الجمهورية الديمقراطية الشعبية الجزائرية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Ecole Supérieure des Sciences Appliquées d'Alger	**وزارة التعليم العالي والبحث العلمي المدرسة العليا في العلوم التطبيقية بالجزائر**

Département du second cycle

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme d'ingéniorat d'état

Filière : **Electrotechnique**

Spécialité : **Traction électrique**

Thème :

Variable speed drive control : study and implementation on an STM32 board

Présenté par : BAHIDJ Ryad

Encadré par : Dr.BENACHOUR Ali

Soutenu publiquement, le : 03/09/2020

Devant le Jury composé de :

Dr.ABERBOUR Adel	MCB, à l'ESSA-Alger	Président
Dr.BENACHOUR Ali	MCB, à l'ESSA-Alger	Encadreur
Dr.GUELLAL Amar	MCB, à l'ESSA-Alger	Examinateur

Binôme N° : 20/PFE. / 2020

الجمهورية الديمقراطية الشعبية الجزائرية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Ecole Supérieure des Sciences Appliquées
d'Alger

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية بالجزائر

Département du second cycle

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme d'ingéniorat d'état

Filière : **Electrotechnique**

Spécialité : **Traction électrique**

Thème :

Variable speed drive control : study and implementation on an STM32 board

Présenté par : BAHIDJ Ryad

Encadré par : Dr.BENACHOUR Ali

Soutenu publiquement, le : 03/09/2020

Devant le Jury composé de :

Dr.ABERBOUR Adel	MCB, à l'ESSA-Alger	Président
Dr.BENACHOUR Ali	MCB, à l'ESSA-Alger	Encadreur
Dr.GUELLAL Amar	MCB, à l'ESSA-Alger	Examinateur

Binôme N° : 20/PFE. / 2020

ملخص:

محركات السرعة المتغيرة (VSD)هي معدات تستخدم للتحكم في سرعة المحرك. هي محولات AC-AC تغير تردد و توتر تيار دخول المحرك. يدرس هذا العمل بعض طرق التحكم في هذه المعدات وينفذه على لوحة STM32. نبدأ هذا العمل باستعراض مستوى التقدم الجاري في بعض طوبولوجيات المحولات AC-AC و اشهر طرق التحكم في المحركات الحثية. تجرى محاكاة عن التحكم القياسي، DTC و DTC-SVM بـSimulink بعد انشاء نموذج محرك حثي. يتم بعدها تنفيذ V/f ثابت وDTC على لوحة STM32F4 Discovery و تختبر بمحاكاة Hardware-in-the-loop. يتم اختبار كل من المحاكاة والتنفيذ على أنواع مختلفة من الحمل وتناقش النتائج لكليهما.

الكلمات المفتاحية : المحرك لحثية، VSD، STM32، HIL.

Abstract:

Variable-speed drives for induction motors control the speed of the motor. They are AC-AC or DC-AC converters that vary the motor input frequency and voltage. This work studies some control methods of these VSDs and implements them on an STM32 board. We begin this study by a state-of-the-art review on some AC-AC converter topologies and the most popular induction motor control methods. A simulation of scalar control, DTC and DTC-SVM is performed with Simulink after establishing an induction motor model. Constant V/f and DTC are then implemented on an STM32F4 Discovery Board and tested with a Hardware-in-the-loop simulation. Both the simulation and implementation are tested on different types of load and the results for both are discussed.

Keywords: Induction motor, VSD, STM32, HIL.

Résumé:

Les variateurs de vitesse des moteurs asynchrones sont utilisés pour commander la vitesse du moteur. Ce sont de convertisseurs AC-AC ou DC-AC qui font varier la fréquence et la tension d’entrée du moteur. Ce travail étudie certaines commandes de ces variateurs de vitesse et les implémentent sur une carte STM32. On commence cette étude par un état de l’art sur certaines topologies des convertisseur AC-AC et les méthodes de commande des moteurs asynchrones. Une simulation de la commande scalaire, de la DTC et de la DTC-SVM est réalisé sur Simulink après avoir établie un modèle de machine asynchrone. Les commandes V/f constante et DTC sont ensuite implémentées sur une carte STM32F4 Discovery et testées avec une simulation Hardware-in-the-loop. La simulation et l’implémentations sont testées avec différents types de charges et les résultats sont discutés.

Mots clés: Moteur asynchrone, Variateur de vitesse, STM32, HIL.

Acknowledgments

I am thankful to Allah for all he blessings, for the will and strength to complete this work.

I would like to express my sincere gratitude to Dr Benachour Ali my thesis supervisor for proposing this subject, for the invaluable knowledge he shared, his advice, his help, for his efforts as my teacher for three years.

I thank Mr Sekhri Sabri for his help and his explanations.

I would also like to thank Mr Boutouche Hicham and Mr Tebib Abdennour for their help and guidance, as well as all the employees of Gatec.

I thank the members of the jury for having accepted to grade this work.

I dedicate this thesis to all the wonderful people in my life. To those who cannot be thanked with just a paragraph I dedicate my hard work.

To my mother who taught me the most important lessons, to my father, to my sisters Yasmine and Naziha.

To all my family, and to all my friends.

To my dearest friend who helped me at all times.

u5TBQA3

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

AC	Alternative Current
ADC	Analog-Digital Converter
ANN	Artificial Neural Network
BBC	Buck-Boost Converter
CMC	Conventional Matrix Converter
CPU	Central Processing Unit
CSI	Current Source Inverter
DAC	Digital-Analog Converter
DC	Direct Current
DSC	Direct Self-Control
DSP	Digital Signal Processor
DTC	Direct Torque Control
DVC	Direct Vector-Controlled
FLC	Fuzzy Logic Controller
FOC	Field-Oriented Control
GA	Genetic Algorithm
GCC	GNU Compiler Collection
GTO	Gate Turn-off Thyristor
HIL	Hardware-In-the-Loop
IDE	Integrated Development Environment
IGBT	Insulated-Gate Bipolar Transistor
IM	Induction Motor
IMC	Indirect Matrix Converter
IoT	Internet of Things
MCU	Micro-Controller Unit
MPC	Model Predictive Control
MRAS	Model Reference Adaptive System
PI/PID	Proportional-Integral /Proportional-Integral-Derivative (Regulator)
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation
SMC	Sliding Mode Control
SVM	Space Vector Modulation
THD	Total Harmonic Distortion
V/f	Voltage over frequency
VFD	Variable Frequency Drive
VSD	Variable Speed Drive
VSI	Voltage Source Inverter

List of Symbols

d,q	Direct, Quadrature
f	Frequency
f_v	Viscous friction coefficient
I	Current
i_d, i_q	Direct current, quadrature current
J	Moment on inertia
K_i	Integral gain
K_p	Proportional gain
L	Leakage inductance
L_s, L_r	Stator leakage inductance, Rotor leakage inductance
M	Mutual inductance
p	Pole pairs
R_s, R_r	Stator resistance , Rotor resistance
T_e	Electromagnetic torque
T_l	Load torque
V	Voltage
V_s	Stator voltage
ζ	Damping ratio
σ	Blondel leakage coefficient
Φ_s, Φ_r	Stator flux, Rotor Flux
Ω	Mechanical speed
ω_n	Natural frequency
ω_r	Rotor speed
ω_s	Stator pulsation
ω_{sl}	Slip speed

GENERAL INTRODUCTION

The induction motor is the most used electric motor in the world. Three-phase squirrel-cage induction motors are widely used as industrial drives because they are self-starting, reliable and economical. Although they have traditionally been used in fixed-speed operation they are increasingly being used in variable-speed applications with variable-speed drives (VSD), these offer important energy savings. About 40% of the world's electricity is consumed by industry, two-thirds of that is used by electric motors. VSDs can reduce their energy consumption by 50% in many applications[1].

The evolution of AC variable speed drive technology has been driven partly by the desire to emulate the excellent performance of the DC motor, such as fast torque response and speed accuracy, while using rugged, inexpensive and maintenance free induction motors [2]. In order to achieve these performances many control methods have been developed. Scalar control uses frequency and voltage to control the IM, this is a low-cost method that does not control torque and offers a low degree of accuracy. Field-oriented control uses "simulated" motor characteristics to emulate a DC motor, it controls torque indirectly and achieves very good performances but requires rotor feedback. Direct torque control (DTC) calculates motor torque directly and doesn't require a modulator nor speed feedback. It achieves a faster torque response and a high dynamic speed accuracy [2]. Another Modification introduces Space Vector Modulation to DTC to increase its performances.

On the hardware side, VSDs for IMs are AC-AC or DC-AC converters with a fixed frequency at the input (generally 50Hz or 60Hz) or DC voltage and a variable frequency at the output. The software (or control) side sets the frequency either directly (constant V/f) or indirectly (DTC, ...).

The objective of this work is the study and implementation of some of the control methods of a VSD on an STM32 board. It is composed of four chapters.

The first chapter of this work gives a state-of-the-art review on some AC-AC converter topologies where different topologies are presented, these can be either direct, indirect through a DC bus or hybrid. The second part of this chapter gives a state-of-art review on the most popular control methods, from scalar, FOC and DTC to more advanced control methods like AI based control.

In the second chapter three control methods, scalar control, DTC and DTC-SVM, where simulated with different types of load. These tests show how the control methods would perform in different applications and in different conditions.

The third chapter introduces the STM32f4 and its different peripherals and how they operate. These are the interrupts, timers and ADCs. The IDE used in this work is the STM32CubeIDE. A brief description of it is given in this chapter.

Finally, in the fourth chapter, scalar control and DTC are implemented on the STM32 MCU. A Hardware-in-the-loop simulation is used to test the controller. The controller runs on the STM32 MCU while the inverter and motor are simulated on Simulink. This allows an easy way to monitor the motor parameters and make quick adjustments and tests without the need for expensive test benches. A general conclusion is then given.

CHAPTER I

STATE OF THE ART ON CONVERTER TOPOLOGIES AND CONTROL

I STATE OF THE ART ON VSD CONVERTER TOPOLOGIES AND IT'S CONTROL STRATEGIES

I.1 Introduction

With the lowering cost of semiconductor power electronics components, the democratization of DSP and microprocessors and the advancement of control algorithms, induction machines have known a great increase in usage in industry, manufacturing and home-appliances.

In this chapter we will give a state-of-the-art review on the main AC-AC converter topologies and a state-of-the-art review on control strategies including scalar control, FOC, DTC, DTC-SVM, intelligent control and sensorless control. We will then give a brief review on VSD economics and markets.

I.2 AC-AC Converters

Induction machines have to be powered through an AC-AC converter to which the control is applied. Many topologies for these converters exist, they can broadly be classified into three categories: indirect (converters with DC-link), direct (matrix converters) and hybrid AC-AC converters, the difference being the presence of a main DC energy storage element.

Figure I.1 represents a diagram of the principal AC to AC converter topologies.

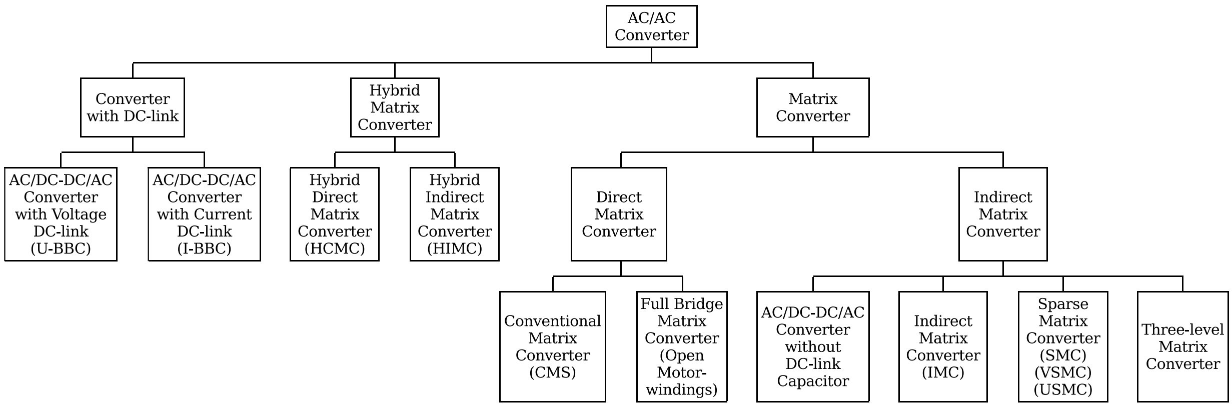

AC/AC Converter
Converter with DC-link
Hybrid Matrix Converter
Matrix Converter
AC/DC-DC/AC Converter with Voltage DC-link (U-BBC)
AC/DC-DC/AC Converter with Current DC-link (I-BBC)
Hybrid Direct Matrix Converter (HCMC)
Hybrid Indirect Matrix Converter (HIMC)
Direct Matrix Converter
Indirect Matrix Converter
Conventional Matrix Converter (CMS)
Full Bridge Matrix Converter (Open Motor-windings)
AC/DC-DC/AC Converter without DC-link Capacitor
Indirect Matrix Converter (IMC)
Sparse Matrix Converter (SMC) (VSMC) (USMC)
Three-level Matrix Converter

Figure I.1 : Classification of AC-AC converters

I.2.1 Direct converters

Matrix converters where introduced in 1976 [3] they use a matrix of bidirectional power switches to produce variable frequency output voltages. They feature a compact volume, bidirectional power flow, controllable input power factor and sinusoidal waveforms [4].

Direct Matrix Converters achieve three-phase AC-AC conversion without any intermediate energy storage element. They perform voltage and current conversion in one single stage.

A standard direct voltage source matrix converter topology is shown in Figure I.2.

Many Different structures of Indirect Matrix converters exist : sparse IMC, bidirectional very sparse IMC, unidirectional very sparse IMC, ultra-sparse IMC and inverter stage IMC [3], [5].

