3] filter with decimation ratios M = 16 and M =32 is presented in Figure 3.15.

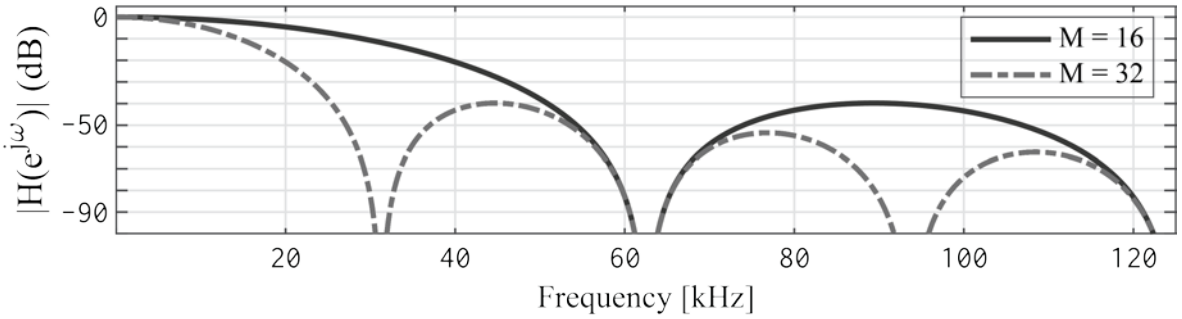

Figure 3.15: Frequency response of Sinc[3] filter for D = 16 and D = 32.

The impulse response of Sinc[3] filter can be found with (3.21) [107], the number of filter coefficients in the impulse response is $c = K(M-2)$, the impulse response of a Sinc[3] filter with M = 32 is presented in Figure 3.16. The ADS1204 usually works with a clock frequency of 10 MHz, from the control point of view, the delay introduced by the ΔΣ ADC is insignificant with a bandwidth of a few kHz.

However, the decimation Sinc[3] filter introduces a delay equal to its K[th] order. In the case of the Sinc[3], the measurement made in n = 1 will be ready at n = 4. Due to the continuous nature of the ADC it is impossible to refer about one defined sampling instant

[111]. The ADS1204 delivers an *effective resolution* i.e. the same resolution of an ideal *N*-bit Nyquist-rate ADC which can be calculated from (3.18), for example a resolution of 11.4 bits is obtained, with a decimation frequency of 312.5 kHz (D = 32).

$$h_n = \frac{n(n+1)}{2} \qquad 1 \le n \le M$$
$$h_n = \frac{M(M+1)}{2} + (n-M)(2M-1-n) \qquad M \le n < 2M \tag{3.21}$$
$$h_n = \frac{(3M-n-1)(3M-n)}{2} \qquad 2M \le n < 3M$$

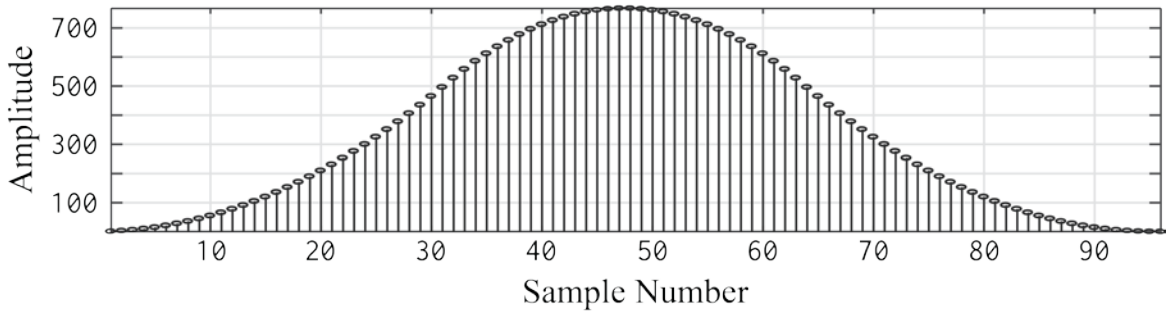

Figure 3.16: Impulse response of Sinc^3 with M = 32.

The decimation filter is a kind of weighted average, which gives more value to the center samples and less at the beginning/end of the sequence, as shown in figure 3.16. Therefore, as shown in Figure 3.17, when using the Sinc^3 filter as decimator the center of its impulse response should be aligned with the PWM, otherwise the measurement of the phase currents will be incorrect, suffering from aliasing [111].

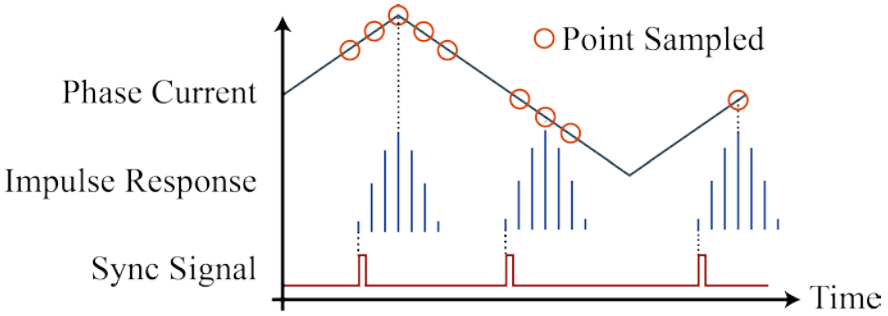

Figure 3.17: Decimator filter response properly aligned to PWM.

3.5 Hardware Description Languages (HDL)

A hardware description language (HDL) is a computer language used to describe the structure and behavior of electronic circuits, most commonly digital logic circuits. The fact that a HDL is used to describe a circuit leads to a change of the programming

paradigm, which happens to be very different from software computer languages. HDLs are principally used to describe concurrent processes actions that occur at the same time. Whereas, most software computer languages are used to implement algorithms or computational procedures in a sequential manner and are executed one instruction at a time [52].

There are two main hardware description languages: VHDL [53] which has more preference in Europe and the east part of the US, and Verilog [54] that has more acceptance in Asia and the west US coast [94]. For digital signal processing with FPGAs, both languages seem well suited. Though, the most recent HDL iteration called System Verilog [55] is an attempt to merge the strengths of both languages and erase the gap between them.

Regardless of the chosen HDL, it is important to understand the concept of concurrent statements. Statements are in the center of most programming languages representing finite quantities of actions to be executed. A statement in a sequential programming such as C, represent an action to be taken by the processor; once the processor concludes one action, it can move onto the next action that is specified in the associated source code. This description lays the foundation for an algorithmic method where the processor follows a set of rules, essentially described by the source code [52].

HDL programming is significantly different because it has the ability to execute virtually an unlimited number of statements at the same time in a concurrent manner i.e. in parallel. This is possible because it describes hardware, where the distinct processes are realized in digital circuits that occupy different regions inside the FPGA chip. This has to be carefully considered, given that HDLs are written as any other sequentially executed languages, yet concurrently executed in hardware.

3.6 FPGA–based Controller Design

The progress made on control of industrial electronic systems has been made in part by the technological revolution leading to powerful components that allow the implementation of more and more complex algorithms. This successful and continuous improvement on the reliability and performance of digital technologies, have made digital control techniques outweigh their analog counterparts. These days, digital control techniques are mostly implemented using general-purpose microcontrollers or a digital signal processor, primarily due to their low cost and software flexibility [56].

Microprocessors have an arithmetic-logic unit dedicated to real-time computations; they also integrate many peripherals like ADCs and timers that are adapted to the needs of the control algorithm. Although control schemes implemented in microcontroller platforms can achieve high dynamic performance, the quality of their results in some aspects is still less than those obtained via analog controllers [6]. An FPGA can also be considered as an appropriate platform that will further improve the performance of the controller and reduce the gap between the analog and digital world.

In fact, when a fast ADC is paired with an FPGA, the control algorithm can be processed within a few microseconds in spite of its complexity. Nevertheless, reaching a high level of control performance can only be achieved with significant design effort. Compared to the standard architectures offered by microcontrollers and DPSs, FPGAs allow the development of application specific hardware architectures. This feature gives the designer a new degree of freedom since building dedicated hardware architectures that meets all the control performance requirements and implementations constraints was not feasible or affordable until now.

Furthermore, all the potentials parallelization of the control algorithms can be exploited and implemented. The design of FPGA-based controllers is still fairly intuitive and the extra degree of freedom demands special skills and expert knowledge of the designers in several subjects e.g. microelectronics, control, power electronics electrical machines [56]. In that sense, it is understandable why most of control engineers prefer to use standard platforms like DSPs for the implementation. Therefore, following a design methodology could ameliorate the design process and make it less cumbersome.

In recent years, the use of complex design tools like MATLAB and Simulink have enabled researchers to make use of FPGAs by translating the code generated in their model-based designs without actually requiring the full knowledge that an FPGA design demands [68], [112]. The simulations results are precisely translated to a hardware implementation, and pave the way to rapid prototyping schemes. Nevertheless, for complex algorithms this solution leads to un-optimized resource architectures [6].

Several design methodologies have been presented [6], [56], [58], [113], the main idea behind these methodologies is to give the designer a specific set of rules in order to reduce the development time, to optimize the resources on the FPGA and to allow the reusability of code. They were specially designed considering that control engineers are not adept experts in microelectronics and for this reason, great part of the design should be done using the Simulink-MATLAB environment.

However, the FPGA code is not automatically generated, because the optimization of the hardware architecture is also taken into account in the design process. Nevertheless, the coding process is highly guided by the methodology. The results are optimized hardware architectures with great dynamic response and reduced development times. Therefore, for the development of this work a similar design methodology was followed.

The HDL Verifier toolbox for MATLAB-Simulink [114] in conjunction with ModelSim [115] allows the designer to create a co-simulation between these two simulation suits. The HDL Verifier toolbox offers the possibility to take full advantage of the MATLAB/Simulink environments, the HDL part of the simulation is processed by ModelSim using the HDL code generated by the user.

By following this approach, the development process can be substantially reduced because the FPGA code can be simulated and verified faster than having to spend time for the compilation every time something has been changed and for the testing in the real system. Furthermore, the design can be optimized following the techniques explained in section 3.2.1 and tailored to the DSP technology of the available FPGA hardware.

3.6.1 Development and Optimization Cycle of the Control Scheme

For this work the chosen hardware description language was VHDL, the target device is a Cyclone V E from Altera which is a low cost FPGA family [116]. These devices contain Variable Precision DSP blocks that are able to perform some arithmetical operations in hardware being mainly based on multiplications and additions. Depending on the word length of the arithmetical operation a DSP block can perform one or several operations [117]. By following these design rules, the code can be optimized for the specific FPGA family.

Figure 3.18 shows the block diagram of the FPGA-based controller development process. First, a model of the electrical drive system is constructed in MATLAB-Simulink, at the same time the desired control structures are written in VHDL. For these purpose, most of the numbers are represented in fixed-point, which are defined in the VHDL-2008 Standard of the IEEE [53]. Nevertheless, Altera has failed to include them in their Quartus Prime [118] HDL synthesis software, so the libraries designed by D. Bishop [119] to work with the VHDL-1993 were used.

During the development process, the controller is simulated and verified, being redesigned and optimized, as explained in 3.2.2 until the response is accurate enough. After that, a soft-core microcontroller, the NIOS II [120] is configured using the Qsys builder [121]. Through the NIOS II data can be accessed over a JTag interface. In addition, the final version of the controller can be accessed through the microcontroller by means of a Hardware Custom Instruction [122] for the NIOS II. After this, the VHDL code is synthesized, placed and routed with Quartus Prime; finally, the design can be experimentally validated.

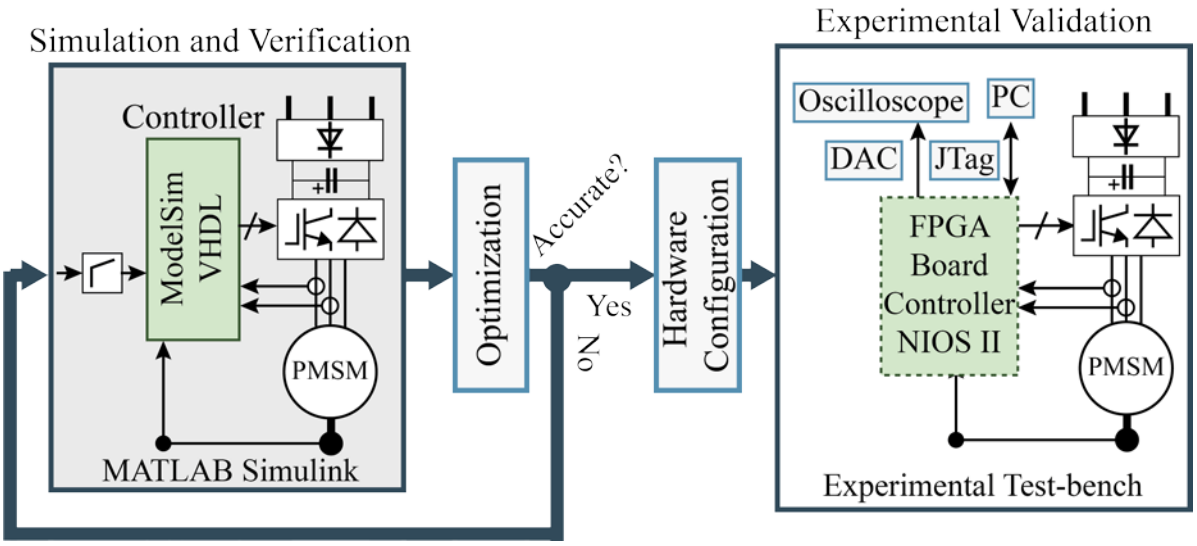

Figure 3.18: Design flow for FPGA-based controller development.

3.7 Summary of the Chapter

This chapter gives an elaborated introduction to FPGA-based controller design. First, the hardware that constitutes an FPGA is described. Following, the reasons that justify that FPGAs are a viable option to realize high performance control schemes for electrical drives and power electronic converters are elucidated. Furthermore, the implementation of digital signal processing by using FPGAs, and the main advantage of FPGAs over other technologies namely their ability to process information in parallel are explained. Subsequently, optimization techniques for algorithm implementation in FPGAs are presented on this chapter.

In addition, the CORDIC algorithm is presented, which is widely used for signal processing with FPGAs. The ΔΣ analog-to-digital converters are then described, which are used in conjunction with the FPGA to obtain the required signals for the control. The resolution and conversion speed of these kind of converters depend on the Sigma-Delta

(ΔΣ) modulator order and oversampling frequency; the decimation filter determines the speed and word-length of the digital signal. Finally, the design process followed to develop FPGA-based control techniques in this work is presented.

4 Predictive Control of the PMSM with a Modulator of Variable Switching Frequency

Recent advancements in microelectronic technology have attracted interest in novel and complex control theories like MPC [4] or artificial intelligence applications [5]. Nevertheless, most of the industrial drives are still based on FOC and DTC and linear PI controllers, because of their ease of implementation and robustness against parameters changes. In particular, field-oriented control uses a modulation technique to generate the voltages applied to the machine. The switching frequency has to be fixed because the fundamental frequency of the pulse-width modulation units on microcontrollers usually cannot be reconfigured if the operation of the whole system is not stopped first.

However, once the switching frequency (f_S) of the inverter is fixed the switching losses cannot be influenced anymore. Furthermore, the dynamic performance of the control will remain fixed according to the selected switching frequency. Aside of the dynamic behavior, the electromagnetic torque ripple (ΔM_e), which is also proportional to the switching frequency is also a measure of the performance of the controller. Conventionally, a compromise between the maximum permissible switching losses and the dynamic performance of the controller is found, in order to select the switching frequency to be used.

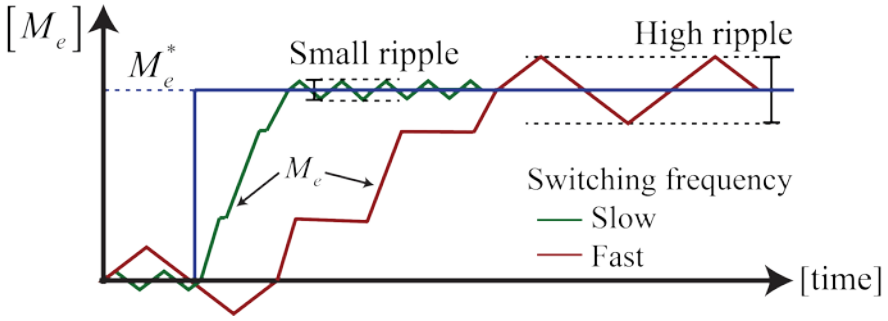

Figure 4.1: Fixed switching frequency performance of the control technique.

As shown in Figure 4.1, when the switching frequency is high, a fast dynamic response, and a small the electromagnetic torque ripple are obtained; nonetheless, the switching losses are high. When the switching frequency is low, the dynamic response is slow, the ΔM_e increases and the switching losses are directly reduced. The advantages of both approaches i.e. a high dynamic response as well as a reduced switching

frequency in the steady-state can be combined if the switching frequency can be modified on-the-fly, i.e. during the operation of the electrical drive.

As already presented in Figure 1.1, the switching frequency usually depends on the computational power of the chosen implementation platform for the control algorithm and on the desired or permissible switching losses. One of the advantages of working with such a powerful and flexible technology as an FPGA is that fixed peripherals in conventional development systems such as the PWM units can be reconfigured to a very precise level of detail. Hence, the main objective of this work is to create a novel field-oriented variable switching frequency modulator-based control technique for the PMSM.

4.1 Control Strategy

The proposed control strategy is based on the FOC and FS-MPC approaches (see sections 1.3.2 and 2.6) since it presents several advantages that make it suitable for the control of power converters:

- Concepts are intuitive and easy to understand.
- Constraints and nonlinearities can be easily considered in the cost function.
- Multivariable systems can be controlled.
- Easy implementation.

Moreover, comparative studies between MPC and classical linear controllers with pulse-width modulation [4], [123], show that the transient response of MPC outperforms that of linear PI with PWM which can be in part attributed to fast sampling times used on MPC implementations. Furthermore, MPC has a better performance in decoupling and controlling the α,β- or d,q-components of the current. However, linear PI controllers with PWM outperform the MPC variant in steady-state operation. MPC also requires a heavy computational power compared to classical control schemes, however, this can be managed by an FPGA control platform.

Conventionally, FS-MPC is implemented without a modulator using high sampling times (50 µs or above) as presented in [4]. As a results, only the discrete voltage space phasor generated by the inverter can be applied. Depending on the operation conditions and the parameters of the control scheme, the same voltage space phasor can remain active for more than one sampling time.

In that sense, FS-MPC generates a variable switching frequency, where the resulting spectrum of voltages and currents will constantly change, depending on the sampling frequency and the operation conditions. However, in most commercial applications a constant switching frequency, referring to a PWM-based control scheme, is preferred for practical reasons. Another disadvantage of FS-MPC is that the quality of the controller depends on the quality of the model [30].

Conversely, these drawbacks can be overcome by combining the strengths of FS-MPC and linear PIs with PWM. The proposed control scheme follows the predictive scheme of FS-MPC and takes advantage of variable switching frequency, however, it does so by generating the switching signals by means of PWM. The fast transient response can be kept by increasing the switching frequency just during the transient states; therefore, switching losses are not greatly increased. Moreover, the steady-state error introduced by classical MPC can be eliminated by replacing the model-based controller with a linear one based on PI controllers.

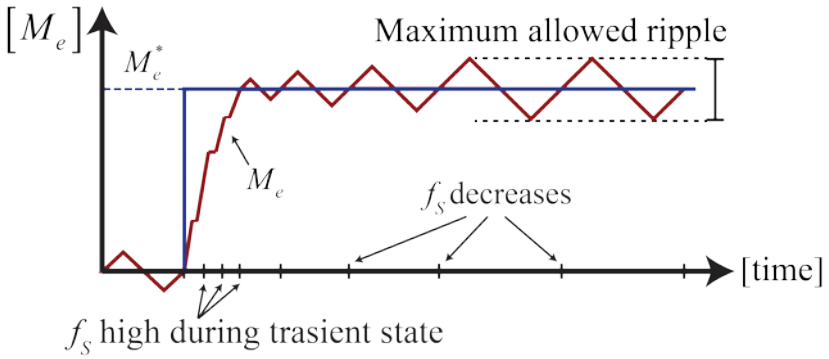

Figure 4.2: Desired variable switching frequency performance of the control technique.

The desired switching pattern of a variable switching frequency control should maximize the switching frequency during the transient states so that the desired reference value is achieved as fast as possible. Once the reference and the real value are close to each other, the switching frequency can be adjusted to a maximum permissible torque ripple, as presented in Figure 4.2. By reducing the switching frequency, the switching losses are also directly reduced. To achieve this kind of dynamic behavior, the bandwidth of the control platform should be very high i.e. sufficient computational power and the analog-to-digital conversion of the measured signals should be very fast; which can be accomplished by using oversampling ΔΣ ADC.

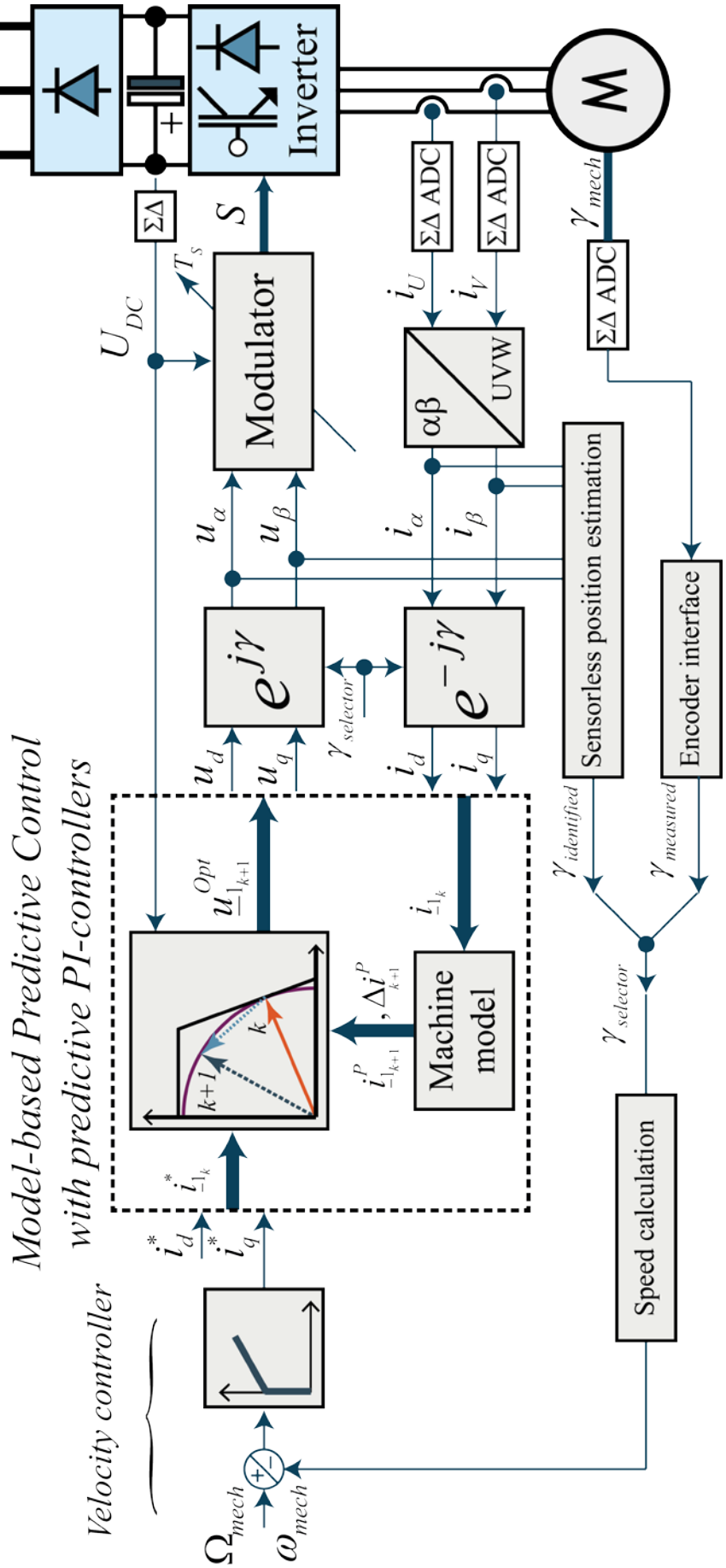

Figure 4.3: Principle of operation of the proposed control scheme.

4.1.1 Operation Principle

The proposed control structure presented in Figure 4.3 follows the field-oriented control approach. The current space phasors are calculated by measuring the phase currents and converting them to the rotating *d,q*-reference frame. A superimposed speed controller tuned according to the symmetrical optimum [124], delivers the reference for i_q^* which in case of the PMSM regulates the electromagnetic torque. All the measurements are made using ΔΣ ADCs.

The PI controllers used in the classical field-oriented control approach are substituted for a predictive controller, which is based on a cost function that minimizes the electromagnetic torque error by minimizing the errors of the currents i_{1d} and i_{1q} in the next sampling state (one step prediction horizon). The cost function include both the estimated electromagnetic torque ripple as well as the switching frequency, to ensure the reduction of the switching losses by directly reducing the switching frequency and maintain the quality of the controller response by regulating the torque ripple. With this information a value is assigned to a finite set of switching frequencies in the operable range of the inverter. The detailed process of the predictive control will be explained in the following sections.

Once the optimal voltage space phasors that will be executed in the next switching cycle are obtained from the predictive control, the switching commands for the inverter are derived from the conventional SPM technique described in section 2.4. The SPM implementation makes use of reconfigurable pulse-width modulation units i.e. the sampling time (T_S) can be varied without stopping the control platform to reconfigure the PWM units. The influence of the inverter dead-time i.e. switching delay is compensated as presented later in section 4.2.5.1; finally these switching signals are sent to a two-level inverter.

4.2 Predictive Torque Control Approach

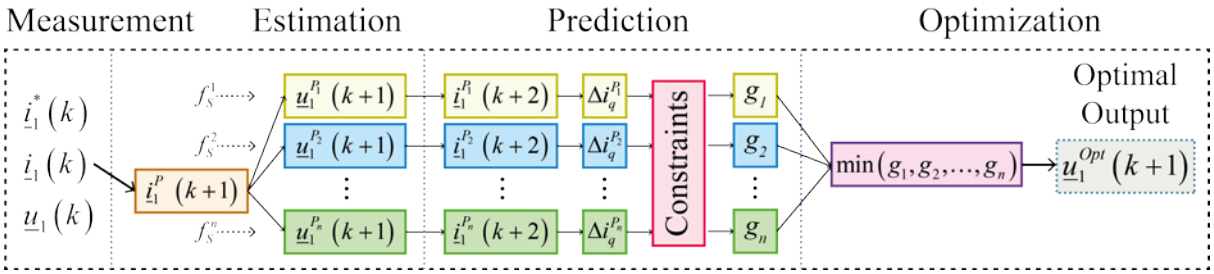

Figure 4.4 Steps for the realization of the predictive control scheme.

The predictive control approach is presented in Figure 4.4. As it can be seen in the diagram the process comprises four main stages: *measurement*, *estimation*, *prediction,* and *optimization*; during this process the value of a cost function g_n for n different sampling times T_S^n is obtained. The super-index ^{P} indicates the estimated or predicted value of the corresponding variable.

First, during the *measurement* stage the values of the actual and reference current space phasors $\underline{i}_1(k)$, $\underline{i}_1^*(k)$ and the current active voltage space phasors $\underline{u}_1^*(k)$ are obtained. The measurement process is synchronized at the middle of the switching period T_S so that the average of the current values can be obtained. However, the prediction algorithm requires of some intermediate values in order to make the estimation more precise and reliable.

During the *estimation* stage depicted in Figure 4.5, the space phasor modulation carrier is represented by the triangles on the top of the image, representing the duration of the T_S. The value of the current space phasor at the end of the current switching period is called $\underline{i}_1^P(k+1)$ whom value is estimated for the remaining half of the active switching period $T_S/2$. With this information, the set of possible references voltage space phasor $\underline{u}_1^{Pn}(k+1)$ can be determined as is explained in detail in section 4.2.3.

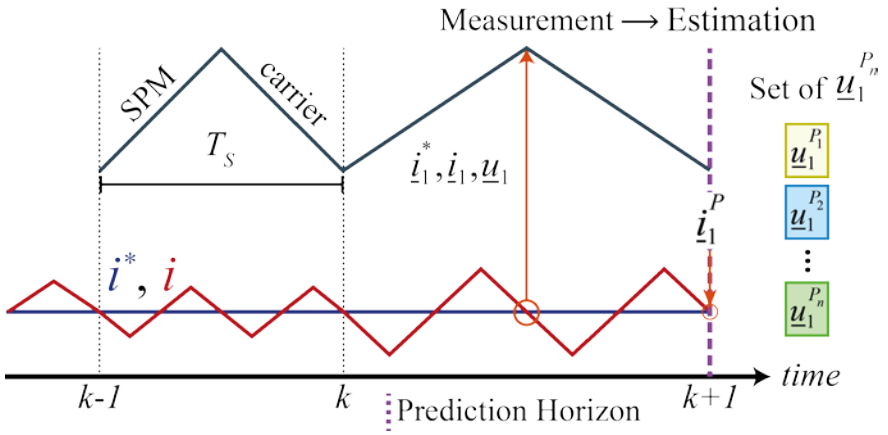

Figure 4.5 Estimation stage of the Predictive Torque Control.

During the *prediction* stage, the measured and estimated values are used to predict the values of the current space phasors at the end of the next switching period $\underline{i}_1^{Pn}(k+2)$ and their corresponding switching ripple Δi_q^{Pn}; in section 4.2.4 it is explained that the electromagnetic torque is proportional to the switching ripple i.e. $\Delta M_e \approx \Delta i_q$. At this point, the final value of the cost function can be calculated by considering the corresponding switching frequency f_S^n and evaluating some constraining functions.

The exact form of the cost function and its constraints is explained in detail in the next section. Finally, during the *optimization* stage depicted in Figure 4.6, the minimum value of the set of n cost functions g_n delivers the optimum control action that will be applied at the beginning of the next sampling time $\underline{u}_1^{Opt}(k+1)$. In the following, the elements of the predictive torque control approach will be explained.

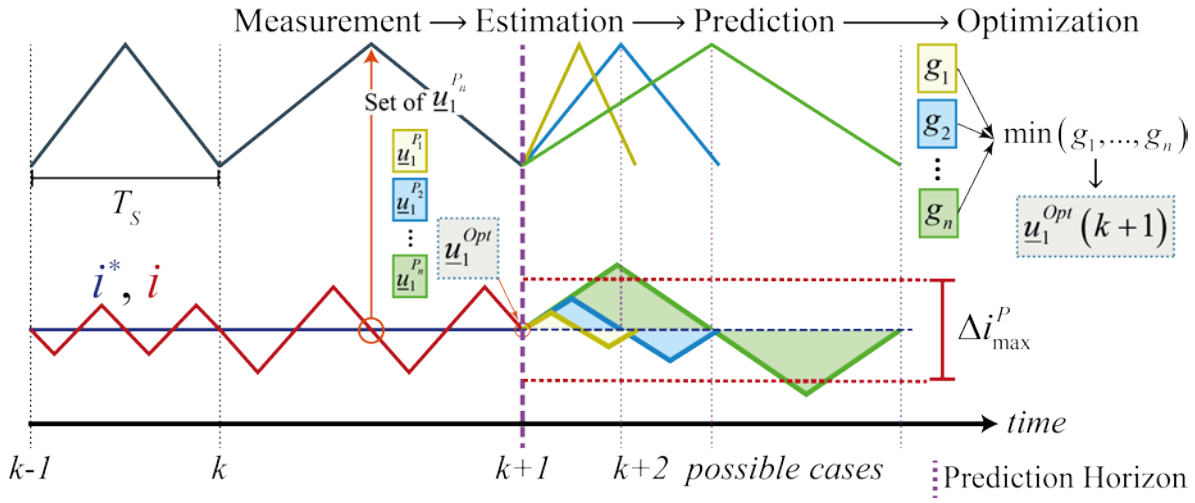

Figure 4.6 Predictive torque control execution.

4.2.1 The Cost Function

The cost function g has a one-step prediction horizon it is defined so that the electromagnetic torque error is reduced by minimizing the errors of the currents i_d and i_q in the next sampling state. The estimated electromagnetic torque ripple, the switching frequency as well as constraints for the desired operation of the control are also included in the cost function.

As previously explained in section 2.6.3, in an strict sense, the future value of the current ought to be known, one possible solution is to estimate the future reference value by extrapolation as in [45]. However, for this work it is considered that the future value of the reference remains equal to the present value i.e. $i^*(k) = i^*(k+1) = i^*(k+2)$. Therefore, the most general form of the cost function is:

$$g = w_q \frac{\left|i_q^*(k) - i_q^P(k+2)\right|}{i_N} + w_d \frac{\left|i_d^*(k) - i_d^P(k+2)\right|}{i_N} + w_{\Delta i_q} \frac{\Delta i_q^P}{i_N} + w_{f_S} \frac{f_S}{f_{S\max}} + f_{pen}\left(i_q^P(k+2)\right) + f_{pen}\left(i_d^P(k+2)\right) + f_{pen}\left(\Delta i_q^P(k+2)\right) + f_{pen}\left(ei_d(k)\right) + f_{pen}\left(ei_q(k)\right) \tag{4.1}$$

$$f_{pen}\left(i^P\right) = \begin{cases} \infty, & \left|i^* - i^P\right| > \varepsilon i^P_{\max} \\ 0, & \left|i^* - i^P\right| \le \varepsilon i^P_{\max} \end{cases} \tag{4.2}$$

$$f_{pen}\left(\Delta i^{P}\right)=\begin{cases}\infty, & \Delta i^{P}>\Delta i^{P}_{\max}\\ 0, & \Delta i^{P}\le\Delta i^{P}_{\max}\end{cases} \tag{4.3}$$

$$f_{pen}\left(ei\right)=\begin{cases}\infty, & \left|i^{*}-i\right|>i_{THLD}\\ 0, & \left|i^{*}-i\right|\le i_{THLD}\end{cases} \tag{4.4}$$

The weights *w* are used to make their respective variable more or less relevant within the cost function i.e. influencing the dynamic behavior of the control. The current errors and the estimated current ripple are normalized to the i_N nominal current of the machine, and the switching frequency f_S is normalized to the maximum frequency of the inverter f_{Smax}.

The $f_{pen}(i_d^P)$ and $f_{pen}\left(i_q^P\right)$ terms (as in equation (4.2)) penalize the corresponding predicted current space phasor, by assigning an infinite value to the cost function in case $|i^* - i^P|$ is outside the εi_{max}^P limit. The i_q current ripple is penalized by $f_{pen}\left(\Delta i_q^P\right)$ (as in equation (4.3)) in case that its value is greater than a allowed limit Δi_{max}^P.

With the terms $f_{pen}(ei_d)$ and $f_{pen}\left(ei_q\right)$ (as in equation (4.4)) the form of the cost function can be changed during transient states so that certain frequencies (previously defined) can be excluded from the minimization process in specific operation ranges. In this case, if the absolute error between the reference and the real *measured* current values during the actual *k* instant of execution $|i^*(k) - i(k)|$ is greater than the i_{THLD} limit, then the respective cost function will be assigned an infinite value, excluding the corresponding switching frequency of the minimization process. During this same operation range w_{f_S} is set to zero i.e. the frequency is not considered in the cost function during these transient states.

In the following, the expected response of the control due to the form of the cost function during transient and steady states is described. If the reference changes by a margin greater than the i_{THLD} limit as in Figure 4.7, then a fast dynamic response is desired, therefore higher switching frequencies should be better favored. However, the fastest switching frequency will not deliver better response because the reference may not be reached within few cycles.

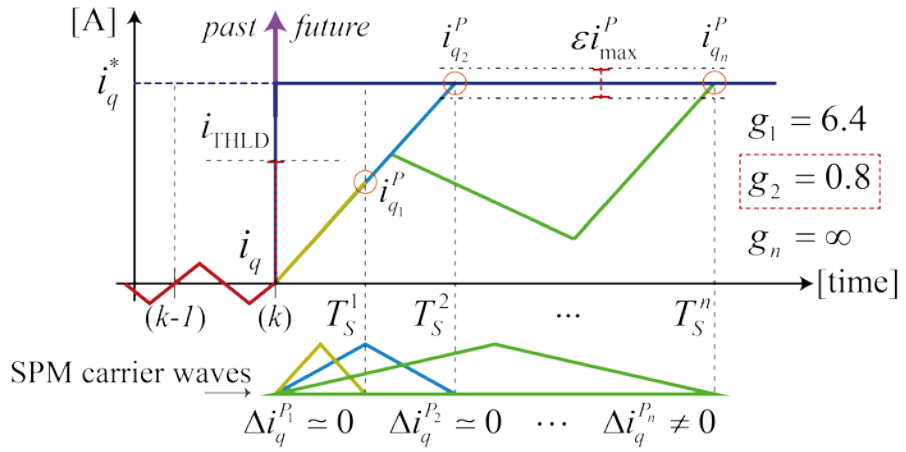

Figure 4.7: Behavior of the current i_q due to the cost function in a transient state.

In fact, the best desired control value will be the one that reaches the references in fewer switching cycles. The lowest switching frequencies are excluded from minimization process because of the slow transitive response, which is not desired. In the case depicted in Figure 4.7, the lowest cost function value g_2 corresponds to the control action with T_S^2, since it represents the best compromise between switching cycles and dynamic response.

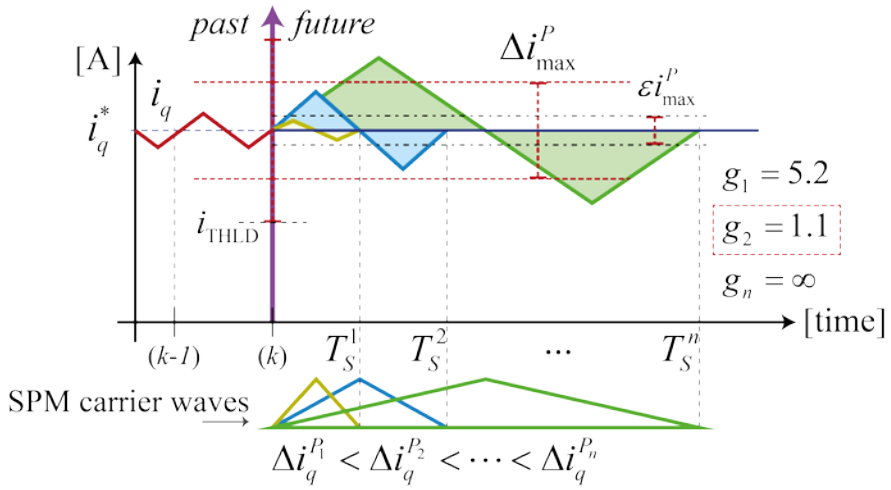

Figure 4.8: Behavior of the current i_q due to the cost function in a steady state.