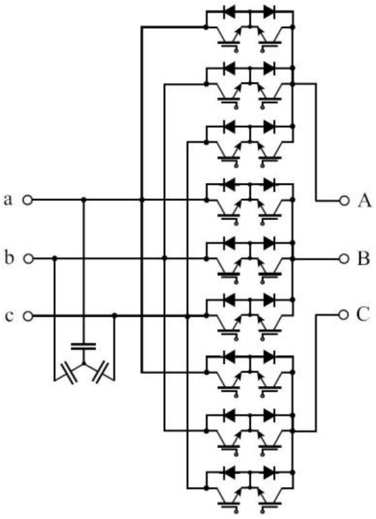


Figure I.2 : Direct voltage source matrix converter

I.2.2 Indirect converters

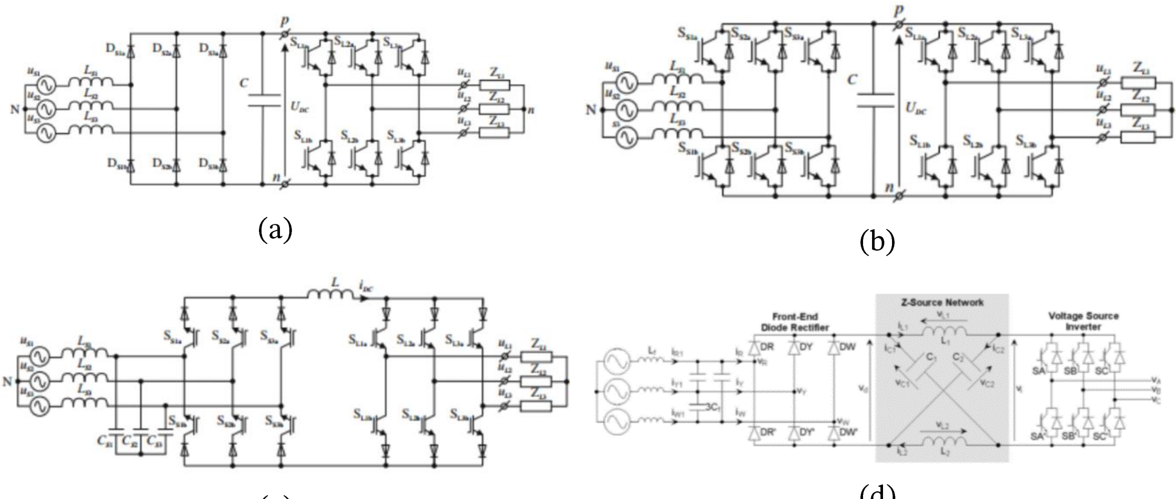
These converters can be either voltage source inverters (VSI) or current source inverters (CSI), the source current or voltage is rectified, filtered in a DC bus and then inverted using semiconductor switches (IGBTs or GTOs usually).

Figure I.3.a is the standard topology of AC-AC converter, it is VSI with a full-bridge diode rectifier and a DC Link capacitor [6].

Figure I.3.b is a back-to-back VSI [7] where the bidirectional flow of energy is possible.

Figure I.3.c is a back-to-back current source inverter (C-BBC) [8], [9].

Figure I.3.d is a topology proposed in [10] by putting a Z-source DC link which allows the system voltage to be stepped down or up as desired.



(a)

(b)

(c)

(d)

Figure I.3 : Main topology of Indirect converters : (a) VSI with a diode rectifier bridge, (b) Voltage-BBC, (c) Current-BBC, (d) AC-DC-AC Converter with Z-source

Table I.1 compares the properties of the four main converter topologies: voltage back to back converter, current BBC, indirect matrix converter and conventional matrix converter

TABLE I.1 : KEY PROPERTIES OF THE BASIC CONVERTER TOPOLOGIES[5]

Properties	V-BBC	C-BBC	IMC	CMC
Number of transistors	12	12	18	18
Number of diodes	12	12	18	18
Number of isolated gate driver supplies	7	12	8 or 10	9
Number of PWM signals	12	12	12	18
Number of devices in current path	4	8	6	4
Minimum number of current sensors	4	3	2	2
Minimum number of voltage sensors	4	6	3	3
Intermediate storage	C_{DC}	L_{DC}	No	No
Maximum output voltage	$>\hat{U}_1$	$> \hat{U}_1$	$0.86\hat{U}_1$	$0.86\hat{U}_1$
Additional protection circuitry	No	Yes	Yes	Yes

I.3 State of the art on the control of induction motors fed by AC-AC converters

I.3.1 Introduction

The induction motor (IM), thanks to its well-known advantages of simple construction, reliability, ruggedness and low cost, has found very wide industrial applications. Furthermore, in contrast to the commutation DC motor, it can be used in an aggressive or volatile environment since there are no problems with spark and corrosion. These advantages, however, are superseded by control problems when using an IM in industrial drives with high performance demands. For several years, academic and industrial research has been carried out to remedy the control problem of the IM and to develop robust and efficient controls.[11]

Scalar controls are simple to implement and offers good steady state response. However, the dynamics are slow because the transients are not controlled. To obtain high precision and good dynamics, vector control schemes have been invented for use with closed-loop feedback controls. At the beginning of 1970s, the principle of flux control was introduced and called ‘field oriented control’ or ‘vector control’ for squirrel cage induction machines and later for synchronous machines. [12]

Other vector control methods and intelligent control techniques were developped since, each having it’s own advantages, drawbacks and use cases. They are classified in a diagram in Figure I.4.

In this section, we will discuss a scalar control strategie (V/f constant), vector control techniques : FOC, DTC and DTC-SVM, intelligent control methods and finally briefly introduce sensorless control.

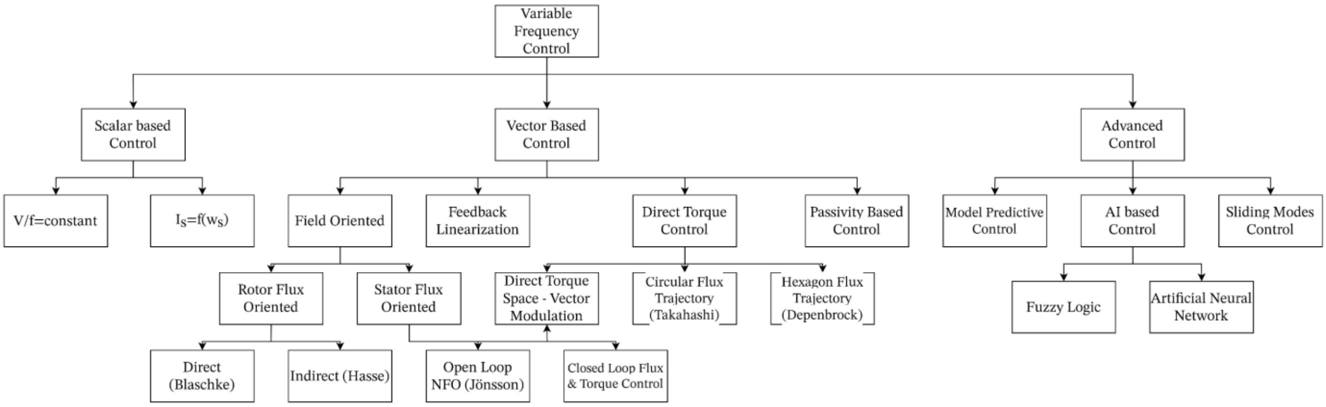


Figure I.4 : Classification of IM Control methods [11]

I.3.2 Scalar Control (V/f constant)

Scalar control is based on the steady-state model of the motor. The control is due to the magnitude variation of the control variables only and disregards the coupling effect in the machine. For example, the voltage of a motor can be controlled to control the flux, and frequency or slip can be controlled to control torque. However, flux and torque are also functions of frequency and voltage respectively. This method is simple and easy to implement, but the inherent coupling effect (i.e., both torque and flux are functions of voltage or current and frequency) gives sluggish response. As a result, the scalar control technique has poor dynamic performance. The scalar controller is usually used in low-cost and low-performance drives.[13]

The principle of this control is to maintain constant the ratio V/f, which means keeping the core stator flux constant. If the value of the resistance of the stator windings is neglected, like it is often the case, the electromagnetic torque-slip characteristic in the stationary regime takes the following form:

$$T_e = \frac{3p}{\omega_s}.V_s^2.\frac{\frac{R_r'}{s}}{\left(\frac{R_r'}{s}\right)^2 + (L\omega_s)^2} \tag{I.1}$$

With s: slip, $\frac{R_r'}{s}$: Equivalent resistance of rotor conductors reduced to the stator, p: number of pole pairs, ω_s: Stator pulsation, V_s: stator voltage, L: leakage inductance converted to the primary side.

Often, we are interested in the maximum value of the torque. To calculate it, we look for the value of s that maximizes the expression of the electromagnetic torque T_e and then replace it in Eq.(I.1) which yields:

$$T_e = \frac{3p}{2L}\left(\frac{V_s}{\omega_s}\right)^2 \tag{I.2}$$

Therefore, the maximum torque in steady-state is proportional to the square of $\frac{V_s}{2\pi f_s}$.

I.3.3 FOC

Scalar controls are simple to implement and offer good steady-state response, however the dynamics are slow because the transients are not controlled and the decoupling effect is not taken into consideration, that is why vector control schemes have been invented for use with closed-loop feedback controls in 1970s and 1960s by Blaschke and Hasse in two different approaches : the Direct Field oriented control (FOC)[14], and the Indirect field oriented control[15].

The concept behind FOC is to control the induction motor similar to a separately excited DC motor, the advantage being that in a DC motor flux and torque are inherently decoupled which allows the control of the DC motor in a simple and more efficient manner[16]. This also allows to control the torque and flux of an IM independently through the proper transformation[17]. The processing of vector control is however complex and computational-heavy, it requires the use of high-speed microcontrollers and DSPs[16].

Figure I.5 shows a general representation of vector control strategy for a three-phase induction motor.

The difference between Indirect or feedforward field-oriented control and the direct or feedback method of field-oriented control is the principle of generation of control angle theta (θ), in the first it is obtained by using rotor position measurement and machine parameter's estimation while in the latter it is obtained by the terminal voltages and currents.[16]

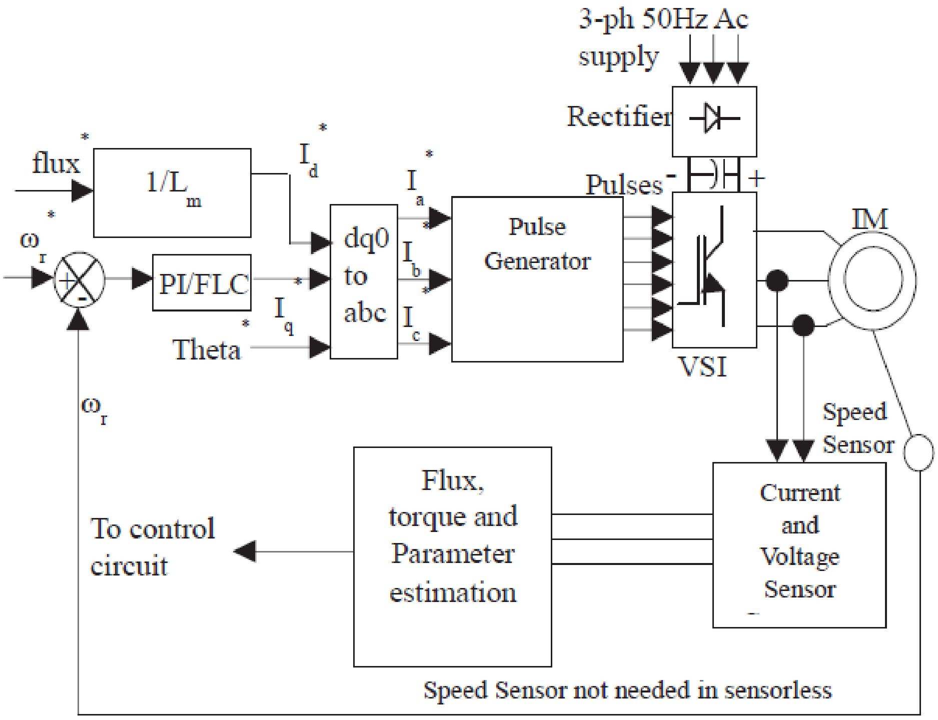


Figure I.5 : Block diagram of general vector control scheme

Indirect Field Oriented Control

It was presented in [15] by HASSE in 1969. In this scheme the unit vector θ_e is calculated in an indirect manner using the measured speed ω_r and the slip speed ω_{sl} as show in Eq.(I.3).

$$\theta_e = \int (\omega_r + \omega_{sl})dt \quad \text{(I.3)}$$

This computation was complex and resulted in high cost, but with the advent of microprocessors, the computation process has been simplified greatly and can now be implemented on high speed processors and DSPs with reduced cost and time [16].

The main research topics in this control scheme is the motor parameter estimation, the modulation strategies, controller performance and efficiency optimization.

In [18] a new identification technique was described where it is possible to calculate the resistance online by injecting the negative current component and identifying the negative voltage component, while [19] proposes an expression involving calculation of rotor parameter through stator current and stator voltage which has the advantage of not requiring the calculation of stator resistance and integration but with a higher cost.

A space vector modulation has been proposed in [20] requiring no time-varying coordinate transformation and no complicated calculation.

In addition, the proposed controller uses the extra information of error derivative to reduce the switching frequency greatly.

[21] proposes a fuzzy adaptive control scheme for vector controlled IM drive to estimate rotor parameters and [22] proposed a combined feedforward and feedback control technique to improve the robustness of vector controlled IM drive system.

A method involving optimization of flux vector based on power loss equation in terms of decupled current component has been proposed in [23].

Direct Field oriented Control

This scheme was first proposed by BLASCHKE in 1972 [14]. In direct FOC the rotor angle is obtained by the terminal voltages and currents directly by using flux estimators. It is also known as feedback vector control. Multiple controllers have been implemented to improve the performance of the drive.

In [24] a feedforward neural network technique for estimation of feedback signals of a direct vector-controlled (DVC) IM drive is proposed, it has the advantage of quicker processing, greater tolerance to fault and lesser harmonic ripple and in [25] a neuro fuzzy approach for a stator flux oriented vector controlled drive is proposed.

I.3.4 DTC

Direct torque control was first patented in Germany on October 20th 1984 by Manfred DEPENBROCK [26][27] and termed the technique direct self-control, however a paper describing a similar technique was described earlier in September of the same year by Isao TAKAHASHI and Toshihiko NOGUCHI in an IEEJ paper termed Direct Torque Control [28], and later published in 1986 in English in an IEEE paper[29]. The difference between the two techniques is the path the flux vector is controlled on, in TAKAHASHI's approach it is circular, while in DEPENBROCK's it follows a hexagon. DSC is best suited for high power applications, as a low inverter switching frequency can justify higher current distortion [9]. Unlike FOC that aims to reproduce the electromagnetic behavior of a DC motor, DTC tries to exploit the torque and flux producing capabilities of the IM fed by a voltage source inverter (VSI).

The basic DTC scheme is characterized by the absence of a coordinate transformation, PWM signal generators and current regulators, but instead uses a Switching Table to determines the switching state selection and requires the estimation of stator flux vector and torque estimation[11].

Figure I.6 shows a block diagram of the conventional "TAKAHASHI" DTC scheme.

This scheme uses two hysteresis controllers. The stator flux controller imposes the time duration of the active voltage vectors, which moves the stator flux along the reference trajectory, and the torque controller determinates the time duration of the zero voltage vectors, which keeps the motor torque in the tolerance band of the hysteresis controller. At every sampling time the voltage vector the inverter switching state (S_a, S_b, S_c) are chosen according to the switching table as to reduce the instantaneous flux and torque errors [11].

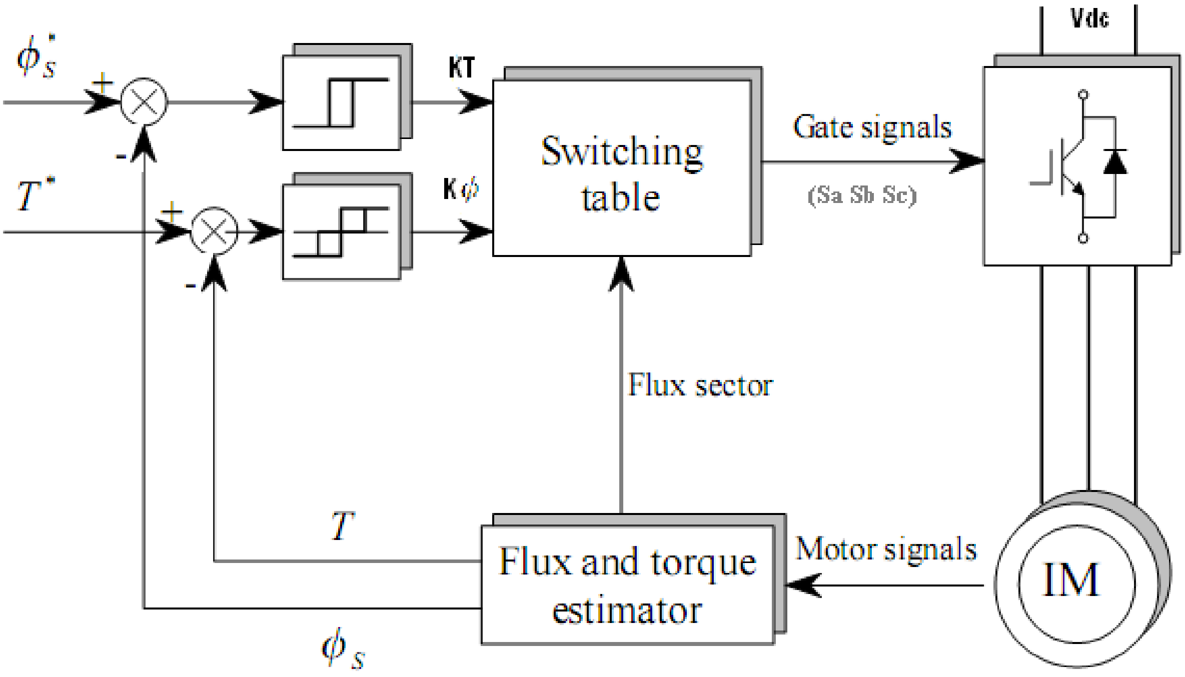


Figure I.6 : Block diagram of the generic DTC

The generic DTC introduced by TAKAHASHI and DEPENBROCK has may benefits, it is robust without the need of a speed/position sensor, the frequent calculation of optimum switching means that the drive can respond rapidly to external influences, no process interruptions after short supply outages, it can start into a motor which is running at any speed without waiting for the flux to decay [30]. It requires no current control loops, no extra block for voltage modulation and no coordinate transformation. This control method is superior to field-oriented control (FOC) in that it only relies on a simplified and qualitative machine behavior and doesn't require and encoder, but achieves good flux and torque dynamic performance [31].

DTC however has many drawbacks. First, the use of a switching table leads to variable switching frequency, which makes it unsuitable for high power application, second is the high noise level, third the complicated starting and low frequency operation and lastly the undesirable torque ripples [31].

A comparison between the primary characteristics of FOC and DTC are presented in Table I.2.

To alleviate these problems many improvements to DTC have been introduced. Figure I.7 classifies these techniques in a diagram [32].

Table I.2 compares between FOC and DTC [33][34].

TABLE I.2 : COMPARISON BETWEEN FOC AND DTC

Comparison property	DTC	FOC
Dynamic response to torque	Very fast	Fast
Coordinates reference frame	alpha, beta (stator)	d, q (rotor)
Low speed (< 5% of nominal) behavior	Requires speed sensor for continuous braking	Good with position or speed sensor
Controlled variables	torque & stator flux	rotor flux, torque current iq and rotor flux current id vector components
Steady-state torque/current/flux ripple & distortion	Low (requires high quality current sensors)	Low
Parameter sensitivity, sensorless	Stator resistance	d, q inductances, rotor resistance
Parameter sensitivity, closed-loop	d, q inductances, flux (near zero speed only)	d, q inductances, rotor resistance
Rotor position measurement	Not required	Required (either sensor or estimation)
Current control	Not required	Required
PWM modulator	Not required	Required
Coordinate transformations	Not required	Required
Switching frequency	Varies widely around average frequency	Constant
Switching losses	Lower (requires high quality current sensors)	Low
Audible noise	spread spectrum sizzling noise	constant frequency whistling noise
Control tuning loops	speed (PID control)	speed (PID control), rotor flux control (PI), id and iq current controls (PI)
Complexity/processing requirements	Lower	Higher
Typical control cycle time	10-30 microseconds	100-500 microseconds

The first typical improvement to be proposed is the use of space vector modulation (SVM) in controlling the voltage inverter, which is referred to as DTC-SVM and was introduced in the early 1990s for constant frequency operation by HABELTER et al. in [35].

In [36] proposed a method of direct torque control of an IM based on fixed switching frequency pulse width modulation (PWM), developed in discrete time to allow easy implementation on a DSP or a microcontroller.

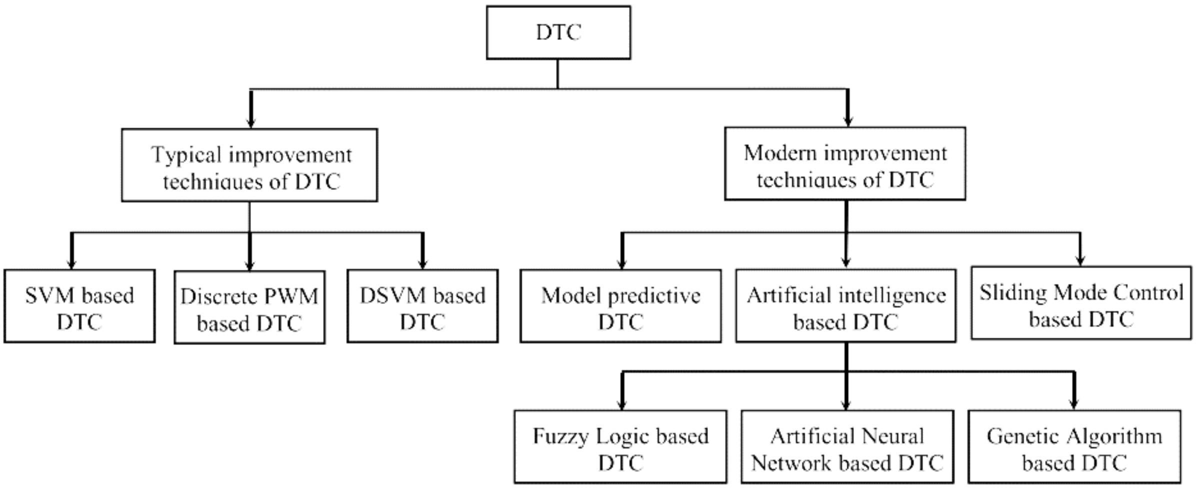

DTC
Typical improvement techniaues of DTC
Modern improvement techniaues of DTC
SVM based DTC
Discrete PWM based DTC
DSVM based DTC
Model predictive DTC
Artificial intelligence based DTC
Sliding Mode Control based DTC
Fuzzy Logic based DTC
Artificial Neural Network based DTC
Genetic Algorithm based DTC

Figure I.7 : Classification of improvement techniques of direct torque control

Other advanced improvements of DTC have been studied, including Sliding Mode Control introduced by Utkin in 1993 in [37], model predictive DTC which is based on the calculation of the future behavior of the system based on the dynamic model of the process inside the real-time controller to calculate the optimal values of the adjustment parameters [32] and other AI based DTC schemes based on artificial neural networks, fuzzy logic, genetic algorithms and expert systems each with its advantages and use cases.

TABLE I.3 : COMPARISON OF THE IMPROVEMENT TECHNIQUES OF DIRECT TORQUE CONTROL

Criterion	Conventional DTC	SVM based DTC	SMC based DTC	DTC-MPC	Fuzzy based DTC	ANN based DTC	GA based DTC
Torque dynamic response	Fast	Fast	Fast	Fast	Very fast	Very fast	Very fast
Torque and flux ripple	High	Low	Medium	Low	Very low	Very low	Medium
Current THD	More distortions	Less distortion	Less distortion	Less distortion	Less distortion	Less distortion	Less distortion
Switching frequency	Variable	Constant	Almost constant	Constant	Constant	Constant	Almost constant
Parameter sensitivity	Insensitive	Sensitive	Insensitive	Insensitive	Insensitive	Insensitive	Insensitive
Switching loss	High	Low	Medium	Low	Low	Low	Medium
Dynamic at low speed	Poor	Good	Good	Good	Very good	Very good	Very good
Algorithm complexity	Simple	Simple	Complex	Simple	More complex	More complex	Complex
Computation time	Low	Medium	High	Medium	High	High	Medium
Precession	Low	Medium	Medium	Medium	High	High	Medium
Regulation	Hysteresis	PI conventional	SMC controller	Hysteresis	FLC	ANN	GA-PI

Table I.3 Shows a comparison between the performances of different improvement techniques of direct torque control [32].

I.3.5 DTC-SVM

Many studies investigated the possibility to associate space-vector modulation techniques with DTC in order to control the switching frequency, SVM being the most popular technique for the control of inverter switches. The controller calculates the required stator voltage vector and then it is realized by SVM technique. The main difference between the generic DTC and DTC-SVM is that DTC is based on instantaneous values whereas DTC-SVM methods are based on averaged value.

Several DTC-SVM methods have been addressed in literature, the simplest and most used is DTC-SVM with PI regulators, but others include schemes with predictive/dead-beat controllers, schemes based on fuzzy logic and/or neural networks [38].

The SVM technique uses eight sorts of different switch modes of inverter to control the stator flux to approach the reference flux circle. Eight switch modes correspond respectively to eight space voltage vectors that contain six active voltage vectors and two zero voltage vectors. The six active voltage vectors form the axes of a hexagon. The two zero voltage vectors are at the origin. The eight vectors are called the basic space vectors. These switch modes are represented in Figure I.8.

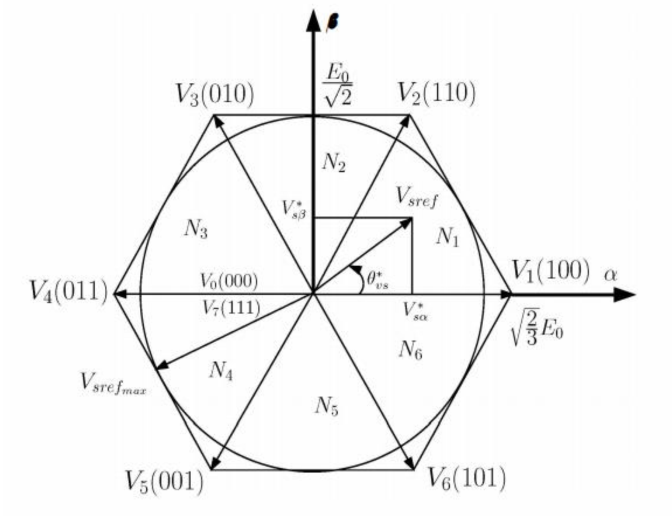


Figure I.8 : Basic switching vectors and sectors

In a PI controlled DTC-SVM IM, many structures exist : DTC-SVM scheme with closed-loop flux control where the rotor flux is assumed as a reference [39], DTC-SVM scheme with closed-loop torque control which was originally proposed for permanent magnet synchronous motor (PMSM), DTC-SVM scheme with closed-loop torque and flux control operating in polar coordinates and DTC-SVM scheme with closed-loop torque and flux control in stator flux coordinates.

TABLE I.4 : ADVANTAGES AND DISADVANTAGES OF CLASSIC FOC, BASIC DTC AND DTC-SVM CONTROL METHODS

Control Methods	Classic FOC	Classic DTC	DTC-SVM
Advantages	-Constant switching frequency -Unipolar inverter output voltage -Low switching losses -Low sampling frequency -Current control loops	-Structure independent on rotor parameters, universal for IM and PMSM -Simple implementation of sensorless operation -No coordinate transformation -No current control loops	-Structure independent on rotor parameters, universal for IM and PMSM -Simple implementation of sensorless operation -No coordinate transformation -No current control loops -Constant switching frequency -Unipolar inverter output voltage -Low switching losses -Low sampling frequency
Disadvantages	-Coordinate transformation -Multiple control loops -Control structure dependent on rotor parameters	-Bipolar inverter output voltage -Variable switching frequency -High switching losses -High sampling frequency	-High CPU time -More complex

Table I.4 gives a comparison between the three main control methods: FOC, DTC and DTC-SVM[38].

I.3.6 Intelligent Control

Intelligent Control include all control techniques other than scalar and vector controls like artificial intelligence-based techniques including artificial neural networks and fuzzy-logic based controllers, sliding modes and predictive control.

AI Based Control

Artificial intelligence (AI) and Biologically-inspired techniques, particularly the neural networks, are recently having significant impact on power electronics and electric drives. Neural networks have created a new and advancing frontier in power electronics, which is already a complex and multidisciplinary technology that is going through dynamic evolution in the recent years [40].

These include neural network-based controllers that feed the input signals to a trained neural network which outputs the control states for the inverter, fuzzy logic which is based on the principles introduced by ZADEH in his 1965 seminal paper [41], but they also include other biologically-inspired techniques like genetic algorithms and brain emotional learning which is inspired by the limbic system of mammalian brain and emotional learning based action selection [40].

Predictive Control

Predictive control predicts the changes in the dependent variable of the modeled system that will be caused by the change in the independent variable. The predictive control techniques are typically classified into dead beat control, hysteresis and trajectory-based control and model predictive control [31].

Dead-beat control makes the error signals go to zero in the next sampling instant. In hysteresis and trajectory-based control, an optimization criterion is used to keep the controlled variables within the band. Model predictive control method combines the use of PI control and the predictive model of the system for the improvement of both steady state and dynamic response of the system, it’s block diagram is shown in Figure I.9.