In case that the real and reference values are close to each other i.e. steady-state operation, the switching frequency should be as low as possible without exceeding the maximum allowed current ripple. Thus, the higher switching frequencies will be heavily penalized as well as the ones that present a current ripple greater than the maximum allowed limit Δi_{max}^P.

At this point, the frequencies with current ripples closer to the limit are the preferred ones, representing the best compromise between performance and reduction

of the losses. In the case depicted in Figure 4.8, the value g_2 of the cost function is again the best choice.

4.2.2 Estimation of the Current Space Phasors

The estimated or predicted value of the current space phasors $\underline{i}_1^P(k+N)$ at the $k+N$ instant are obtained from the model of the machine, by replacing the current derivative di/dt by a forward Euler approximation in the voltage equations (2.25) and (2.26) of the PMSM as follows:

$$\frac{di}{dt} \approx \frac{\Delta i}{\Delta t} = \frac{i(k+1) - i(k)}{T_S} \tag{4.5}$$

$$\begin{aligned} i_d^P(k+N) &= i_d(k+N-1) + \frac{T_S}{L_1}\left[-i_d(k+N-1)\cdot R_1 + \omega \cdot L_1 \cdot i_q(k+N-1) + u_d(k+N-1)\right] \\ i_q^P(k+N) &= i_q(k+N-1) + \frac{T_S}{L_1}\left[-i_q(k+N-1)\cdot R_1 - \omega \cdot \left(i_d(k+N-1)\cdot L_1 + \psi_{d0}\right) + u_q(k+N-1)\right] \end{aligned} \tag{4.6}$$

From (4.6) it can be seen that the mechanical speed is required to estimate the current space phasors. The mechanical time constant of the system is determined by the inertia of the machine (2.34), which is usually larger when compared to the sampling time of the current controller. Therefore, it is assumed that during short period of time the mechanical speed will not change i.e. $\omega(k+1) = \omega(k) = \omega$.

More specifically, when the prediction horizon $N = 1$, the predicted current space phasors $\underline{i}_1^P(k+1)$ (see Figure 4.6) at the end of the present switching period are estimated with the measurement of $\underline{i}_1$, made at $T_S/2$ and the voltage space phasors $\underline{u}(k)$ that are being applied during the actual T_S, therefore the (4.6) can be rewritten as:

$$\begin{aligned} i_d^P(k+1) &= i_d(k) + \frac{T_S}{2L_1}\left[-i_d(k)\cdot R_1 + \omega \cdot L_1 \cdot i_q(k) + u_d(k)\right] \\ i_q^P(k+1) &= i_q(k) + \frac{T_S}{2L_1}\left[-i_q(k)\cdot R_1 - \omega \cdot \left(i_d(k)\cdot L_1 + \psi_{d0}\right) + u_q(k)\right] \end{aligned} \tag{4.7}$$

In the case of the current space phasors $\underline{i}_1^P(k+2)$ at the end of the next switching period (see Figure 4.6), the prediction can be obtained by modifying (4.6) as follows:

$$
\begin{aligned}
i_d^P(k+2) &= i_d^P(k+1) + \frac{T_S}{L_1}\left[-i_d^P(k+1)\cdot R_1 + \omega \cdot L_1 \cdot i_q^P(k+1) + u_d^P(k+1)\right] \\
i_q^P(k+2) &= i_q^P(k+1) + \frac{T_S}{L_1}\left[-i_q^P(k+1)\cdot R_1 - \omega \cdot \left(i_d^P(k+1)\cdot L_1 + \psi_{d0}\right) + u_q^P(k+1)\right]
\end{aligned}
\tag{4.8}
$$

Here, the reference voltage space phasors $\underline{u}_1^{P_n}(k+1)$ that can be applied at the beginning of the next switching period are needed to make the estimation.

4.2.3 Voltage References

The predicted voltage space phasor $\underline{u}_1^{P_n}(k+1)$ for different lengths of the switching periods T_S^n can be obtained by the following two approaches: either linear by computing a PI control algorithm or by using the discrete voltage equations of the mathematical model of the machine. As already presented in section 2.5, the voltage equations of the PMSM from the control perspective are presented again for better understanding:

$$
\begin{aligned}
u_d &= i_d \cdot R_1 + L_1 \cdot \frac{di_d}{dt} - \omega \cdot L_1 \cdot i_q \\
u_q &= i_q \cdot R_1 + L_1 \cdot \frac{di_q}{dt} - \omega \cdot \left(L_1 \cdot i_d + \psi_{d0}\right)
\end{aligned}
\tag{4.9}
$$

4.2.3.1 PI Current Control

The linear PI controllers of the proposed scheme are tuned according to the optimum of the magnitude criterion [125]–[127], Figure 4.9 shows the diagram of the implemented scheme, equation (4.10) shows how the controller gains are calcualted. The advantage of the PI controllers is that they ensure a zero steady-state error and are robust against parameters changes.

$$
K_P = \frac{L_1}{2T_S} \quad K_I = \frac{1}{T_N} = \frac{R_1}{L_1} \tag{4.10}
$$

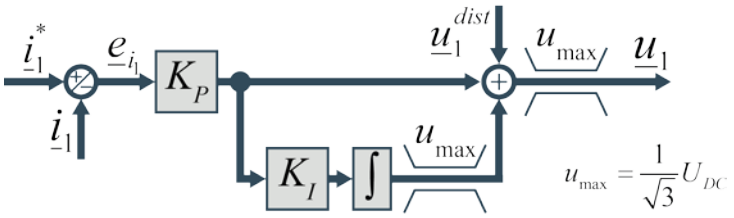

Figure 4.9: Linear PI current regulator diagram.

Special attention has to be taken when the PI-controller are used to generate the predicted voltage space phasors because of the integral action that takes into account the

whole error trajectory; otherwise, the regulator will operate erroneously. The integral action can be approximated as follows for purposes of its digital implementation as follows:

$$\begin{aligned} I(t) &= K_I \int_0^t K_P \cdot \underline{e}_1(\tau) d\tau \\ \frac{dI}{dt} &= K_I \cdot K_P \cdot \underline{e}_1(t) \quad \rightarrow \quad \frac{dI}{dt} \approx \frac{\Delta I}{\Delta t} \\ \frac{\Delta I}{\Delta t} &= \frac{I(k+1) - I(k)}{T_S} \end{aligned} \tag{4.11}$$

$$I(k+1) = I(k) + K_I \cdot K_P \cdot T_S \cdot \underline{e}_1(k) \tag{4.12}$$

Which leads to the recursive equation (4.12) for the integral term, here it is easier to see that the term $I(k)$ accumulates all the past information, which is something that has to be considered when the PI-controller is used estimate the set of possible predicted voltage space phasor $\underline{u}_1^{P_n}(k+1)$ for different lengths of the switching periods T_S^n.

Therefore, when the PI operates in prediction mode, the value of the $I(k)$ term is retained in a different register $I_{pred}(k)$, and $I(k)$ will only be updated until the optimal voltage space phasors is found. In this way, the PI algorithm can be used correctly in the prediction stage. Finally, the PI algorithms for the computation of the voltage references can be written as:

$$\begin{aligned} u_d^P(k+1) &= K_P \cdot e_d(k+1) \cdot (1 + K_I \cdot T_S) + I_d(k) - \omega \cdot L_1 \cdot i_q^P(k+1) \\ u_q^P(k+1) &= K_P \cdot e_q(k+1) \cdot (1 + K_I \cdot T_S) + I_q(k) - \omega \cdot \left[L_1 \cdot i_d^P(k+1) + \psi_{d0} \right] \end{aligned} \tag{4.13}$$

$$\underline{e}_1(k+1) = \underline{i}_1^*(k+2) - \underline{i}_1^P(k+1) \tag{4.14}$$

4.2.3.2 Classical Model-based Voltage References

Another form to obtain the voltage reference for the next sampling instant, is to discretize the equations presented in (4.9). The current derivative is replaced by a forward Euler approximation as in (4.5), and subsequently the current space phasor $i(k+1)$ is replaced by the current reference space phasor $i^*(k+2)$, so that the future error can be calculated as in (4.14). The model-based voltage references for the next sampling time can be expressed as follows:

$$u_d^P(k+1) = i_d^P(k+1)\cdot R_1 + \frac{L_1}{T_S}\cdot e_d(k+1) - \omega\cdot L_1\cdot i_q^P(k+1) \tag{4.15}$$

$$u_q^P(k+1) = i_q^P(k+1)\cdot R_1 + \frac{L_1}{T_S}\cdot e_q(k+1) - \omega\cdot\left[L_1\cdot i_d^P(k+1) + \psi_{d0}\right] \tag{4.16}$$

A drawback of this approach is the sensitivity to errors in the parameters of the model and the lack of an integral action to ensure that the steady-state error reaches zero.

4.2.4 Estimation of the Electromagnetic Torque

As already presented in section 2.2.3, the electromagnetic torque is proportional to the i_q current; this relationship is presented again in (4.17).

$$M_e = \frac{3p}{2}\cdot\psi_{d0}\cdot i_q \tag{4.17}$$

Therefore, it can also be expected that the electromagnetic torque ripple be proportional to the i_q current ripple i.e. $\Delta M_e \approx \Delta i_q$. Consequently, by estimating the current ripple generated by the switching signals of the inverter as is in the following explained, the electromagnetic torque ripple can be calculated.

4.2.4.1 Estimation of the Current Ripple

The switching ripple generated on the current by a voltage obtained by means of pulse-width modulation is caused by the error between the reference ideal value of the voltage space phasor and the natural space phasor that the VSI can generate and is applied to the machine. For a two-level VSI the error corresponds to the difference between one of the eight voltage-space phasors and the desired voltage space phasor.

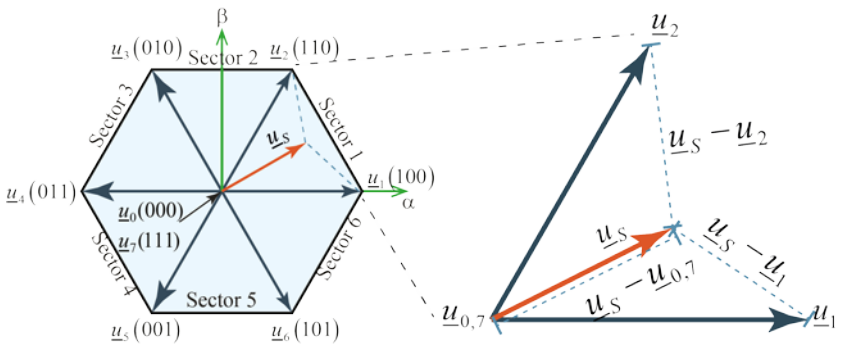

Figure 4.10: Error voltage phasors corresponding to phasors: $\underline{u}_S$ $\underline{u}_1$, $\underline{u}_2$, $\underline{u}_0$ and $\underline{u}_7$.

In the case of a conventional SPM implementation, there are always two active space voltage vectors and two zero space voltage vectors applied. As an example, for a given voltage space phasor $\underline{u}_S$ in sector 1, the error voltage space phasors corresponding to the active voltage space phasors $\underline{u}_1$-$\underline{u}_2$ and zero voltage space phasors $\underline{u}_0$-$\underline{u}_7$ are shown in Figure 4.10.

Compared to the inductive voltage drop, the voltage drop in the resistance can be neglected, therefore, the voltage drop equation can be written as in (4.18), where $\underline{u}_k$ is one of the voltage space phasors generated by the VSI with the magnitude described as in (4.19) [83].

$$\frac{di}{dt} \approx \frac{\underline{u}_\nu - \underline{u}_S}{L} \tag{4.18}$$

$$u_\nu = \begin{cases} \frac{2}{3} U_{DC} & \nu = 1,...6 \\ 0 & \nu = 0,7 \end{cases} \tag{4.19}$$

Subsequently, assuming that the desired voltage space phasor $\underline{u}_S$ is an average within a switching cycle (T_k), and by making incremental changes, the currents for each switching phasor can be approximated as follows:

$$\Delta i_k = \frac{\underline{u}_\nu - \underline{u}_S}{L} \cdot T_k \tag{4.20}$$

Following the example of the voltage space phasor $\underline{u}_S$ in sector 1, the active voltage space phasors $\underline{u}_1$-$\underline{u}_2$ and zero voltage space phasors $\underline{u}_0$-$\underline{u}_7$ are applied their respective T_k time, where $k = 0..7$, as in (4.19). From equation (2.37), it can be seen that the λ_k duty cycle are required for the computation of (T_k), which can be obtained according to the SPM equations (2.38)-(2.40). The current ripple due to zero voltage space phasors is opposite in the direction of the desired $\underline{u}_S$ voltage phasor, while the current ripple flow along difference between the applied active phasor and the desired $\underline{u}_S$ phasor; this can be better understood on the Figure 4.11.

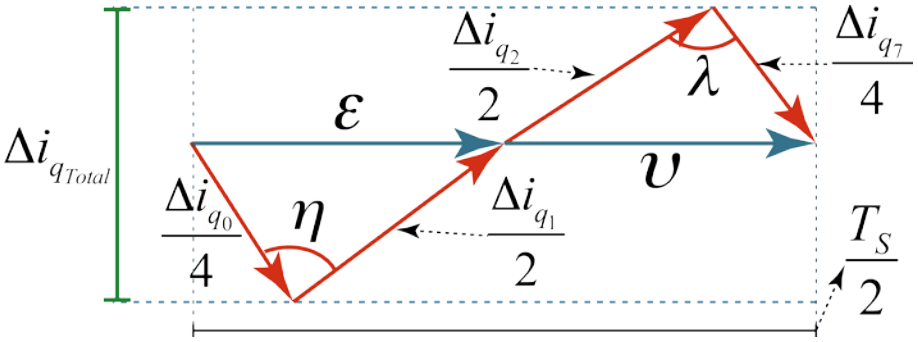

Figure 4.11: Current ripple estimation derived from the modulator switching signals.

Nevertheless, as depicted in Figure 4.11, the expected behavior of the currents cannot be directly derived from the modulator switching signals; the system is undetermined as the magnitudes of the angles η and λ or the sizes of ε and υ are unknown, and therefore the exact ripple cannot be predicted. In spite of this, the current ripple derived from each voltage space phasor is obtained with (4.20), and the total current ripple is approximated as in (4.21), and depicted in Figure 4.12.

$$\Delta i_{Total} \approx \frac{\Delta i_1 + \Delta i_2 + \Delta i_{0,7}}{3} \tag{4.21}$$

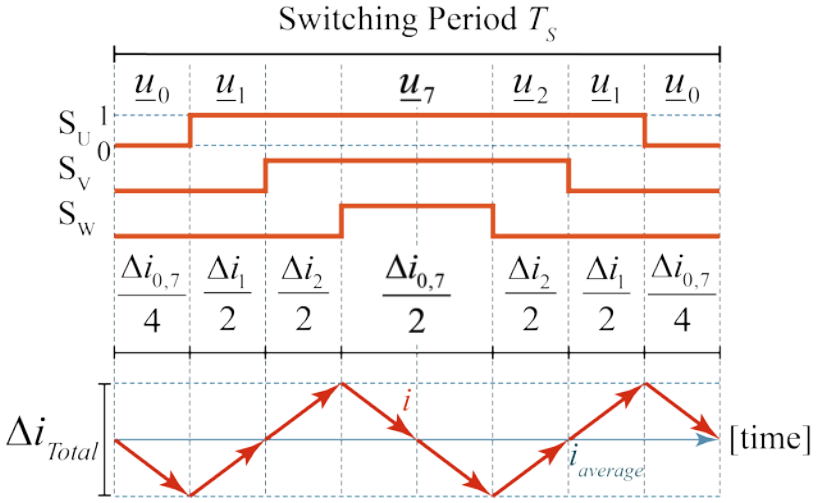

Figure 4.12: Current ripple approximation from the switching pattern generated by SPM in the first sector

4.2.4.2 Enhanced Current Ripple Estimation

Precise control of the torque ripple is crucial for applications that require accurate tracking of the position. Among them are arc welding, laser cutting, high precision machine tools, printing and engraving machines and antenna tracking. Once more, taking advantage of the of the oversampling capabilities of the ΔΣ ADCs, the current ripple of the i_q current can be measured in real-time. As it is well known, the electromagnetic torque is proportional to the i_q current and thus, the electromagnetic

torque ripple ΔM_e, being also proportional to i_q, can be estimated with a higher time resolution.

The estimation method introduced in section 4.2.4.1 can be enhanced with the measurement of the real switching ripple on the i_q current. As explained in section 2.4, in the conventional SPM the switching period is the double of the sampling time T_S or pulse-width modulation period. Therefore, within one sampling time, it is possible to measure the switching ripple two times.

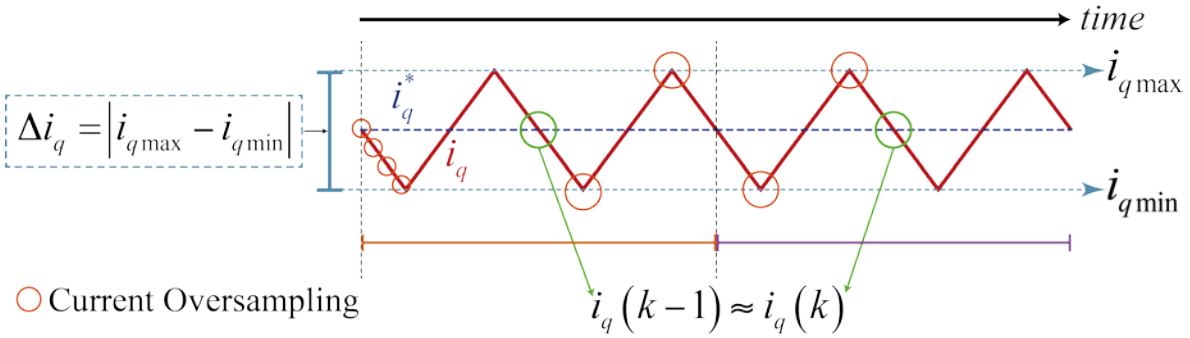

Figure 4.13: Steady-state operation of the control system.

For a precise measurement of the current ripple, the steady-state operation of the system must be ensured; this is assessed by the equation (4.22) i.e. the absolute error between the present and the last measurement of the i_q current is smaller or equal than a predefined *threshold*, as depicted in Figure 4.13. If this condition is fulfilled, the measured switching ripple Δi_q, which is defined by (4.23), can be used to improve the precision of the predicted current ripples for the different switching periods of the controller. To reduce the measurement noise and increase the precision of the measurement, the last two current ripple measurements are averaged as in (4.24).

$$\left|i_q(k-1) - i_q(k)\right| \leq threshold \tag{4.22}$$

$$\Delta i_q = \left|i_{q\max} - i_{q\min}\right| \tag{4.23}$$

$$\Delta i_q = \frac{\Delta i_q(k) + \Delta i_q(k-1)}{2} \tag{4.24}$$

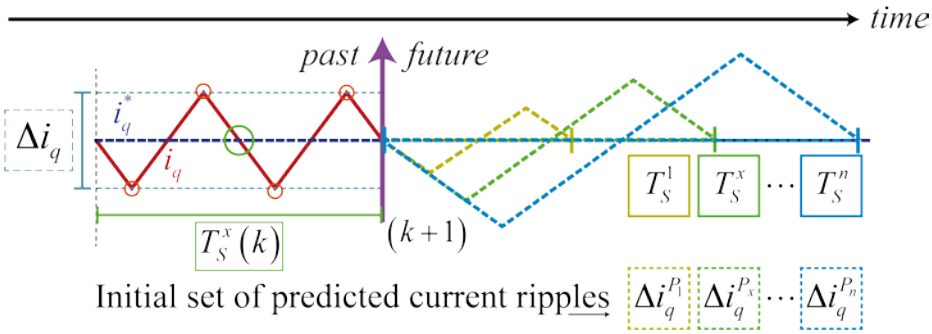

Figure 4.14: Execution of the predictive control, calculation of the first predicted set of current ripples.

Following, when the predictive torque controller is executed in the middle of the T_S, the set of predicted current ripples Δi_q^{Pn} is calculated. As the switching period $T_S^x(k)$ that is active during the execution of the torque controller is also known, the corresponding predicted current ripple Δi_q^{Px} can be extracted from the set as presented in Figure 4.14.

$$\Delta i_{qratio} = \frac{\Delta i_q}{\Delta i_q^{P_n}} \tag{4.25}$$

Therefore, by calculating a *scaling factor* Δi_{qratio} between the real and predicted current ripples as in (4.25), the set of predicted current ripples can be adjusted by multiplying it by Δi_{qratio}; this process is graphically described in Figure 4.15.

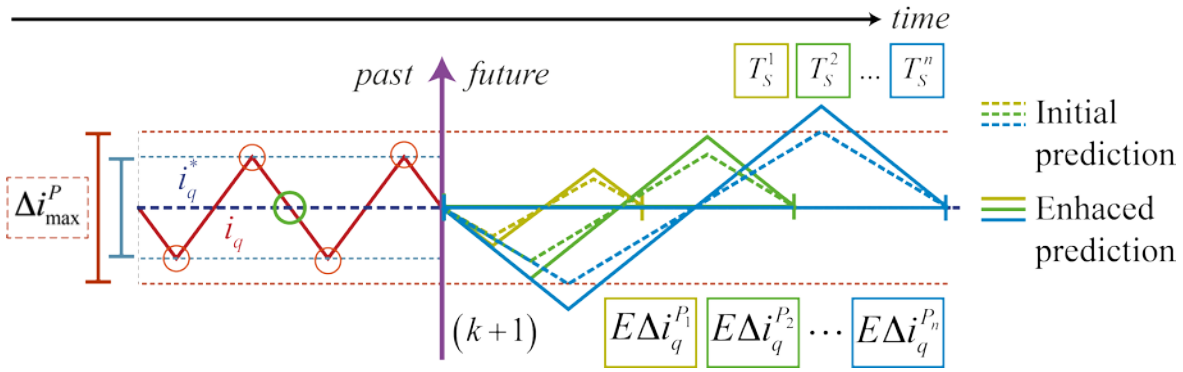

Figure 4.15: Enhanced prediction of the i_q current ripple.

The idea behind this algorithm is to make the prediction of the current ripple more accurate and make the Δi_{max}^P limit a *hard* limit i.e. the i_q ripple cannot be greater than the desired maximum limit. In this way, the initial approximation of the i_q current ripple can be corrected by a real-time measurement of the ripple. In some cases, the prediction alone is smaller or bigger than the real value, and thus the optimal switching frequency is not selected. With the aid of the real-time correction the precision of the prediction

increases greatly. The real-time execution of the ripple correction algorithm can be made possible thanks to the high computational power and parallel processing capabilities of the FPGA. The algorithm is implemented in silicon, taking 24 clock cycles to be executed.

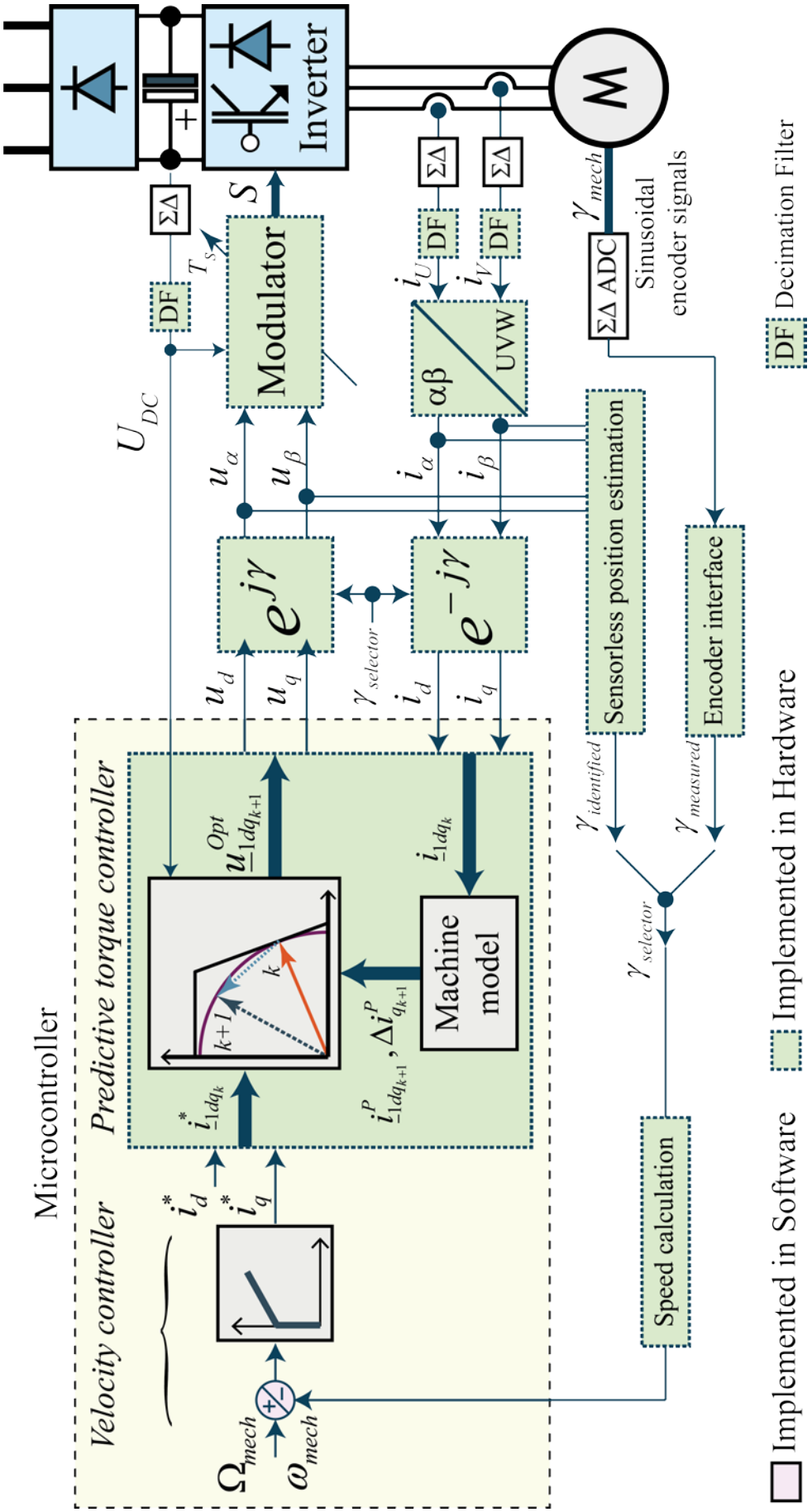

Figure 4.16: FPGA-based implementation of the proposed controller.

4.2.5 Implementation of the proposed FPGA-based Control

The parallel processing capabilities of the FPGA and the oversampling rates of the ΔΣ ADCs can be a huge advantage to develop a high-performance implementation of the control scheme proposed here. Nevertheless, the resources in the FPGA are finite, so a compromise between performance and implementation size (number of utilized cells) has to be done; otherwise, the controller will not fit the FPGA target. For this work, the chosen hardware description language was VHDL and the target FPGA device is a Cyclone V E from Altera, which is a low-cost FPGA family [116].

The FPGA-based implementation of the proposed controller is presented in Figure 4.16, the main blocks such as the predictive torque controller and the modulator are implemented in digital circuits. A soft-core microcontroller unit, the NIOS II [120] was also used so that functions with less priority can be performed. Some of these functions are for example: The speed control, the determination of the initial position of the rotor or the other computations that are not so critical for the control.

Hence, the resources for these functions are minimized and shared because they are programmed in software and executed by the hardware allocated to the microcontroller unit. The general purpose microcontroller can be enhanced using a Hardware Custom Instructions [122]. They can be integrated into the sequential execution process of the NIOS II microcontroller although their execution is performed in a digital circuit. This is the reason why the predictive torque controller block appears inside the microprocessor unit as presented in Figure 4.16.

The last stage of the ΔΣ ADCs is a decimation filter (DF) (see section 3.4.2.2) that reduces the rate in which the measurements are delivered and increases the word length from one bit to several bits, which has to be implemented inside the FPGA. Finally, the modulator unit generates the interrupts for the microcontroller and maintains the system synchronized.

4.2.5.1 Compensation of the Inverter Switching Delay

In an inverter with ideal switches when the upper switch in the half bridge is turned off (negative edge in signal T_P), the lower transistor will be immediately turned on (positive edge in T_N), the same will happen every time the T_P and T_N on/off states alternate. However, in real power converters the semiconductor switches like IGBT's reach their on and off states in a certain period, called turn-on (T_{ON}) and turn-off (T_{OFF})

times. To avoid a short-circuit in the legs of the inverter, the switch-on control signal must be delayed with respect to the turn-off ones. This time delay is usually called blanking or *dead-time* (T_D), its effects [83], [89], [128] are shown in Figure 4.17.

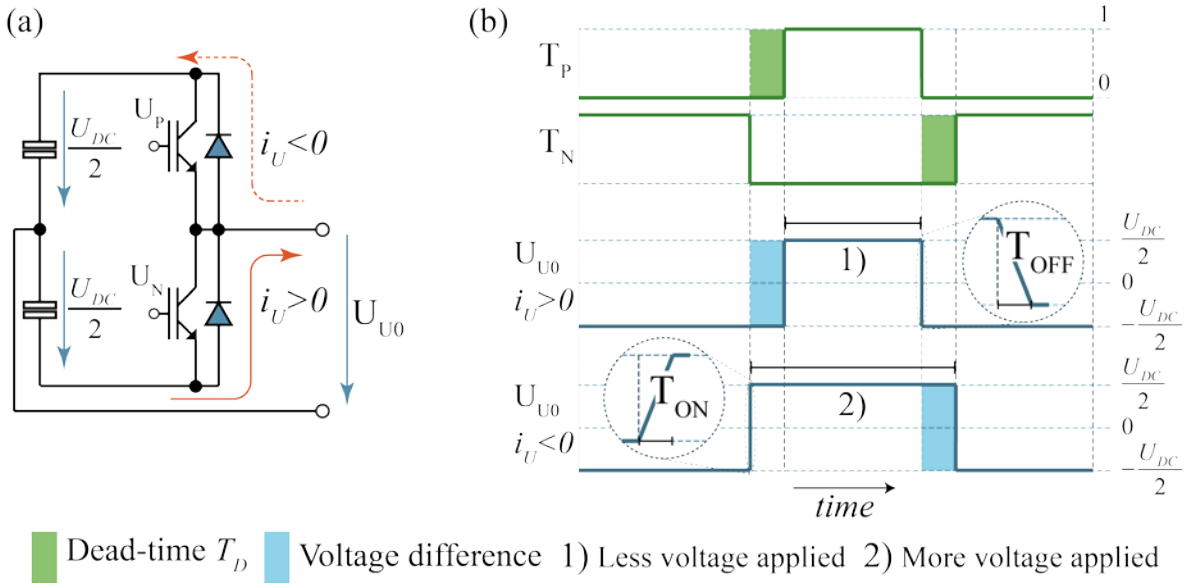

Figure 4.17: Inverter switching delay effects: a) effect of the current flow over the $u^*_{desired,U}$ and the $u^*_{real,U}$ voltage, b) effect of the dead-time on the phase voltage and its correction.

The resulting effect is a difference between the reference phase voltages $u^*_{desired}$ and the actual voltages generated by the inverter. The voltage error caused by the dead-time (T_D) on every phase can be calculated by (4.26), where $u^*_{desired,U}$ is the desired phase voltage, $u^*_{real,U}$ is the real phase voltage after the effects of the dead-time, T_{ON} and T_{OFF} are subtracted. Where T_S is the modulation period, U_{DC} is the DC-link voltage of the inverter and $sign(i)$ is the sign of phase current i_U.

$$u^*_{real,U} = u^*_{desired,U} - \frac{T_D}{T_S} \cdot U_{DC} \cdot sign(i_U) \qquad sign(i) = \begin{cases} + & i > 0 \\ - & i < 0 \end{cases} \tag{4.26}$$

Figure 4.17 depicts the real and the reference switching patters. If carefully analyzed, it can be derived that the real $u^*_{real,U}$ applied to the H-Bridge can be expressed in terms of the duty-cycle as $D_{real} = D_{desired} + D_{losses}$. Where D_{losses} is expressed in (4.28), and the duty-cycle $D_x = T_x/T_S$ is the percentage of the switching time T_S in which the signal x remains active, D_D is the duty-cycle that corresponds to the dead-time, D_{On} to the time a switch takes to get turned on and D_{Off} to the time a switch takes to get turned off. Therefore, the duty-cycles are compensated so that the effect of the losses due to switching delay get canceled as presented in (4.27), therefore, $D_{real} \approx D_{desired}$ is obtained.

$$D_{real} = D_{desired} + D_{losses} - D_{losses} \tag{4.27}$$

$$\begin{aligned} i > 0 \quad & D_{losses} = -D_D - D_{On} + D_{Off} \\ i < 0 \quad & D_{losses} = +D_D + D_{On} - D_{Off} \end{aligned} \tag{4.28}$$

4.2.5.2 Sinusoidal Rotary Encoder Interface

Another enhancement to the classical implementation of the field-oriented control is the way the encoder signals are processed. For manufacturers, it is easier to design optical encoders with output sinusoidal signals [129] rather than digital encoders with high resolutions. These sinusoidal signals are conventionally converted to squared signals and fed to counters, losing all the high resolution that is provided by the analog signals. Furthermore, if the mechanical speed of the motor is obtained by differentiating the position of the shaft, a short sampling time dramatically reduces the resolution of the measurement of the angular speed.

With the FPGA and the oversampling frequencies of the ΔΣ ADCs it is possible to implement a high-resolution measurement of the shaft position. For the implementation of the proposed encoder interface, the sinusoidal signals are measured using ΔΣ ADCs. The arctangent function was implemented using the CORDIC algorithm [100], [130]. Even though the arctangent function delivers results in the $(-\pi, \pi]$, they can be mapped to $[0, \pi)$ by adding 2π to the negative results.

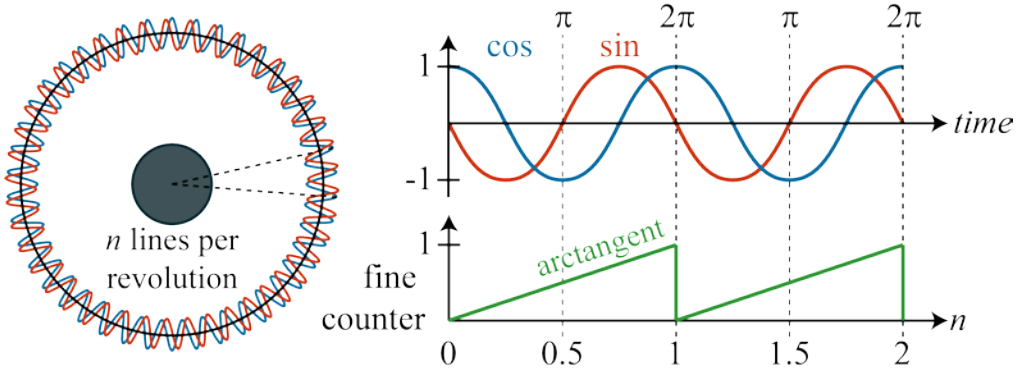

Figure 4.18: Enhanced encoder interface: resolutions increases from 12 to 23 bits.

Thus, the CORDIC arctangent function is used as a *fine* counter i.e. the counter that measures the precision of one line or increment on the encoder, which is subsequently accumulated to obtain the position of the shaft. Every time the arctangent function reaches a period i.e. at every zero crossing, a step in the count is increased (or decreased.) as shown in Figure 4.18. Therefore, every time that a new measurement of

the encoder signals is delivered by the last stage of the acquisition system, a circuit in charge of computing the rotor position is executed.

For this work, an encoder with a resolution of 2048 increments was used. With the conventional method of counting the edges of the rectangular functions, a maximum resolution of 13 bits for the position of the shaft is obtained. With this method, the sinusoidal encoder signals are sampled with a resolution of 12 bits, increasing the resolution of the position up to 23 bits.

4.2.6 Optimization of the Proposed Control Scheme

In the following, it is described how the performance of the proposed control system, introduced in section 4.2, can be optimized by implementing it in the FPGA. The performance of classical field-oriented control approach can be effortlessly improved just by implementing it in an FPGA. Thus, the bandwidth of the controller can be greatly increased leaving the power converter as the main constraint. Moreover, because of the oversampling nature of ΔΣ ADCs the measured signals can be treated as quasi-analog waveforms.

The execution of the proposed control scheme introduced in section 4.2, can be divided into four stages: *measurements*, *coordinate transformation*, *controller execution* and *modulation* as presented in Figure 4.19. Each one of these stages has an individual execution time, and the sum of them is the *total execution time*. Regardless of the implementation platform, these functions have to be sequentially executed.

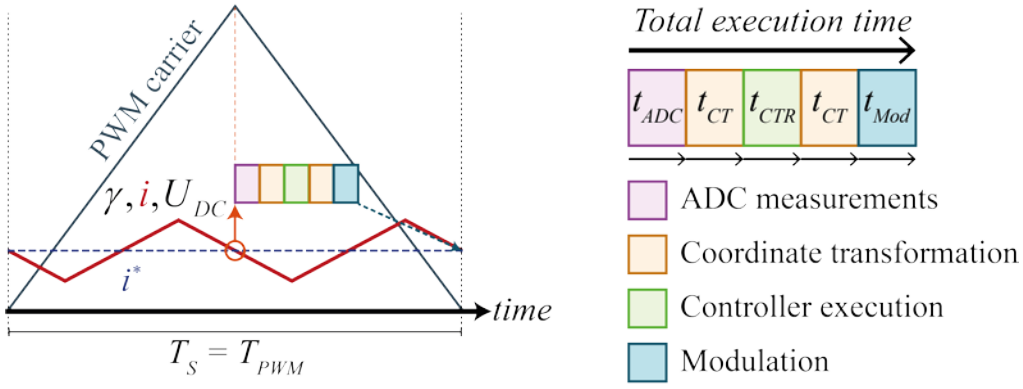

Figure 4.19: Control execution process with a software implementation.