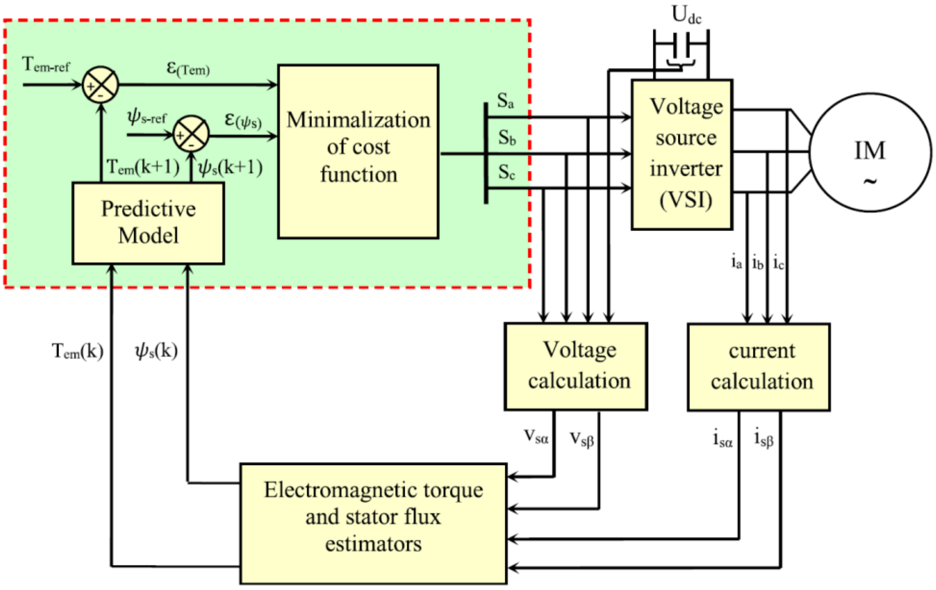


Figure I.9 : Model predictive Torque Control block diagram

The advantages of this technique are: the use of online optimization technique, switching tables are not used, switching losses are reduced and it features an improved steady state and dynamic response. However, it requires a tedious tuning work [31].

I.3.7 Sensorless Control

In recent years, the application of sensorless AC motor drives is expanding in areas ranging from industrial applications to household electrical appliances. The advantages of sensorless motor drives include lower cost, increased reliability, reduced hardware complexity, better noise immunity and less maintenance requirements.

Sensorless Control aims at removing the need for a mechanical position/speed sensor in a vector control system, this increases the reliability of drive systems and reduces cost and complexity [42].

Figure I.10 show the block diagram of a typical IM vector control system where the speed and flux are estimated through an observer.

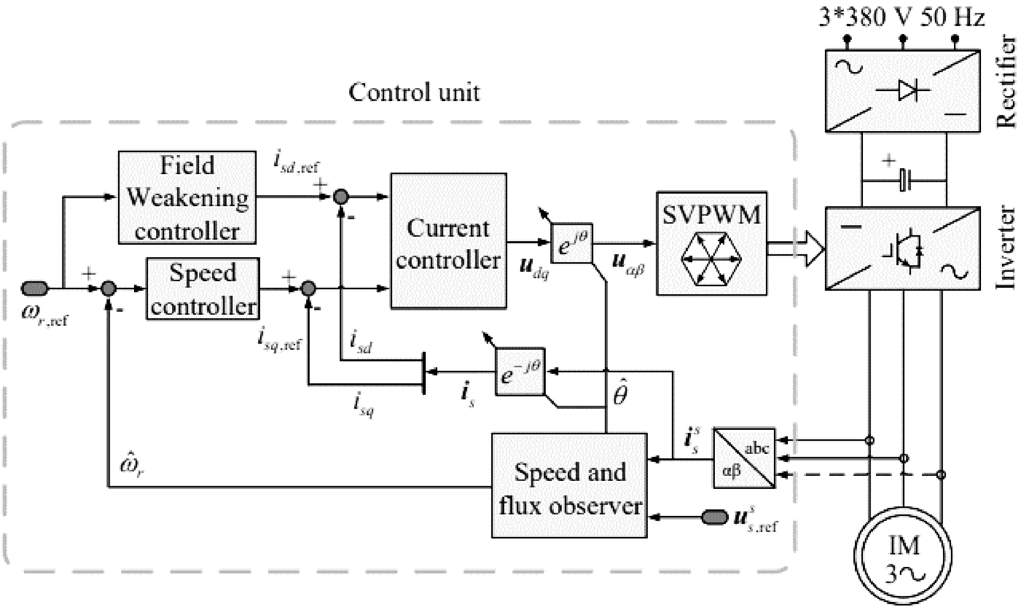


Figure I.10 : Block diagram of typical IM vector control System

A lot of researches have been done on flux and speed observer of IM sensorless drive system, mainly includes: low frequency signal injection method [43], high frequency signal injection method [44] and model reference adaptive system (MRAS) [45], full order flux observer [46], reduced order observer [47], sliding mode observer [48], neural network [49], and Kalman filter [50]. These methods can be classified into two categories:

- Signal injection-based flux and speed observer which works by injecting a signal and uses rotor slot harmonic and leakage inductance to extract the rotor position information.
- IM model-based flux and speed observer which works by first establishing the mathematical model of the IM and then estimating the rotor flux and rotor speed.

I.4 Variable Speed Drives' market, evolution and major industrial manufacturers

The variable frequency drives (VFDs) market was valued at 9.47 billion USD in 2019 and is expected to reach a value of 12.44 billion USD by 2025 according to [51].

The advance of IoT (Internet of Things) is a global trend in every industry, it has accelerated and simplified the flow of information from machine to a technician and back again having an impact on VFDs.

The rising demand for process optimization is driving the market. Communication between modern motor controllers, such as VFD and process automation controllers, it is now possible to achieve much higher efficiency along with process improvement with reducing peak demand charges on an electricity bill by understanding in greater detail where and why power consumption varies.

Rising demand for energy efficiency is driving the market. VFDs use an AC motor to control pumps and fans in a system without using valves and dampers. Traditionally, fixed-speed motors had to be controlled using valves and dampers, but they are a source of large amounts of energy loss. VFDs can create the same energy flow without the high pressure and power needed to adjust the flow, which saves a high amount of energy, raising the appeal of upgrading older equipment.

Additionally, advanced and modern drives integrate networking and diagnostic capabilities leading to better performance and increased productivity. Intelligent motor control, energy savings, and reduction of peak-current draws are some of the major reasons to choose a VFD as the controller in every motor-driven system.

Asia Pacific is expected to be the largest market for VFDs in 2023 and is projected to grow at the highest CAGR. China accounted for the largest share of the VFD market in the Asia Pacific region in 2017. Moreover, in January 2017, the government of China planned to invest about 361 billion USD in clean energy by 2020 as China aims to supply 20% of power from non-fossil sources by 2030, which shows a positive growth for the VFD market. All these factors are expected to drive the growth of the VFD market in Asia Pacific [51].

The European market is also expected to rise as the European ecodesign directives are maturing and industries are gearing towards more energy efficient processes and drives [52].

The primary end-user sectors for VFD are: Oil & Gas, industrial, infrastructure and power generation sectors.

VFDs are mainly used for pumps, fans, compressors, conveyors and others including extruders, elevators & escalators, cranes & hoists, crushers, mills, and mixers.

The primary manufacturers of VFDs today are ABB (Switzerland), Siemens (Germany), Schneider Electric (France), Danfoss (Denmark), Rockwell (US), Crompton Greaves (India), Eaton (Ireland), Fuji, Hitachi, NIDEC, TMEIC (Japan), WEG SA (Brazil), and Yaskawa (Japan)[53].

I.5 Conclusion

In this chapter, a state-of-the-art review was presented on the main topologies of AC-AC converters, the main control methods for induction machines and an overview of the market for variable speed drives.

The main control strategies were presented from scalar control to vector control methods to advanced control schemes. Scalar control is the simplest control method and remains in use in many applications, while vector controls are more performant but are more complex and more resource intensive. Advanced control methods are varied in approach and offer many advantages. New advancements are made in this area and research is very active, as the market for variable speed drive is expending with the increase in production levels, the modernization of production facilities and the advance of IoT as a global trend.

It is expected for the variable speed drive technology to progress as research, manufacturing and market demand converge.

CHAPTER II

INDUCTION MOTOR CONTROL

II INDUCTION MOTOR CONTROL

II.1 Introduction

In this chapter an induction motor model will be proposed and used for the simulation of three control methods: V/f constant control, DTC and SVM-DTC.

V/f constant is simple yet effective and is still in use in many applications. DTC offers a lot of advantages especially for its implementation, SVM-DTC is an improvement to the DTC through the use of an SVM modulator and PI controllers but comes with more complexity.

These control schemes will be presented and then simulated on the MATLAB/SIMULINK environment. A comparison of the performances of these three methods will be made.

II.2 Induction machine model

The three-phase induction motor model is obtained from the application of the electromagnetic and mechanical principles. The three-phase stator voltage equations, represented in a stationary (a, b, c) frame, can be expressed as:

$$V_{sabc} = R_s[I_{sabc}] + \frac{d[\Phi_{sabc}]}{dt} \tag{II.1}$$

And

$$V_{rabc} = R_r[I_{rabc}] + \frac{d[\Phi_{rabc}]}{dt} \tag{II.2}$$

Where V_{sabc} and V_{rabc} are the voltages of the stator and rotor phases respectively, R_s and R_r are the stator and rotor resistances, I_{sabc} and I_{rabc} are the currents of the stator and rotor phases and Φ_{sabc} and Φ_{rabc} are the stator and rotor flux respectively.

The magnetic fluxes are given by:

$$\Phi_{sabc} = [L_{os}][I_{sabc}] + [M_{osr}][I_{rabc}] \tag{II.3}$$

and

$$\Phi_{rabc} = [L_{or}][I_{rabc}] + [M_{osr}][I_{sabc}] \tag{II.4}$$

L_{os}, L_{or}, M_{osr} are the stator inductance, rotor inductance and mutual inductance respectively.

The park transformation matrix is given by:

$$P(\theta_e) = \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \tag{II.5}$$

By applying the Park transform to the equations above with θ_s the (d,q) frame angular position and θ_r the rotor relative angular position we get:

$$V_{sdq} = R_s[I_{sdq}] + \frac{d[\Phi_{sdq}]}{dt} - \frac{d\theta_s}{dt} I[\Phi_{sdq}] \tag{II.6}$$

And

$$V_{rdq} = 0 = R_r[I_{rdq}] + \frac{d[\Phi_{rdq}]}{dt} - \frac{d\theta_r}{dt} I[\Phi_{rdq}] \tag{II.7}$$

And the magnetic flux equations yield:

$$\Phi_{sdq} = L_s[I_{sdq}] + M_{sr}[I_{rdq}] \tag{II.8}$$

and

$$\Phi_{rdq} = L_r[I_{rdq}] + M_{sr}[I_{sdq}] \tag{II.9}$$

Where L_s and L_r are the stator and rotor cyclic inductances and M_{sr} the mutual cyclic inductance.

The mechanical equation of the IM is given by:

$$\frac{d\Omega}{dt} = \frac{T_e}{J} - \frac{T_l}{J} - \frac{f_v}{J}\Omega = \frac{1}{J}\{\frac{pM_{sr}}{L_r}(\phi_{rd}i_{sd} - \phi_{rq}i_{sq})\} - \frac{T_l}{J} - \frac{f_v}{J}\Omega \tag{II.10}$$

From these equations we get the induction motor model in the rotating (d,q) frame given by :

$$\begin{aligned} \frac{d[I_{sdq}]}{dt} &= \left\{-\frac{R_s}{L_s\sigma}I_{2\times2} - \frac{R_rM_{sr}^2}{L_r^2L_s\sigma}I_{2\times2} + \omega_s I\right\}[I_{sdq}] \\ &\quad + \left\{\frac{R_rM_{sr}}{L_r^2L_s\sigma}I_{2\times2} + \frac{M_{sr}}{L_rL_s\sigma}p\Omega I\right\}[\phi_{rdq}] + \frac{1}{L_s\sigma}[V_{sdq}] \\ \frac{d[I_{sdq}]}{dt} &= \left\{\frac{R_rM_{sr}}{L_r}\right\}[I_{sdq}] + \left\{-\frac{R_r}{L_r}I_{2\times2} + (\omega_s - p\Omega)I\right\}[\phi_{rdq}] \\ \frac{d\Omega}{dt} &= \frac{1}{J}\{\frac{pM_{sr}}{L_r}(\phi_{rd}i_{sd} - \phi_{rq}i_{sq})\} - \frac{T_l}{J} - \frac{f_v}{J}\Omega \end{aligned} \tag{II.11}$$

With: I the [1 -1] vector for considering the different signs in the d and q equations, I_{2x2} the identity matrix of dimension 2, w_s the stator pulsation given by:

$$\omega_s = p\Omega - \frac{aM_{sr}i_{sq}}{\phi_{rd}} \tag{II.12}$$

And σ the Blondel leakage coefficient given by:

$$\sigma = 1 - (\frac{M_{sqr}^2}{L_s L_r}) \qquad \text{(II.13)}$$

and

$$\frac{d\theta_s}{dt} = \omega_s \ , \quad \frac{d\theta_m}{dt} = \Omega \ , \quad \frac{d\theta_r}{dt} = \omega_r = \omega_s - p \qquad \text{(II.14)}$$

State-space model of the IM:

By defining the simplifying parameters: a $= \frac{\mathrm{R_r}}{L_r}$, b $= \frac{\mathrm{M_{sr}}}{\mathrm{\sigma L_s L_r}}$, c $= \frac{\mathrm{f_v}}{J}$, $\gamma = \frac{\mathrm{L_r^2 R_s + M_{sr}^2 R_r}}{\mathrm{\sigma L_s L_r^2}}$,

m $= \frac{\mathrm{pM_{sr}}}{JL_r}$, m1 $= \frac{1}{\mathrm{\sigma L_s}}$.