The position of the rotor shaft γ, two phase currents (i_U and i_V for example) and the DC-link voltage U_{DC} are measured at the middle of the sampling time T_S. Afterwards, the required coordinate transformations are carried out, in order to execute the control in the rotating reference frame. Then, the outputs of the controller are transformed back

to the stator-fixed reference frame so that the PWM can be executed and the corresponding switching signals sent to the inverter.

The performance of the control can be affected because its outputs are transformed to the stator reference frame with a measurement that is made in the middle of the T_S but take-effect at the beginning of the next T_S remain constant during the whole period. This effect can be especially noted when the machine is operated at high speeds and the position changes faster. However, this delay can be compensated [22] by:

$$\gamma_{back} = \gamma_{measured} + \alpha \cdot \omega \cdot T_S \tag{4.29}$$

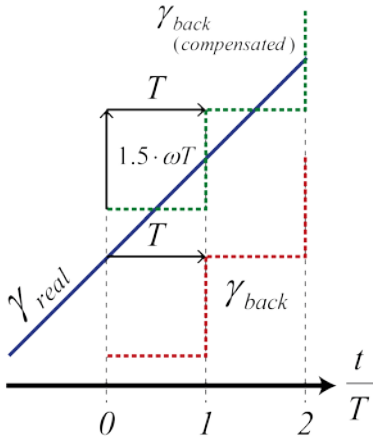

Figure 4.20: Correction of the rotor position.

Here, γ_{back} is the estimated position that will be used in the backward coordinate transformation and α is a factor of the sampling time, which in this case if $\alpha = 1.5$ the future value of the position at the end of the next T_S can be estimated, as shown in Figure 4.20. However, the horizon time of this estimation can be reduced because of the oversampling of the rotor position.

By implementing the proposed control scheme on an FPGA, the process in which the control is executed can be optimized and the execution time reduced. Since some of the functions are implemented in digital circuits and concurrently executed, which allows to decentralize and to lessen the computational work made by the microcontroller unit that is conventionally in charge of performing all these calculations. Besides, the ΔΣ ADCs generate a continuous stream of measurements.

For the implementation of the proposed control scheme, the encoder signals, the currents and the DC-link voltage are measured using ΔΣ ADCs. Thus, every time that a new measurement of the encoder signals is delivered by the decimation filter, a circuit in charge of computing the rotor position is executed. The same process is done with the

current measurements and the coordinate transformations, which are executed when a new measurement is obtained.

The execution process of the proposed control scheme can be seen in Figure 4.21. The PWM carrier function is synchronized with the ADC measurements (see section 3.4.2.2), so no phase delay is introduced in the current measurement. The oversampling process of the rotor position is used to reduce the delay introduced by the backward coordinate transformation. The decimation factor for the encoder signals is of 16 and for the current is of 32, which means that if the ΔΣ ADCs operate at 10 MHz, the sampling frequencies are 1.6μs and 3.2μs respectively (or 625 kHz and 312.5 kHz).

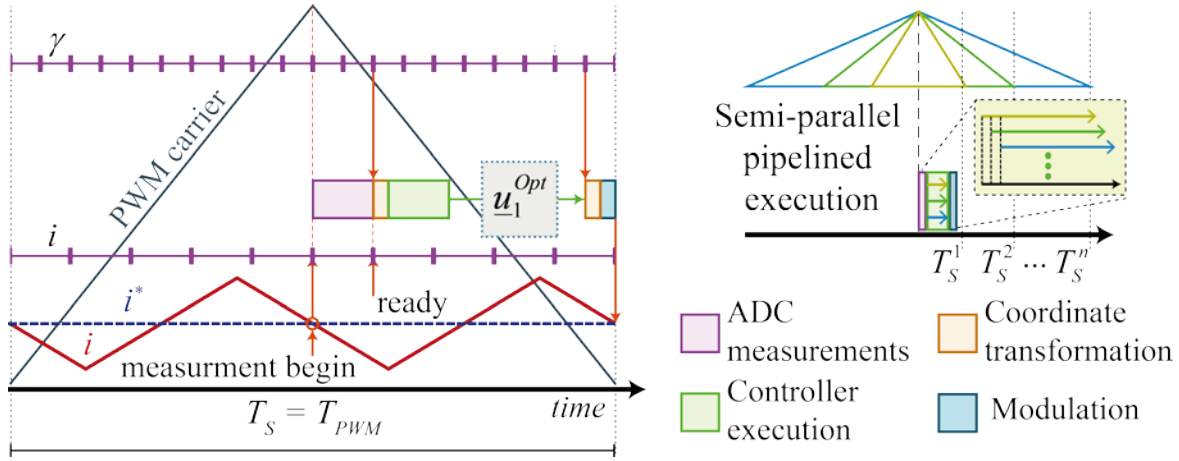

Figure 4.21: Control execution when implemented in an FPGA.

Because the rotor position is being oversampled, it is possible to reduce $\alpha = 1.0$ in (4.29). Finally, there is just enough remaining time to execute the modulation process and send the signals to the inverter before the next switching period begins. By performing the control in this way, the switching frequency can be dynamically changed without driving the PMSM unstable.

$$T_E = N_u \cdot T_u \tag{4.30}$$

$$T_E = \left(N_P + N_u\right) \cdot clk_{FPGA} \tag{4.31}$$

The semi-parallel pipelined structure of the predictive torque control presented in Figure 4.22, can receive new information every clock cycle of the FPGA. If the algorithm would be executed sequentially, the total execution time T_E can be obtained by (4.30), where N_u is the number of voltage space phasors with different switching times to be evaluated, and T_u is the time it takes to evaluate one of this voltage space phasors sequentially. In the case of the semi-parallel pipelined execution, the T_E can be

obtained by (4.31), where N_P is the number of iterations of the pipeline and clk_{FPGA} is the period of the FPGA clock.

With a clk_{FPGA} of 20ns (50 MHz) the T_u is 1.5µs (75 iterations), if N_u is 32, it could take T_E = 48µs to execute the algorithm sequentially. With the proposed architecture the T_E = 3.84µs i.e. if N_u is 32, then if N_P is 160 iterations (or 3.2µs). Furthermore, in the practical implementation, the clk_{FPGA} was reduced to 15ns (66.666MHz) achieving an execution time of the predictive torque controller of 2.88µs. In this way, the total execution time of the algorithm is reduced allowing the controller to achieve a higher bandwidth than using a microcontroller for the implementation.

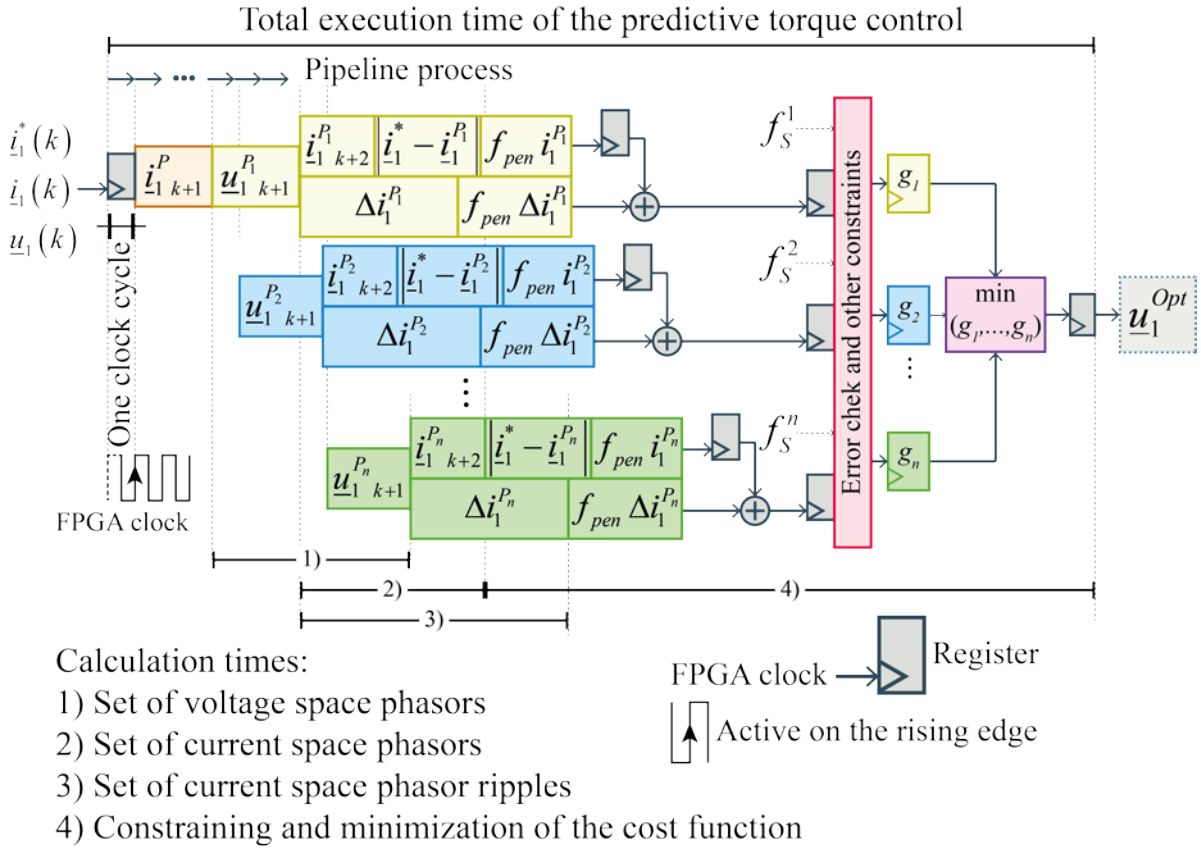

Figure 4.22: Pipelined execution of the predictive torque controller

4.2.7 Reconfigurable Pulse-width Modulation Unit

The proposed predictive control scheme with variable switching frequency technique demands pulse-width modulation units capable of being reconfigured during the execution of the controller for its implementation. Therefore, a pulse-width modulation (PWM) unit, whose switching frequency and duty cycle can be reconfigured on-demand is presented in the following. Thereby, the conventional space phasor

modulation technique for a two-level VSI as previously explained in section 2.4 is implemented.

Because of the nature of the SPM technique, symmetrical PWM are used, with the advantage of a lower harmonic distortion. The switching period counter of a conventional PWM unit was modified so that it can be reconfigured at the beginning of each period. Considering that the switching frequency of the inverter is in the range of tens of kilohertz while the working frequency of the FPGA is of several megahertz, i.e. $f_S \ll clk_{FPGA}$ and therefore the switching period can be expressed in FPGA clock cycles as:

$$T_{Scc} = \frac{T_S}{clk_{FPGA}} \tag{4.32}$$

For this work, the maximum switching frequency of the inverter is 20 kHz, and the frequency of the FPGA clock is 66.666 MHz. The controller has a resolution of around 11.7 bits or approximately 180 mV with a DC-link voltage of 600V. The advantage of developing the PWM in an FPGA is that the designer has control over all the signals. It is also possible to execute different processes simultaneously or in parallel. Therefore in one process the switching period counter sums the FPGA clock cycles until it reaches the limit T_{SccLim} corresponding to the switching frequency currently active.

If the switching frequency is to be changed, then a new value must be available before the counter resets. A second process will be activated every time the switching counter is reset or $T_{Scc} = 0$ and will load the new value of T_{SccLim} this way it is ensured that the switching counter limit does not get changed any other instant which could result in a dangerous operation of the inverter. The minimum and maximum switching frequencies of the inverter are limited to the range $0.8 \leq f_S \leq 20$ kHz.

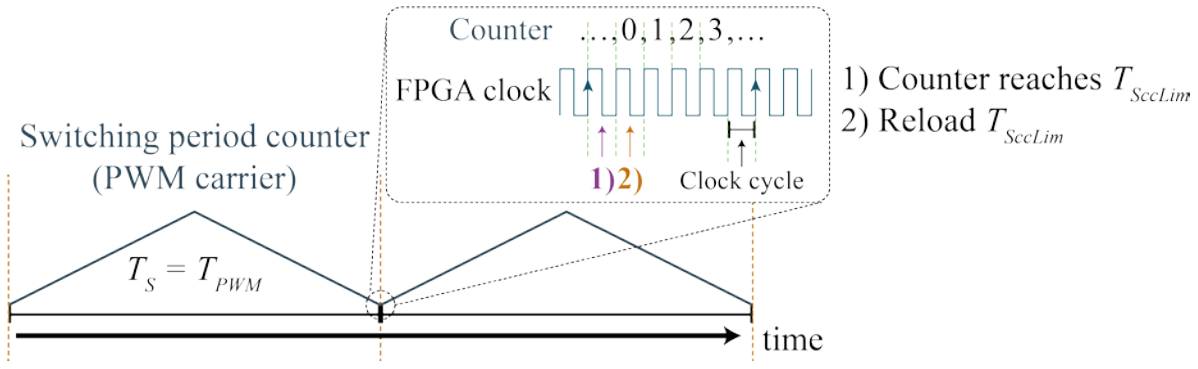

Figure 4.23: Changing the switching frequency during operation by reloading of the switching period counter limit.

With this easy but very efficient modification, the switching frequency of the inverter can be changed without stopping the execution of the controller in order to reconfigure the PWM units (as depicted in Figure 4.23) and therefore new possibilities of control techniques can be created. The resulting operation is expected to be as presented in Figure 4.24a where the condition $T_S(k-1) \neq T_S(k) \neq T_S(k+1)$ is fulfilled.

However, a problem arises when the current reference changes and the system is working at a low switching frequency. When a transient occurs, a delay in the current response is introduced as shown in Figure 4.24b. Most of the time, these transient states mean that the change in the current reference is greater than the i_{THLD} limit and therefore a higher switching frequency is desired because the real value will reach the reference value faster (see section 4.2.1). However, the switching frequency will not be changed until the end of the current switching period. Conversely, if the PWM counters could be reconfigured on-the-fly this delay could be significantly reduced.

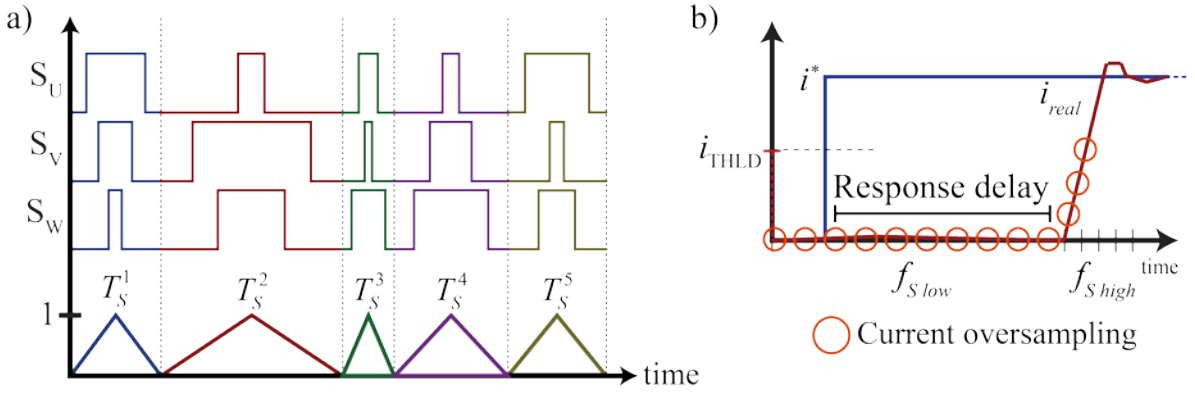

Figure 4.24: Dynamic change of the switching frequency of the PWM: a) Different switching periods b) Delay of the current when a low switching frequency is operating

Because of the current oversampling, the torque controller output can be updated every 3.2 µs; therefore a delay in the response can be considerably reduced. The on-the-fly reconfiguration of the PWM units due to the *sampling time*, is triggered by a process that is permanently sensing if the value of i_q^* is bigger than the i_{THLD} limit. A flow diagram presented in Figure 4.25a, describes the on-line *sampling time* reconfiguration process of the PWM. While changing the switching frequency at the beginning of each period can be done with relative ease, changing it on-the-fly has to be done through an error-checking procedure to ensure that it can be safely performed, otherwise the system could be severely damaged.

During this error-checking procedure, the actual switching frequency is verified and in case that the maximum frequency is active then the PWM will not be

reconfigured. Once this procedure is finished, then the process will wait until a zero voltage space phasor is active to continue the execution. Regarding the SPM technique, zero voltage space phasors are present at the beginning, center, and end of the switching period.

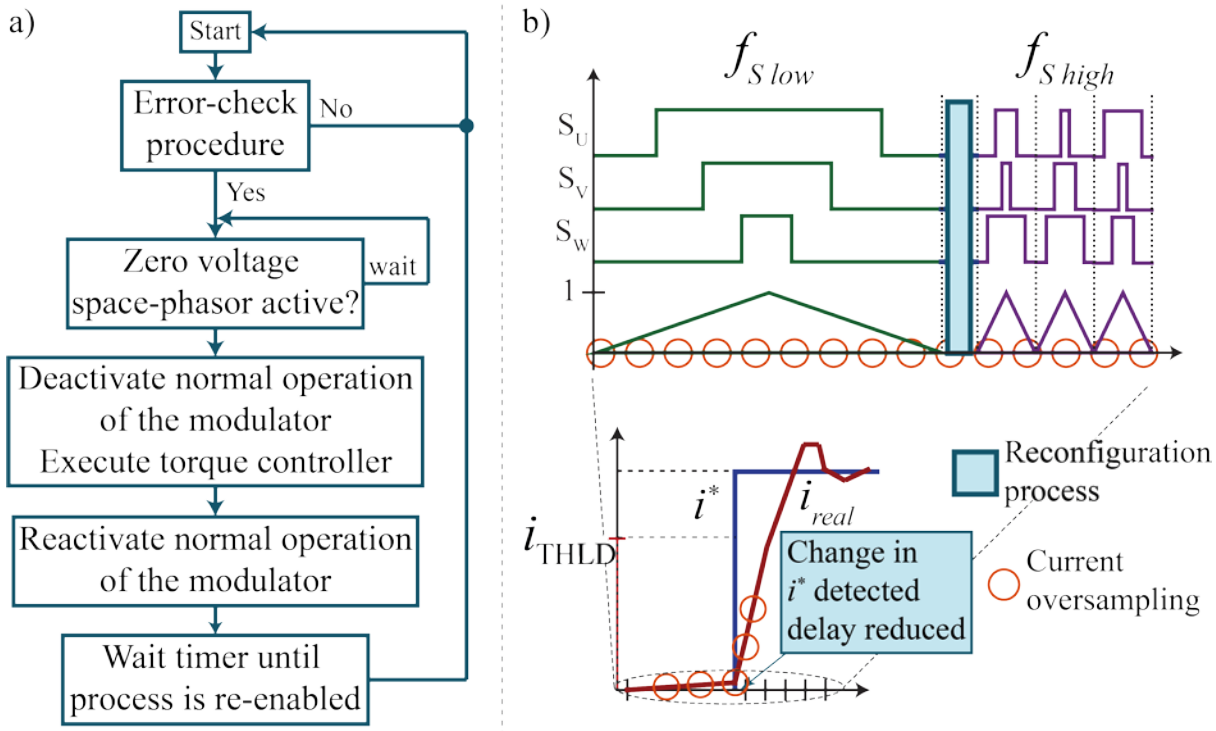

Figure 4.25: On-the-fly *sampling time* reconfiguration of the PWM units: a) flow diagram of the process b) graphical description of the switching signals and the expected current response.

Once the modulation process reaches an active zero voltage space phasor, then the normal operation of the modulator will be deactivated, and the predictive torque controller will be executed. The first sampling period following an on-the-fly *sampling time* reconfiguration of the PWM units constraints the torque controller to a 50 µs switching period, to ensure a high dynamic response.

As soon as the response of the predictive torque controller is obtained, the operation of the modulator will be reactivated. Following this, the SPM circuit will be executed again. Finally, a timer will be started, disabling the *sampling time* reconfiguration process of the modulator for a predefined period. This reconfiguration process is coded in a state-machine inside the FPGA, taking a few FPGA clock cycles to be activated. It is expected that the biggest delay (in the order of µs) be introduced by the microcontroller that executes the predictive torque controller circuit after a software interrupt, still reducing the response delay of the controller by a huge margin.

4.3 Sensorless Control

Sensorless AC drives or drives that work without the need of a mechanical angular position sensor to obtain the position of the rotor shaft have several advantages. The hardware can be simplified and therefore, the reliability can be increased and less maintenance requirements are needed. Furthermore, a motor without mechanical position sensor can work in hostile environments [2]. As mentioned in section 1.3.2, sensorless control schemes have been thoroughly researched in the past decades. The most recent efforts on this topic of research are the development of sensorless drives with a full range of operation.

Fundamental wave models can be applied in the high-speed region. However, at low-speed ranges the problem is that they become unobservable at zero stator frequency and therefore a *signal injection* technique has to be used to cover the whole speed range of a sensorless drive for AC machines. Regardless of all the disadvantages, fundamental wave models have become the most accepted ones by the industry and implemented in almost any commercial available sensorless drive [3].

4.3.1 Fundamental Wave Models

Sensorless control was mainly originated as a step forward in high dynamic vector control, i.e. field-oriented control or DTC for AC machines. Both methods are based on the knowledge of the stator flux or rotor positon in case of the PMSM. Therefore, the first proposals of field-oriented control were followed by the necessary procedures to determine the fluxes with and without sensors. One of the first proposed solutions was the integration of the stator induced voltages (4.33), which delivers the stator flux and the subsequent calculations lead to other state variables.

$$\begin{aligned} \underline{u}_1 &= \underline{i}_1 \cdot R_1 + \frac{d\underline{\psi}_1}{dt} \\ \underline{\psi}_1 &= \int \left(\underline{u}_1 - \underline{i}_1 \cdot R_1 \right) dt \end{aligned} \tag{4.33}$$

As depicted in Figure 4.26, this structure based on the induced voltages is the foundation of the *fundamental wave* sensorless schemes [3]. At the first look, obtaining the stator fluxes through the voltage model of the machine appears to be a quite straightforward operation. Nevertheless, as thoroughly studied in [22] the open-loop integration lead to drift effects, as any DC component at the input will accumulate at the

integrator output. DC components at the input may be caused by the measurement circuit and as much as it is desired, cannot be avoided in practice.

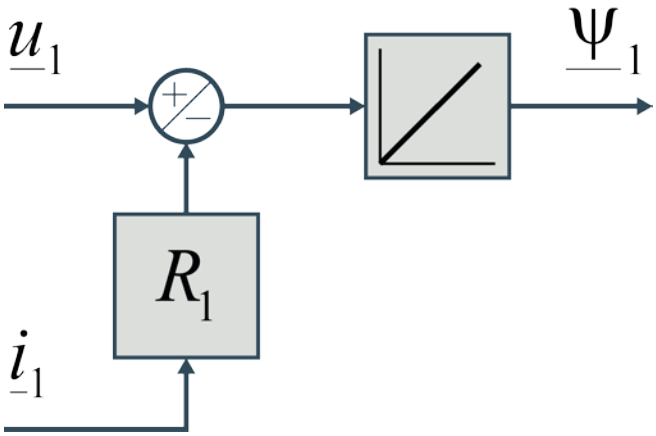

Figure 4.26: Diagram of the voltage model used on the fundamental wave models in sensorless control.

4.3.1.1 Voltage Model Correction through Low-pass Filter

A common solution to the drift effect is to replace the open-loop integration with a negative $1/\tau$ feedback as shown in Figure 4.27a that behaves as a first-order low-pass filter as shown in Figure 4.27b. The frequency response of an open-loop integrator and a first-order low-pass filter are shown in Figure 4.28 [131], [132].

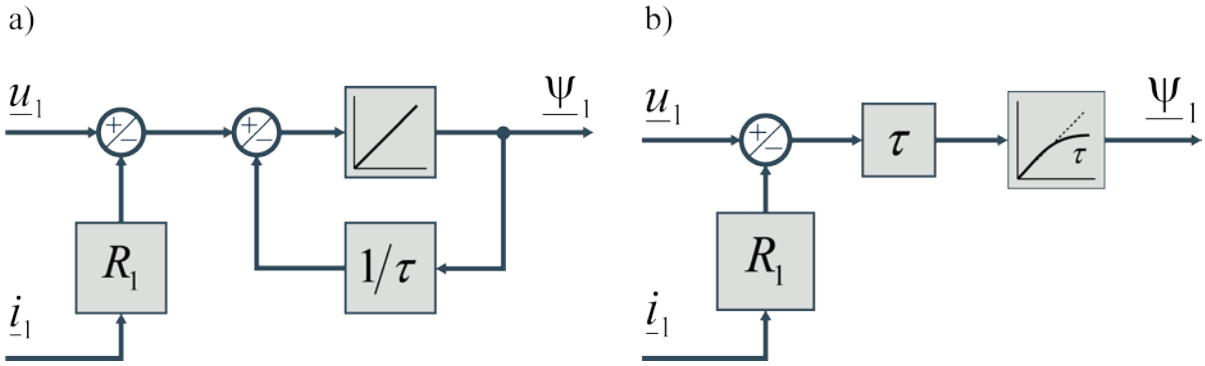

Figure 4.27: Diagram of the voltage model corrected through a) constant $1/\tau$ feedback, b) equivalent first order low pass filter.

The low pass filter works as an integrator at frequencies much higher that the $1/\tau$ cut-off frequency of the filter. However, the model becomes imprecise when the frequency gets closer to the cut-off frequency of the filter. When $\tau = 1$, the obtained flux will differ strongly from the actual machine flux for frequencies below 1 Hz and the control may become unstable. If the damping effect is increased, like in the case of $\tau = 0.1$ the model will only work at frequencies above 10 Hz.

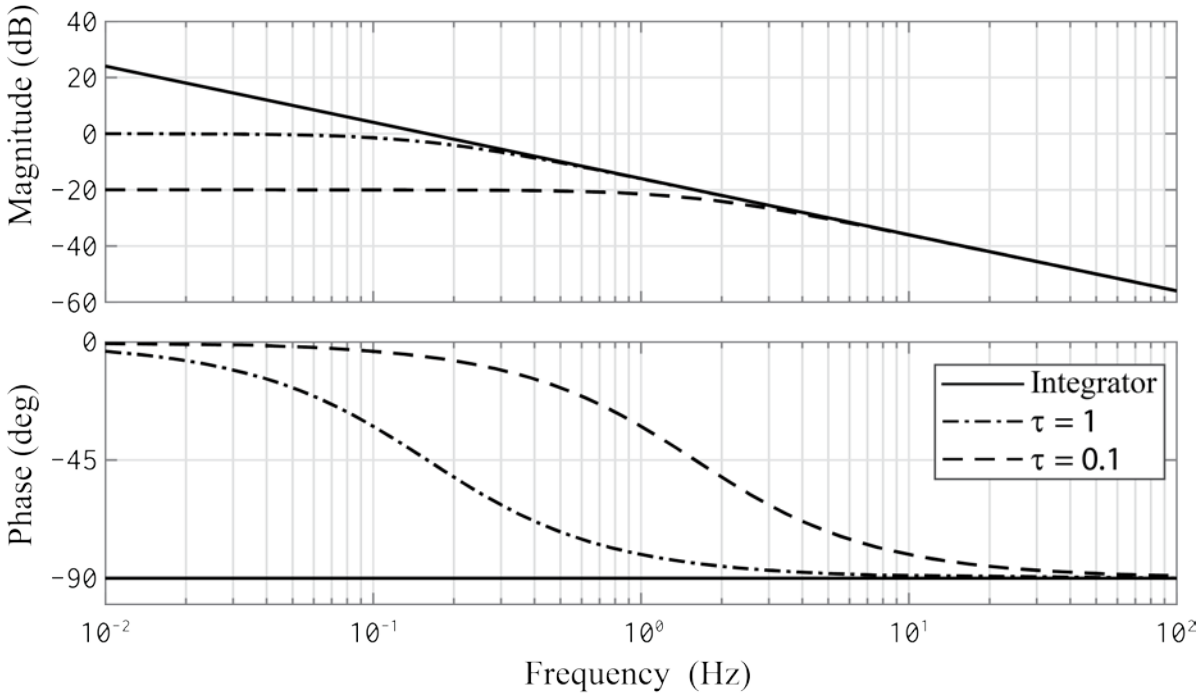

Figure 4.28: Bode diagram comparison of an open-loop integrator and a first-order low-pass filter with different cut-off frequencies.

4.3.1.2 Voltage Model Correction through the Current Model

Another possibility to correct the drift on the voltage model is to use the current model, which it does not presents diverting problems. The system presented in Figure 4.29 was proposed by Harnefors in [133] and later improved by Feuersänger et al. so that the position of the rotor can be estimated and during sensorless operation of electrically excited synchronous machine (EESM) drives, the detailed explanation of this technique can be found in [134]. However, this system has the advantage to work with any synchronous machine, since the voltage and current models used for the estimation can be adapted to the specific machine.

The correction is achieved using a model reference adaptive system (MRAS) in which normally the voltage model works as the reference model and the current model gets adapted. The rotor position γ_i required by the current model is calculated by a phase-locked loop (PLL). This PLL aligns the stator flux space phasor derived from the current model $\underline{\psi}_{1,I}$ with the reference stator flux obtained from the voltage model $\underline{\psi}_{1,U}$ by tuning the estimated rotor position as shown in the space phasor diagram in Figure 4.29.

In this way, the rotor position is identified based on the utilization of both: voltage and current model of the machine. Additionally, in this method the voltage model is slightly corrected by the current model to suppress any drift effects of the integration.

This correction affects only the amplitude of the space phasor of the stator flux, but not its phase. In other words, the drift of the voltage model is compensated by the drift free current model. The current model needs to be tuned by the PLL by gaining the angular information from the voltage model [134].

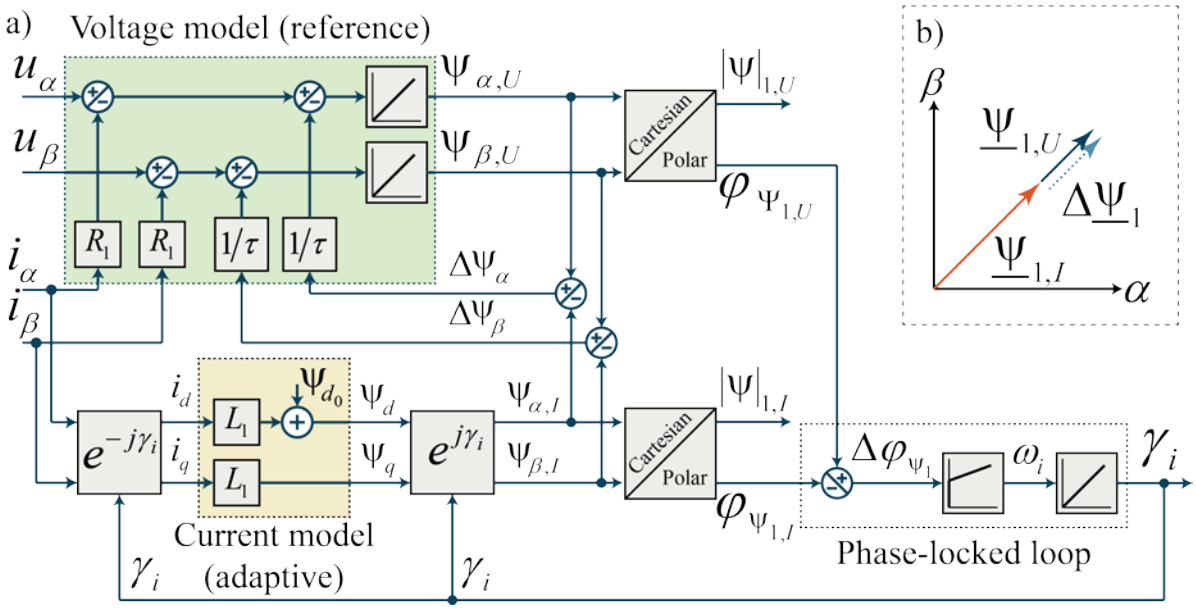

Figure 4.29 Correction of the stator flux by the voltage model through the current model in sensorless operation.

4.3.2 Sensorless Control at Low-Speed Range without Signal Injection

In a conventional implementation of the voltage model, the voltage space phasors obtained from the PWM signals in conjunction with the current space phasors measured at the middle of the sampling time T_S, are used to estimate the rotor position. Once the rotor position γ_i is correctly identified, it can be used for the execution of the control algorithm. The elimination of the rotor sensor affects directly the computational complexity of the control algorithm. However, due to the parallel processing capabilities of FPGAs the identification of the rotor position can be performed without substantially increasing the execution time of the whole control procedure.

So far, different sensorless approaches have to be combined to cover the whole speed range of an AC machine. More specifically at low-speed ranges or even standstill, the operation becomes very challenging since the induced stator voltages become almost zero and therefore fundamental wave models cannot be used for the estimation of the rotor position.

As stated in [2] the lower limit of stable operation lies in the 0.5 – 2 Hz range of the stator frequency (or 0.33 – 1.33 % of the nominal speed). However, practical

implementations fail even when operated at higher limits, between 3 – 5 % of the nominal speed or 4.5 – 7.5 Hz of the stator frequency. Therefore, for the low-speed range usually, a signal injection technique has to be used for the correct operation of the drive.

Nevertheless, it could be expected that the conventional sensorless schemes for the identification of the rotor position, based only the fundamental of the voltage and current space phasors, can be able to operate at a lower speed range because of the continuous oversampling of the currents by means of the ΔΣ ADCs combined with computational power of the FPGA. Thus, the range of operation without signal injection could be extended and the signal injection techniques would be only necessary for the standstill identification of the rotor position and for very low speeds.

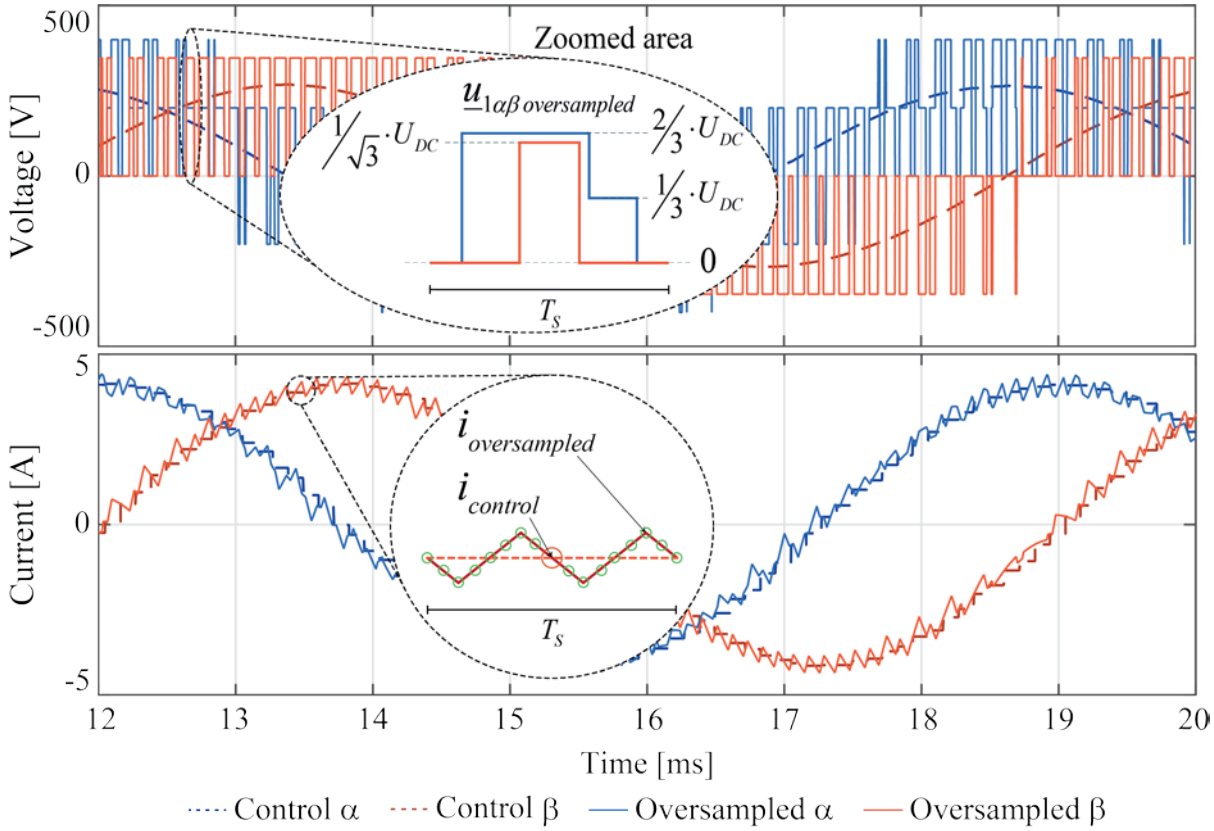

Figure 4.30: Signals used for the estimation of the rotor position.

Faster measurements have to be done if during one switching period not only the average values but also additional harmonic components are considered during the identification process. As presented in Figure 4.30, the generated voltages by the inverter that are applied to the machine are square wave signals, far from the sinusoidal form of the reference signals.

In the case of the currents the ripple is not considered in a conventional approach. While the currents can be effortlessly oversampled by the ΔΣ ADCs, the stator voltages

have to be either reconstructed by measuring the DC-link voltage in conjunction with the inverter switching signals or use the direct output of the torque controller. In both cases, the compensation of the inverter switching delays (as presented in section 4.2.5.1) is mandatory in order to avoid differences between the applied and the commanded or estimated values.

4.3.2.1 FPGA Implementation of the Sensorless Control Scheme

The MRAS sensorless control scheme as presented in section 4.3.1.2 was implemented in the FPGA; its execution takes 60 FPGA clock cycles (i.e. with a 15ns clock cycle the execution time is 0.9μs). For comparison purposes, the conventional execution of the algorithm was also realized, so the outputs of the torque controller and the current measurement at the middle of the sampling time were used, i.e. only the average values during one switching period of the voltage and current space phasors are used to identify the rotor position. This execution mode is conventionally implemented using a DSP or μC, likewise the advantages and/or disadvantages of using an FPGA for the execution of the algorithm are compared in section 5.3.

Additionally, two types of implementations with oversampling were done:

- The voltage control signals generated by the torque controller are used in conjunction with the oversampling of the currents, which happen every 3.2μs, where only the current harmonics are considered.
- The stator voltages are reconstructed by means of the voltage DC-link and the inverter switching signals, thus considering harmonics in both voltage and current, which is explained in the following.

The estimation of the stator voltages through the reconstructed voltage signals using the voltage DC-link and the inverter switching signals is commonly used in conjunction with DTC or MPC without modulator schemes, as presented in Figure 4.31. The voltage space phasor applied to the stator is one of the natural voltage space phasors that the VSI can generate and remains constant during the whole sampling time T_S. Even though the inverter switching delays can be compensated so that the commanded control values do not differ too much from the real applied voltages by the inverter, their physical effect remains present.

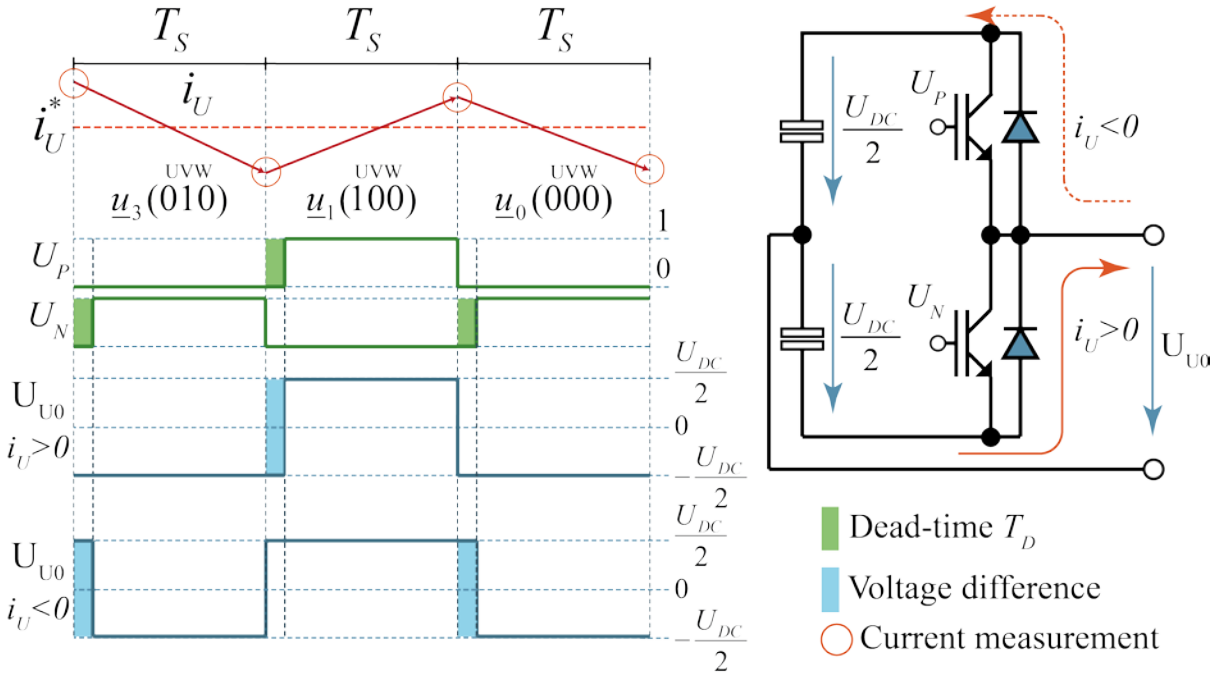

Figure 4.31: Reconstruction of one phase-to-ground stator voltage in control schemes without modulation stage.

The dead-time T_D will generate a voltage difference every time the IGBTs in the inverter change their state. Nevertheless, T_S is usually constant and this facilitates the integration of the estimated voltages over this period of time. As T_D is also generally constant, its effect can be considered during the integration or furthermore, if $T_S \gg T_D$ its effect can be neglected.

This estimation becomes more complex when a modulation technique like SPM is used, as the natural voltage space phasors of the inverter are combined to generate the desired reference voltage. In this case, the natural voltage space phasors of the inverter are combined to realize the desired voltage space phasors and thus, the active times of the switches on the inverter will not always be the same as the T_S.

One possibility is to reconstruct the stator voltages through small integration steps, as fast as the FPGA allows it whose limit happens to be the FPGA clock cycle, as presented in Figure 4.32; for this work, it is done with a clk_{FPGA} of 15ns (66.6 MHz). In this way the voltages that are being applied to the stator can be estimated in real-time and the dead-time T_D can also be considered. This can be done without losing information because the FPGA can execute several processes in parallel. A small delay is introduced, in the order of hundred of nanoseconds, because of the pipeline processing, which can be ignored without a significant loss of information of the identified rotor position.

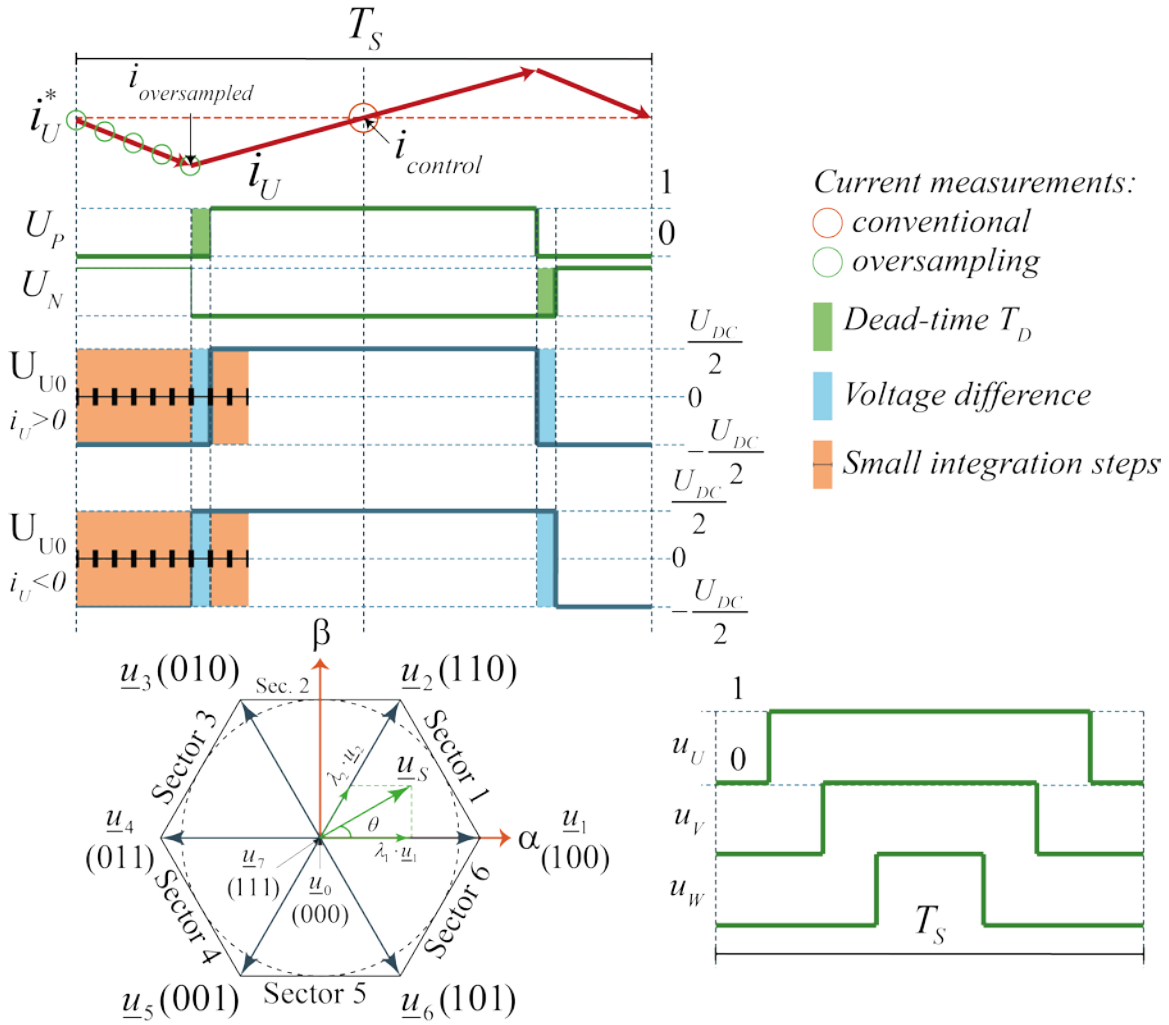

Figure 4.32: Reconstruction of one phase-to-ground stator voltage in control schemes with modulation stage.

4.4 Summary of the Chapter

In this chapter, a predictive torque control based on field-oriented control and model predictive control is introduced. The main proposal is a control scheme with a variable switching frequency of the modulation stage. For this purpose a pulse-width modulation unit was designed that allows the control the change of the switching frequency during its execution.

This control strategy aims on the one hand to have a very high dynamic response while reducing the switching losses by directly decreasing the switching frequency and on the other hand to maintain the torque ripple within given limits. An FPGA is proposed for its implementation because its parallel processing capabilities and dedicated hardware for the execution of algorithms. The control strategy is comprehensible

explained for the operation with and without rotor position encoder i.e. sensorless control.

5 Experimental Results

The proposed concepts for the realization of a sensorless predictive control of a PMSM with variable switching frequency, introduced in the previous chapters were experimentally verified. In the following, the laboratory setup and later the experimental results are explained in detail.

5.1 Experimental Setup

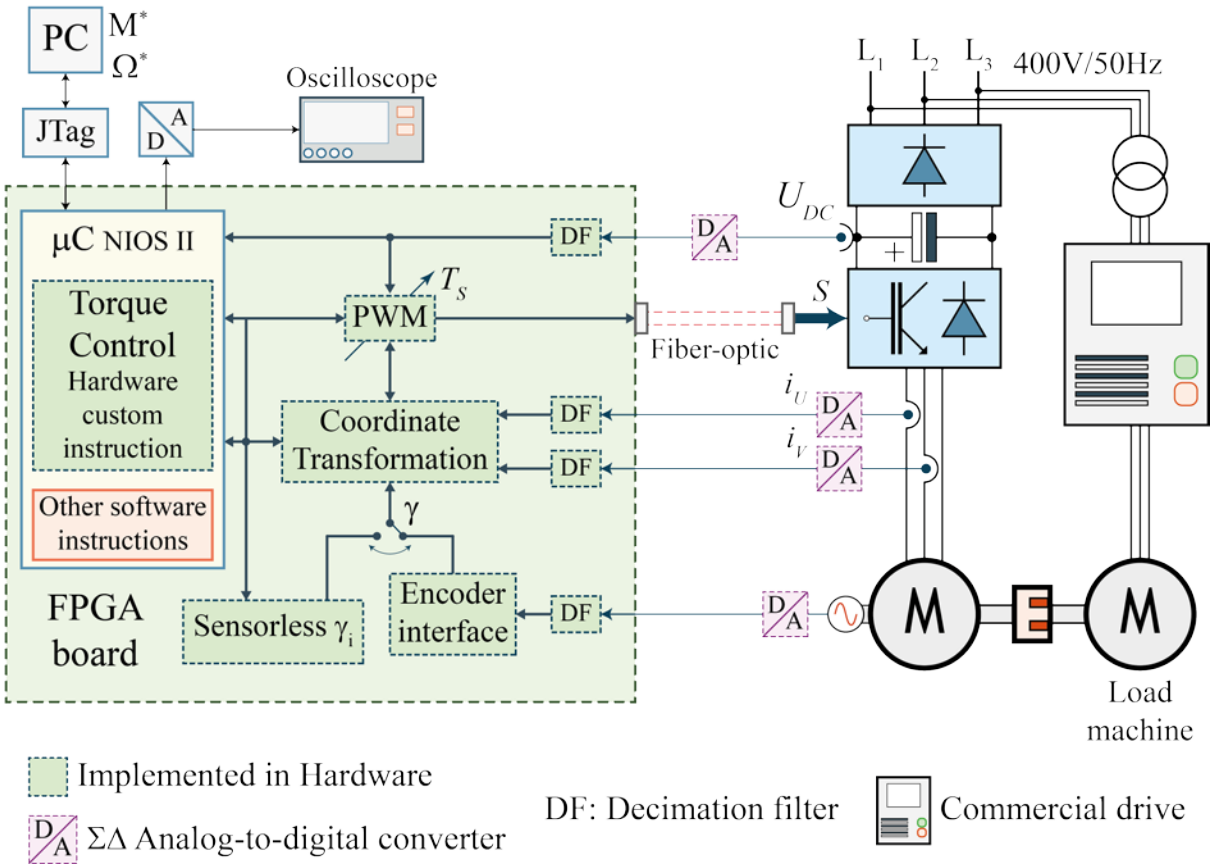

Figure 5.1: Block diagram of the experimental test-bench.

A simplified diagram of the experimental setup is shown in Figure 5.1; the proposed control scheme was digitally implemented by using in a DBC5CEFA7 commercial board [135]. This board is based on the low-cost Cyclone V FPGA model 5CEFA7F27C7N from Altera [116] and contains several peripherals as well as general purpose input-output (GPIO) pins. Through these GPIO pins, additional external boards are connected to extend the I/O capabilities of the FPGA: two ADC boards, each one containing four channels making a total of 8 ΔΣ ADCs channels and a DAC board with eight 16-bit 5V output channels. The FPGA programs are transferred to the target

hardware through a JTag interface, which provides some basic bidirectional communication for debugging of the microcontroller code and monitoring of the system.

The core of the torque control is also fixed in a digital circuit, which can be accessed through the microcontroller (see section 4.2.5). The modulator of the ΔΣ ADC is an additional chip outside of the FPGA while the decimation filters are implemented inside. The DAC board is also outside the FPGA; it converts digital information into analog signals that can be displayed on an oscilloscope. To capture and display the signals inside the FPGA, the software SignalTap II [136] from Altera was used. It makes use of some resources inside the FPGA, like RAM memory and registers among many others, to capture signals which are predefined by the user, and later transfers them through the JTag and display them on a computer.

For the measurement of the angular position of the shaft a sinusoidal incremental encoder ROD486 with 2048 positions per revolution [129] is attached to the shaft of the PMSM. The stator currents are measured with two Hall–effect current transducers (LEM LT100-P) placed in a PCB near the FPGA board to avoid large cables. The power electronics consists of a two level inverter, which is a MIPAQ Serve model IFS150V12PT4 of Infineon [137] mounted on a heatsink with a fan for the cooling of the components. The control signals are optically isolated to reduce electromagnetic interference. The DC-link voltage was kept at a constant level of $570V_{DC}$ by an active rectifier unit.

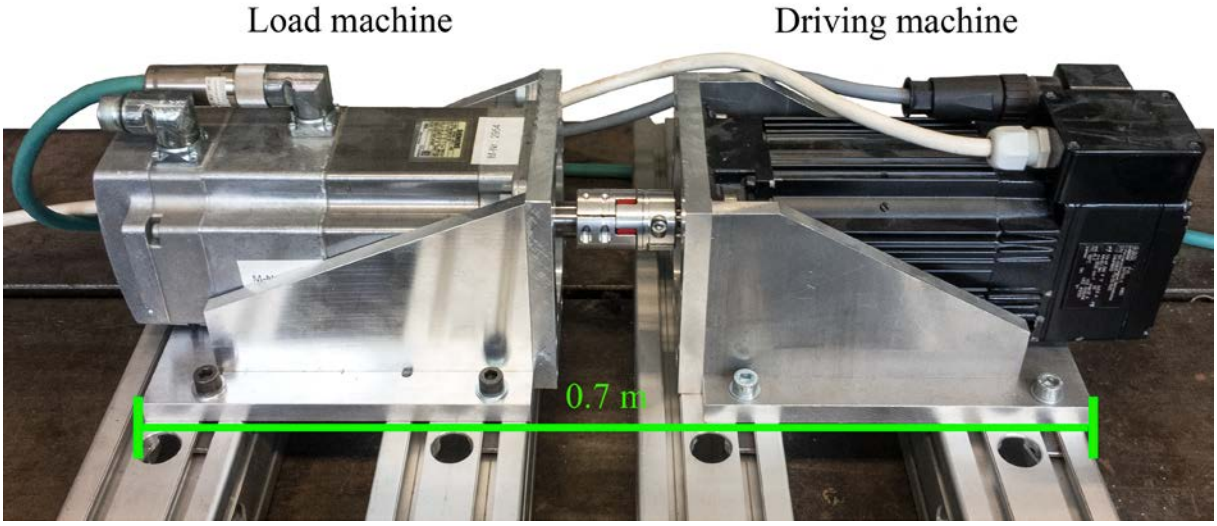

Figure 5.2: Mechanical Setup: two PMSMs mechanically coupled.

Two commercially available PMSMs were mechanically coupled as depicted in Figure 5.2. The *driving machine* (see Table 6.1 for parameters) is controlled by the

proposed control scheme developed in this work. The second PMSM (see Table 6.2), is used as the *load* or *braking machine* and is controlled by a commercial drive (parameters in Table 6.3). The parameters both PMSMs and of the commercial drive are summarized in the A.

The experimental results of the predictive control with variable switching frequency were carried out with a set of 32 switching frequencies, as presented in Table 6.4. The function that executes the torque control also returns a signal that represents the switching frequency and can be visualized with help of the DAC. In this way, it possible to know the actual switching frequency during the whole operation, as presented in Figure 6.1; this information is summarized on the B. For certain experiments, the currents were oversampled with a period of 1.6 µs.

5.2 Predictive Control with Variable Switching Frequency

The proposed predictive control scheme with variable switching frequency introduced in section 4.1 was implemented as explained in section 4.2.5. In order to examine the response of the current control loop, the transformation angle was set to zero i.e. $\gamma = 0$. Therefore, the control is made with the machine in standstill in the revolving *d,q*-reference frame. A step with a duration of 12 ms was applied to the i_q^* from zero to the nominal current (4.1A) while the i_d^* current reference remains zero.

5.2.1 Current Response against conventional FOC in Standstill

For all measurements at standstill a step with a duration of 12 ms was applied to i_q^* from zero to the nominal current (4.1A) while the i_d^* current reference remains zero. For the purpose of comparison, the classical field-oriented control with linear PI control was executed with switching frequencies of 1.8 kHz and 20 kHz. Figure 5.3 shows the response of the control at a switching frequency of 1.8 kHz.

As expected, the response exhibits a slow behavior with a settling time of approximately 2 ms of the i_q current. The overshoot in both cases, rising and falling edges, is around 22% of i_N. The response of the i_d current remains almost unaffected. A slow dynamic response is a tradeoff for a low fixed switching frequency, though the switching losses are kept low.

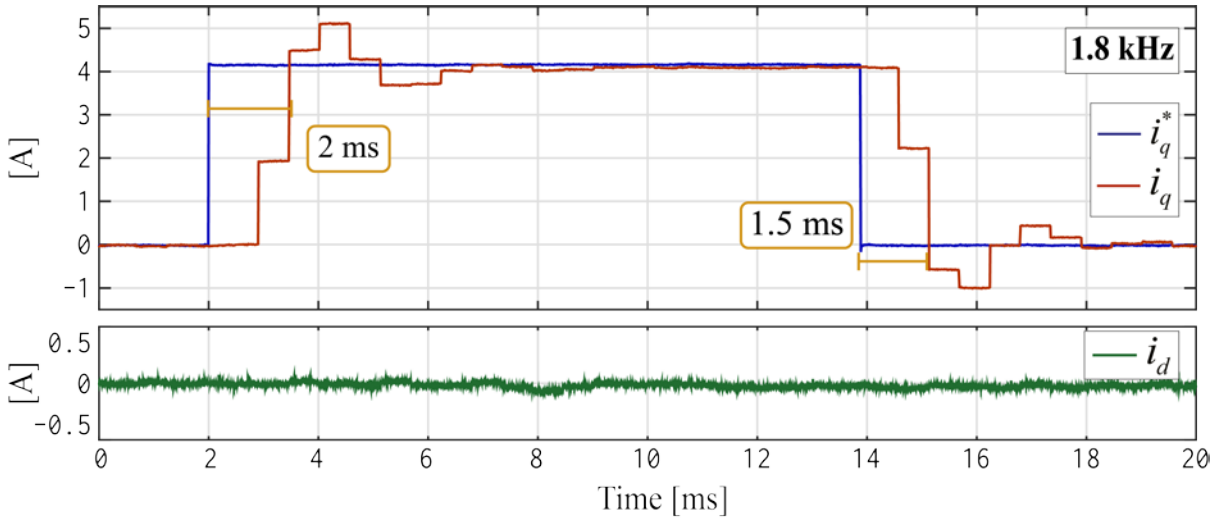

Figure 5.3: Current response of the classical field-oriented control with linear PI-Control at 1.8 kHz.

As presented in Figure 5.4, an increase of the switching frequency to 20 kHz leads to an improvement of the dynamic response. Obviously, the switching losses are also directly affected but the settling time is reduced to 200 µs, the overshoot is smaller but still present and around 17.78% of i_N.

The response of the i_d current remains almost unaffected, yet it shows a low frequency oscillation, which also in some cases appears in i_q, and will be present in most of the measurements carried out in this work, especially pronounced in the sensorless ones. Since this effect was not present in the results of simulations, it can be conjectured that it is produced by systematical error in the hardware implementation that could not be identified in the course of the work. Imperfections in the acquisition and conversion of the currents is one of the possible sources of error that should be investigated in future works.

In the next experiment, the proposed predictive control scheme with variable switching frequency was used. It was configured with the parameters presented in Table 5.1, and the results of the measurements are presented in Figure 5.5. For this test the on-the-fly adaption of the sampling time (T_S) on the SPM circuit was not activated, i.e. if the current reference has a unexpected change, the control will not take effect until the next T_S.

The transient response obtained with the predictive control scheme with variable switching frequency is clearly better than of the one with the classical field-oriented control operating at 1.8 kHz, on the rising edge of the i_q^* current the settling time is of about 250 µs while on the falling edge it is less than 100 µs.

Table 5.1: Configuration of the predictive control scheme with variable switching frequency configuration parameters for the current control mode experiments.

Current control	w_q	w_d	$w_{\Delta i_q}$	w_{f_s}	εi^P_{max}	Δi^P_{max}	i_{THLD}
PI	1.0	0.2	1.0	1.0	0.25 A	0.5 A	0.5A
On-the-fly adaption of T_S	**Enhanced prediction of the current ripple**						
Off	Off						

The overshoot is very small, which can be attributed to the fast switching frequency during the transient state and the prediction of the current. On the rising edge its value is 5.26% and on the falling edge 11.94% of i_N, respectively. The response of the switching frequency is also as expected. During the transient the frequency increases rapidly and accordingly, will also rapidly decrease once the i_q current reaches the εi^P_{max} limit vicinity (εi^P_{max} is defined in section 4.2.1). The response of the i_d current remains almost unaffected whereas the average switching frequency increases to 2.24 kHz, which is an increase of 24.44% if compared to the 1.8 kHz used in the conventional FOC implementation.

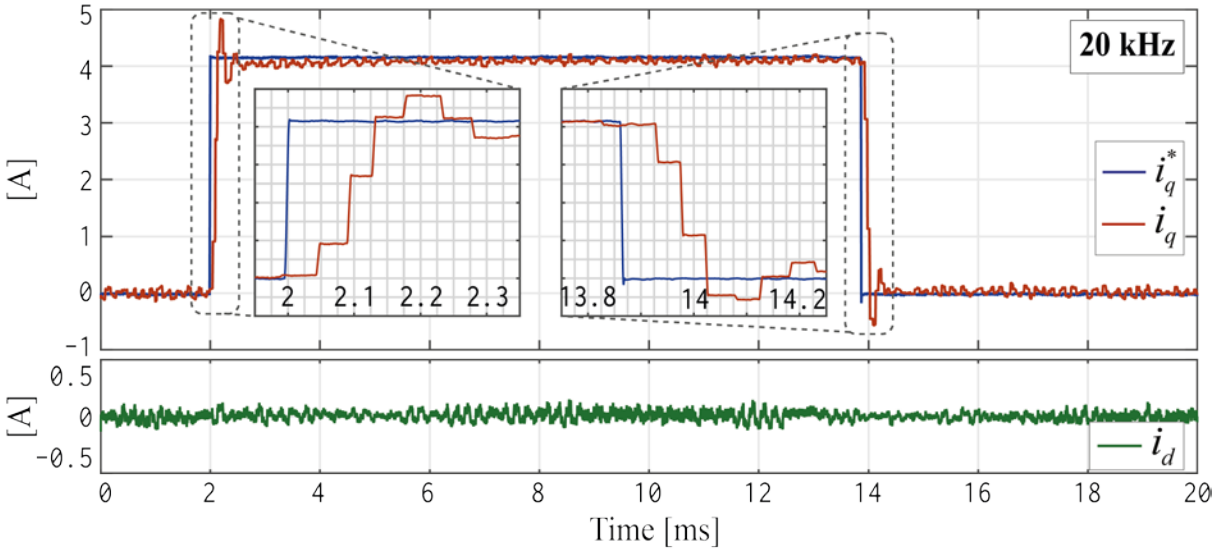

Figure 5.4: Current response of the classical field-oriented control with linear PI-control at a switching frequency of 20 kHz

This increase in the average switching frequency represents a good compromise between the improvement in the transient response and the increase of the switching losses, the desired good dynamic response is obtained, and the switching losses are kept as low as possible. Nevertheless, a delay in the response after the change of the reference signal for the current component i_q^* can be clearly noted.

On the rising edge of i_q^* it takes 1.5 ms and on the falling edge roughly 0.5 ms before the control reacts and the value of i_q begins to change. As explained in section

4.2.7, this lag in the response results due to the fact that a sudden change in the i_q^* current occurs while the system is sampling with low frequency and yields a dead time in the reaction of the control. However, this delay in the response of the control for a change in the reference channel, can be eliminated when the on-the-fly reconfiguration circuit of the sampling time is activated.

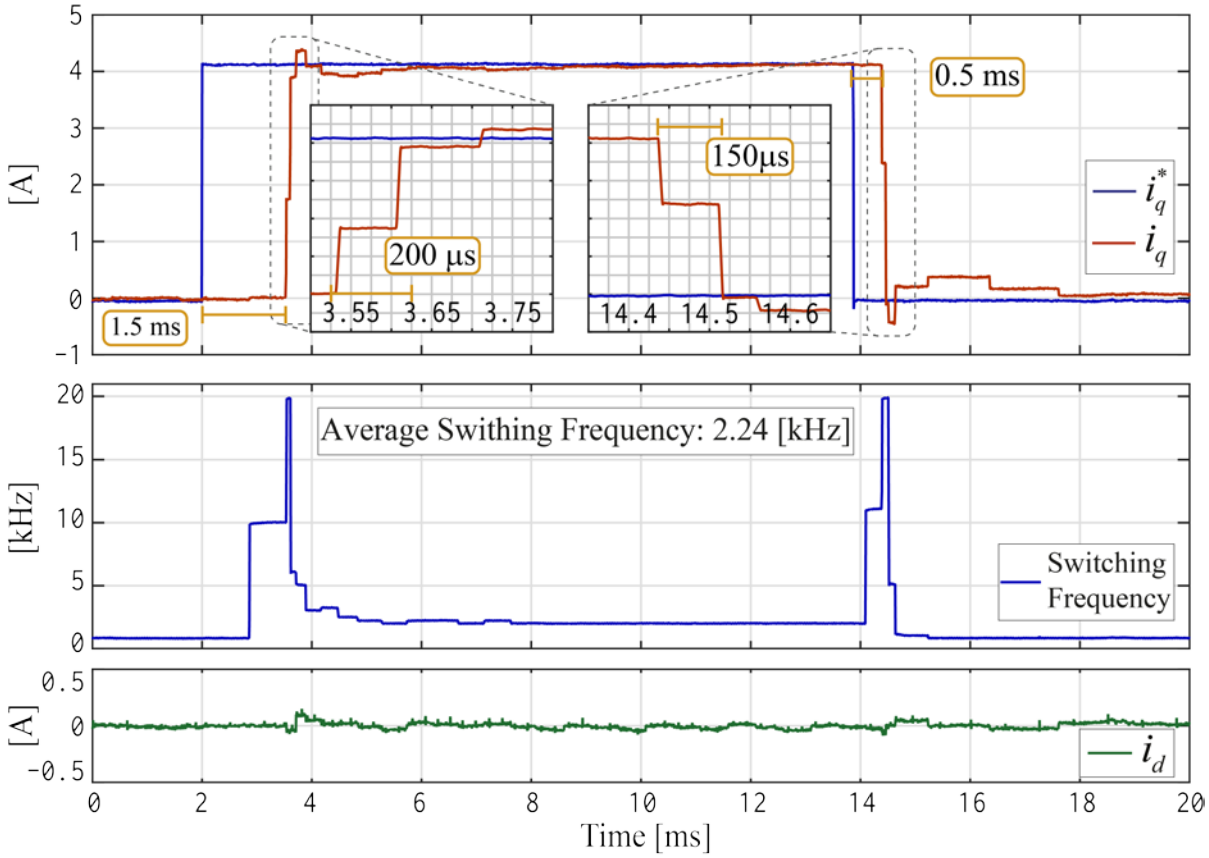

Figure 5.5: Current response of the predictive control scheme with variable switching frequency without on-the-fly reconfiguration of the sampling time.

The following measurement shows a case, in which the on-the-fly reconfiguration circuit of the PWM units and sampling time. As depicted in Figure 5.6, once the reconfiguration circuit is enabled, the response of the predictive control scheme with variable switching frequency is much faster, the reference and the real value of the current are close to each other.

The switching frequency is kept as low as possible, except when a change in the current reference greater than i_{THLD} limit (defined in section 4.2.1) is presented. During these transient states a high dynamic response of the controller is obtained, higher switching frequencies are favored by the controller, as desired. The dead time in the response during the transient state is now reduced as a result of the reconfiguration of the sampling time and increase in the sampling frequency.

As it can be seen in the zoom depicted in upper side of Figure 5.6, the waveform of the i_q current, obtained with higher time resolution by means of oversampling shows that the real current ripple is within approximately 0.3 A, slightly higher than the predicted value of Δi_q^P of 0.25 A. However, it is kept below the $\Delta i_{max}^P = 0.5A$ limit set in the control. The current overshoots are of 10.70% of i_N on the rising edge and 8.33% of i_N on the falling edge, the response of current the i_d current remains constant, almost unaffected.

The average switching frequency was 1.92 kHz i.e. an increase of only 6.66% compared with the 1.8 kHz of the classical FOC implementation. The switching losses are kept at a minimum by maintaining the switching frequency as low as possible; however, the good performance of the control is preserved and the current ripple is kept below the desired limit. Finally, a high dynamic response is obtained just when it is required, during the transient states.

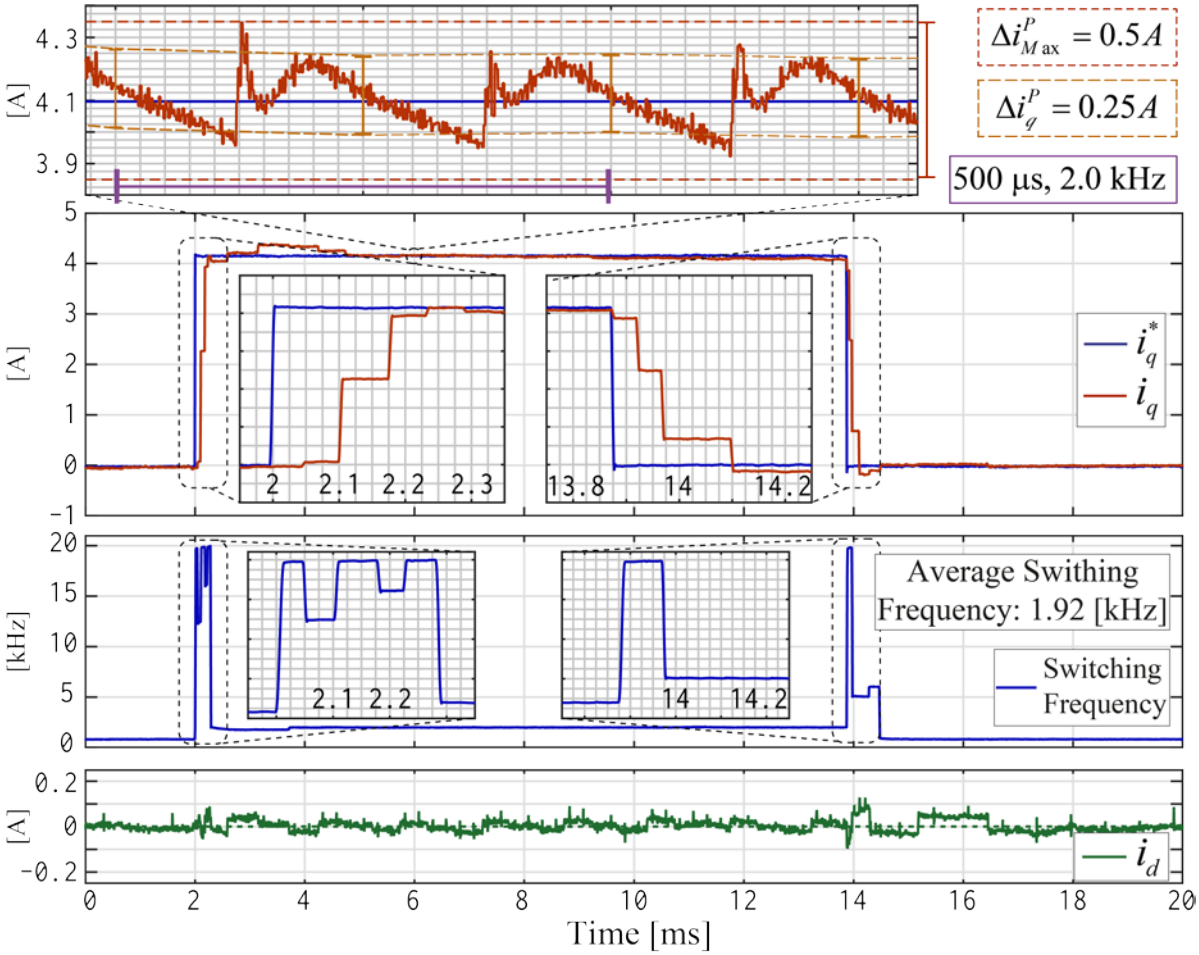

Figure 5.6: Current response of the predictive control scheme with on-the-fly reconfiguration of the PWM units and sampling time.

5.2.2 Reconfiguration of the Pulse-width Modulation Units in Detail

A close-up to the rising edge on the current reference i_q^* presented in Figure 5.7 shows that the reconfiguration process of the PWM units and the sampling time takes approximately 17.67 µs to be completed, thus, the dead time is substantially reduced. This execution time will always have slightly variations as not all the functions are implemented in hardware but also in software with a not very well defined execution time. Nevertheless, these variations are smaller than one µs.

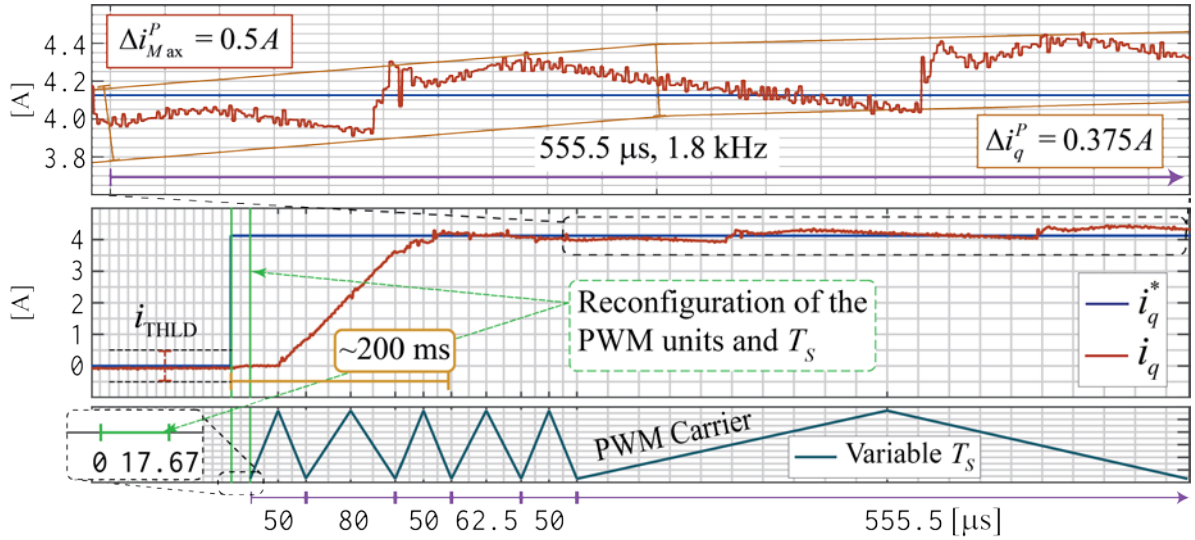

Figure 5.7: Close-up to the dynamic change in the current response of the predictive control scheme with on-the-fly reconfiguration of the PWM units and the sampling time.

After the reconfiguration process, the i_q current will reach the desired i_q^* reference value in three switching periods that combined make 180 µs. Once the i_q current reaches the reference value, the control will take some extra periods to stabilize and then the switching frequency will be reduced. Figure 5.7 also shows a closer look to the oversampled form of the current, once i_β reaches the reference value and the switching frequency is reduced to 1.8 kHz, the predictive control scheme estimates a current ripple Δi_q^P of 0.375A. This measurement corroborates that the real switching ripple is almost identical to the predicted value, furthermore, it stays below the desired Δi_{max}^P limit of 0.5A.

To validate the online reconfiguration process of the modulator, its internal signals were registered, as presented in Figure 5.8. In this test, the reconfiguration process is completed in 18.435 µs. The process begins when the reconfiguration signal receives an impulse, which means that a change in the current reference is larger than the i_{THLD} limit. Within a few clock cycles of the FPGA (75 ns), the normal operation of the

modulator is stopped, the signal that enables the modulator is set to low and a signal to trigger a control interrupt in the µC is generated.