We define the space vector $\mathrm{x_{dq}}$, the input vector u and the output vector y as:

$$x_{dq} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \phi_{rd} \\ \phi_{rq} \\ \Omega \end{bmatrix} \ , \ u = \begin{bmatrix} v_{sd} \\ v_{sq} \\ T_l \end{bmatrix}, \ \ y = h(x_{dq}) = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \Omega \end{bmatrix}$$

The state-space dynamics of the IM in the (d, q) frame is given by:

$$\begin{bmatrix} \frac{di_{sd}}{dt} \\ \frac{di_{sq}}{dt} \\ \frac{d\phi_{rd}}{dt} \\ \frac{d\phi_{rq}}{dt} \\ \frac{\mathrm{d\Omega}}{\mathrm{dt}} \end{bmatrix} = \begin{bmatrix} -\gamma i_{sd} + \omega_s i_{sq} + ba\phi_{rd} + bp\Omega\phi_{rq} \\ -\omega_s i_{sd} + \gamma i_{sq} - bp\Omega\phi_{rd} + ba\phi_{rq} \\ aM_{sr} i_{sd} - a\phi_{rd} + (\omega_s - p\Omega)\phi_{rq} \\ aM_{sr} i_{sq} - a\phi_{rq} - (\omega_s - p\Omega)\phi_{rd} \\ \mathrm{m}(\phi_{rd} i_{sq} - \phi_{rq} i_{sd}) - \mathrm{c}\Omega \end{bmatrix} + \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} v_{sd} \\ v_{sq} \\ T_l \end{bmatrix} \qquad \text{(II.15)}$$

$$y = h(x_{dq}) = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \Omega \end{bmatrix}$$

II.3 Constant V/f control

II.3.1 Working principle

Both voltage and frequency can be controlled to control the motor flux. With a fixed number of poles, the speed of an induction motor is changed with the frequency, according to the equation:

$$N = \frac{120f}{n_p}(1-s) \tag{II.16}$$

With N being the motor speed, f the supply frequency, n_p the number of pole pairs and s the slip.

The air gap induced electromotive force (EMF) in an IM is given by:

$$E_g = 4.44 f N_s K_{N_s} \Phi_m \tag{II.17}$$

Where Ns is the number of turns per phase in the stator, K_{Ns} is the winding cording factor and Φm is the peak air gap flux. Neglecting the stator impedance (resistance and leakage inductance) the induced EMF approximately equals the supply phase voltage[13], hence:

$$V \approx E_g = 4.44 f N_s K_{N_s} \Phi_m \tag{II.18}$$

Therefore, if the frequency f is constant, the air gap flux is approximately proportional to the supply voltage, meaning that to maintain the flux constant, the ratio between the supply voltage and frequency has to remain constant[13].

When frequency is increased, the flux will decrease and the torque developed by the motor will decrease. When frequency is decreased, the flux will increase and may lead to the saturation of magnetic circuit. When the speed is below the base speed, the motor runs at constant torque operation range. The phase voltage-amplitude command V_s^* is produced by the frequency control command through the gain factor G. For low-speed operation, the voltage drop along the stator winding is significant and, consequently, the produced magnetic flux is limited. To avoid this phenomenon and to maintain the value of the magnetic flux near the nominal for low-speed operation, a boost voltage Vo is added to the reference amplitude of the phase voltage. The impact of the boost voltage is negligible in higher frequencies[54].

Figure II.1 shows the different operation regions for motor operation: a boost voltage for maintaining operation at low speeds, constant flux region when the voltage saturation is not yet reached and the constant power region where the field weakening is performed and torque decreases with the increase in speed. It also shows the voltage curve (frequency=f(Vs)) for this control method with boost voltage added for the low-speed operation.

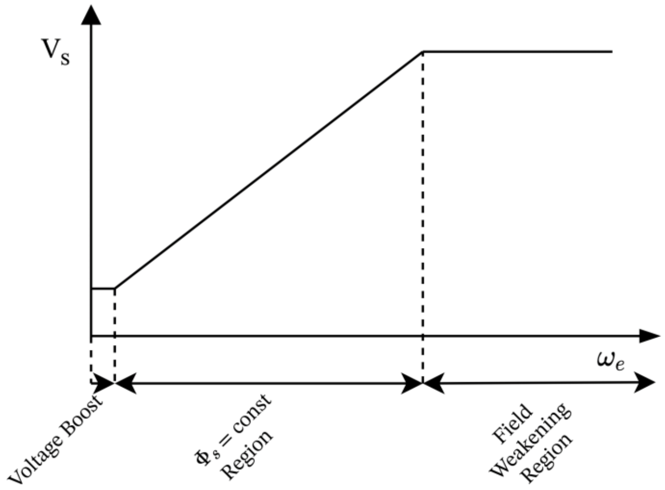


Figure II.1 : V/f constant control operation regions for motor operation

Figure II.2 shows a block diagram for V/f constant control scheme.

PWM modulator can be replaced with an SVM modular for better performance, in this case the reference speed signal w_1^* is integrated, producing the reference angle signal θ_e and the respective reference phase voltages V^*a, V^*b, and V^*c are produced as:

$$\begin{cases} V_a^* = \sqrt{2}V_s \sin\,(\theta_e) \\ V_b^* = \sqrt{2}V_s \sin\,(\theta_e - \frac{2\pi}{3}) \\ V_c^* = \sqrt{2}V_s \sin\left(\theta_e + \frac{2\pi}{3}\right) \end{cases} \qquad \text{(II.19)}$$

These are then fed to SVM modulator to generate the appropriate gate signals that are fed to the VSI. The voltage reference is passed through a defluxing block that limits its maximum value and adds a boost voltage for low speed operations which is proportional to the reference speed (Constant V/f). A PI regulator regulates the speed error to generate a speed offset that compensates the slip in order to get a speed matching exactly the value of the reference.

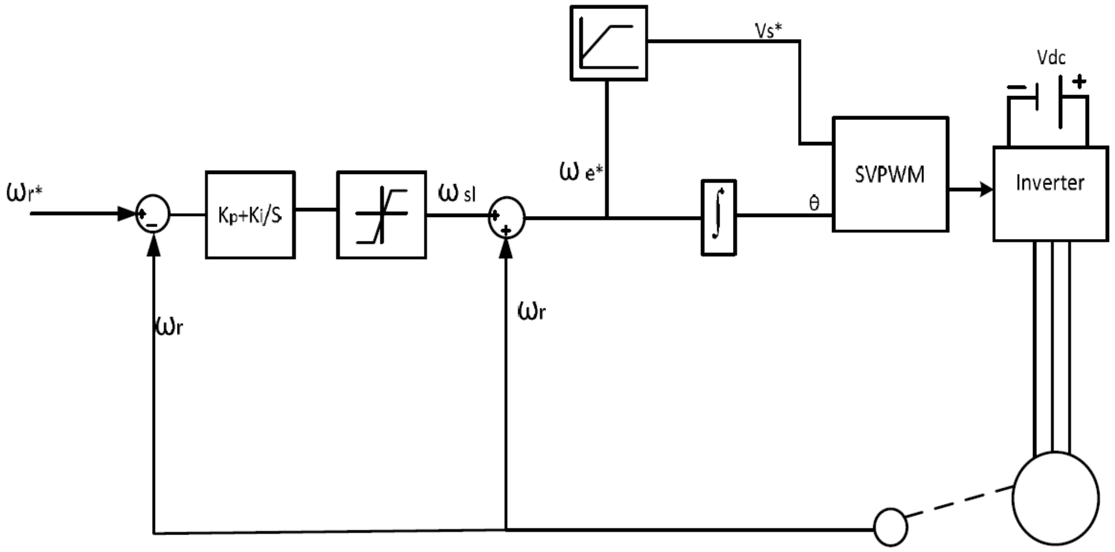


Figure II.2 : Block diagram of the V/f open loop control with a three-phase voltage fed inverter

II.3.2 Simulation

To test the Constant V/f control method, a simulation was performed with different load types. The simulation was performed using the MATLAB/Simulink environment, using the induction machine model described in II.2.

The motor parameters used for the simulation are given in Appendix A: Induction motor parameters.

Simulation results of V/f constant control method for an induction motor are given in the figures below. Three types of load where simulated : screw compressors conveyors and feeders with a constant load (Figure II.3), mixers with a load torque propotional the motor speed (Figure II.4) and fans and pumps with a load torque proportional to the square of the speed of the motor (Figure II.5).

An acceleration/deceleration rate of 150 rad/s^2 has been imposed to the speed. The speed reference is set to the nominal value at startup, a reverse nominal speed reference is imposed at t=3s and at t=6s a a stopping of the motor imposed.

The constant load changes values twice during the simulation at t=4s and t=8s going from nominal load to no-load and then from no-load to negative nominal load.

The switching frequency of the Space Vector Modulation is set to 5 kHz.

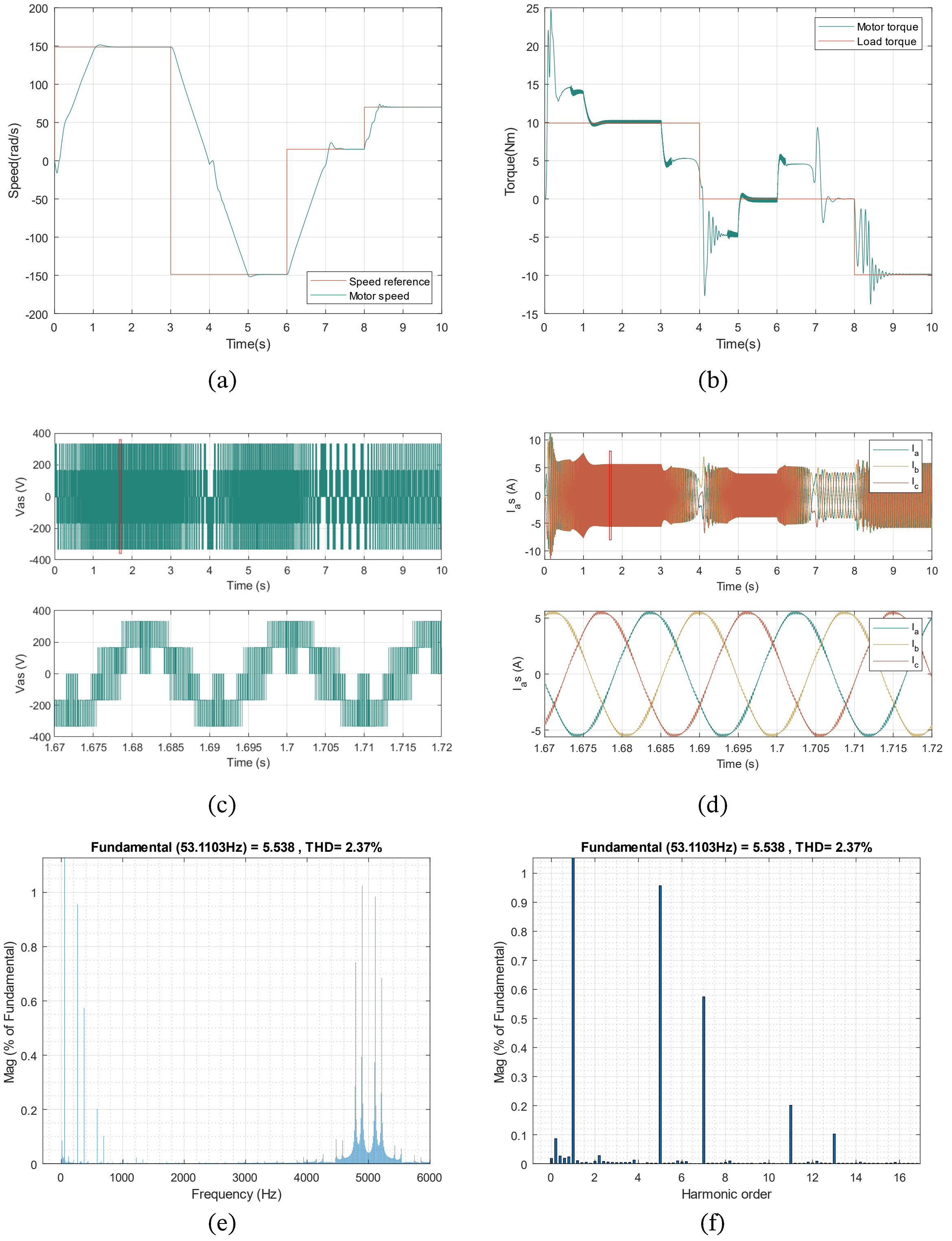
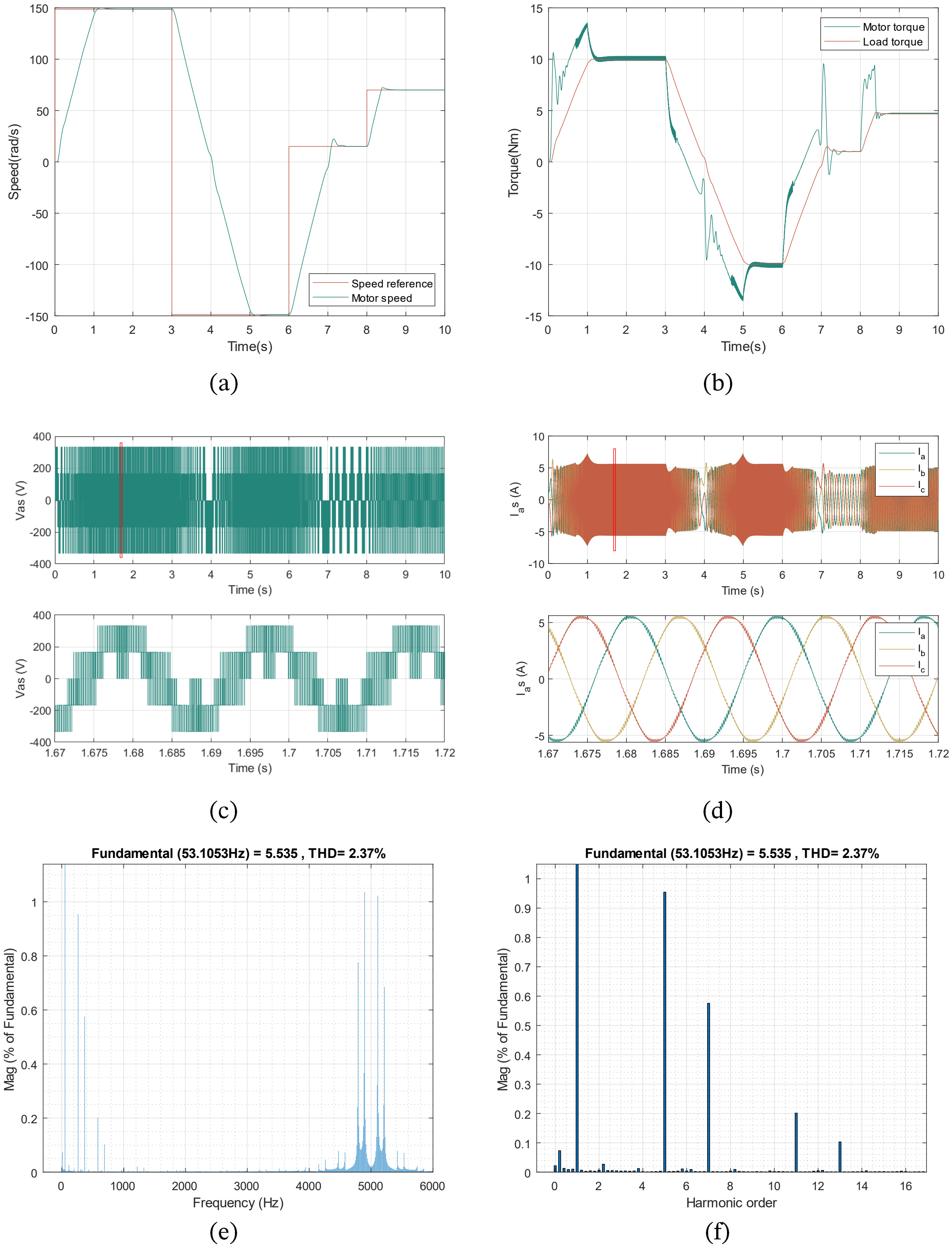