It will then take 8.84 µs before the µC initiates to torque controller, as carefully explained in section 3.4.2.2, the synchronization of the PWM timer with the Sinc^3 decimation filter of the ΔΣ ADCs introduce a delay of 4.8 µs, therefore, the µC has to wait this time before the current measurement is available. The forward coordinate transformation process can take up to 1.5 µs to be completed when the drive operates in sensorless mode; therefore two µs are added to the delay, and finally the response of the µC to the interrupt takes approximately two µs.

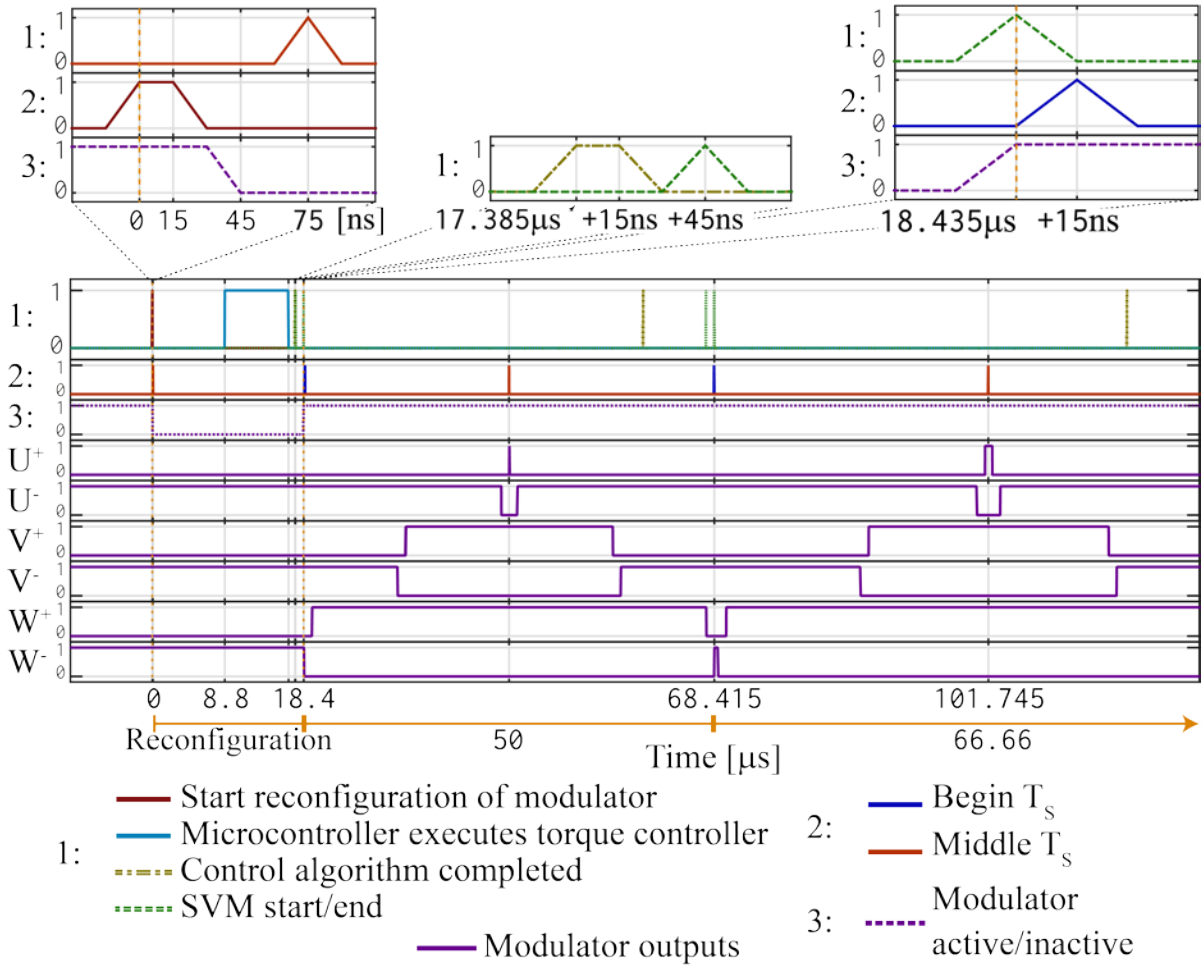

Figure 5.8: Signals of the modulator during the reconfiguration process of the PWM units.

Once the torque control is executed, and the signals are sent to the backward coordinate transformation circuit, the µC generates a flag to let know the reconfiguration circuit that the new control signals are ready; this happens at 17.835 µs. The SPM circuit is then initialized and needs roughly 1 µs to execute; finally, the modulator can be reactivated after 18.435 µs. When the normal execution of the controller is restarted, the first switching period will always be of 50 µs.

After the first 50 µs switching period, the controller can choose a different switching frequency and period, which will vary depending on the configuration parameters and the reference value, in this case, the controller chose a T_S = 66.66 µs corresponding to a f_S = 15 kHz switching frequency. As depicted in Figure 5.8, most of the delay during this procedure is generated by the ADC measurement process and the µC. However, in power electronic systems a delay of approximately 20 µs microseconds ca be considered to be suitable.

5.2.3 Speed Control Mode

For these tests, a speed controller connected in cascade to the torque controller delivers the reference for the i_q^* current, which in case of the PMSM regulates the electromagnetic torque. The current control was realized in the rotating *d,q*-reference frame. For the first test, the machine was slowly accelerated and decelerated from 0 to 3000 min^{-1} (100% of the nominal speed) without load torque. The predictive control scheme was configured as presented in Table 5.2.

Table 5.2: Configuration of the predictive control scheme with variable switching frequency configuration parameters for the experiments in speed control mode with variable speed reference.

w_q	w_d	$w_{\Delta i_q}$	w_{f_s}	εi_{max}^P	Δi_{max}^P	i_{THLD}	On-the-fly adaption of T_S	Enhanced prediction of the current ripple
1.0	0.25	1.0	1.0	0.75 A	1.0 A	3.0A	On	Off

Figure 5.9 shows measurement the mechanical speed, the variable switching frequency, the errors $i_q^* - i_q$ and $i_d^* - i_d$, with the current controller configured to use the PI-control. The dynamic response is as expected, the errors $i_q^* - i_q$ and $i_d^* - i_d$ remains within the limits. However, it is interesting to see how the switching frequency changes with the mechanical speed. When the mechanical speed reaches 2000 min^{-1} the switching frequency is 5 kHz and at 3000 min^{-1} around 4.5 kHz.

The performance of the control is maintained by regulating the maximum allowed electromagnetic torque ripple, which in turn is adjusted by controlling the i_q current switching ripple, as explained in section 4.2.4. Finally, as depicted on the right side of Figure 5.9, the measured current ripple is 12.5% bigger than the predicted current ripple $\Delta i_q^P = 0.64A$.

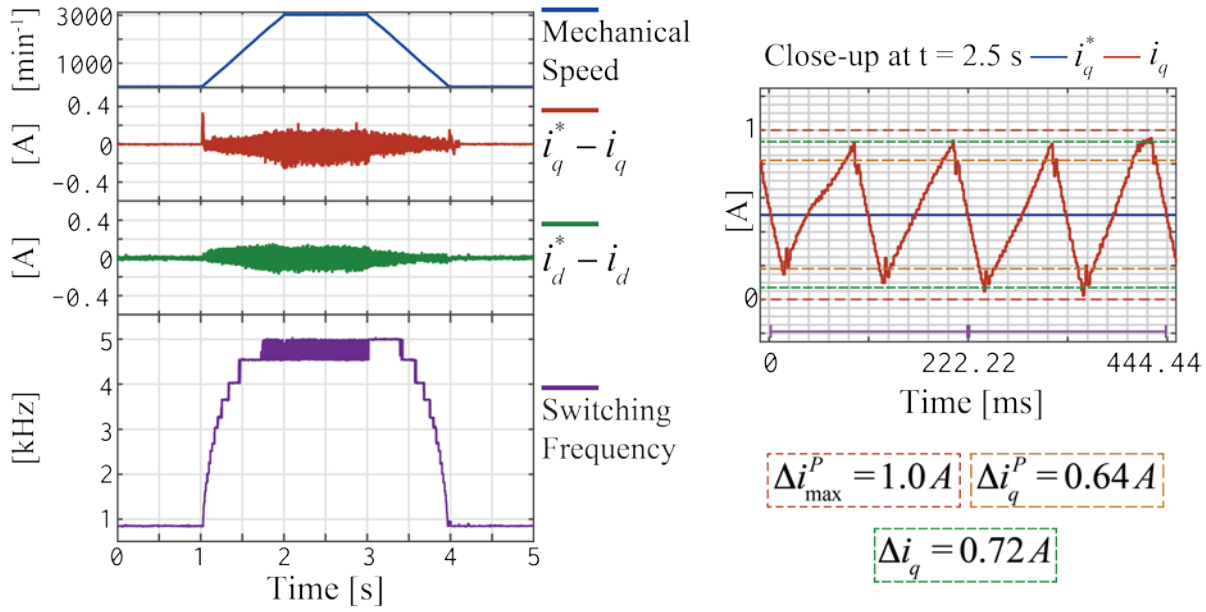

Figure 5.9: Mechanical speed, variable switching frequency and current responses of the predictive control scheme with PI-Control, the maximum current ripple is 1A without the load torque.

For the next test, the maximum allowed i_q current ripple limit Δi_{max}^P was lowered to 0.5A, the other parameters remained unaffected, Figure 5.10 presents the response. While the errors $i_q^* - i_q$ and $i_d^* - i_d$ are almost the same as with the previous configuration, the switching frequency presents an increase of almost two times. When the mechanical speed is around the 2000 min^{-1} the switching frequency reaches 8 kHz that is almost a 100% of increase, due to the reduction of the Δi_{max}^P limit.

However, when the speed reaches 3000 min^{-1} the commutation frequency is around the 6 kHz that represents an increase of 33%. Nevertheless, as presented on the right side of Figure 5.10 the measured current ripple is 66.6% greater than the Δi_q^P of 0.48A, furthermore, it is also outside of the maximum desired limit of $\Delta i_{max}^P = 0.5A$.

For the next test, the enhanced prediction of the current ripple was activated, while the other configuration parameters remained unchanged, the results are presented in in Figure 5.11. The errors $i_q^* - i_q$ and $i_d^* - i_d$ remain zero, with slightly more perturbations but still within a tolerable limit. It is interesting to see that the switching frequency also presents heavy noise, which can be caused because of the oversampling in the measurement of the currents that allows the regulation of the ripple in real-time. Here a careful design of the filter of the ΔΣ ADC is mandatory for obtaining a compromise between bandwidth and SNR (Signal-to-Noise Ratio).

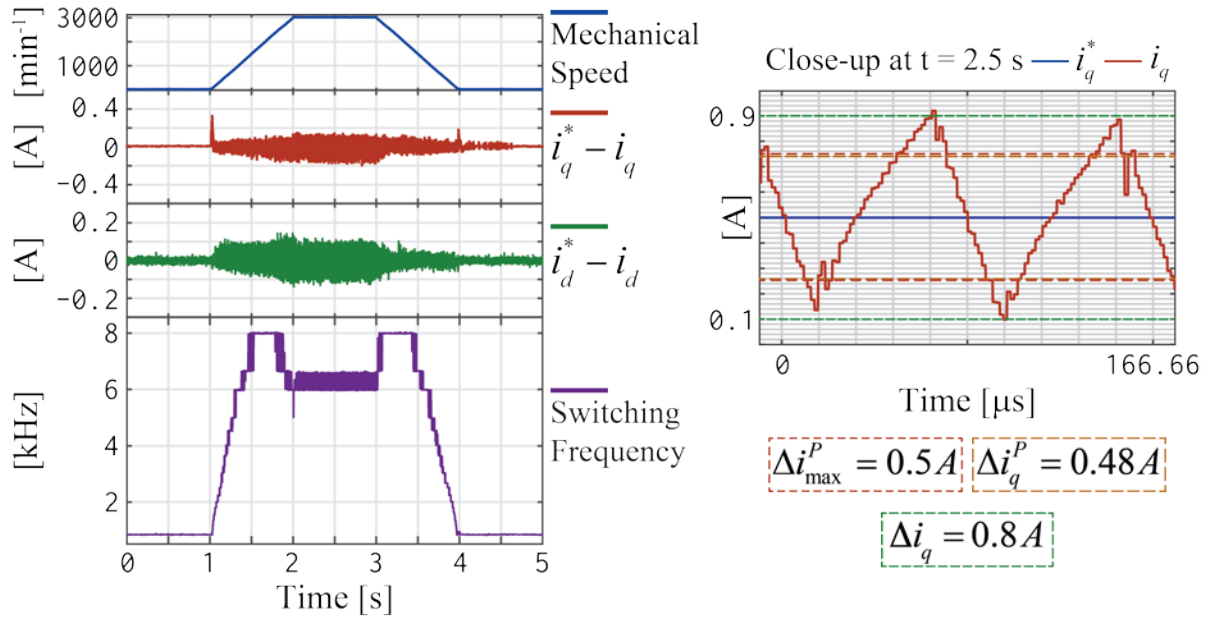

Figure 5.10: Mechanical speed, variable switching frequency and current responses of the predictive control scheme with PI-Control, the maximum current ripple is 0.5A without the load torque.

When the speed is around the 2000 min^{-1} the switching frequency has an average of 12 kHz, which represents an increase of 140% in comparison to the first case. Once the nominal speed is reached, the average switching frequency is 10 kHz that is an increase of 122.22% when compared with the first case. Nonetheless, in spite of the heavy variations on the switching frequency, the current ripple is effectively controlled, as presented on the right side of the Figure 5.11. In this case, the measured current ripple is 14.63% smaller than the predicted value, furthermore, it stays below the desired maximum limit of 0.5A.

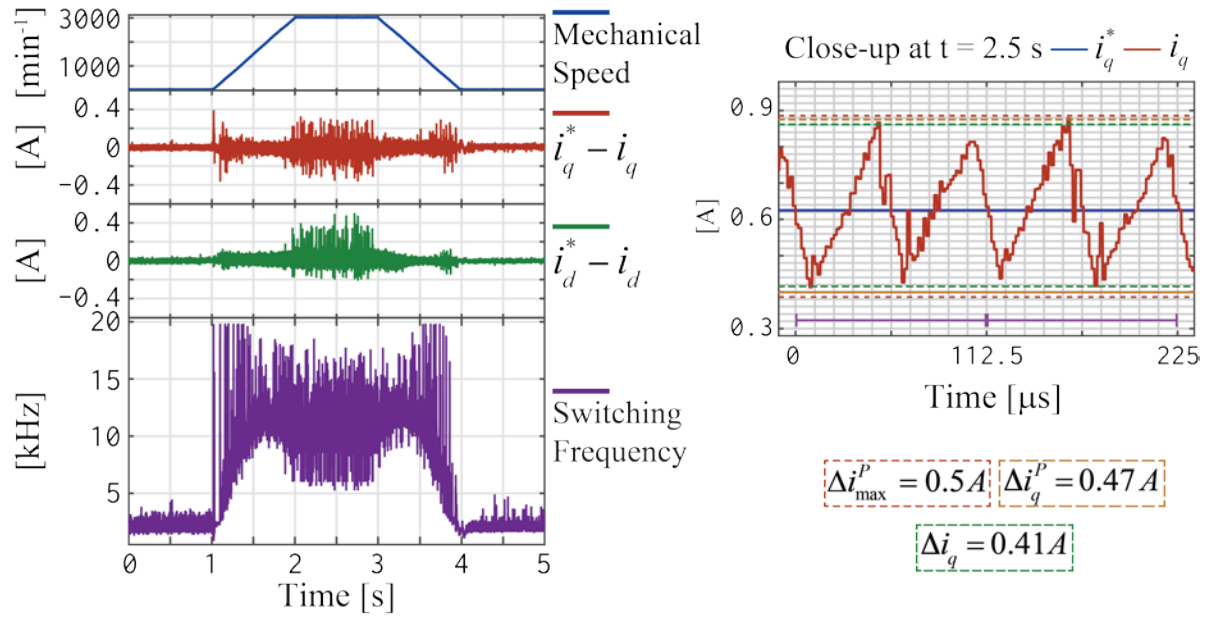

Figure 5.11: Mechanical speed, variable switching frequency and current responses of the predictive control scheme with PI-Control, the maximum current ripple is 0.5A with enhanced prediction of the current ripple and without the load torque.

5.2.3.1 Step Response without Load

For the following experiments, a step change from 0 to 900 min^{-1}, which is 30% percent of the nominal speed, was applied to the reference of the angular speed of the drive without external load. The control is operated with the mechanical encoder. For reference purposes, the classical implementation of the field-oriented control was executed at 3.6 kHz, as shown in Figure 5.12. From the close-up on the i_q current it can be seen that the torque control takes around 700 µs to reach the desired the desired reference value.

The overshoot in the current is approximately of 21%; the i_d current remains almost unaffected. The oversampled waveform of the i_q current at 180 milliseconds, shows that the calculated current ripple Δi_q^P is of 0.7 A, however, the real current ripple is larger than the predicted value, around 1.0 A (42.3% bigger than the predicted value).

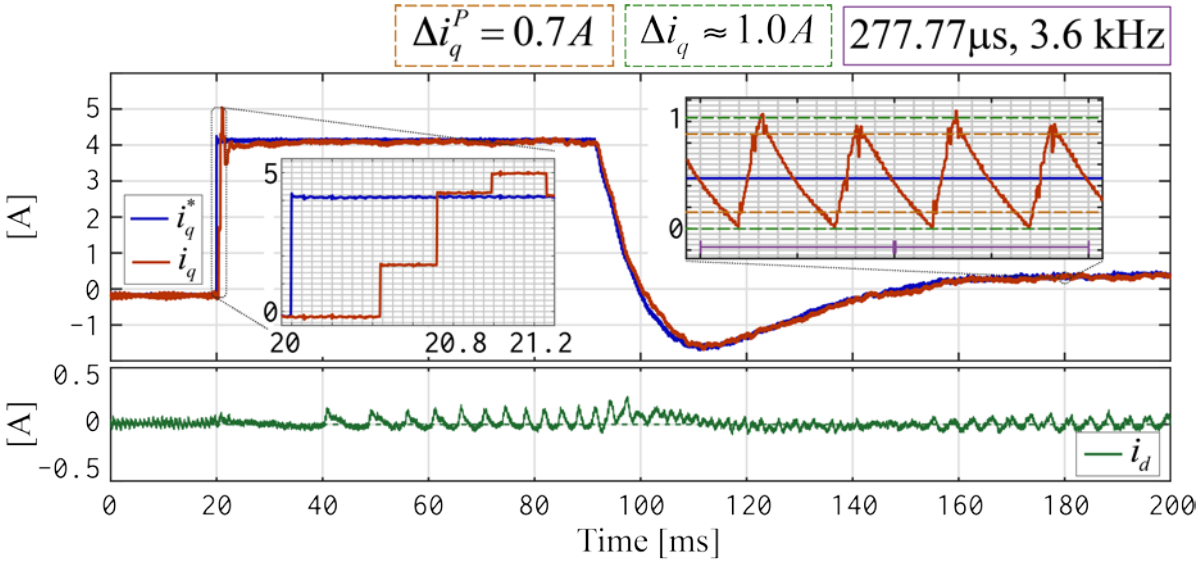

Figure 5.12: Current response of the classical field-oriented control with linear PI-control at 3.6 kHz.to a step change in the speed reference from 0 to 900 min^{-1} without external load.

For the following tests, the predictive control scheme was configured as shown in Table 5.3. The desired performance to be obtained from these parameters is a fast transient, and a low average switching frequency.

Table 5.3 Configuration of the predictive control scheme configuration parameters for the experiments in speed control mode without load.

w_q	w_d	$w_{\Delta i_q}$	w_{f_s}	εi_{max}^P	Δi_{max}^P	i_{max}	On-the-fly reconfiguration	Enhanced prediction of the current ripple
1.0	0.25	1.0	1.0	0.75 A	2.0 A	3.0A	On	Off

The current response of the predictive control scheme with variable switching frequency is presented in Figure 5.13; the current control was configured to use the PI-controllers. The i_q current will reach the desired reference value in 200 µs, which is 250% less than the classical implementation of the field-oriented control. It is interesting to see the selection of switching frequencies that the control makes during the transient state.

After the reconfiguration of the sampling time, where the frequency is set to 20 kHz, the switching frequency is reduced to 10 kHz. With this step on the commutation frequency, the i_q current practically reaches the desired reference value and then it rapidly stabilized. Even though the switching frequency is fluctuating, the control manages to reach the desired reference value in three sampling periods. The predictive control scheme with variable switching frequency selects the next commutation period so that the switching losses are reduced, and the performance of the control is optimized; which can be mainly attributed to the cost function.

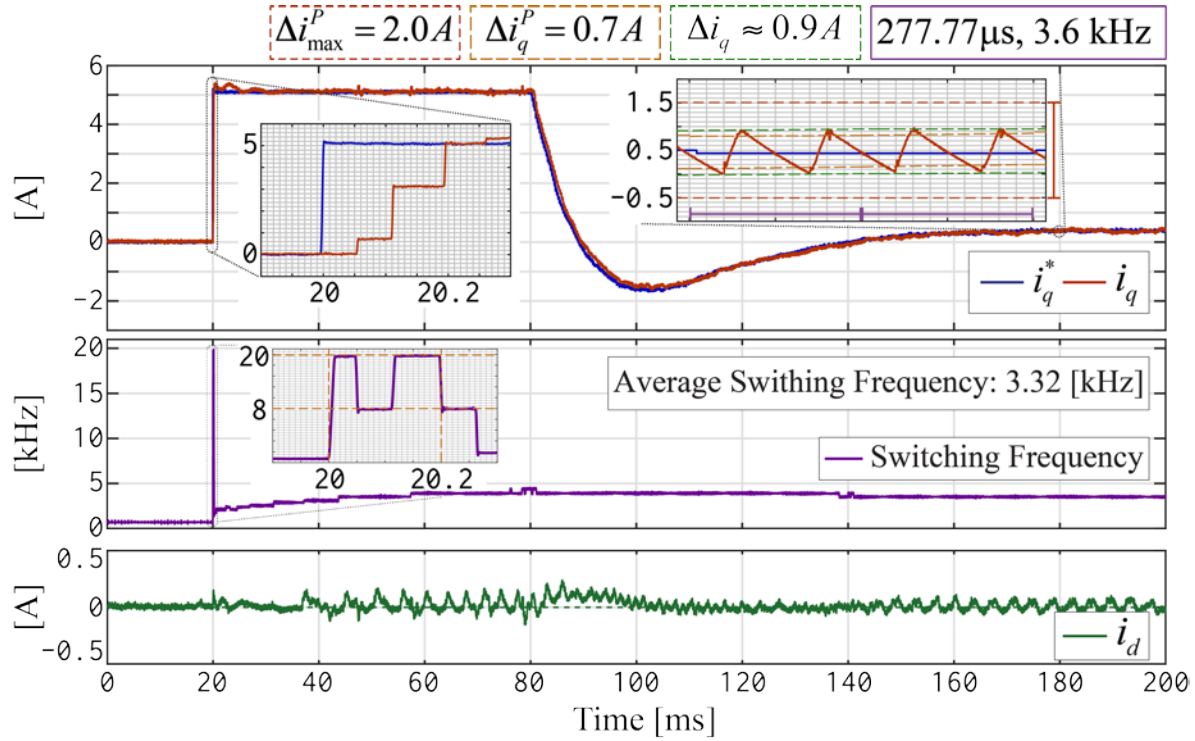

Figure 5.13: Current response of the predictive control scheme to a step change in the speed reference from 0 to 900 min^{-1} without external load. The current controller is configured to work with PI-control and a maximum current ripple of 2.0A.

A small oscillation on the i_q current can be seen, caused by the sudden decrease in the switching frequency. This happens because the gains of the PI-control are adapted proportionally to the switching frequency. The average switching frequency is 3.32 kHz, 8.43% smaller than the classical FOC. This is possible because the switching frequency

is reduced to 800 Hz when the speed reference is zero. Once the speed reference is reached, the switching frequency will settle to 3.6 kHz.

The i_d current remains almost unaffected. The switching ripple of the i_q current measured at 180 milliseconds, shows that the predicted current ripple Δi_q^P is of 0.7 A, nevertheless, the real current ripple is bigger than the predicted value, around 0.9 A, which is 28.57% bigger than the predicted value. Nonetheless the ripple stays whiting the desired maximum limit.

For the next test the current ripple limit Δi_{max}^P was reduced to 0.5A, the rest of the configuration parameters of the control remain as presented in Table 5.3. The current response is very similar to the previous configuration; as depicted in Figure 5.14. The i_q current will reach the desired reference value in around 200 µs, the i_d current remains barely unaffected. However, the average switching frequency will increase to 4.77 kHz (32.5% more than the 3.6 kHz of the classical FOC implementation), in order to regulate the maximum allowed i_q current ripple.

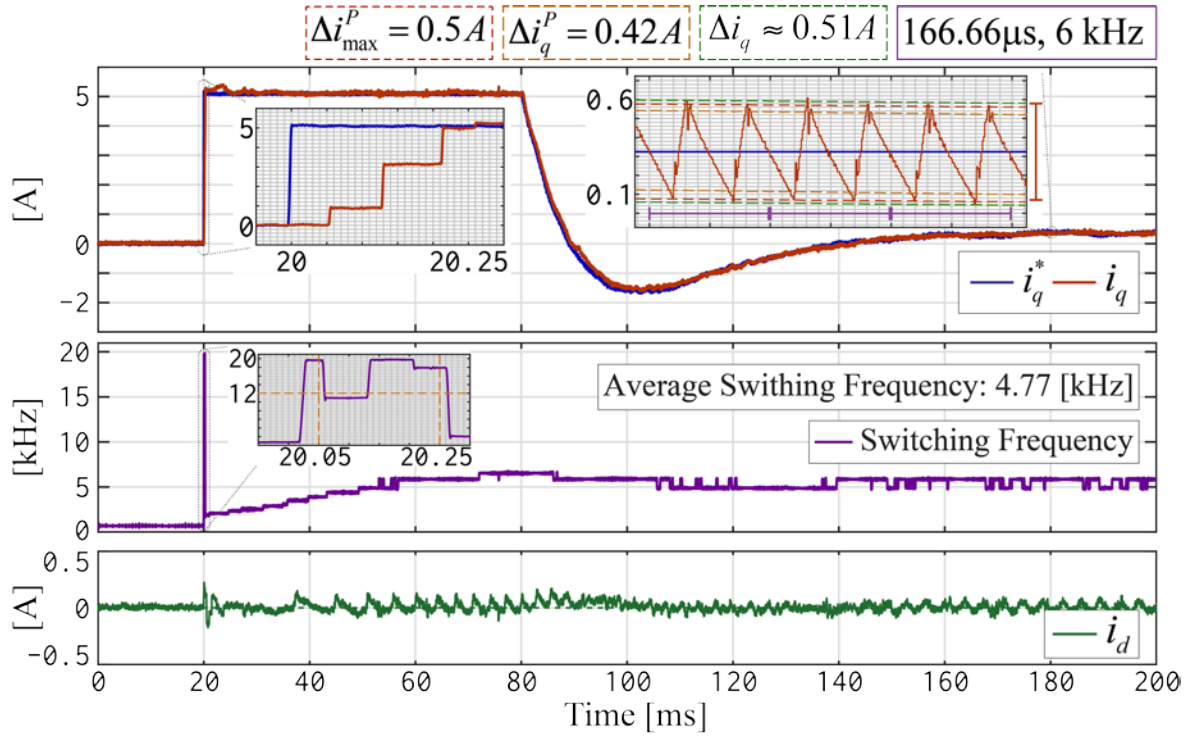

Figure 5.14: Current response of the predictive control scheme to a step change in the speed reference from 0 to 900 min^{-1} without external load. The current controller is configured to work with PI-Control and a maximum current ripple of 0.5A.

Once the steady-state operation is reached, the switching frequency settles at 6 kHz. The predicted current ripple is 21.43% smaller than the measured current ripple; however, the limit is only exceeded by 2%. If the maximum current ripple can not be

surpassed, then the enhanced prediction of the current ripple can be activated, but this would increase the average switching frequency and therefore, the switching losses.

For the sake of comparision the variable frequency control that includes the ripple in the cost function as presented in this work was implemented by using a classical scheme of model-based i.e. without PI-controllers. It was configured as presented in Table 5.3 with the *Current control* directive set to MBC (model-based control); the voltage space phasors are calculated as presented in section 4.2.3.2.

The overall dynamic response is as expected, as shown in Figure 5.15. However, it presents a small ripple in both i_d and i_q currents, which could be attributed to the fact that model-based control is very sensitive to errors in the parameters of the model and this implementation does not include a system to adapt the parameters online. The overshoot on the i_q current is of 24.87%; however, the current reaches the desired reference value in roughly 200 µs.

A steady-state error, typical of model-based controllers lacking an integral component can be better noticed on the i_d current, that settles at 0.18A or 4.4% of the i_N. The average switching frequency is 3.17 kHz, 13.57% smaller than the classical FOC. The lack of the integral component on the controller allows it to make fast changes to the commutation frequency; the switching frequency finally settles at 3.6 kHz.

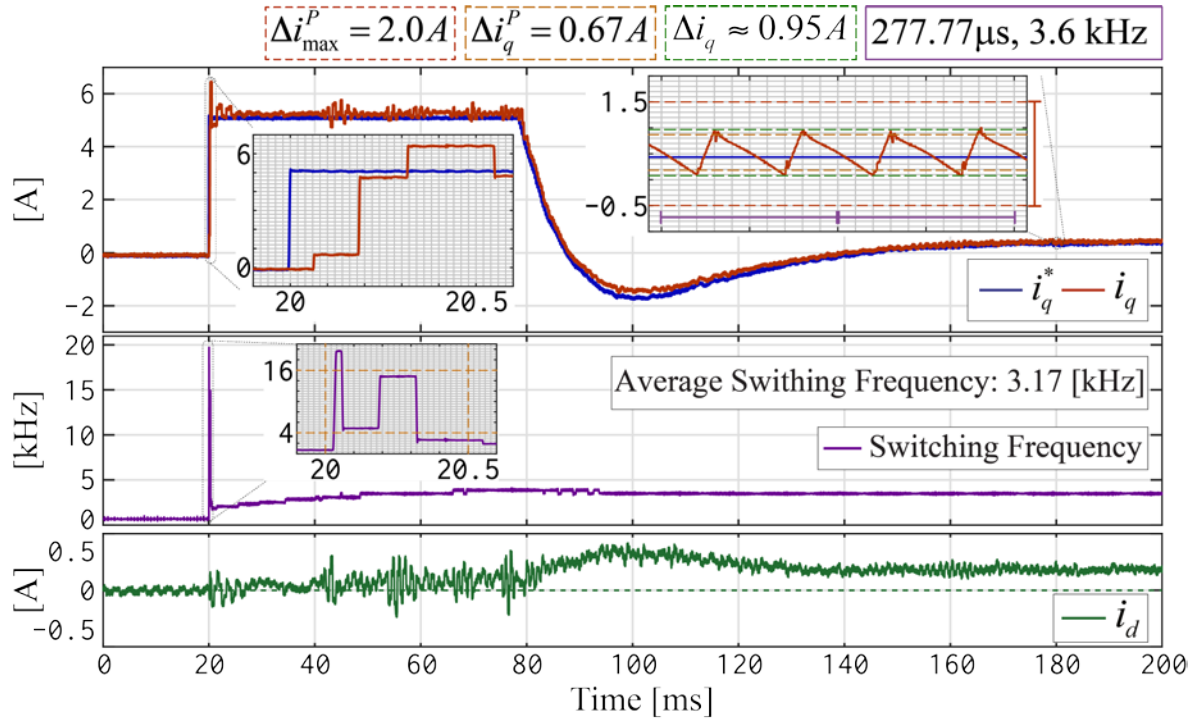

Figure 5.15 Current response of the predictive control scheme to a step change in the speed reference from 0 to 900 min^{-1} without external load. The current controller is configured to work with model-based control and a maximum current ripple of 2.0A

The switching ripple of the i_q current measured at 180 milliseconds, shows that the predicted current ripple Δi_q^P is of 0.67 A, however, the current ripple is bigger than the predicted value, around 0.95 A, which is 41.8% bigger than the predicted value. Nonetheless the ripple stays whiting the desired maximum limit.

In the following experiment, the current ripple limit Δi_{max}^P was set to 0.5A, while the rest of the configuration parameters remained unchanged. The current response is very similar to the previous case; it is presented in Figure 5.14. However, the ripple in both i_d and i_q has an slight increase in persistence and magnitude. The overshoot on the i_q current is of 24.35%, the transient presents the same desired fast dynamics as in the previous experiments.

The settling time of the i_q is again about 200 µs, the steady-state error remains present, it can be better noticed on the i_d current, which settles again at 0.18A or 4.4% of the i_N. The average switching frequency is 4.76 kHz, 32.22% more than the 3.6 kHz reference, which is about the same as when the PI-controllers are used. The switching ripple of the i_q current measured at 180 milliseconds, shows that the predicted current ripple Δi_q^P is of 0.44 A, while the real current ripple is 29.55% bigger than the predicted value, approximately 0.57 A; thus, the maximum desired limit is exceeded by 14%.

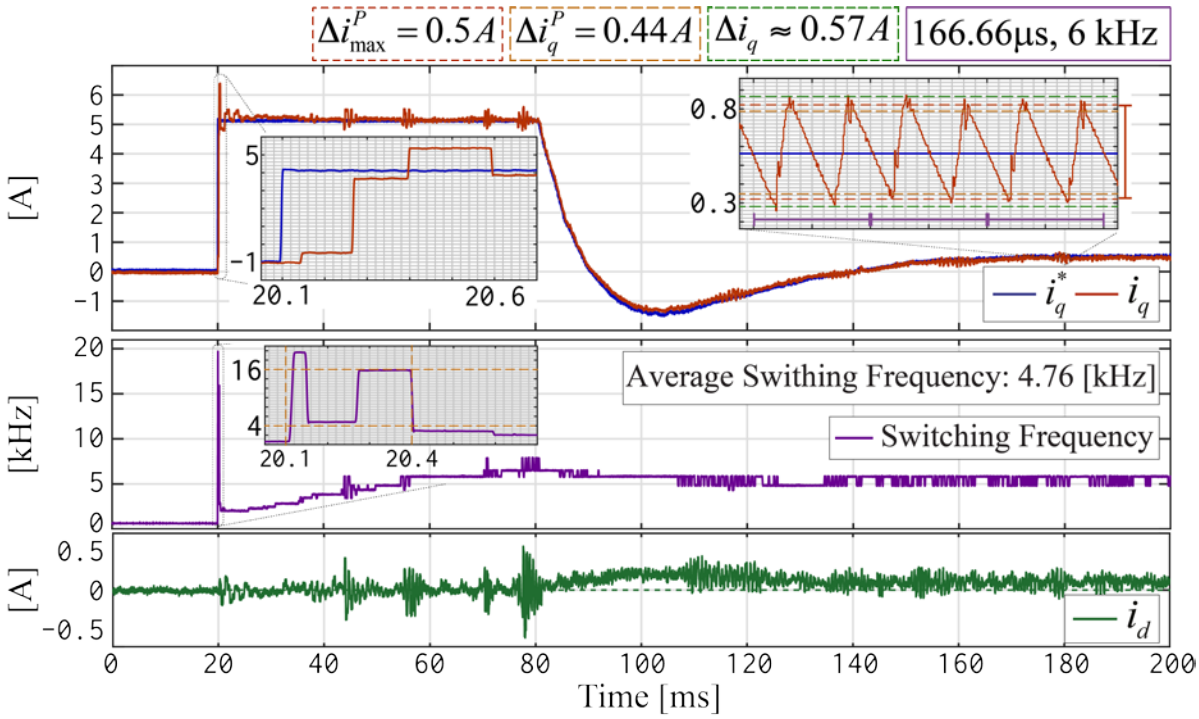

Figure 5.16: Current response of the predictive control scheme to a step change in the speed reference from 0 to 900 min^{-1} without external load. The current controller is configured to work with model-based control and a maximum current ripple of 0.5A

5.2.3.2 Speed Reversal Response at Full Load

For these experiments, the machine was loaded with 100% of the nominal torque (corresponding to $i_q = 4{,}1A$, the amplitude of the stator current was limited to $\underline{i}_1^* = 5.125A$). The machine was rapidly accelerated to -900 min^{-1} and then the speed reference is suddenly changed to 900 min^{-1}; as a result, a high dynamic response of the current control is required. The control parameters are presented in Table 5.4.

Table 5.4 Configuration of the predictive control scheme with variable switching frequency configuration parameters for the experiments in speed control mode during speed reversal with 100% torque load.

w_q	w_d	$w_{\Delta i_q}$	w_{f_s}	εi_{max}^P	Δi_{max}^P	i_{max}	PWM on-the-fly reconfiguration	Enhanced current ripple prediction
1.0	0.25	1.0	1.0	0.5 A	1.0 A	3.0A	On	Off

Figure 5.17 shows the current response when the control is configured to use the PI-controllers. It presents a very high dynamical response, without overshoot in the i_q current and only a small perturbation on the i_d current during the speed reversal. The average switching frequency is 3.75 kHz, the speed reversal can also be noted when the switching frequency is reduced to 1 kHz before it starts to increase again.