Figure II.3 : Simulation results of V/f constant control for a **constant load**; (a) Speed reference, motor speed, (b) Electromagnetic torque developed by the motor and load torque, (c) stator voltage for one phase, (d) stator current for all three phases, (e) current harmonics spectrum, (f) low order current harmonics



(a)

(b)

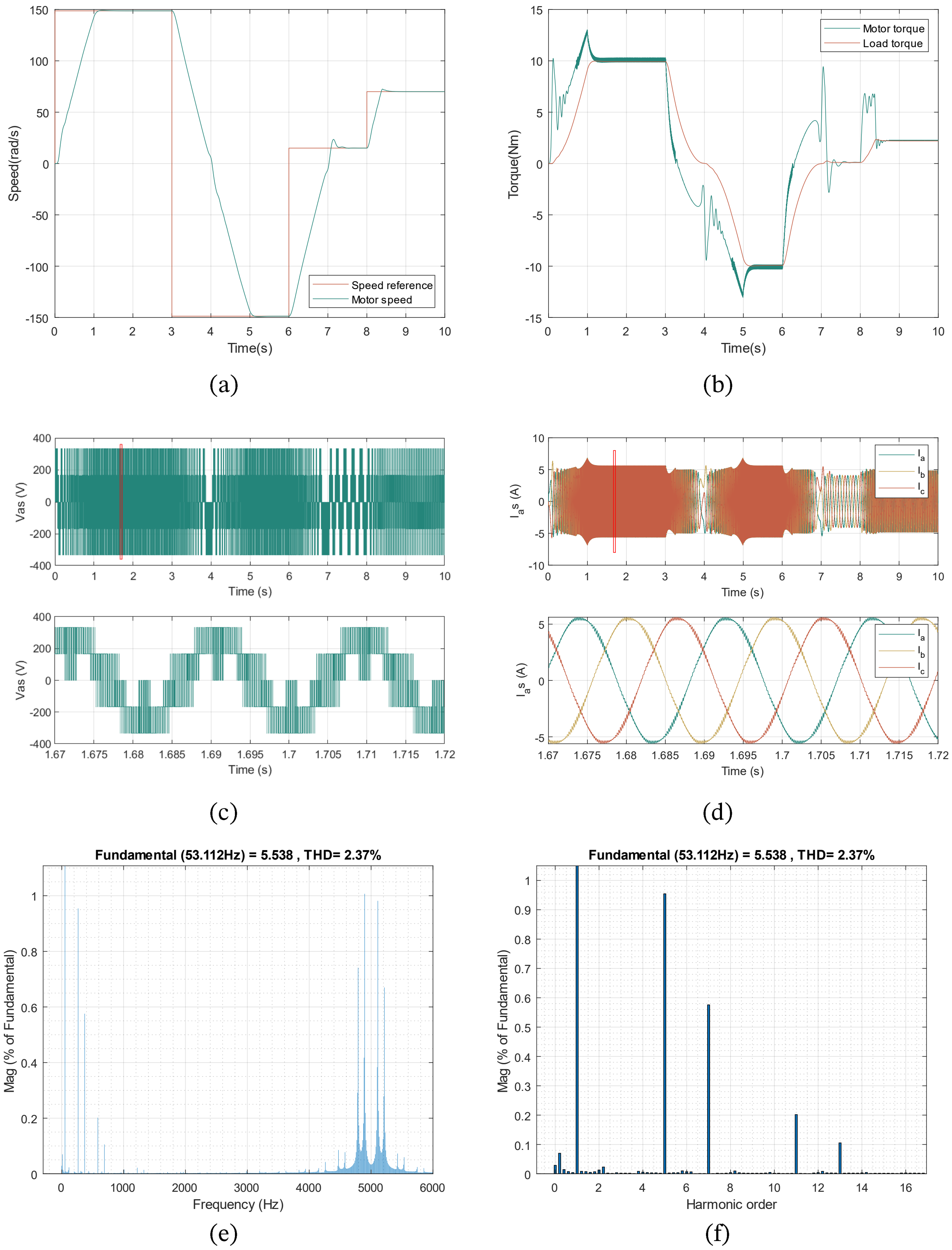
(c)

(d)

(e)

(f)

Figure II.4 : Simulation results of V/f constant control for a **load torque proportional to speed**; (a) Speed reference, motor speed, (b) Electromagnetic torque developed by the motor and load torque, (c) stator voltage for one phase, (d) stator current for all three phases, (e) current harmonics spectrum, (f) low order current harmonics



(a)

(b)

(c)

(d)

(e)

(f)

Figure II.5 : Simulation results of V/f constant control for a **load propotional to the square of the motor speed**; (a) Speed reference, motor speed, (b) Electromagnetic torque developed by the motor and load torque, (c) stator voltage for one phase, (d) stator current for all three phases, (e) current harmonics spectrum, (f) low order current harmonics

Figures II.3, II.4 and II.5 show that the control method was able to correctly impose a speed reference for all three types of load, it was able to accelerate at a steady rate, decelerate and reverse direction and operate in low speeds. The steady-state error was compensated with the PI controller. Torque matches the load torque in constant speeds, it can however exhibit large ripples during speed or load variation. The voltage frequency and amplitude follow the variation of speed, while the current amplitude follows the variation of torque.

The control methods show good performances in starting and low speed conditions. The starting current is low since voltage is proportional to speed, which is an advantage that VFDs offer compared to Direct-on-line starting or star-delta starting methods.

Current distortion is minimal thanks to the use of Space Vector Modulation with a high switching frequency having a THD of less than 2.4%. Low order harmonics in current are very low. The 5th harmonic has a maximum value of less than 1% of the fundamental, all the harmonics are pushed to the switching frequency of 5 kHz as shown by plots (e) and (f) of the figures.

II.4 DTC

II.4.1 Working principle of DTC

The principle of DTC is based on a direct determination of the control pulses applied to the switches of the voltage inverter. It has good robustness with respect to the parameters of the machine as well as remarkable dynamic performances [55], [56].

The basic scheme for DTC is given in Figure II.6.

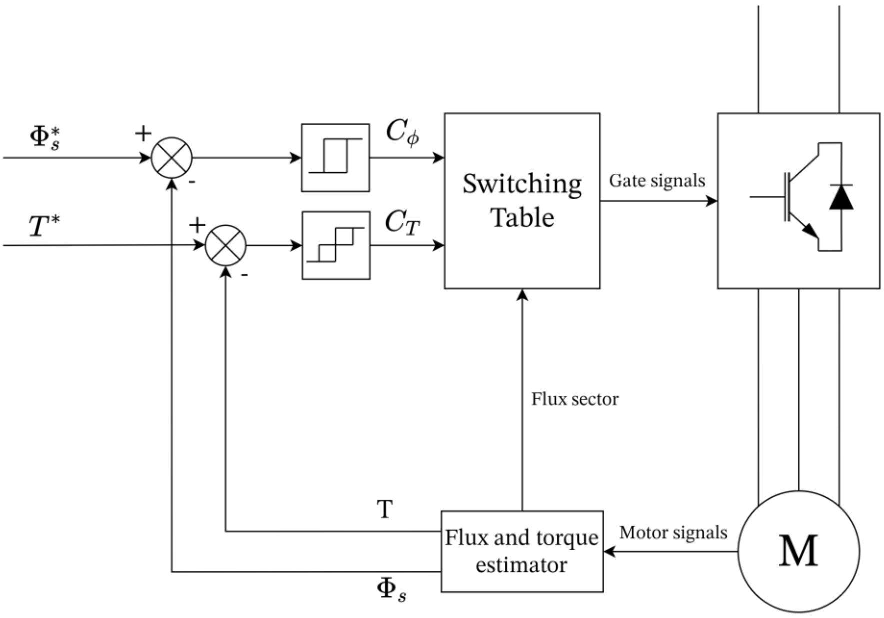


Figure II.6 : Basic ST-DTC Scheme.

Direct torque control does not require a complex coordinate transform. The decoupling of the nonlinear AC motor structure is obtained by the use of on-off control, which can be related to the on-off operation of the inverter power switches. Similarly to field-oriented control, the flux and the torque are either measured or estimated and used as feedback signals for the controller. However, as opposed to field-oriented control, the states of the power switches are determined directly from the measured and the reference torque and flux signals.

Direct torque control is achieved by means of a selection table, the inputs of which are the torque error, the error in magnitude of the stator flux space vector, and the angle of the stator flux space vector. The magnitude error signal of the stator flux is discretized into two levels by means of a hysteresis comparator. The torque error signal is discretized into three levels by means of a three-stage hysteresis comparator. The stator flux angle is discretized into six 60 degrees sections. The outputs of the selection table are the settings for the power switching devices of the inverter.[57]

In a DTC motor drive torque and flux are controlled directly without a current controller [56]. It is based on the equation of the electromagnetic torque developed by the machine since:

$$T_e = n_p Im\{\vec{\phi}_s^* \vec{\imath}_s\} \tag{II.20}$$

and:

$$\vec{\imath}_s = \frac{1}{\sigma L_s}\vec{\phi}_s - \frac{L_m}{\sigma L_s L_r}\vec{\phi}_r\,, \qquad (\sigma = 1 - \frac{L_m^2}{L_s L_r}) \tag{II.21}$$

Substituting Eq. (II.20) in Eq. (II.21) yields:

$$T_e = n_p \frac{L_m}{\sigma L_s L_r} |\vec{\phi}_s||\vec{\phi}_r| \sin\beta \tag{II.22}$$

β being the angle between the stator and rotor flux vectors [58].

The derivative of Eq. (II.22) can be approximated by:

$$\frac{dT_e}{dt} = n_p \frac{L_m}{\sigma L_s L_r} |\vec{\phi}_s||\vec{\phi}_r| \frac{d\beta}{dt} \cos\beta \tag{II.23}$$

Eq. (II.23) shows that the torque dynamics depend on the variation of β, and therefore a fast torque control can be achieved by rapidly changing β.

In a referential linked to the stator of the machine, the stator flux can be expressed by the following equation:

$$\phi_s(t) = \int_0^t (V_s - R_s I_s)dt + \Phi_{s0} \tag{II.24}$$

If the voltage vector applied during the interval [0, t_e] is non zero, by neglecting the voltage drop $R_s I_s$, Eq. (II.24) yields:

$$\phi_s(t) = \phi_s(0) + V_s t_e \tag{II.25}$$

Therefore, by applying a suitable stator voltage vector the stator flux can be rotated.

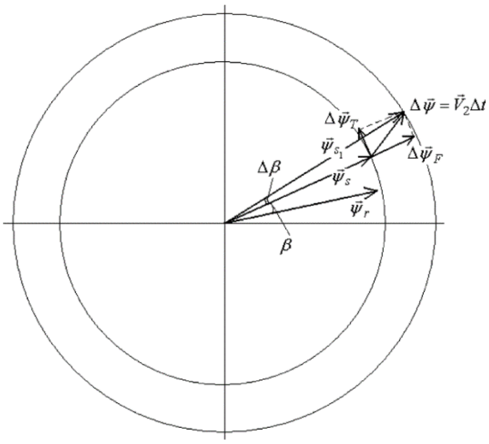


Figure II.7 : Flux vector variation during a switching period in DTC.

Figure II.7 shows how the flux vector rotates inside a ring which width is the hysteresis band, during one switching period in DTC.

This is achieved through a two-level hysteresis controller for which the output response is shown in Figure II.8 : when the stator flux is too high the controller's output goes high, and the voltage vector is selected by the switching table so as to decrease the flux and vice versa.

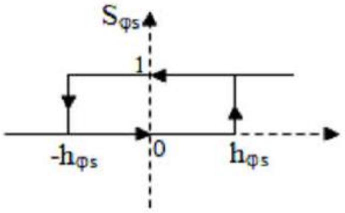


Figure II.8 : Output response of a two-level hysteresis controller.

To control torque, a three-level hysteresis controller is used since it was demonstrated in Eq. (II.23) that the variation in torque is directly dependent on the variation of angle of stator flux, its output response is shown in Figure II.9.

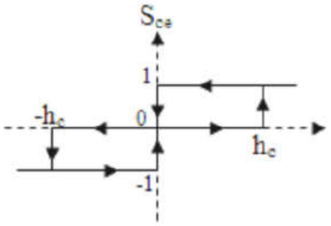


Figure II.9 : Output response of a three-level hysteresis controller.

In DTC, the accuracy of electromagnetic torque and stator flux estimation is very important to ensure satisfactory performance [59], [60]. So, several parameters must be determined, the stator current is measured while the stator voltage depends on the switching state (Sa, Sb and Sc) produced by the switching table and the DC link voltage Vdc [61]. These parameters are transformed into coordinates (α,β) by the Concordia transformation using Eq.(II.26), which are suitably adapted to the DTC algorithm [32].

$$\begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \tag{II.26}$$

In this reference frame the stator flux components' expressions are:

$$\begin{cases} \phi_{s\alpha} = \int_0^t (V_{s\alpha} - R_s I_{s\alpha}) dt \\ \phi_{s\beta} = \int_0^t (V_{s\beta} - R_s I_{s\beta}) dt \end{cases} \tag{II.27}$$

Currents and voltages in (α, β) coordinates are as follows:

$$\begin{cases} I_\alpha = I_a \\ I_\beta = \frac{\sqrt{3}}{3}(I_a + 2I_b) \end{cases} \quad \text{(II.28)}$$

$$\begin{cases} V_\alpha = \frac{V_{dc}}{3}(2S_a - S_b - S_c) \\ V_\beta = \frac{\sqrt{3}}{3}V_{dc}(S_b + S_c) \end{cases} \quad \text{(II.29)}$$

The stator flux φ_s is estimated from the following equations:

$$\begin{cases} \phi_{s\alpha} = \phi_{s\alpha_{old}} + (V_{s\alpha} - R_s I_{s\alpha})t_e \\ \phi_{s\beta} = \phi_{s\beta_{old}} + \left(V_{s\beta} - R_s I_{s\beta}\right)t_e \end{cases} \quad \text{(II.30)}$$

$$\phi_s = \sqrt{\phi_{s\alpha}^2 + \phi_{s\beta}^2} \quad \text{(II.31)}$$

With the stator flux angle θ_s calculated from :

$$\theta_s = \tan^{-1}(\frac{\phi_{s\beta}}{\phi_{s\alpha}}) \quad \text{(II.32)}$$

While the torque is estimate with the formula:

$$T_{em} = \frac{3}{2}n_p(\phi_{s\alpha}i_{s\beta} - \phi_{s\beta}i_{s\alpha}) \quad \text{(II.33)}$$

The angle calculated by Eq.(II.32) determines the operation sector illustrated by Figure II.10. The combination between the output hysteresis controllers and the sector is then applied to an optimal switching table, given by Table II.1, known as Takahashi table, which will give the voltage vector to be applied to the two-level voltage source inverter (VSI) [55].