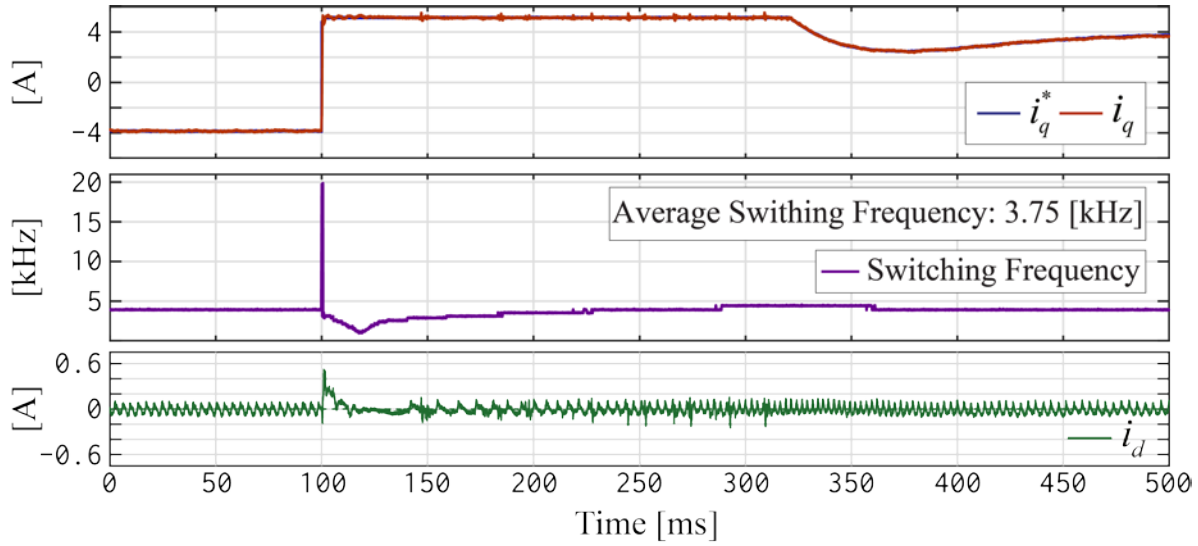

Figure 5.17: Current response of the predictive control scheme to a step change in the speed reference from -900 to 900 min^{-1} at full load. The current controller is configured to work with PI-Control and a maximum current ripple of 1.0A

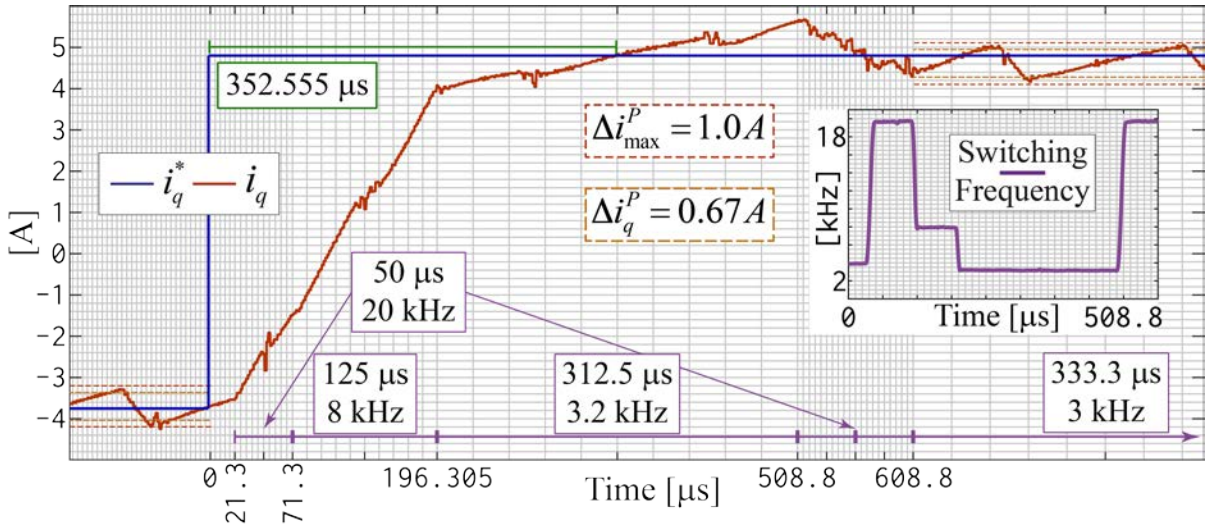

Figure 5.18 Close-up to the i_q current response of the predictive control scheme to a step change in the speed reference from -900 to 900 min^{-1} at full load. The current controller is configured to work with PI-Control and a maximum current ripple of 1.0A

A close-up of the i_q current is obtained with a high time resolution by using the oversampling capability of the ΔΣ-ADC and is presented in Figure 5.18 during the speed reversal. It shows a fast transient, after 200 µs the i_q current reaches around 78% of the reference value. However, since the Δi^P_{max} limit is set to 1A, the remaining 22% takes 152.5 µs to be reached i.e. the selected switching periods are 50, 125 and 315 µs to reach the desired torque reference. This long settling time can be explained with the larger change of the current reference and with the smaller control reserve due to the higher induced voltage in the machine.

In Figure 5.18, it can also be appreciated that the overshoot is very small. The switching ripple of the i_q current shows that the predicted current ripple Δi^P_q is of 0.67 A, while the measured current ripple is between 0.86 to 0.95A that is 29 to 42% bigger than the predicted value, however, the maximum desired limit is not exceeded.

For the following experiment, the Δi^P_{max} limit is reduced to 0.3A, results are presented in Figure 5.19 and Figure 5.20. The same desired high dynamic response of the last case is achieved, the i_q current has no overshoot and only the perturbation on the i_d current during the speed reversal is smaller as in the previous configuration. The average switching frequency increases to 9.7 kHz, which is an increase of 158.6% when compared to the previous configuration.

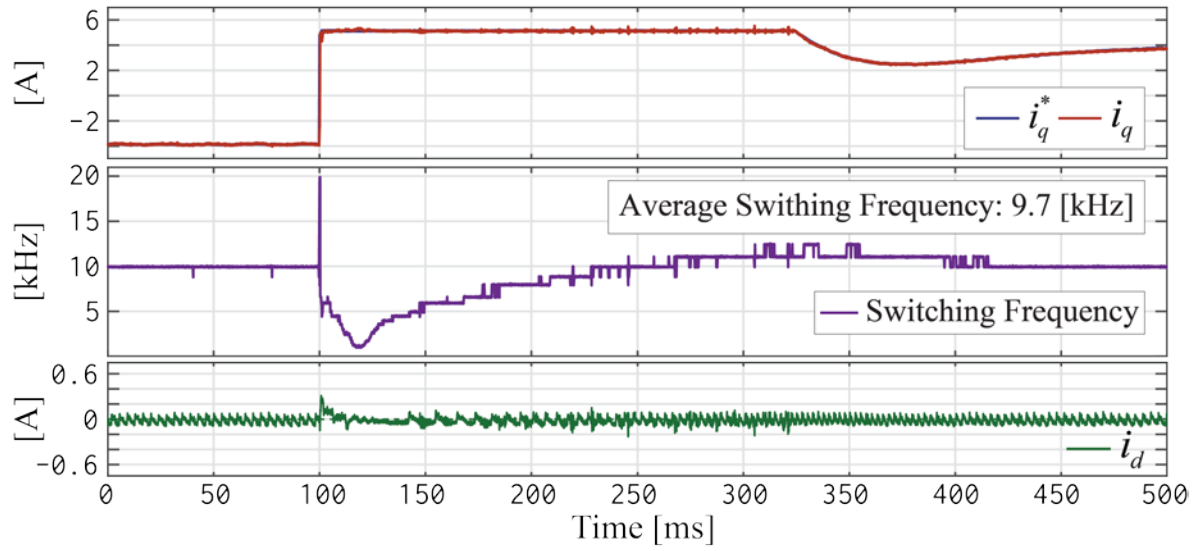

Figure 5.19: Current response of the predictive control scheme to a step change in the speed reference from -900 to 900 min^{-1} at full load. The current controller is configured to work with PI-Control and a maximum current ripple of 0.3A

Figure 5.20 depicts the i_q current as obtained with a high time resolution by using the oversampling capability of the ΔΣ-ADC during the speed reversal. In this case, the selected switching periods to reach the desired torque reference are 50, 125 and 50 µs. This reduces the transient time to 217.92 µs when the time to reconfigure the PWM units is added, which is a reduction of 61.8%. Nonetheless, the Δi_{max}^P limit set at 0.3A is surpassed in a range between 0.48 and 0.58A that is an excess ranging 60 to 93%.

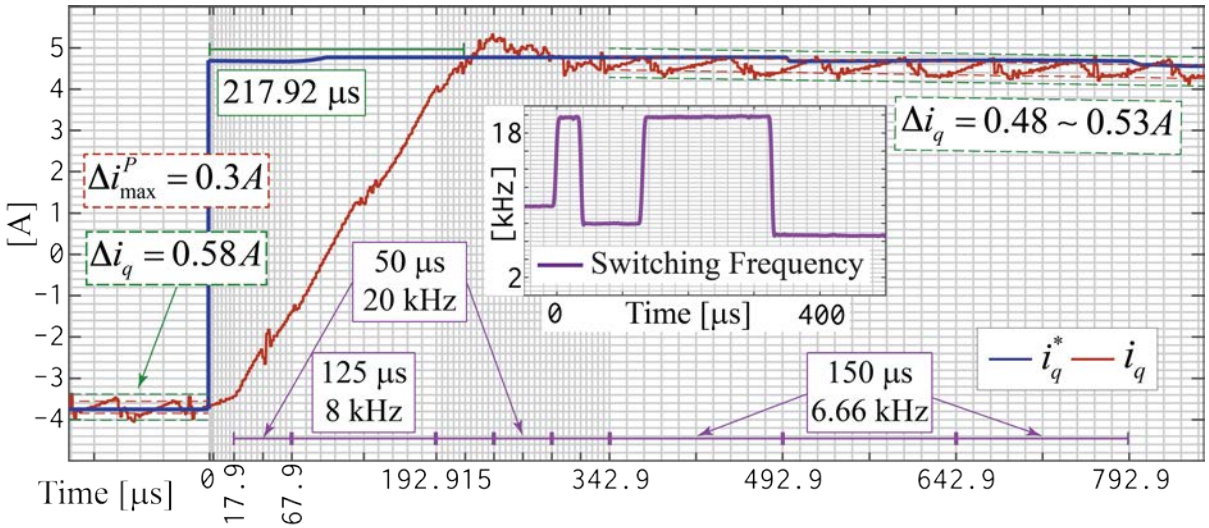

Figure 5.20: Close-up to the i_q current response of the predictive control scheme to a step change in the speed reference from -900 to 900 min^{-1} at full load. The current controller is configured to work with PI-Control and a maximum current ripple of 0.3A

In the subsequent experiments, the current controller was configured to work with model-based control (i.e. without PI-Controllers) by keeping the features of variably switching frequency and control of the ripple. The maximum current ripple is set to 1A. The response presented in Figure 5.21 is as expected. While a similar high dynamic

response is obtained, both i_q and i_d currents present a bigger ripple when compared to the experiment where the PI controllers are used. The overshoot on the i_q current reaches 6.77A, which is 32.1% more than the maximum allowed torque.

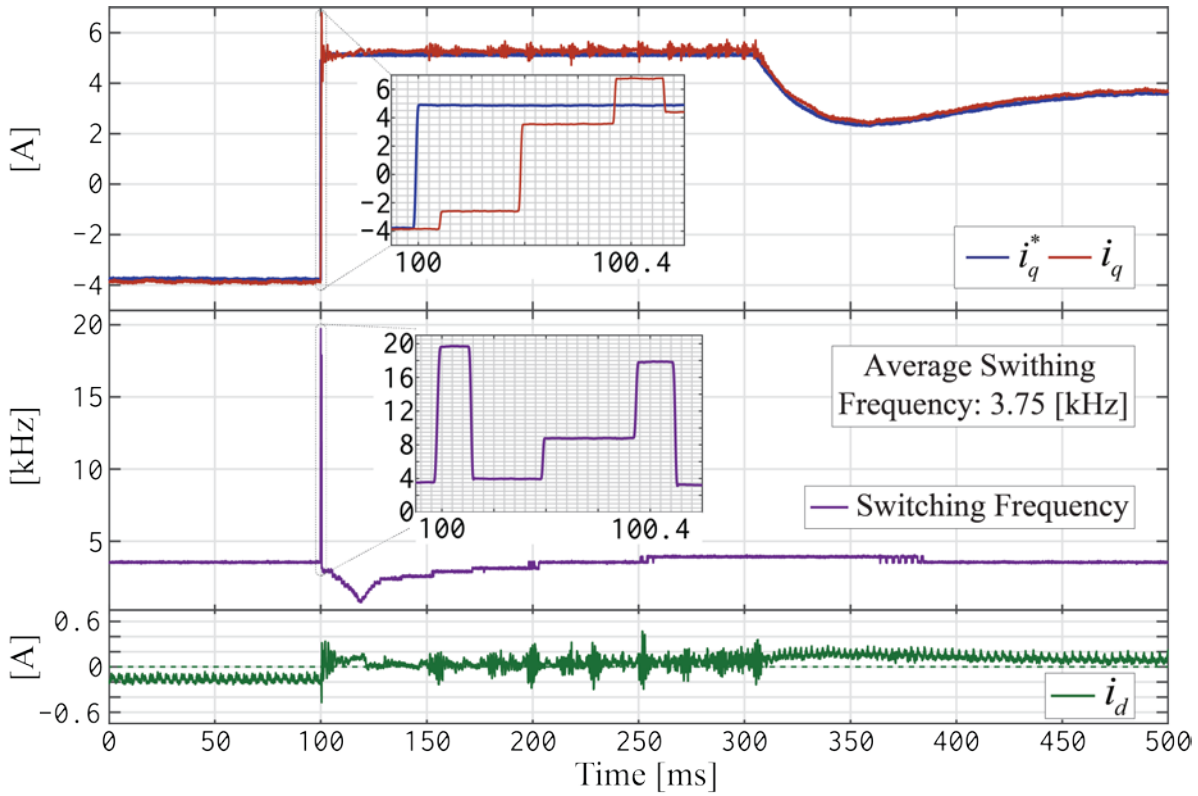

Figure 5.21: Current response of the predictive control scheme to a step change in the speed reference from -900 to 900 min^{-1} at full load. The current controller is configured to work with model-based control and a maximum current ripple of 1.0A.

The settling time for i_q is approximately 300 µs, it also takes longer to stabilize around the desired value. A steady-state error is also present, which can be better noticed in i_d the current with an absolute magnitude average of 0.15A, a 3.65% of the nominal current. The average switching frequency is 3.75 kHz, the same value that is obtained when the control is configured to work with the PI-controllers.

Finally, the Δi_{max}^P limit was reduced to 0.3A. Figure 5.22 shows an improved response of the predictive control scheme with variable switching frequency when the model-based control with variable switching frequency is used. The overall steady-state error is substantially reduced almost imperceptible. The ripple exhibited in both i_q and i_d currents is also lessened if compared with the previous configuration that allowed a bigger current ripple. However, the transient is the same as in the previous configuration; the overshoot on the i_q current is 6.7A or 30.73% bigger than the maximum allowed one. The average switching frequency is 9.75 kHz, a slight increase over the same configuration of the control using the PI-Control.

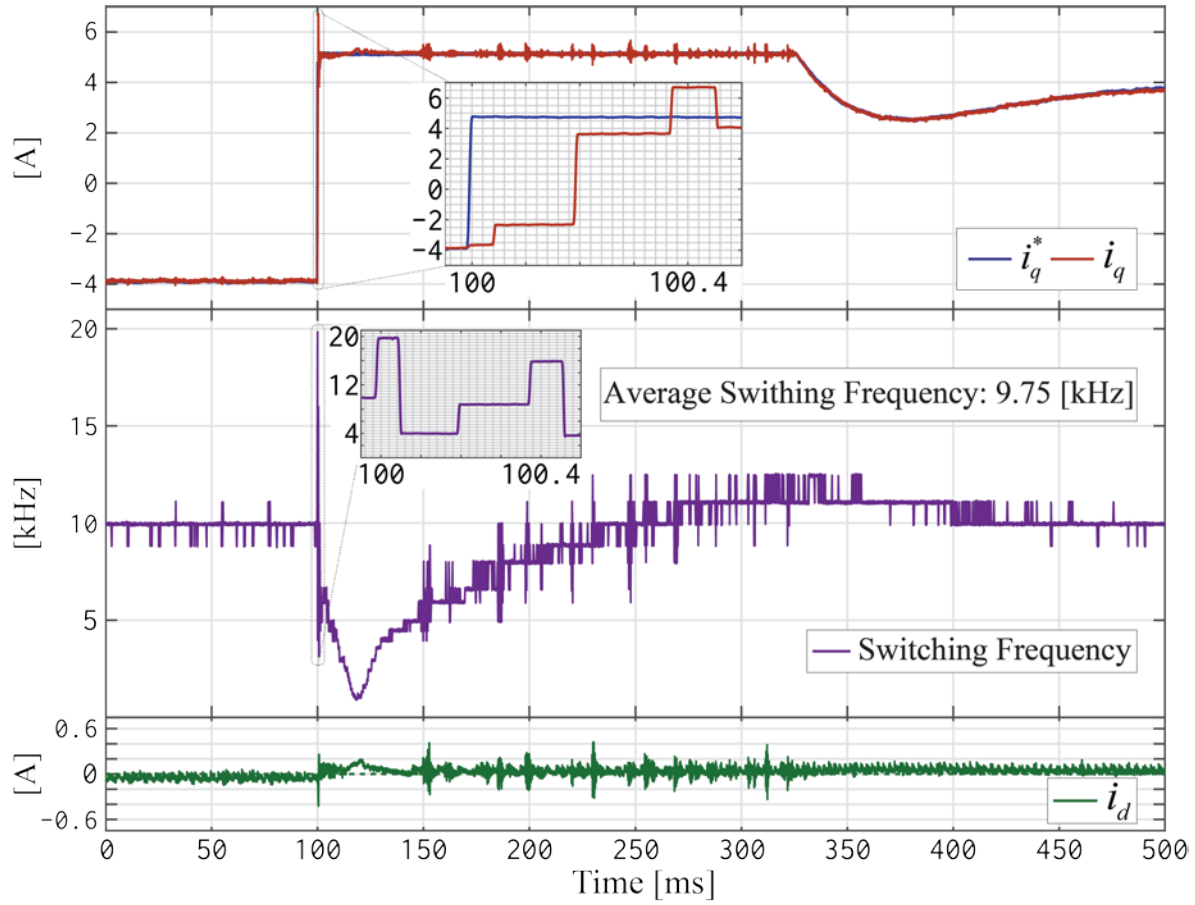

Figure 5.22: Current response of the predictive control scheme to a step change in the speed reference from -900 to 900 min^{-1} at full load. The current controller is configured to work with model-based control and a maximum current ripple of 0.3A.

The experiments carried out in sections 5.2.1 to 5.2.3 show that the performance of the predictive control scheme with variable switching frequency is certainly better than that of the conventional FOC. The switching frequency can be substantially reduced without losing quality in the performance during transients. Furthermore, the use of PI-Control, as opposed to conventional model-based control, solves many inconveniences of MBC, like the sensitivity to the parameters of the model. Nevertheless, the conventional model-based approach improves its response when the average switching frequency is increased.

5.2.4 Enhanced Current Ripple Estimation

Even though the predictive control scheme with variable switching frequency exhibits a great dynamic response, there are some cases where the Δi_{max}^{P} limit is surpassed. The main reason is that the prediction of the switching current ripple, as presented in section 4.2.4.1 is just an approximation. For certain applications this could be undesirable, therefore, the algorithm presented in section 4.2.4.2, which aims to increase the precision of the ripple prediction, is evaluated in this section. The predictive

control scheme was configured with the parameters shown in Table 5.4, the current control was done using PI-controllers.

5.2.4.1 Enhanced Current Ripple Estimation in Steady-State Operation

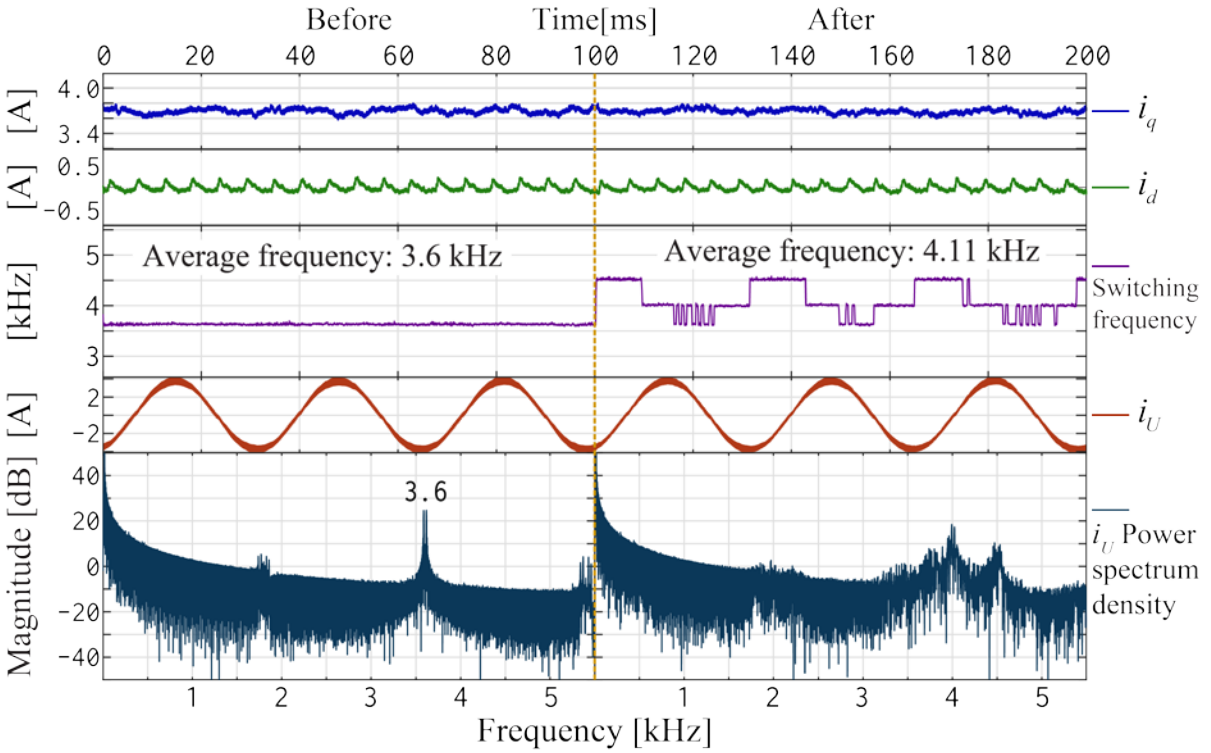

Figure 5.23: Measurements at a speed of 600 min^{-1}, the enhanced prediction of the current ripple is activated at t = 100 ms.

The machine was loaded at approximately 90% of its maximum torque. For the first test, the speed reference of the control was set constant at 600 min^{-1} (20% of the nominal speed). The maximum allowed ripple was set to $\Delta i^{P}_{max} = 1.0A$, the enhanced prediction of the current ripple is activated at t = 100 ms. As it can be seen in figure 5.23, the algorithm has no major impact on the performance of the control.

However, the average of the switching frequency increases from 3.6 to 4.11 kHz and presents more dynamic changes, to regulate the desired electromagnetic torque ripple. The power spectrum density of the phase current i_U is shown in the lower part of Figure 5.23. First, the switching frequency is centered at 3.6 kHz, after the compensation system is activated, the primary active frequencies are 3.6, 4 and 4.5 kHz as is presented in the diagram above, which shows the instantaneous commutation frequency.

In Figure 5.24a a closer look at the oversampled form of the i_q current before the proposed correction system is applied, shows that the predicted current ripple is about 11.11% smaller than the measured ripple. In this case, the switching ripple does not

exceed the Δi_{max}^{P} desired maximum limit; however, the desired behavior should predict a ripple of similar or slightly greater magnitude than the real switching ripple. Once the correction system is applied the desired behavior is obtained, as presented in Figure 5.24b, the measured ripple is 7.14% smaller than the predicted ripple.

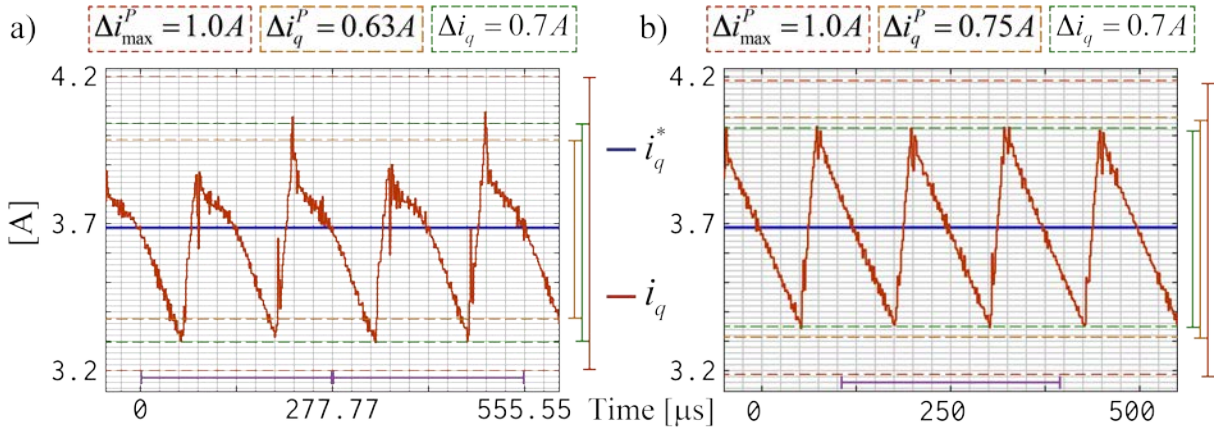

Figure 5.24: Measurements at a speed of 600 min^{-1}, close-up of the i_q current a) Before corrections. b) After the correction of the ripple prediction.

For the next test, the speed reference of the control was set at 1500 min^{-1} (50% of the nominal speed), the maximum allowed ripple was set to $\Delta i_{max}^{P} = 0.5$, and the enhanced prediction of the current ripple is activated at the 50 ms. The measurement of the i_q current before the correction algorithm is activated is depicted in Figure 5.25a. It shows that the real-time measurement of the i_q current ripple is 47.72% bigger that the predicted value and surpasses the desired maximum limit by 30%. However, once the correction system is activated as presented in Figure 5.25b, the desired behavior of the system is obtained. The switching ripple is 9.52% smaller than the predicted ripple, and furthermore, it stays within the desired maximum limit.

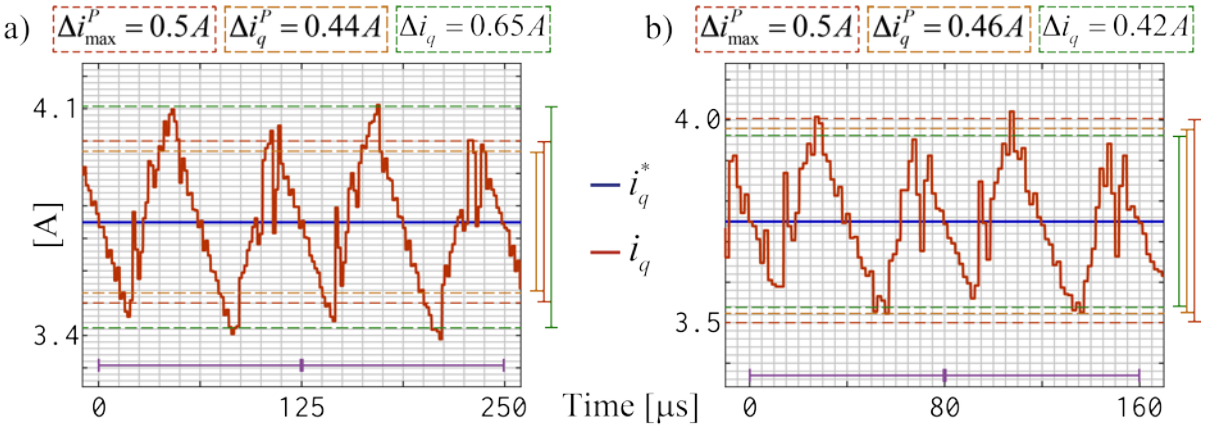

Figure 5.25: Measurements at a speed of 1500 min^{-1}, close-up of the i_q current a) Before corrections. b) After the correction of the ripple prediction.

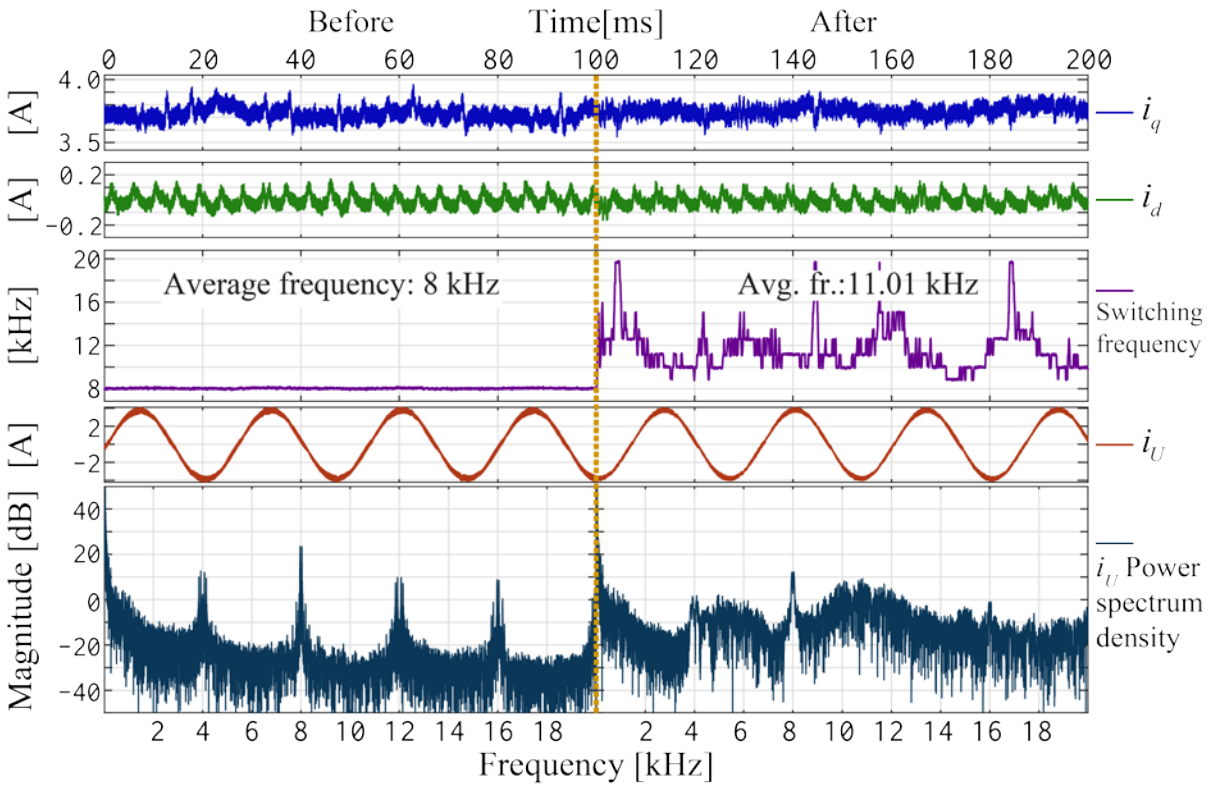

Figure 5.26: Measurements at a speed of 1500 min^{-1}, the enhanced prediction of the current ripple is activated at t = 50 ms.

Figure 5.26 shows again that there is no impact on the performance of the control can be noted, apart from the increase in the average switching frequency and its endless variations, caused by the permanent control of the current ripple. The power spectrum density of the i_U current show that first the commutating frequency is centered around 8kHz (with its correspondent harmonics). However, when the correction system is activated, the magnitude of the 8 kHz is reduced, and some power is transferred to the 10 to 12 and 16 kHz.

For the last test, the enhanced prediction system is permanently activated, the speed reference is 1500 min^{-1}, the initial current ripple is 1.0A and at t = 50 ms this limit is changed to 0.5A. The results are presented in Figure 5.27. When the current ripple limit is 1.0A the average switching frequency at the beginning of the test is 6 kHz, once the ripple limit is reduced to 0.5A the average switching frequency increases up to 10.48 kHz. The results show a minor improvement in the performance of the control, once the ripple limit is reduced, as the i_d and i_q currents present less disturbances.

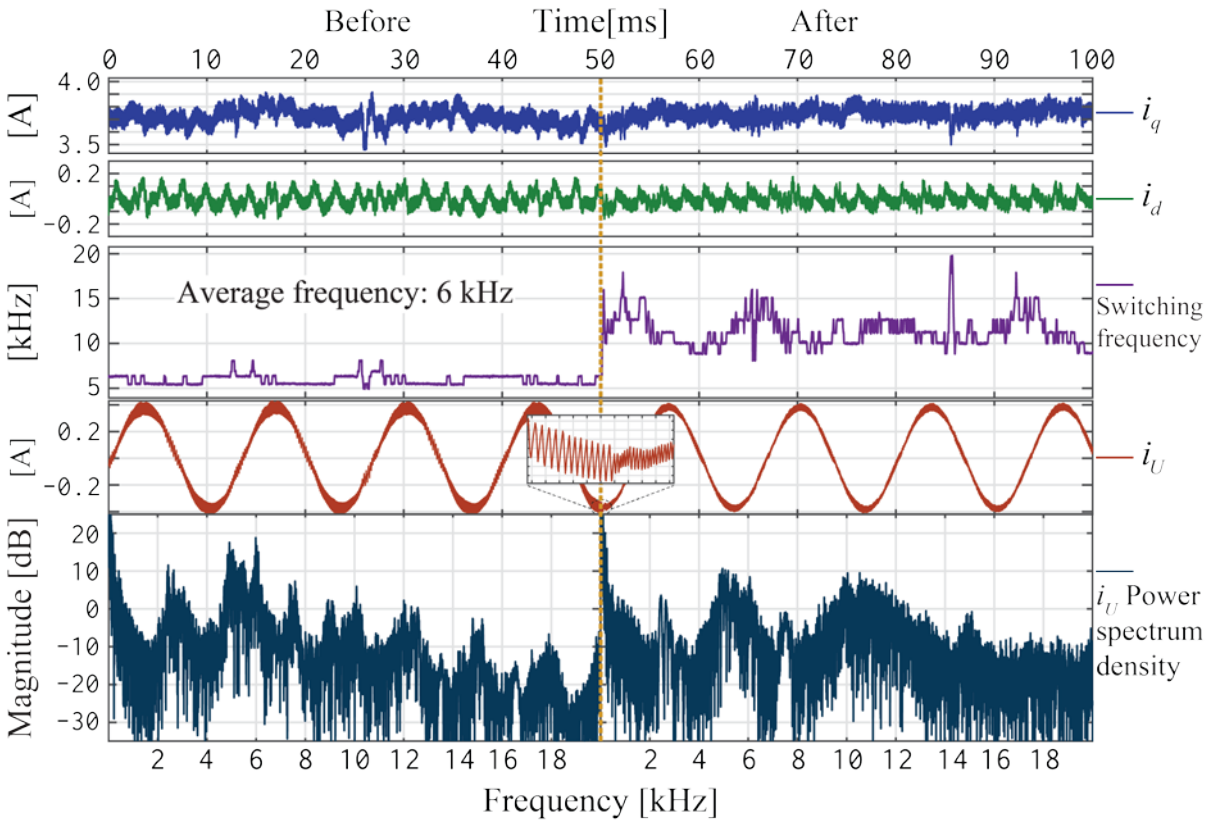

Figure 5.27: The enhanced prediction of the current ripple is permanently activated, speed 1500 min^{-1}, the initial current ripple limit is 1.0A, at t = 50 ms the limit is changed to 0.5A.

A close-up in the i_U current waveform taken at aproximately t = 50 ms show how once the current ripple limit Δi^P_{max} is reduced, the i_U current ripple gets instantaneously affected. It is also interesting to notice that the power spectrum density of the i_U current present the biggest magnitude around 5, 5.5 and 6 kHz. When the ripple limit is changed, the magnitude of the 5 and 6 kHz frequency is slightly reduced and this power is transferred to the 10 and 11 kHz range of frequencies.

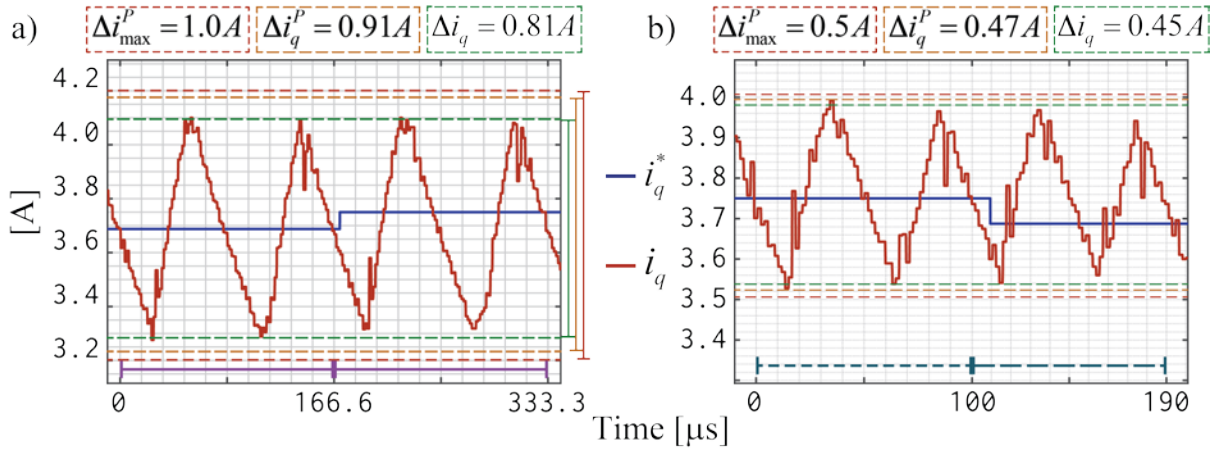

Figure 5.28: Close-up of the i_q current, the enhanced prediction of the current ripple is permanently activated, speed 1500 min^{-1}: a) With current ripple limit of 1.0A. b) With current ripple limit of 0.5A.

Finally, Figure 5.28 shows how the i_q current ripple is effectively controlled and maintained below the desired limits. In figure 5.28a when the limit is $\Delta i_{max}^{P} = 1.0A$, the predicted and real values of the switching ripples are 0.91 and 0.81A respectively, which is a 12.3% difference. As shown in Figure 5.28b, when the is $\Delta i_{max}^{P} = 0.5A$ the predicted and real values of the switching ripple differ only 4.44%.