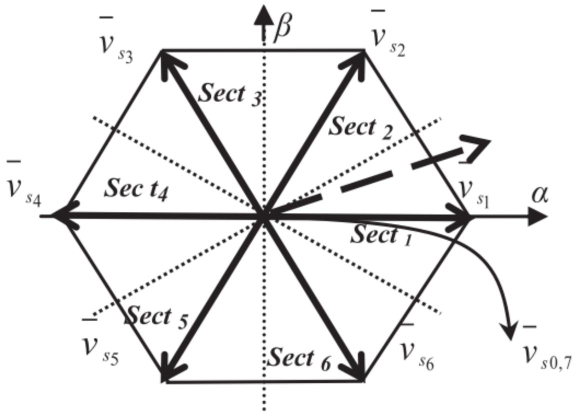


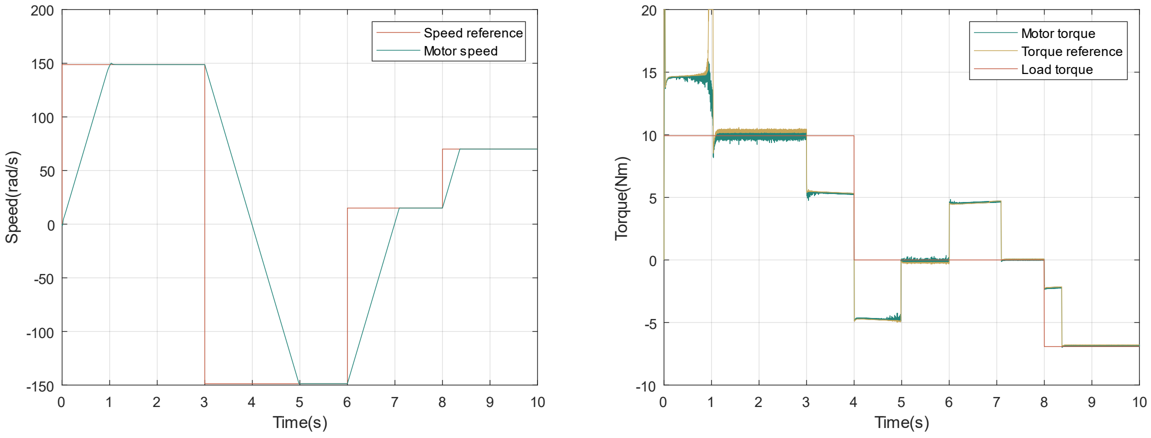
Figure II.10 : Location of the different flux sectors and voltage vectors in (α, β).

Table II.1 : Takahashi Switching Table for classical DTC.

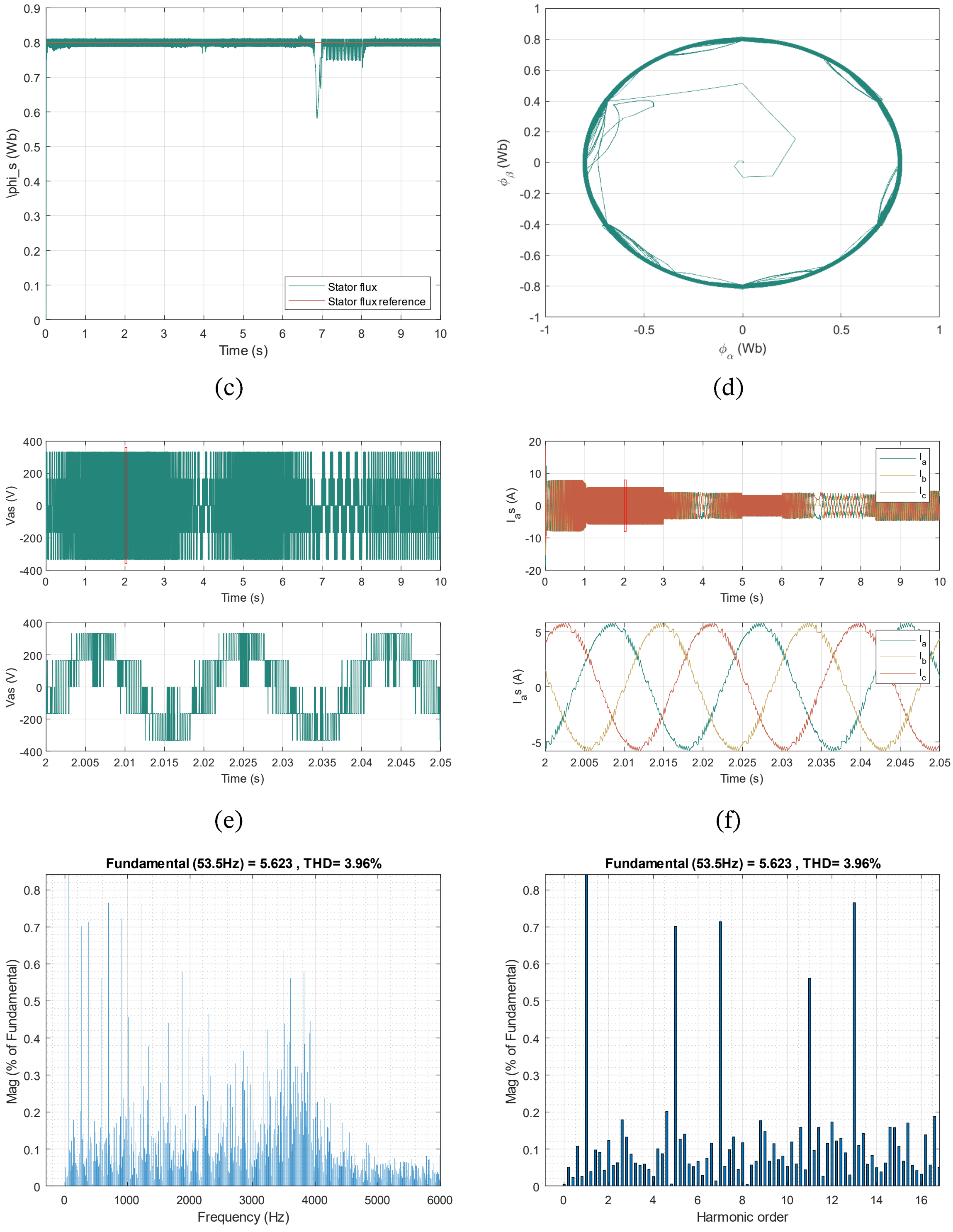
		Sector					
$E_{\Phi s}$	E_{Te}	1	2	3	4	5	6
1	1	$\bar{v}_{s2}$	$\bar{v}_{s3}$	$\bar{v}_{s4}$	$\bar{v}_{s5}$	$\bar{v}_{s6}$	$\bar{v}_{s1}$
	0	$\bar{v}_{s0}$	$\bar{v}_{s7}$	$\bar{v}_{s0}$	$\bar{v}_{s7}$	$\bar{v}_{s0}$	$\bar{v}_{s0}$
	-1	$\bar{v}_{s6}$	$\bar{v}_{s1}$	$\bar{v}_{s2}$	$\bar{v}_{s3}$	$\bar{v}_{s4}$	$\bar{v}_{s5}$
-1	1	$\bar{v}_{s3}$	$\bar{v}_{s4}$	$\bar{v}_{s5}$	$\bar{v}_{s6}$	$\bar{v}_{s1}$	$\bar{v}_{s2}$
	0	$\bar{v}_{s7}$	$\bar{v}_{s0}$	$\bar{v}_{s7}$	$\bar{v}_{s0}$	$\bar{v}_{s7}$	$\bar{v}_{s0}$
	-1	$\bar{v}_{s5}$	$\bar{v}_{s6}$	$\bar{v}_{s1}$	$\bar{v}_{s2}$	$\bar{v}_{s3}$	$\bar{v}_{s4}$

II.4.2 Simulation

Results of the simulation of a DTC control scheme are shown in figures Figure II.11, Figure II.12 and Figure II.13. A speed control loop was added with a wrapped-state PI controller. The acceleration rate was limited to 150 rad/s^2 and the torque reference generated was limited to 20N.m, the stator flux reference was set to 0.8 Wb with a hysteresis band of 0.01 and the torque hysteresis band to 0.1. The speed refenrence is the same as with the V/f constant control simulation as well as the load. The motor parameters used for the simulation are given in Appendix A: Induction motor parameters.



(a) (b)

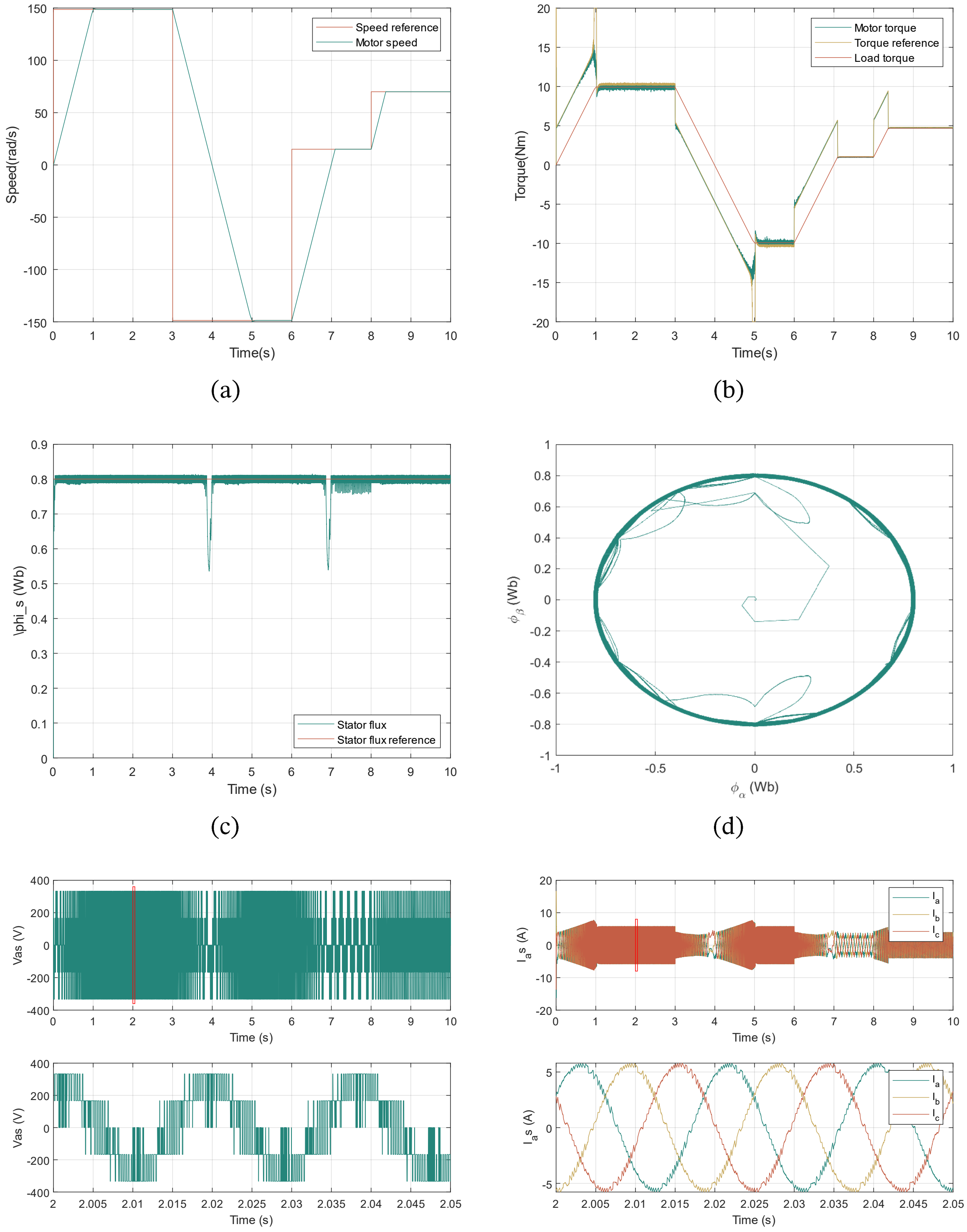


(c) (d)

(e) (f)

(g) (h)

Figure II.11 : Simulation results of DTC with a **constant load torque**;(a)motor speed and it's reference, (b) motor torque, it's reference and the load torque, (c) amplitude of the stator flux vector, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics


Speed reference
Motor speed
Speed(rad/s)
Time(s)
Motor torque
Torque reference
Load torque
Torque(Nm)
Time(s)
Stator flux
Stator flux reference
\phi_s (Wb)
Time (s)
ϕ_β (Wb)
ϕ_α (Wb)
Vas (V)
Time (s)
I_a^s (A)
I_a
I_b
I_c
Time (s)

(a)

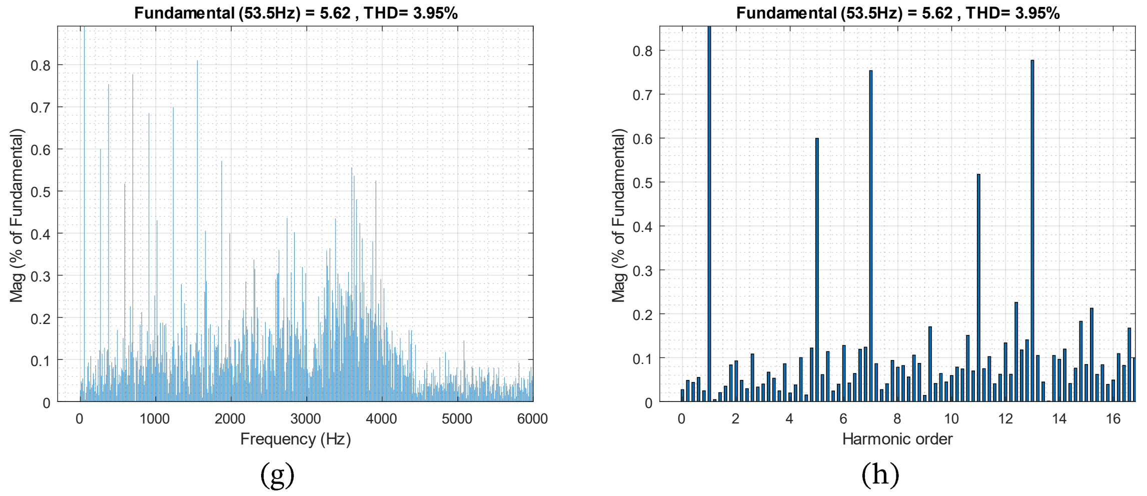
(b)