All results show that the proposed corrections to the current ripple estimation improve its precision. Furthermore, the desired effect is achieved as the predicted values are greater or equal to the measured ripple. Moreover, the execution of the correction does not affect the performance of the control. However, due to the constant regulation of the switching ripple, the commutating frequency is constantly changing. Depending on the operating conditions, these variations can be quite large, spreading the harmonic distortion in wider a range of frequencies.

5.2.4.2 Enhanced Current Ripple Estimation: Speed Reversal at Full Load

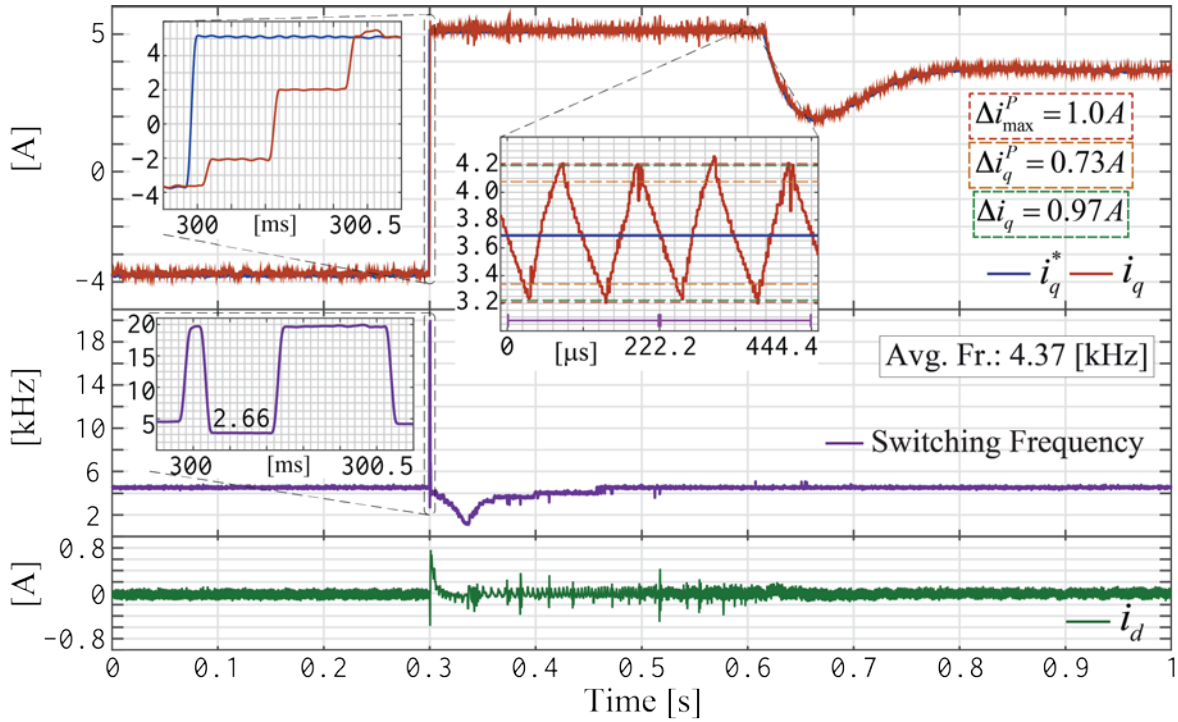

Figure 5.29: Current response of the predictive control scheme to a step change in the speed reference from -1500 to 1500 min^{-1} at full load. The current controller is configured to work with PI-Control and a maximum current ripple of 1.0A, the enhanced control of the current ripple is deactivated.

For these experiments, the machine was loaded with 100% of the nominal torque, a speed reversal from 1500 to 1500 min^{-1}was performed, the predictive control scheme parameters are presented in table 5.4. Figure 5.29 presents the current response when the enhanced control of the switching ripple is deactivated; the control presents a high

dynamic response. The i_q current reaches the desired reference in approximately 450 µs; the overshoot is minimal almost non-present.

The i_d current presents an overshoot of 0.6A during the speed reversal, which is 14.63% of the nominal current. However, it is quickly stabilized, afterwards it remains unaffected. The average switching frequency is 4.37 kHz, a close-up to the i_q current shows that the real switching ripple is very close to the desired limit, and the predicted value is smaller than the real value by 32.87%.

The results of the same test with the enhanced prediction of the current ripple are presented in Figure 5.30. The same high dynamic response of the control is obtained, the i_q current reaches the desired reference value in around 450 µs whit almost no overshoot. The i_d current presents the same overshoot of 0.6A during the speed reversal that after correcting the calculation of the current ripple remains almost unaffected.

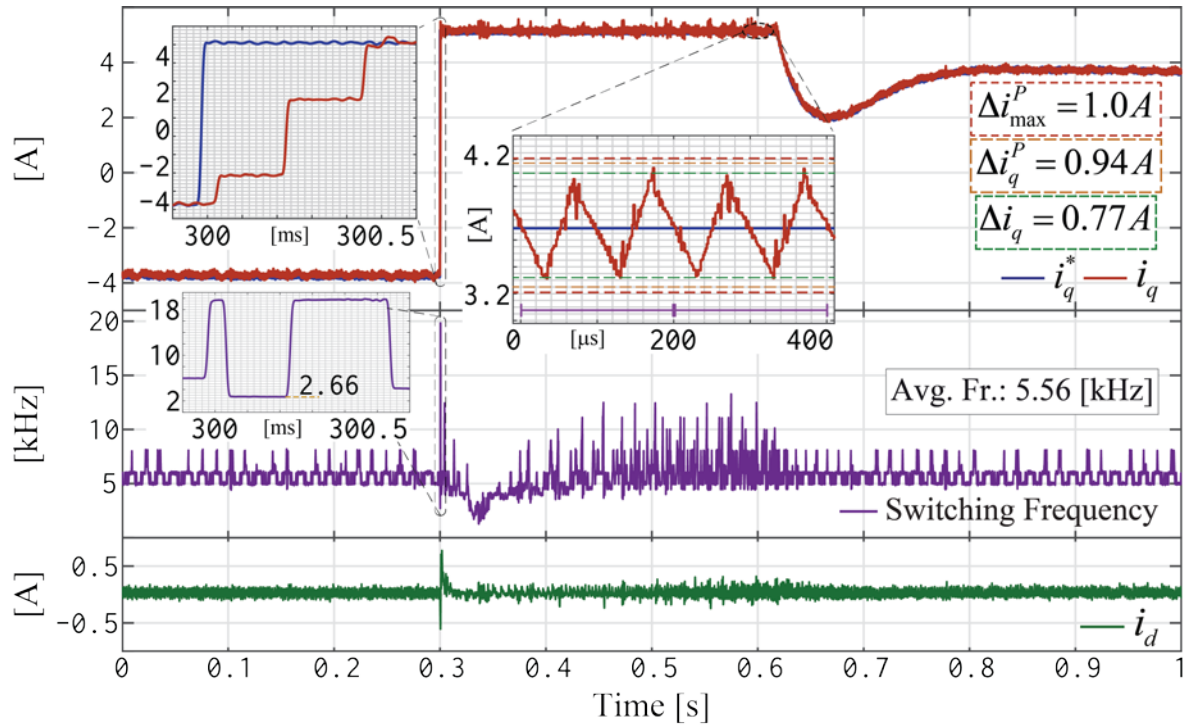

Figure 5.30: Current response of the predictive control scheme to a step change in the speed reference from -1500 to 1500 min^{-1} at full load. The current controller is configured to work with PI-Control and a maximum current ripple of 1.0A, the enhanced control of the current ripple is activated.

However, the average switching frequency increases to 5.56 kHz, which is a 27.23% of increase. It can also be noted that the commutation frequency is constantly changing due to the permanent regulation of the switching ripple. A zoom into the i_q current shows that the real switching ripple is reduced by 22% with respect to the predicted value.

These experiments show that the enhanced estimation of the current ripple can also be successfully used during high dynamic scenarios. Nonetheless, as also commented in previous cases, the precision of the current measurement becomes critical as the electromagnetic noise can introduce noise and an unnecessary high variation on the switching frequency.

5.3 Sensorless Control

The ideas presented in section 4.3 about sensorless control are evaluated in this section. A comparison between a classical implementation of field-oriented control and the proposed predictive control scheme with variable switching frequency is made, the implementation is explained in detail in section 4.3.2.1.

Unless otherwise noted the experiments were executed as follows: the absolute position of the rotor position (and thus, the permanent flux) is detected through the mechanical encoder. Once the initial position was found, the machine is quickly accelerated to the desired speed, loaded and then the control switches to sensorless operation.

5.3.1 Sensorless Control at High-Speed Ranges

For the first test, an instantaneous speed reversal at 600 min^{-1} (20% of the nominal speed) was generated, with the machine loaded at 100% of the nominal torque. For these experiments, the predictive control scheme with variable switching frequency with PI-Control for the current control was configured with the parameters presented in table 5.4.

The results of the predictive control scheme with oversampling in the current are presented in Figure 5.31. The margin of error during the steady-state operation is around 6º, however, before the speed reversal it has an offset of -6º, and after the speed reversal, it reaches -10º. This offset in the error has to be further examined; however, it could be caused by a DC-component introduced by the ΣΔ ADCs.

At the zero crossing of the speed the angular error reaches a maximum value of 45º; though, it will be quickly compensated. This perturbation can be noticed on the i_d current, over the speed reversal, it presents an overshoot slightly less than 3A (almost 75% of the nominal current) followed by an instant where the value remains around 1A (or approximately 25% of i_N), after that the i_d current remains unaffected.

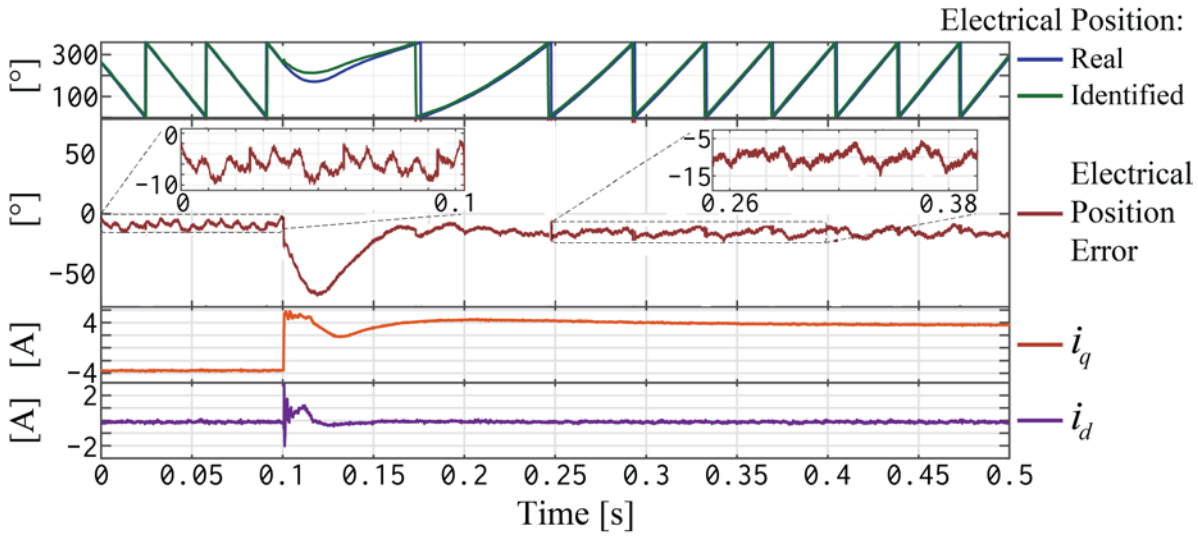

Figure 5.31: Speed reversal at 600 min^{-1} at full load, predictive control scheme with PI-Control, sensorless implementation with oversampling in the current.

For the next test, an instantaneous speed reversal at 1500 min^{-1} (50% of the nominal speed) was generated, with the machine loaded at 100% of the nominal torque. The results of the implementation with oversampling in both voltage and current are shown in Figure 5.32. The error range is very large during steady-state operation, around 15º.

Nevertheless, this implementation was the only one that completed the instantaneous speed reversal test at 1500 min^{-1}. As it can be seen, the maximum error during the speed reversal reaches the -35º, which is rapidly corrected. During the acceleration process, the angle error suffers an offset with an average of -12º, though, when the desired speed reference is reached the offset vanishes.

The dynamic response is very similar, compared to the results presented in section 5.2.4.2, where the same speed reversal test is performed with the use of the mechanical encoder. The i_q current reaches the desired reference in less than 500 µs, without a notable overshoot. The i_d current presents an overshoot of 1.5A (36.58% of i_N) during the speed reversal; however, it rapidly stabilized. During the steady-state operation, an oscillation can be noticed, which can be regarded to the oscillation in the identified position of the rotor.

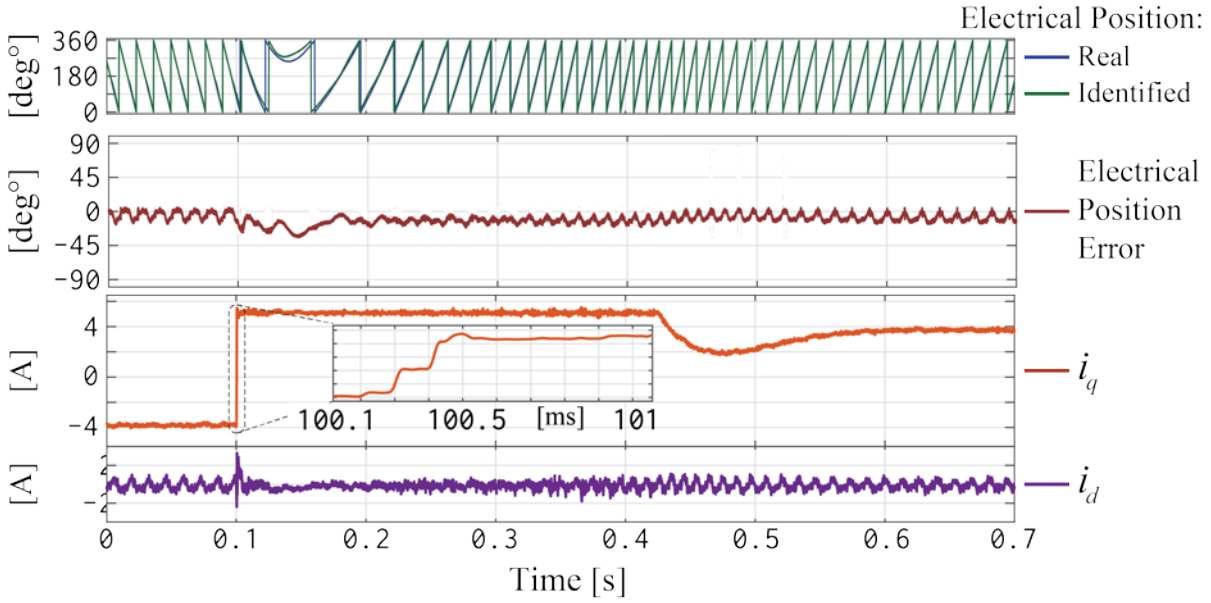

Figure 5.32 Speed reversal at 1500 min^{-1} at full load, predictive control scheme with variable switching frequency with PI-Control, sensorless implementation with oversampling in both voltage and current.

The experiments show that the sensorless algorithm at higher speeds affects the behavior of the enconderless identification system; an offset is introduced in the identified position, which is something that has to be further studied, one possible cause could be that a DC-component is introduced by the ΣΔ ADCs. Also the low frequency oscillation, that was pointed out in previous sections has to be examined in further works. In fact, contrary to the initial assumptions, the use oversampling of the voltage signals does not yield a better identification of the position. Nonetheless, the control can operate in high dynamic situations.

5.3.2 Sensorless Control at Low-Speed Ranges

For these experiments, the implementations of the conventional sensorless FOC, and the predictive control scheme with variable switching frequency with oversampling in the currents were tested and compared. The implementation of the predictive control scheme with oversampling in voltages and currents introduces an oscillation on the identified position of the rotor (as presented on the results in section 5.3.1), which at low-speed ranges increases, and therefore it was not considered.

5.3.2.1 Influence of Faster Processing Times on the Identification

For comparison purposes, the conventional implementation of FOC was tested with and without oversampling, the speed reference was set to 3 Hz electrical, the switching frequency to 4.5 kHz and 100% of the load torque; results are presented in Figure.5.33. On the first sight, there appears to be no benefit from the faster processing

of the sensorless algorithm and the oversampling of the currents. However, there is a small performance improvement of the sensorless algorithm when the oversampling of the current is present, as it can be seen in the zoom of the image it will track the position of the rotor with better accuracy.

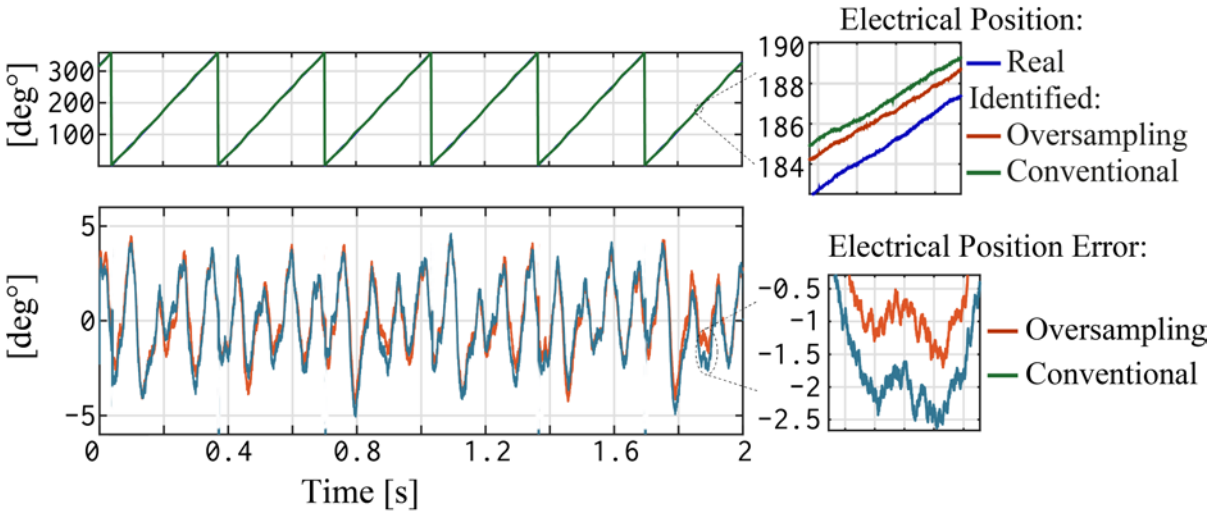

Figure.5.33: Comparison of the sensorless performance at 2% of the nominal speed (3 Hz electrical or 60 min^{-1}) at full load.

Despite the differences, both implementations perform well, with an error below the five electrical degrees. Once the speed reference is further reduced to 30 min^{-1} corresponding to an f_1=1.5 Hz at full load, the drive with FOC and no oversampling becomes unstable with the conventional implementation of the sensorless algorithm. The oversampling implementation kept working as presented in Figure 5.34, with an oscillation of approximately 8° peak-peak and a small offset with an average of -1°.

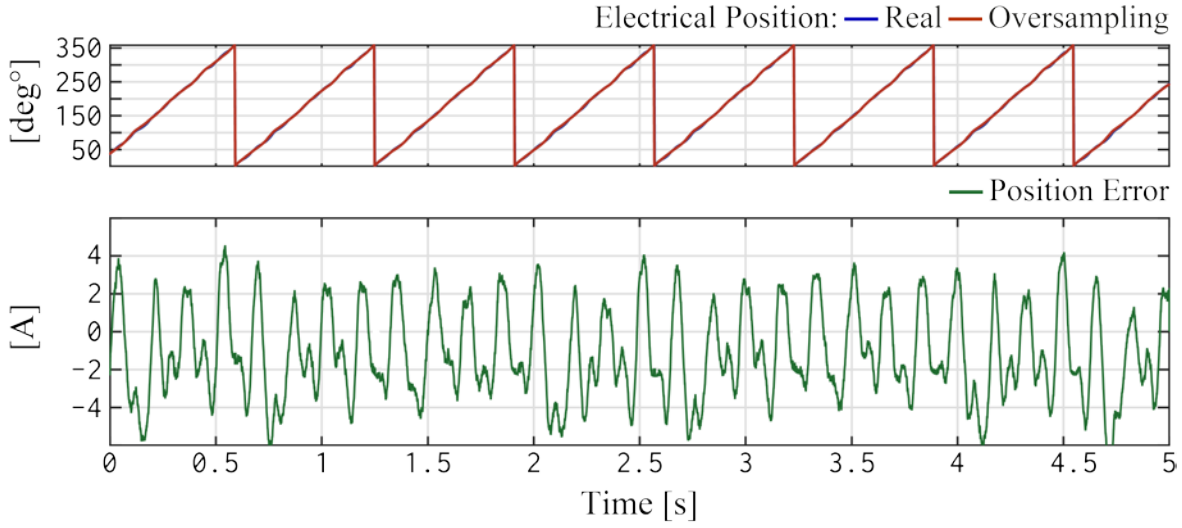

Figure 5.34: Sensorless performance with oversampling of the currents at 1% of the nominal speed (1.5 Hz electrical or 30 min^{-1}) at full load.

The lowest possible speed of operation with FOC was reached at 0.75Hz electrical, the load hat to be reduced to 80% of the load torque and the switching frequency to 1.25 kHz due to the effects of the non-linearities on the inverter, although they were compensated. The faster implementation of the sensorless algorithm presents an acceptable margin of error, below the 10º, as it can be seen in Figure 5.35.

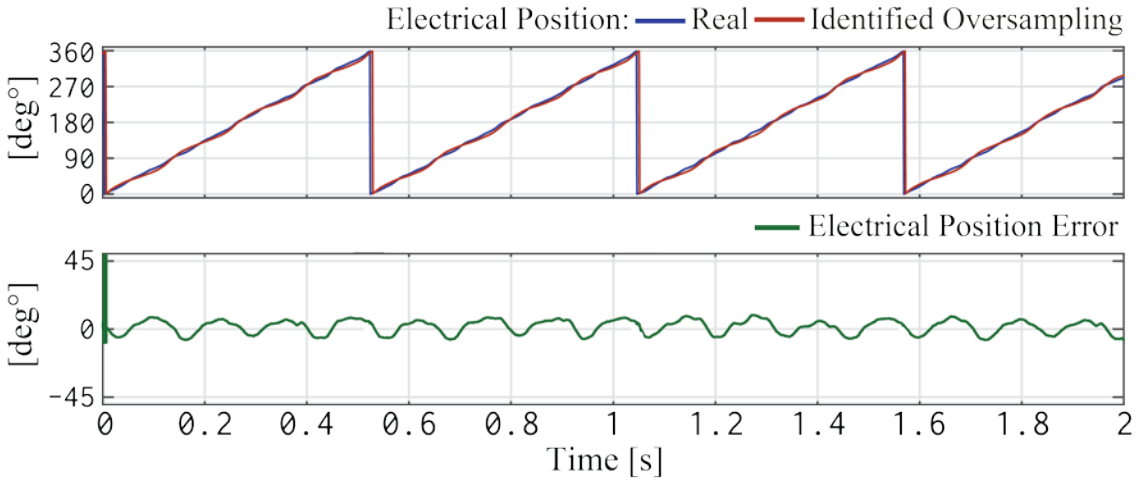

Figure 5.35: Sensorless performance with oversampling of the currents at 15 min^{-1} at 0.5% of the nominal speed (0.75 Hz electrical or 15min^{-1}) at 80% load torque.

The experiments have shown that the faster execution of the sensorless algorithm, as well as the oversampling of the currents, allows it to operate at slower speed ranges that a conventional implementation.

5.3.2.2 Performance of the Sensorless Predictive Control scheme with PWM of Variable Switching Frequency at Low-Speed Ranges

In this section, the performance of the predictive control scheme with variable switching frequency with PI-Control and oversampling on the currents for the encoderless identification of the rotor position is measured; the predictive control scheme with variable switching frequency is configured with the parameters shown in Table 5.5.

Table 5.5 Configuration of the predictive control scheme with variable switching frequency configuration parameters for the experiments in speed control mode during speed reversal with 100% load torque.

w_q	w_d	$w_{\Delta i_q}$	w_{f_s}	εi^P_{max}	Δi^P_{max}	i_{max}	On-the-fly reconfiguration	Enhanced prediction of the current ripple
1.0	0.25	1.0	1.0	0.75 A	2.0 A	3.0A	On	Off

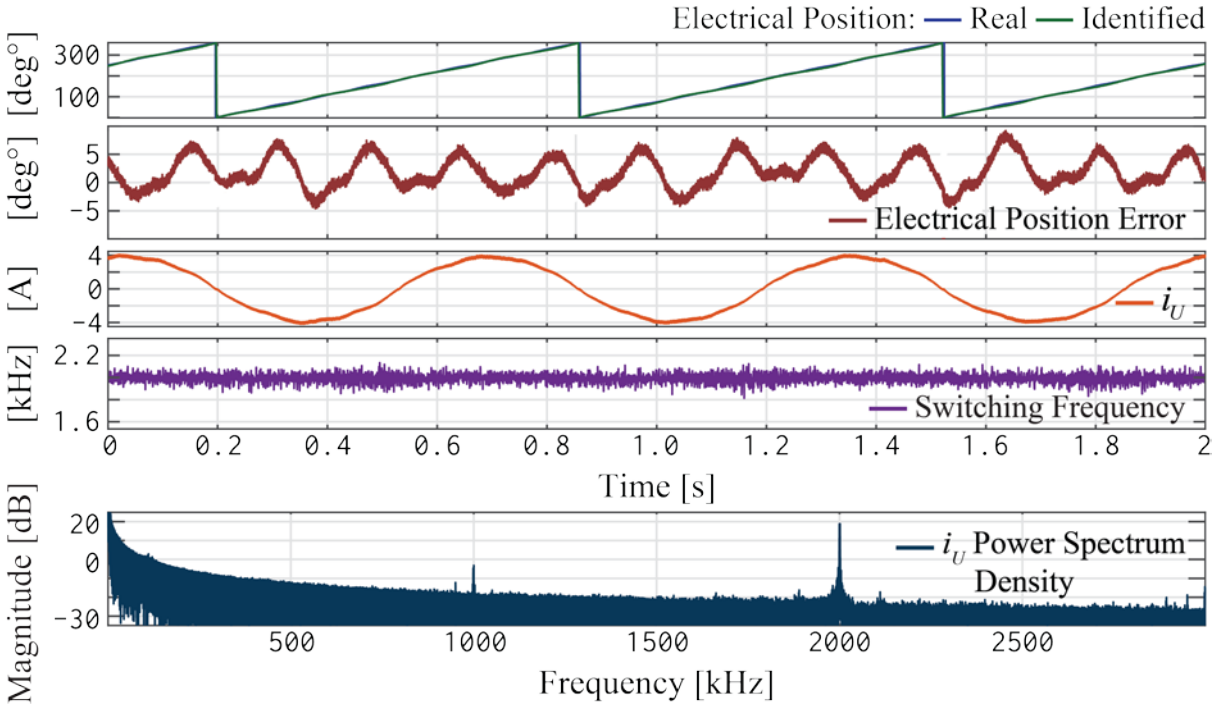

Figure 5.36 Sensorless performance of the predictive control scheme with oversampling on the currents at 1% of the nominal speed (1.5 Hz electrical or 30 min^{-1}) at full load.

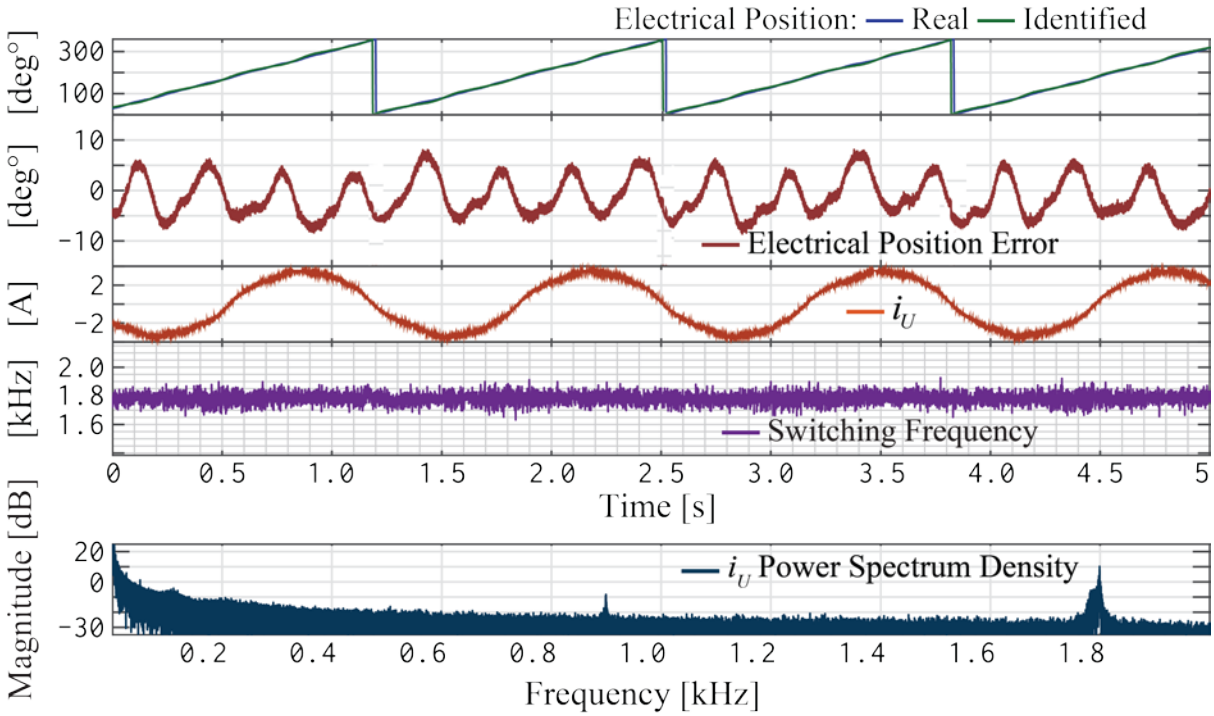

Figure 5.37 Sensorless performance of the predictive control scheme with oversampling on the currents at 0.5% of the nominal speed (0.75 Hz electrical or 15 min^{-1}) at 90% load torque.

Figure 5.36 shows the results when the speed reference is set to 30 min^{-1} corresponding to an f_1=1.5 Hz. The performance is as expected, the margin of error is within the 5º; however, an offset with an average of 1.88º is present; the average switching frequency is 2 kHz. The result is very similar to the one obtained with the conventional FOC, previously shown in Figure 5.34. Nevertheless, once the speed

reference is reduced to 15 min^{-1}, as presented in Figure 5.37, the load torque could be increased up to 90%, which is 10% more as with the conventional implementation.

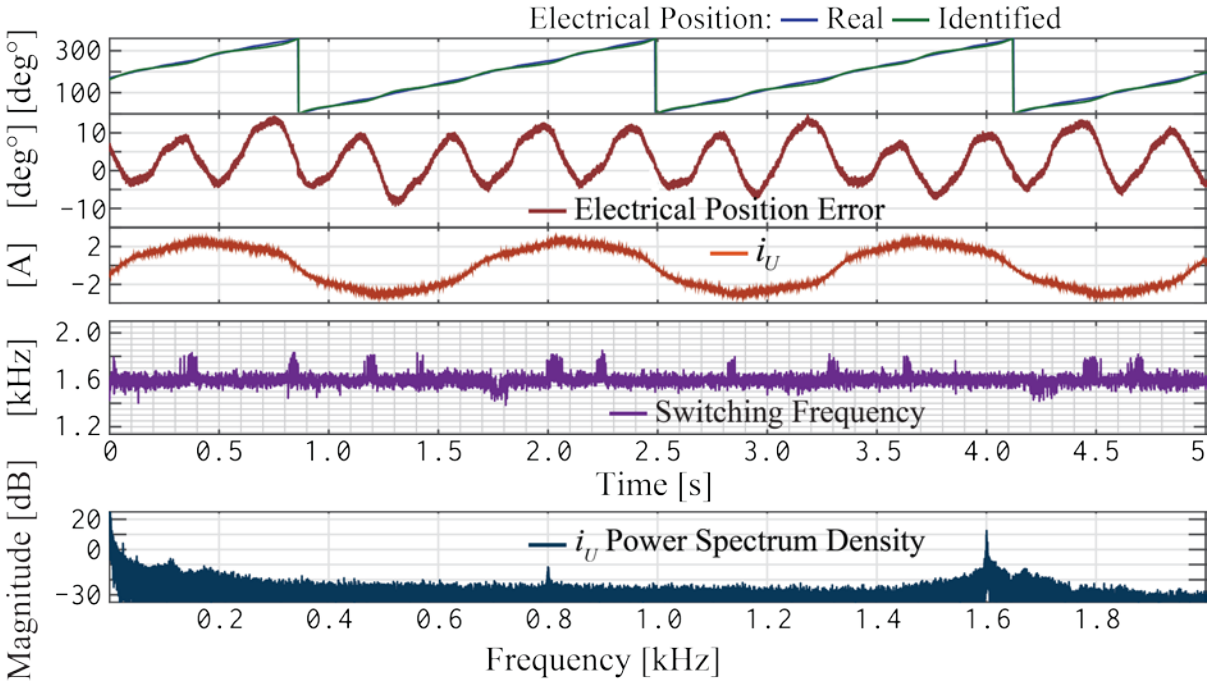

Figure 5.38 Sensorless performance of the predictive control scheme with oversampling on the currents at 0.4% of the nominal speed (0.6 Hz electrical or 12 min^{-1}) at 65% load torque.

The margin of error is maintained within the 6º, and the average switching frequency is reduced to 1.8 kHz. The speed reference was then further decreased to 12 min^{-1} corresponding to an $f_1 = 0.6$ Hz, results are presented in Figure 5.38. The sensorless operation of the drive is still stable, although the load torque had to be decreased to 65%. The margin of error increases to 10º, with an average offset of 3º; the average switching frequency is also reduced to 1.6 kHz.

These results show that the predictive control scheme with variable switching frequency further increase the performance of the sensorless identification of the rotor position, allowing an increase in the load torque and a reduction on the speed reference.

The following experiment is a speed reversal at 60 min^{-1}, where the speed reversal should take about 600 ms to be completed (2 min^{-1}/ms), the results are shown in Figure 5.39. Before the speed reversal, the margin of error is approximately 10º, however, an offset with an average of -6º is present. During the speed reversal the maximum, absolute error reaches the 60º, which can be better noticed in the overshoot on the i_q current. The response of the i_d current remains almost unaffected.

In spite of the high margin of error during the speed reversal, the control and the sensorless identification systems are able to correct and maintain the operation. After

the direction of rotation is successfully changed, the margin of error is reduced to 6º, with an average offset of minus three degrees.

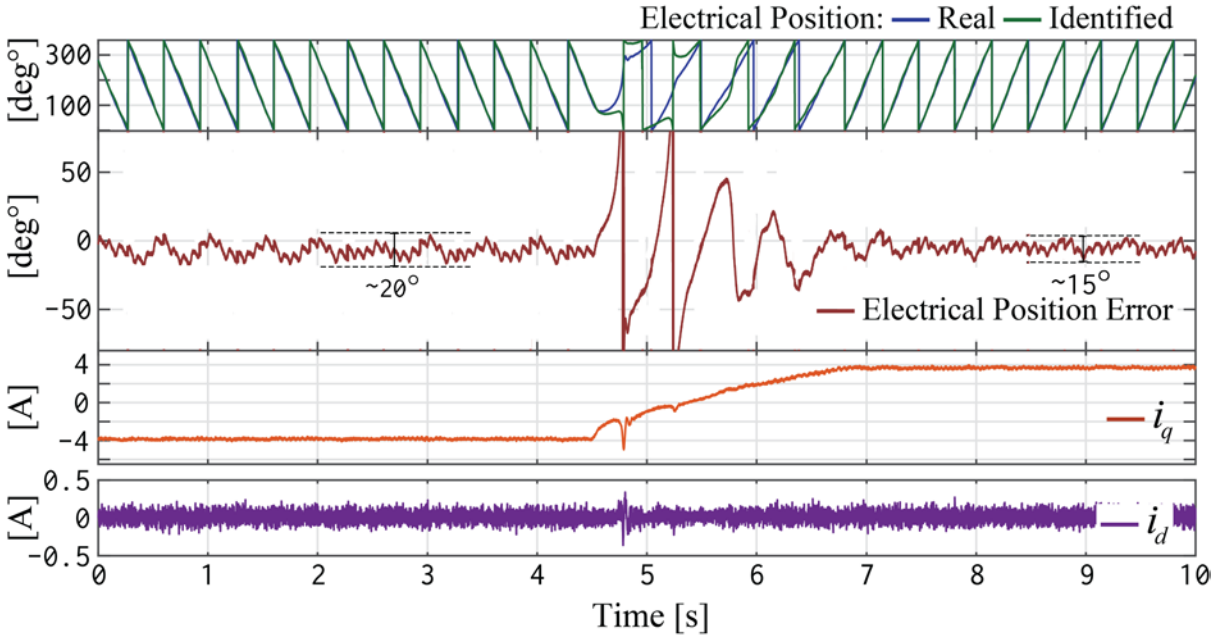

Figure 5.39 Sensorless performance of the predictive control scheme with oversampling on the currents during a speed reversal at 2% of the nominal speed (3 Hz electrical or 60 min^{-1}) at full load.

Finally, a speed reversal at 30 min^{-1}, where the speed reversal should take around 300 ms to be completed (2 min^{-1}/ms), results are shown in Figure 5.40. Before the speed reversal the margin of error is 6º; nonetheless, it presents an offset with an average of -14º. During the speed reversal, the maximum error reaches -65º, which can be noticed on the overshoot of the i_q current at $t \simeq 4.2$ s. The response of the i_d current is almost unaffected.

Once the speed reversal is completed, the identified position of the rotor takes about 3 seconds to stabilize, after that, the margin of error is reduced to 4º without offset. These results show that the execution of the sensorless algorithm at higher speeds and the oversampling of the currents indeed improve the response of the identification system. Furthermore, opposed to the initial assumption, the oversampling on the voltage signals did not increase the performance at low-speed ranges.

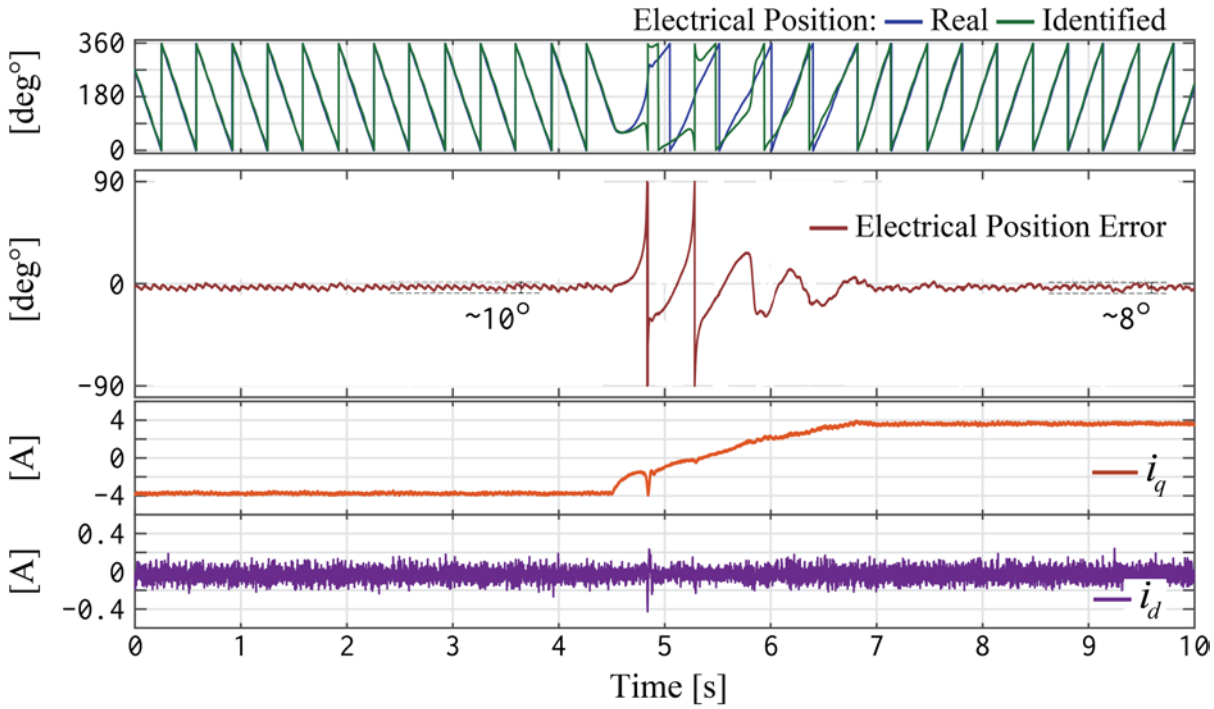

Figure 5.40 Sensorless performance of the predictive control scheme with oversampling on the currents during a speed reversal at 1% of the nominal speed (1.5 Hz electrical or 30 min^{-1}) at full load.

6 Conclusions

The main objective of this work was the research of control techniques for power converters that take advantage of the powerful processing capabilities of FPGAs and the possibility to modify conventional peripherals to a very precise level. As a result, a novel predictive control scheme for the PMSM with variable switching frequency modulator-based is presented.

On a first approach, a pulse-width modulation stage that can be reconfigured without stopping the execution of the whole system was developed; thus, the switching frequency of the inverter and the sampling time of the control can be changed during the execution. This opens new possibilities for control techniques, which for this work were applied in order to improve the performance of field-oriented control.

Following, the PI controllers conventionally used in field-oriented control were modified so they could be used during the prediction approach with different sampling times, without saturating the integral part of the controller. This first approach of the controller was first presented in [138]. Subsequently, a predictive control with variable switching frequency based on the Model-based Predictive Control theory was developed and presented in [139]

In the initial approach, the predictive control did not include the ripple of the electromagnetic torque ripple in the cost function and thus only the switching losses were affected. Further enhancements of the proposal increased the complexity of the control, also adding constraints, all aimed to reduce the switching losses by directly reducing the switching frequency and maintaining a minimum level of desired performance mainly regulated by the maximum allowed torque ripple.

It then became clear that the control scheme with of the predictive PI-Controllers successfully eliminates some of the drawbacks of conventional MPC and exploits the features of a FPGA implementation by calculating most of the algorithms in silicon and in parallel. Furthermore, the predictive PI implementation outperformed the conventional MPC and improved the dynamic response, as presented in [140].

Some of these improvements are: the substantial reduction of overshoot, no steady-state error at lower switching frequencies and a fast-dynamic response. Nevertheless, it does introduce some oscillations in the current when the switching frequency is reduced,

as opposed to the model-based variation. In the final steps, the oversampling of the current provided by the ΣΔ ADCs was exploited, developing high bandwidth control system that could also affect the harmonics i.e. the switching ripple, these results were presented in [141].

Such manipulations of the controlled variable can hardly be achieved by using only a microcontroller or DSP for the implementation. Furthermore, the execution of algorithms can be integrated into an already complex control system, without a great impact on the overall performance of the system. This can be accomplished thanks to the parallel processing capabilities of the FPGAs.

Finally, to complement the proposed predictive control strategy, a sensorless implementation was developed, evaluated and presented in [142]. Furthermore, the faster processing capabilities of the FPGA allowed the sensorless identification of the rotor position to be executed at very low speeds without the need of a signal injection.

As it has been presented in this work, a viable predictive control scheme for the PMSM with PWM of variable switching frequency is proposed, which overcomes the majority of the drawbacks of conventional field-oriented control. As it is well known, FOC relies on a modulation scheme like the space phasor modulation. Therefore, it features several advantages of FOC and SPM like zero steady-state error, reduced torque ripple, and reduced THD.

Moreover, the strengths of other control approaches like DTC and MPC are combined to further increase the performance of the proposed control. These advantages are variable switching frequency and ease on the tuning of the performance of the control using a cost function.

The main drawback of this approach is the use of an FPGA for the implementation. The learning curve toward developing functional FPGA systems is way steeper than that of microcontrollers or digital signal processors based systems. There are so many layers in which the developer must become an expert, so that she or he is able to successfully build an FPGA-based controller.

Out of the box, a microcontroller development board comes with all the necessary hardware and software to develop a motion control system. In contrast, the FPGA board requires external hardware like the ΣΔ ADCs, the internal peripherals like the PWM units, DAC interface, among others, have to be either written from scratch or instantiated from IP of the FPGA vendor.

So why bother? As it has been shown in this work, FPGAs are a viable platform to build control platforms for AC electrical drives. An FPGA enables the designer to create powerful custom DSPs, which cannot be entirely duplicated using commercial microcontrollers. These custom DSPs can take advantage of digital circuits, which can work in parallel to process information at unmatched speed-rates and if it also required, one or more microcontroller units can be implemented, so the flexibility of software is always at hand.

A Machine Parameters

Table 6.1: Driven PMSM machine parameters

	Value	Unit
Manufacturer	ABB	
Nominal Power	2.01	kW
Nominal Voltage	360	V
Nominal Current	4.1	A
Nominal Torque	6.4	Nm
Nominal Speed	3000	min^{-1}
Pole pairs	3	
Winding resistance R_{UV}	4.0	Ω
Winding inductance L_{UV}	15.2	mH
Moment of inertia	8.9	kg cm^2
Induced voltage	100	V/1000 min^{-1}

Table 6.2: Load PMSM machine parameters

	Value	Unit
Manufacturer	Siemens	
Nominal Power	2.3	kW
Nominal Voltage	400	V
Frequency	50	Hz
Nominal Current	5.6	A
Nominal Torque	7.3	Nm
Nominal Speed	3000	min-1
Pole pairs	4	
Moment of inertia (with break)	17.3	kg cm2
Brake current	0.75	A

Table 6.3: Commercial drive parameters, drives the load machine.

	Value	Unit
Manufacturer	ABB	
Model	ACS850-04-014a-5+E200+J400	
P_n	5.5/7.5	kW/HP
U_1	3~380…500	V
I_{1n}	18	A
f_1	48…63	Hz
U_2	3~0…U1	V
I_{2n}	14	A
f_2	0…500	Hz

B Switching Frequency Values

Table 6.4: Set of valid switching frequency values for the operation of the VSI.

	Frequency [kHz]		[kHz]		[kHz]
1	0.8	**12**	2.222	**23**	8.0
2	0.9	**13**	2.5	**24**	8.888
3	1.0	**14**	2.666	**25**	10.0
4	1.111	**15**	3.0	**26**	11.111
5	1.25	**16**	3.2	**27**	12.5
6	1.333	**17**	3.6	**28**	13.333
7	1.5	**18**	4.0	**29**	15.0
8	1.6	**19**	4.5	**30**	16.0
9	1.777	**20**	5.0	**31**	18.0
10	1.8	**21**	6.0	**32**	20.0
11	2.0	**22**	6.666		

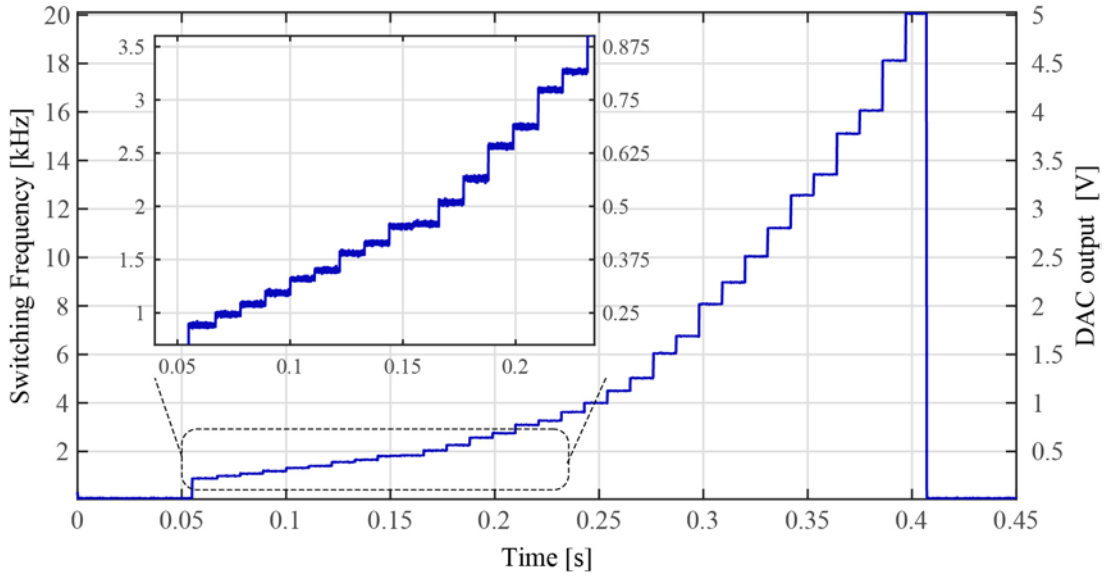

Figure 6.1: Variable frequency values relation between DAC voltage output and switching frequency in kHz.

Bibliography

[1] W. Leonhard, *Control of Electrical Drives*, Third edit. Berlin: Springer-Verlag, 2001.

[2] J. Holtz, "Developments in Sensorless AC Drive Technology Keynote Paper," in *International Conference on Power Electronics and Drives Systems, 2005. PEDS 2005.*, 2005, vol. 2, pp. 9–16.

[3] M. Pacas, "Sensorless drives in industrial applications," *IEEE Ind. Electron. Mag.*, vol. 5, no. 2, pp. 16–23, 2011.

[4] J. Rodriguez *et al.*, "State of the art of finite control set model predictive control in power electronics," *IEEE Trans. Ind. Informatics*, vol. 9, no. 2, pp. 1003–1016, 2013.

[5] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, 2011.

[6] E. Monmasson and M. N. Cirstea, "FPGA Design Methodology for Industrial Control Systems - A Review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, 2007.

[7] F. Blaschke, "Das Prinzip der Feldorientierung, Grundlage für die Transvektor-Regelung von Drehfeldmaschinen," *Siemens Zeitschrift*, vol. 45, no. 10, pp. 757–760, 1971.

[8] I. Takahashi and T. Noguchi, "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor," *Ind. Appl. IEEE Trans.*, vol. IA-22, no. 5, pp. 820–827, 1986.

[9] T. G. Habetler and D. M. Divan, "Acoustic Noise Reduction in Sinusoidal PWM Drives Using a Randomly Modulated Carrier," *IEEE Trans. Power Electron.*, vol. 6, no. 3, pp. 356–363, 1991.

[10] A. M. Trzynadlowski, "Space vector PWM technique with minimum switching losses and a variable pulse rate," *IEEE Trans. Ind. Electron.*, vol. 44, no. 2, pp. 173–181, 1997.

[11] R. L. Kirlin, C. Lascu, and A. M. Trzynadlowski, "Shaping the noise spectrum in power electronic converters," *IEEE Trans. Ind. Electron.*, vol. 58, no. 7, pp. 2780–2788, 2011.

[12] A. Peyghambari, A. Dastfan, and A. Ahmadyfard, "Selective Voltage Noise Cancellation in Three-Phase Inverter Using Random SVPWM," *IEEE Trans. Power Electron.*, vol. 31, no. 6, pp. 4604–4610, 2016.

[13] X. Mao, R. Ayyanar, and H. K. Krishnamurthy, “Optimal variable switching frequency scheme for reducing switching loss in single-phase inverters based on time-domain ripple analysis,” *IEEE Trans. Power Electron.*, vol. 24, no. 4, pp. 991–1001, 2009.

[14] D. Jiang and F. F. Wang, “Variable switching frequency PWM for three-phase converters based on current ripple prediction,” *IEEE Trans. Power Electron.*, vol. 28, no. 11, pp. 4951–4961, 2013.

[15] Z. Wang, J. Chen, M. Cheng, and K. T. Chau, “Field-oriented control and direct torque control for paralleled VSIs Fed PMSM drives with variable switching frequencies,” *IEEE Trans. Power Electron.*, vol. 31, no. 3, pp. 2417–2428, 2016.

[16] J. R. Frus and B. C. Kuo, “Closed-loop control of step motors using waveform detection,” in *Proceedings of the international conference on stepping motors and systems*, 1976, pp. 77–84.

[17] V. Barinberg and F. Götz, “Verbessertes Spannungs-Modell zur sensorlosen Ansteuerung von Antrieben mit permanentmagneterregten Synchronmotoren,” *SPS IPC Drives*, 2008.

[18] P. P. Acarnley and J. F. Watson, “Review of position-sensorless operation of brushless permanent-magnet machines,” *IEEE Transactions on Industrial Electronics*, vol. 53, no. 2. pp. 352–362, 2006.

[19] M. Jansson, L. Harnefors, O. Wallmark, and M. Leksell, “Synchronization at startup and stable rotation reversal of sensorless nonsalient PMSM drives,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 379–387, 2006.

[20] C. Olivieri and M. Tursini, “A novel PLL scheme for a sensorless PMSM drive overcoming common speed reversal problems,” in *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2012 International Symposium on*, 2012, pp. 1051–1056.

[21] Y. Zhang and J. Liu, “An improved Q-PLL to overcome the speed reversal problems in sensorless PMSM drive,” in *2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia)*, 2016, pp. 1884–1888.

[22] S. P. Feuersänger, “Drehgeberlose Identifikation der Rotorlage der elektrisch erregten Synchronmaschine in Mittelspannungsantrieben,” Siegen, 2015.

[23] M. Schroedl, “Sensorless control of AC machines at low speed and standstill based on the ‘INFORM’ method,” in *Industry Applications Conference, 1996. Thirty-First IAS Annual Meeting, IAS ’96., Conference Record of the 1996 IEEE*, 1996, pp. 270–277.

[24] M. Linke, R. Kennel, and J. Holtz, “Sensorless speed and position control of synchronous machines using alternating carrier injection,” *Electr. Mach. Drives Conf. 2003. IEMDC’03. IEEE Int.*, vol. 2, pp. 1211–1217 vol.2, 2003.

[25] Z. Ma and R. Kennel, “FPGA based signal injection sensorless control of SMPMSM using Delta-Sigma A/D conversion,” in *Sensorless Control for Electrical Drives (SLED), 2012 IEEE Symposium on*, 2012.

[26] D. Paulus and J.-F. Stumper, “Sensorless Control of Synchronous Machines Based on Direct Speed and Position Estimation in Polar Stator-Current Coordinates,” *IEEE Trans. Power Electron.*, vol. 28, no. 5, pp. 2503–2513, 2013.

[27] Z. Chen, X. Cai, R. Kennel, and F. Wang, “Enhanced sensorless control of SPMSM based on Stationary Reference Frame High-Frequency Pulsating Signal injection,” in *2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia)*, 2016, pp. 885–890.

[28] S. Feuersänger and M. Pacas, “Initial rotor position identification in medium voltage synchronous machines,” in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 1852–1857.

[29] S. Feuersänger and M. Pacas, “Rotor position identification in synchronous machines by using the excitation machine as a sensor,” in *2016 IEEE Symposium on Sensorless Control for Electrical Drives (SLED)*, 2016, pp. 1–6.

[30] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, “Predictive Control in Power Electronics and Drives,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 12, pp. 4312–4324, 2008.

[31] S. V. Emeljanov, *Automatic control systems with variable structure (Automatische Regelsysteme mit verändlicher Struktur)*. München: Oldenburg Verlag, 1969.

[32] M. Morari and J. H. Lee, “Model predictive control: Past, present and future,” in *Computers and Chemical Engineering*, 1999, vol. 23, no. 4–5, pp. 667–682.

[33] R. Kennel, A. Linder, and M. Linke, “Generalized predictive control (GPC)-ready for use in drive applications?,” *Power Electron. Spec. Conf. 2001. PESC. 2001 IEEE 32nd Annu.*, vol. 4, pp. 1839--1844 vol. 4, 2001.

[34] A. Linder and R. Kennel, “Model Predictive Control for Electrical Drives,” *Power Electron. Spec. Conf. 2005. PESC ’05. IEEE 36th*, pp. 1793–1799, 2005.

[35] A. Linder, R. Kanchan, R. Kennel, and P. Stolze, *Model-based Predictive Control of Electric Drives*. Muenchen: Cuvillier Verlag Goettingen, 2010.

[36] E. Camacho and C. Bordons, *Model Predictive Control*. Sevilla, Spain: Springer-Verlag, 1999.

[37] S. Mariethoz, A. Domahidi, and M. Morari, “High-Bandwidth Explicit Model Predictive Control of Electrical Drives,” *IEEE Trans. Ind. Appl.*, vol. 48, no. 6, pp. 1980–1992, 2012.

[38] S. Saeidi and R. Kennel, “Continuous set nonlinear model predictive control for an induction motor,” in *15th European Conference on Power Electronics and Applications (EPE), 2013*, 2013, pp. 1–10.

[39] G. Torrisi, S. Grammatico, M. Morari, and R. S. Smith, “Model Predictive Control approaches for centrifugal compression systems,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 4320–4325.

[40] S. Almér, S. Mariéthoz, and M. Morari, “Dynamic Phasor Model Predictive Control of Switched Mode Power Converters,” *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 349–356, 2015.

[41] J. Rodriguez *et al.*, “Predictive Current Control of a Voltage Source Inverter,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 495–503, 2007.

[42] P. Correa, M. Pacas, and J. Rodríguez, “Predictive torque control for inverter-fed induction machines,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1073–1079, 2007.

[43] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, “Model Predictive Control - A Simple and Powerful Method to Control Power Converters,” *Ind. Electron. IEEE Trans.*, vol. 56, no. 6, pp. 1826–1838, 2009.

[44] S. Vazquez *et al.*, “Model predictive control with constant switching frequency using a discrete space vector modulation with virtual state vectors,” in *Proceedings of the IEEE International Conference on Industrial Technology*, 2009.

[45] J. Rodriguez and P. Cortes, *Predictive Control of Power Converters and Electrical Drives*. Valparaiso, Chile: Wiley, 2012.

[46] C. Bordons and C. Montero, “Basic Principles of MPC for Power Converters: Bridging the Gap Between Theory and Practice,” *IEEE Industrial Electronics Magazine*, vol. 9, no. 3. pp. 31–43, 2015.

[47] S. Kouro, M. Perez, J. Rodriguez, A. Llor, and H. Young, “Model Predictive Control, MPC’s Role in the Evolution of Power Electronics,” *IEEE Industrial Electronics Magazine*, vol. 9, no. 4, pp. 8–21, 2015.

[48] T. Mouton, P. Stolze, T. Geyer, M. Tomlinson, and R. Kennel, “Predictive control vs. linear control for current control of a single-leg inverter,” in *2013 IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE),* 2013, pp. 1–8.

[49] M. Pozo, “Finite Set Model Predictive Control of the PMSM with Sine-Wave Filter,” Siegen, 2015.

[50] R. P. Aguilera, P. Lezana, and D. E. Quevedo, “Finite-control-set model predictive control with improved steady-state performance,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 2, pp. 658–667, 2013.

[51] H. Bähring, *Anwendungsorientierte Mikroprozessoren Mikrocontroller und Digitale Signalprozessoren*. Springer Berlin Heidelberg, 2010.

[52] F. Mealy, Bryan; Tappero, *Free Range VHDL*. Electronically Published, 2016.

[53] IEEE Computer Society, *IEEE Standard VHDL Language Reference Manual*, vol. 2008. 2009.

[54] D. Automation and S. Committee, *IEEE Standard for Verilog Hardware Description Language*, vol. 2005, no. April. 2006.

[55] IEEE Computer Society, *IEEE Standard for SystemVerilog - Unified Hardware Design, Specification, and Verification Language*, no. November. 2013.

[56] M. W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhodja, and N. Patin, "FPGA-based current controllers for AC machine drives - A review," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4. pp. 1907–1925, 2007.

[57] A. Maalouf, S. Le Ballois, E. Monmasson, J. Y. Midy, and C. Bruzy, "FPGA-based sensorless control of brushless synchronous starter generator at standstill and low speed using high frequency signal injection for an aircraft application," *IECON Proc. (Industrial Electron. Conf.*, pp. 4003–4008, 2011.

[58] I. Bahri, L. Idkhajine, E. Monmasson, and M. El Amine Benkhelifa, "Hardware/Software codesign guidelines for system on chip FPGA-based sensorless AC drive applications," *IEEE Trans. Ind. Informatics*, vol. 9, no. 4, pp. 2165–2176, 2013.

[59] E. Monmasson, I. Bahri, L. Idkhajine, A. Maalouf, and W. M. Naouar, "Recent advancements in FPGA-based controllers for AC drives applications," in *Proceedings of the International Conference on Optimisation of Electrical and Electronic Equipment, OPTIM*, 2012, vol. 33, no. 0, pp. 8–15.

[60] J. J. Rodriguez-Andina, M. D. Valdes-Pena, and M. J. Moure, "Advanced Features and Industrial Applications of FPGAs - A Review," *Ind. Informatics, IEEE Trans.*, vol. 11, no. 4, pp. 853–864, 2015.

[61] C. Altera, "Implementing FPGA Design with the OpenCL Standard," 2011.

[62] M. Zhixun, "FPGA-Based High Performance Sensorless Control for PMSM Drives," TU Muenchen, 2013.

[63] E. Monmasson, L. Idkhajine, and M. W. Naouar, "FPGA-based controllers," *IEEE Ind. Electron. Mag.*, vol. 5, no. 1, pp. 14–26, 2011.

[64] J. Millan, P. Godignon, X. Perpina, A. Perez-Tomas, and J. Rebollo, "A survey of wide bandgap power semiconductor devices," *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2155–2163, 2014.

[65] L. Fu, X. Zhang, M. Scott, Y. Chengcheng, and W. Jin, "The evaluation and application of wide bandgap power devices," in *Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2014 IEEE Conference and Expo*, 2014.

[66] L. A. Fernandez-Serantes, H. Berger, W. Stocksreiter, and G. Weis, "Ultra-High Frequent Switching with GaN-HEMTs using the Coss-Capacitances as non-dissipative Snubbers," in *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management; Proceedings of*, 2016, pp. 296–303.

[67] G. Königsmann, R. Herzer, S. Beutow, M. Rossberg, and Semikron, "High power, high frequency gate driver for SiC-MOSFET modules," in *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management; Proceedings of*, 2016, pp. 728–734.

[68] L. Idkhajine, E. Monmasson, and A. Maalouf, "Fully FPGA-based sensorless control for AC drive using an extended kalman filter," *Ind. Electron. 2009. IECON '09. 35th Annu. Conf. IEEE*, vol. 59, no. 10, pp. 2925–2930, 2009.

[69] M. Abdellatif, M. W. Naouar, I. Slama-Belkhodja, and E. Monmasson, "FPGA-based vector PI regulator for electrical drives control," *2007 Eur. Conf. Power Electron. Appl. EPE*, vol. 729, 2007.

[70] "FPGA-Based Model Predictive Controller for Direct Matrix Converter," *IEEE Trans. Ind. Electron.*, 2016.

[71] Z. Zhang and R. Kennel, "FPGA-based Direct Model Predictive Power and Current Control of 3L NPC Active Front Ends," in *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management; Proceedings of*, 2016, pp. 629–636.

[72] J. Druant, T. Vyncke, F. De Belie, P. Sergeant, and J. Melkebeek, "Adding Inverter Fault Detection to Model-Based Predictive Control for Flying-Capacitor Inverters," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2054–2063, 2015.

[73] P. Kovacs, *Transient Phenomena in Electrical Machines*. Hungary: Elsevier, 1984.

[74] P. K. Kovacs and I. Racz, *Transiente Vorgänge in Wechselstrommaschinen*. Budapest: Verlag der Ungarischen Akademie der Wissenschaften, 1959.

[75] J. Stepina, "Raumzeiger als Theorie der elektrischen Maschinen," *Elektrotechnische Zeitschrift*, vol. 88, pp. 584–588, 1967.

[76] L. Serrano-Iribarnegaray, "The modern space-phasor theory, part I: Its coherent formulation and its advantages for transient analysis of converter-fed AC machines," *Eur. Trans. Electr. Power*, vol. 3, no. 2, pp. 171–180, 1993.

[77] L. Serrano-Iribarnegaray, "The modern space-phasor theory, Part II: Comparison with the generalized machine theory and the space-vector theory," *Eur. Trans. Electr. Power*, vol. 3, no. 3, pp. 213–219, 1993.

[78] L. Serrano-Iribarnegaray, "The Space Phasor Theory," in *Modern Electrical Drives*, A. Bülent Ertan, H.; Yildirim Üçtuğ, M.; Colyer, R.; Consoli, Ed. 2000, pp. 393–423.

[79] J. F. Gieras and M. Wing, *Permanent magnet motor technology: design and applications*, Second. New York: Marcel Dekker, Inc., 2002.

[80] S. Morimoto, “Trend of Permanent Magnet Synchronous Machines,” *IEEJ Trans. Electr. Electron. Eng.*, vol. 2, no. 2, pp. 101–108, 2007.

[81] K. R. Weeber *et al.*, “Advanced permanent magnet machines for a wide range of industrial applications,” *Power Energy Soc. Gen. Meet. 2010 IEEE*, pp. 1860–1867, 2010.

[82] F. Giri, *AC Electric Motors Control: Advanced Design Techniques and Applications*. Wiley, 2013.

[83] R. Krishnan, *Permanent Magnet Synchronous and Brushless DC Motor Drives*. Boca Raton: CRC Press, 2010.

[84] R. Krishnan, *Electric Motor Drives: Modeling, Analysis, and Control*. New Jersey: Prentrice Hall, 2001.

[85] J. Weber, “Prädiktive Direkte Drehmomenteregelung der Synchronmaschine,” Siegen, 2004.

[86] B. K. Bose, *Power Electronics And Motor Drives*. Elsevier, 2006.

[87] M. H. Rashid, *Power Electronics: Circuits, Devices and Applications*. Pearson Education, 2004.

[88] H. W. an der Broeck, H. C. Skudelny, and G. V. Stanke, “Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space Vectors,” *IEEE Trans. Ind. Appl.*, vol. 24, no. 1, pp. 142–150, 1988.

[89] N. P. Quang and J.-A. Dittrich, *Vector Control of Three-Phase AC Machines*. Berlin: Springer-Verlag, 2008.

[90] K. Hasse, “Zur Dynamik drehzahlgeregelter Antriebe mit stromrichtergespeisten Asynchron-Kurzschlußläufermaschinen,” TH Darmstadt, 1969.

[91] F. Blaschke, “The principle of field orientation as applied to the new TRANSVEKTOR closed loop control system for rotating field machines,” *Siemens Rev.*, vol. 34, pp. 217–220, 1972.

[92] R. Soeterboek, *Predictive Control: A Unified Approach*, First. Cambridge, UK: Prentrice Hall, 1992.

[93] Xilinx, “Putting Things in Perspective,” *Xcell*, 1999. [Online]. Available: http://www.xilinx.com/publications/archives/xcell/Xcell32.pdf. [Accessed: 06-Mar-2016].

[94] U. Baese-Meyer, *Digital Signal Processing with Field Programmable Gate Arrays*, Second. Tallahaessee: Springer-Verlag, 2004.

[95] J. Serrano, “Digital Signal Processing using Field Programmable Gate Arrays,” *Digit. Process.*, pp. 346–355, 2008.

[96] Y. Bai *et al.*, “FPGA vs DSP: A throughput and power efficiency comparison for Hierarchical Enumerative Coding,” *IEEE/IFIP Int. Conf. VLSI Syst. VLSI-SoC*, pp. 318–321, 2013.

[97] U. Sinha, D. S. P. Product, and M. Specialist, “Enabling Impactful DSP Designs on FPGAs with Hardened Floating-Point Implementation Evolution of Floating-Point Implementation in Altera FPGAs,” no. August, pp. 1–7, 2014.

[98] J. E. Volder, “The CORDIC Trigonometric Computing Technique,” *Electron. Comput. IRE Trans.*, vol. EC-8, no. 3, pp. 330–334, 1959.

[99] J. S. Walther, “A unified algorithm for elementary functions,” in *Proceedings of the May 18-20, 1971, spring joint computer conference on - AFIPS ’71 (Spring)*, 1971, p. 379.

[100] P. K. Meher, J. Valls, T. B. Juang, K. Sridharan, and K. Maharatna, “50 years of CORDIC: Algorithms, architectures, and applications,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 56, no. 9, pp. 1893–1907, 2009.

[101] P. M. Aziz, H. V. Sorensen, and J. vn der Spiegel, “An overview of sigma-delta converters,” *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 61–84, 1996.

[102] W. Kester, “Which ADC Architecture Is Right for Your Application ?,” *Analog Dialogue*, vol. 39, no. 6, pp. 1–8, 2005.

[103] J. M. De La Rosa, “Sigma-delta modulators: Tutorial overview, design guide, and state-of-the-art survey,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 1. pp. 1–21, 2011.

[104] H. Inose, Y. Yasuda, and J. Murakami, “A Telemetering System by Code Modulation- Δ-Σ Modulation,” *IRE Trans. Sp. Electron. Telem.*, vol. 8, no. 3, pp. 204–209, 1962.

[105] E. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 2, pp. 155–162, 1981.

[106] J. Candy, “Decimation for Sigma Delta Modulation,” *IEEE Trans. Commun.*, vol. 34, no. 1, pp. 72–76, 1986.

[107] J. C. Candy, “A Use of Double Integration in Sigma Delta Modulation,” *IEEE Trans. Commun.*, vol. 33, no. 3, pp. 249–258, 1985.

[108] M. Shiraishi, “Simultaneous coefficient calculation method for Sinc/sup N/ FIR filters,” *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 50, no. 4, pp. 523–529, 2003.

[109] A. V Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete Time Signal Processing*, vol. 1999. New Jersey: Prentrice Hall, 1999.

[110] Texas Instruments, “Four 1-Bit, 10MHz, 2nd-Order Delta-Sigma Modulators.” pp. 1–28, 2009.

[111] Analog Devices, “Sigma-Delta Conversion used for Motor Control,” 2015.

[112] Y. P. Siwakoti and G. E. Town, “Design of FPGA-controlled power electronics and drives using MATLAB Simulink,” in *ECCE Asia Downunder (ECCE Asia), 2013 IEEE*, 2013, pp. 571–577.

[113] L. Idkhajine and E. Monmasson, “Design methodology for complex FPGA-based controllers - Application to an EKF sensorless AC drive,” *19th Int. Conf. Electr. Mach. ICEM 2010*, 2010.

[114] I. The MathWorks, “HDL Verifier,” 2016. [Online]. Available: http://de.mathworks.com/products/hdl-verifier/. [Accessed: 19-Apr-2016].

[115] Mentor Graphics, “ModelSim.” [Online]. Available: https://www.mentor.com/products/fpga/verification-simulation/modelsim/. [Accessed: 19-Apr-2016].

[116] C. Altera, “Cyclone V Device Overview,” San Jose, CA, USA, 2015.

[117] C. Altera, “Enabling High-Performance DSP Applications with Arria V or Cyclone V Variable-Precision DSP Blocks,” San Jose, CA, USA, 2011.

[118] C. Altera, “Quartus Prime Design Software Overview.” [Online]. Available: https://www.altera.com/products/design-software/fpga-design/quartus-prime/overview.html. [Accessed: 20-Apr-2016].

[119] D. Bishop, “Fixed point package user’s guide,” 2008. [Online]. Available: www.vhdl.org/fphdl/vhdl.html. [Accessed: 10-Sep-2013].

[120] C. Altera, “Nios II Processor: The World’s Most Versatile Embedded Processor.” [Online]. Available: https://www.altera.com/products/processors/overview.html. [Accessed: 20-Apr-2016].

[121] C. Altera, “Qsys - Altera’s System Integration Tool.” [Online]. Available: https://www.altera.com/products/design-software/fpga-design/quartus-prime/features/qts-qsys.html. [Accessed: 20-Apr-2016].

[122] C. Altera, “Nios II Custom Instruction User Guide,” San Jose, CA, USA, 2015.

[123] J. C. R. Martinez and R. Kennely, “Boosting the performance of field-oriented control by using a model predictive direct current strategy,” in *Proceedings of EPE-PEMC 2010 - 14th International Power Electronics and Motion Control Conference*, 2010.

[124] C. Kessler, “Das Symmetrische Optimum,” *Regelungstechnik*, vol. 6, no. 11,12, pp. 395-400-436, 1958.

[125] H. Sartorious, “Die zweckmässige festlegung der frei wählbaren regelungkonstanten,” Technische Hochscule, Stuttgart, Germany, 1945.

[126] R. C. Oldenbourd and H. Sartorious, “A Uniform Approach to the Optimum Adjustment of Control Loop,” *Trans. ASME*, vol. 8, pp. 1265–1279, 1954.

[127] K. G. Papadopoulos, *PID controller tuning using the magnitude optimum criterion*. 2015.

[128] U. Nuß, *Hochdynamische Regelung elektrischer Antriebe*. VDE Verlag, 2010.

[129] Heidenhein GmbH, “Rotary encoders with integral bearing,” *Rotary encoders from HEIDENHAIN*, 2015. [Online]. Available: http://www.heidenhain.de/fileadmin/pdb/media/img/349529-2F_Rotary_Encoders_en.pdf. [Accessed: 12-Jul-2016].

[130] F. De Dinechin and M. Istoan, “Hardware Implementations of Fixed-Point Atan2,” in *Proceedings - Symposium on Computer Arithmetic*, 2015, vol. 2015–Augus, pp. 34–41.

[131] J. Hu and B. Wu, “Field oriented control CSI synchronous motor drive without shaft sensors,” in *Power Electronics and Drive Systems, 1997. Proceedings., 1997 International Conference on*, 1997.

[132] S. Li, Q. Ge, X. Wang, and Y. Li, “Implementation of sensorless control with improved flux integrator for wound field synchronous motor,” in *ICIEA 2007: 2007 Second IEEE Conference on Industrial Electronics and Applications*, 2007, pp. 1526–1530.

[133] L. Harnefors, “On Analysis, Control and Estimation of Variable-Speed Drives,” Royal Institute of Technology Stockholm, 1997.

[134] S. Feuersänger and M. Pacas, “Enhanced estimation of the rotor position of MV-synchronous machines in the low speed range,” in *Energy Conversion Congress and Exposition (ECCE), 2015 IEEE*, 2015.

[135] devboards GmbH, “DBC5CEFA7 Boards,” 2016. [Online]. Available: http://www.devboards.de/startseite/produkte/produkte-details/article/dbc5cefa7/. [Accessed: 10-Oct-2016].

[136] C. Altera, “Debugging Nios II Systems with the SignalTap II Embedded Logic Analyzer,” 2011.

[137] Infineon, “MIPAQ Serve,” *Technical Information*, 2016. [Online]. Available: http://www.infineon.com/dgdl/Infineon-IFS150V12PT4-DS-v02_01-en.pdf?fileId=db3a30432a40a650012a461b84052b57. [Accessed: 10-Oct-2016].

[138] F. D. Ramirez-Figueroa and M. Pacas, “FPGA Implementation of a Predictive Control for a PMSM with Variable Switching Frequency,” in *2015 Intl Aegean Conference on Electrical Machines & Power Electronics (ACEMP), 2015 Intl Conference on Optimization of Electrical & Electronic Equipment (OPTIM) & 2015 Intl Symposium on Advanced Electromechanical Motion Systems (ELECTROMOTION)*, 2015, pp. 317–322.

[139] F. D. Ramirez-Figueroa and M. Pacas, “Model based control of a PMSM with variable switching frequency and torque ripple control,” in *Industrial Electronics Society, IECON 2015 - 41st Annual Conference of the IEEE*, 2015, pp. 1418–1423.

[140] F. Ramirez and M. Pacas, “Finite Control Set Model Based Predictive Control of a PMSM with Variable Switching Frequency and Torque Ripple Optimization,” in *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management; Proceedings of*, 2016.

[141] F. Ramirez and M. Pacas, “Enhanced Control of the Torque Ripple in a PMSM Drive with Variable Switching Frequency,” in *2017 International Conference and Exposition on Electrical and Power Engineering (EPE)*, 2017.

[142] F. Ramirez, C. Gonzalez, and M. Pacas, “FPGA–based Sensorless Control of a PMSM at Low–Speed Ranges,” in *PCIM Europe 2017; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management; Proceedings of*, 2017, pp. 121–128.