(c)

(d)

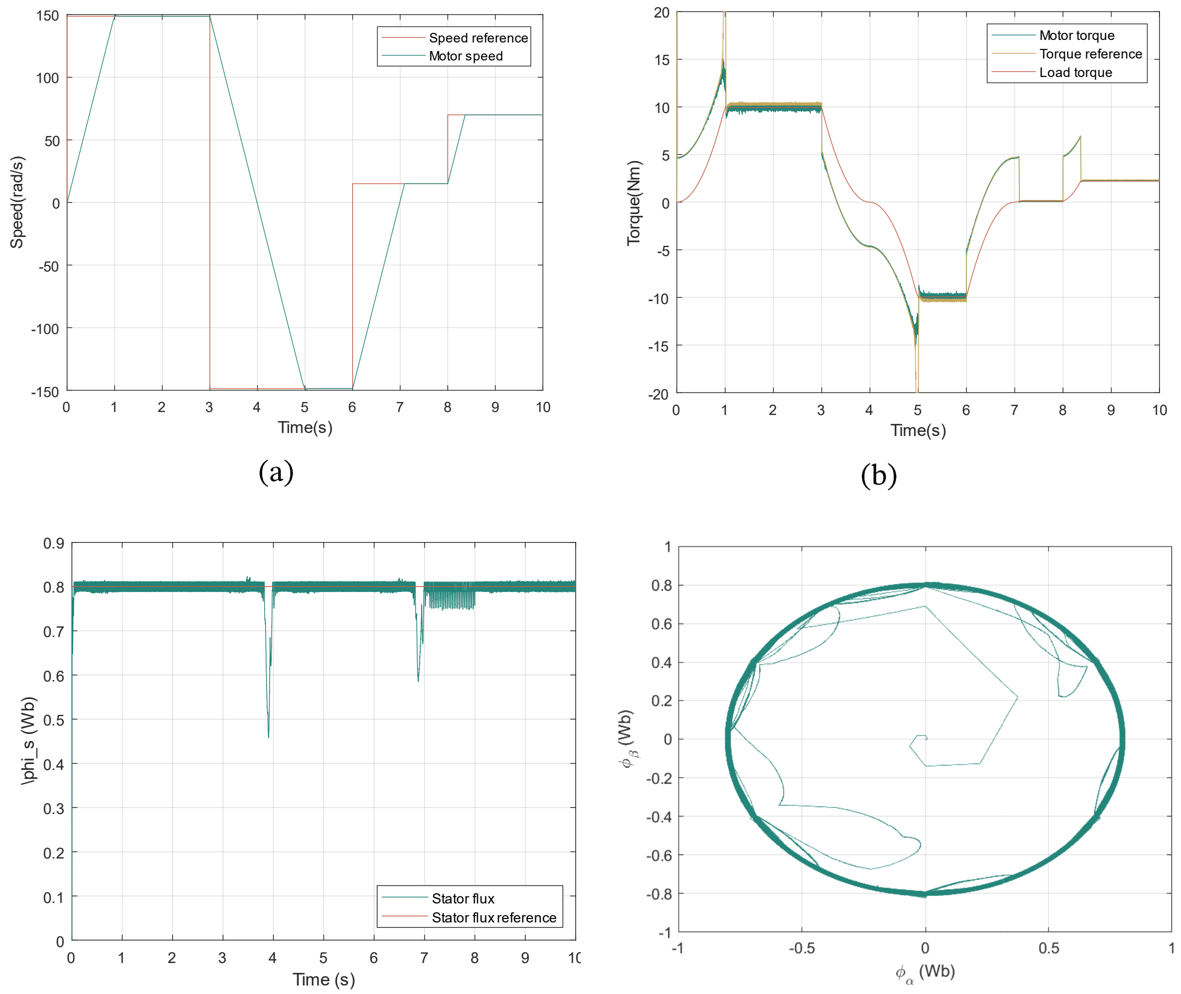
(e)

(f)



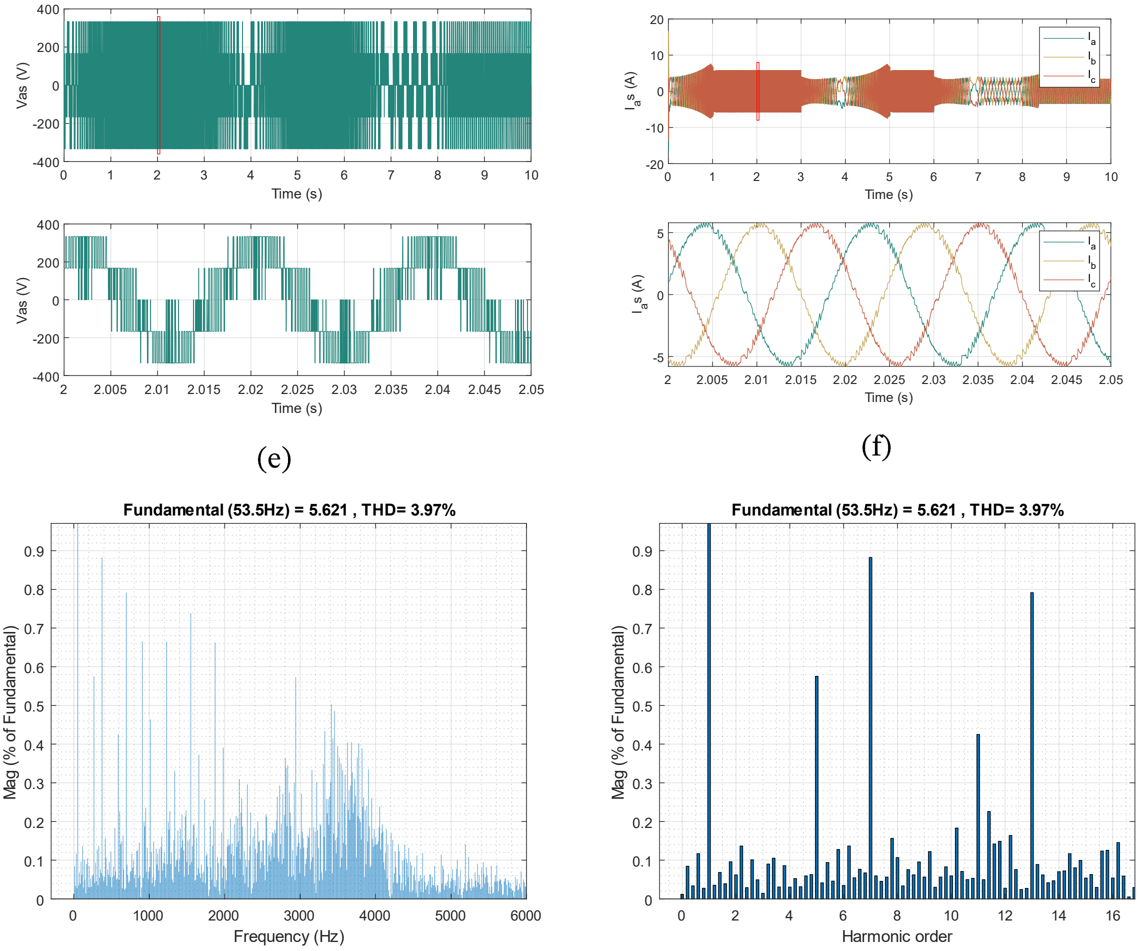
(g) (h)

Figure II.12 : Simulation results of DTC with a **load torque proportional** to speed;(a)motor speed and it's reference, (b) motor torque, it's reference and the load torque, (c) amplitude of the stator flux vector, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics



(a) (b)

(c) (d)



(e) (f)

(g) (h)

Figure II.13 : Simulation results of DTC with **load torque proportional to the square of motor speed**; (a) motor speed and it's reference, (b) motor torque, it's reference and the load torque, (c) amplitude of the stator flux vector, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics

Figures II.11, II.12 and II.13 show that the motor speed follows the reference perfectly in steady and transient states with a maximum overshoot of less than 0.1 rad/s and the imposed acceleration rate of 150 rad/s^2 is respected as seen in plots (a). Flux remains close to its reference but dips dramatically with quick speed changes. Torque remains limited by the hysteresis band of 0.1 N.m with occasional ripples but doesn't exceed the limit imposed of 20 N.m as shown in plots (b) of the figures. The frequency and amplitude of the stator voltage varies with respect to speed, while current amplitude varies with the variation of torque. Currents have lots of distortion compared to scalar control, with a superior THD of 3.97% and harmonics in all frequencies, although low order harmonics (5^{th}, 7^{th} order) are of the same magnitude as shown in plots (g) and (h) of the figures.

In all three cases the speed is very well adjusted to the reference with negligeable overshoot and no steady-state error. Torque is also well adjusted to its reference. This control method is superior to scalar control in that it shows great performance in transient state with small torque ripples and minimal startup inrush current, but has more current harmonics in all frequencies as the switching frequency is variable.

II.5 DTC-SVM

II.5.1 Working principle

DTC-SVM improve upon classical DTC by operating at a constant switching frequency, the controllers calculate the required stator voltage vector that is realized with the Space Vector Modulation technique. Several methods of generating this voltage vector exist: PI controllers, neural networks/fuzzy logic, predictive ... etc. or a combination of them.

Space Vector Modulation (SVM):

SVM considers a complex voltage vector as the reference waveform to be generated. This reference signal is sampled with a constant frequency and the converter generates it using a linear combination of possible switching states that can be achieved by the power converter. A switching sequence formed by several switching states of the converter is performed and the average value of the output voltage must coincide with the desired reference one.

A graphical representation of the output voltages of all the discrete switching configurations of the power converter helps to determine the switching sequence and the time of each switching state, as shown in Figure II.14 where Vref is the reference voltage; its position in each sector determines the duration the switching states have to be applied. In the two-level three-phase VSI only eight possible switching states are possible depending on the values of S1, S2, and S3 [62].

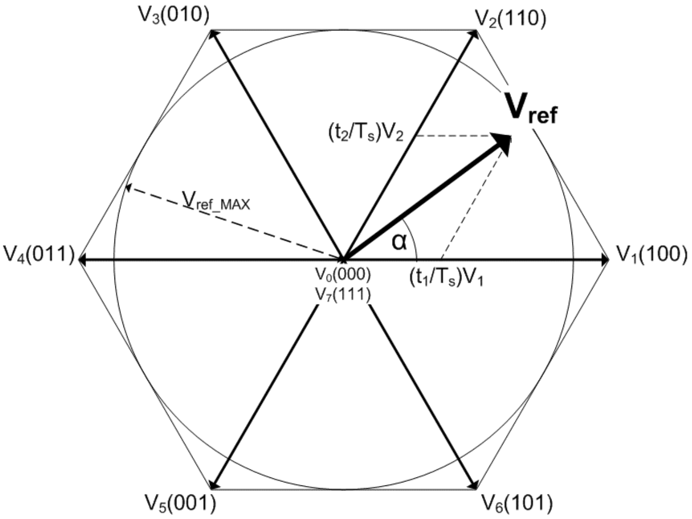


Figure II.14 : Principle of space vector modulation for a three phase two level VSI

A summary of the possible switching states and corresponding phase voltages is shown in Table II.2

TABLE II.2 : SWITCHING STATES OF THE THREE-PHASE TWO-LEVEL POWER CONVERTER

S_1	S_2	S_3	V_{a0}/V_{DC}	V_{b0}/V_{DC}	V_{c0}/V_{DC}
0	0	0	-1/2	-1/2	-1/2
1	0	0	1/2	-1/2	-1/2
1	1	0	1/2	1/2	-1/2
0	1	0	-1/2	1/2	-1/2
0	1	1	-1/2	1/2	1/2
0	0	1	-1/2	-1/2	1/2
1	0	1	1/2	-1/2	1/2
1	1	1	1/2	1/2	1/2

The two state vectors 000 and 111 are placed in the same location, they are equal seen from the load. The choice between these two vectors has to be done as to avoid unnecessary switching.

DTC-SVM:

Many DTC-SVM schemes have been reported in literature. The torque and flux closed loop DTC-SVM with PI controllers is represented in Figure II.15.

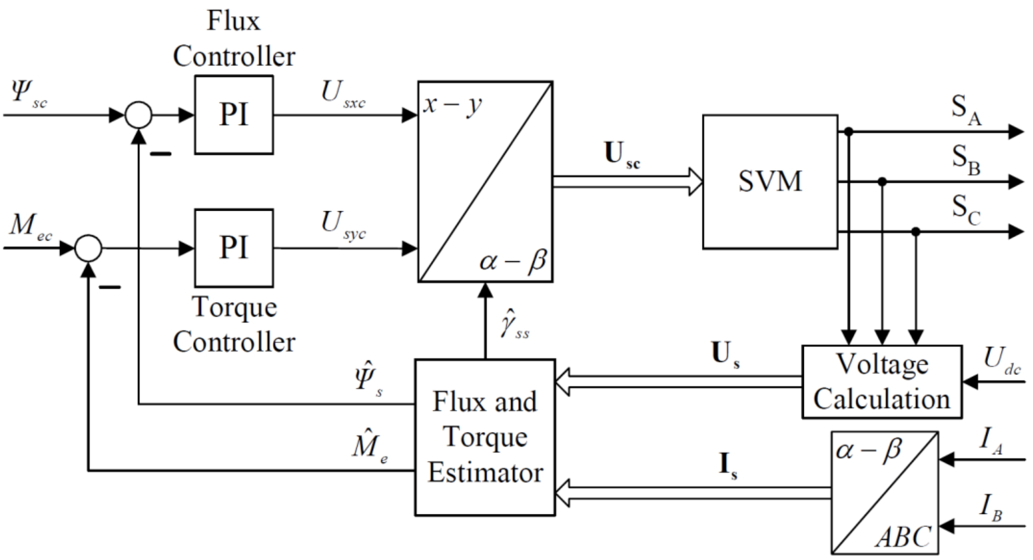


Figure II.15 : DTC-SVM scheme with closed-loop flux and torque control in stator flux coordinates

Flux and torque are estimated similarly to classic DTC, the angle of the estimated flux vector is the angle of the dq to αβ transformation of voltage. In the xy (dq) refence frame flux is expressed according to [63] by:

$$\frac{d\phi_s}{dt} = U_{sx} - R_s I_{sx} \tag{II.34}$$

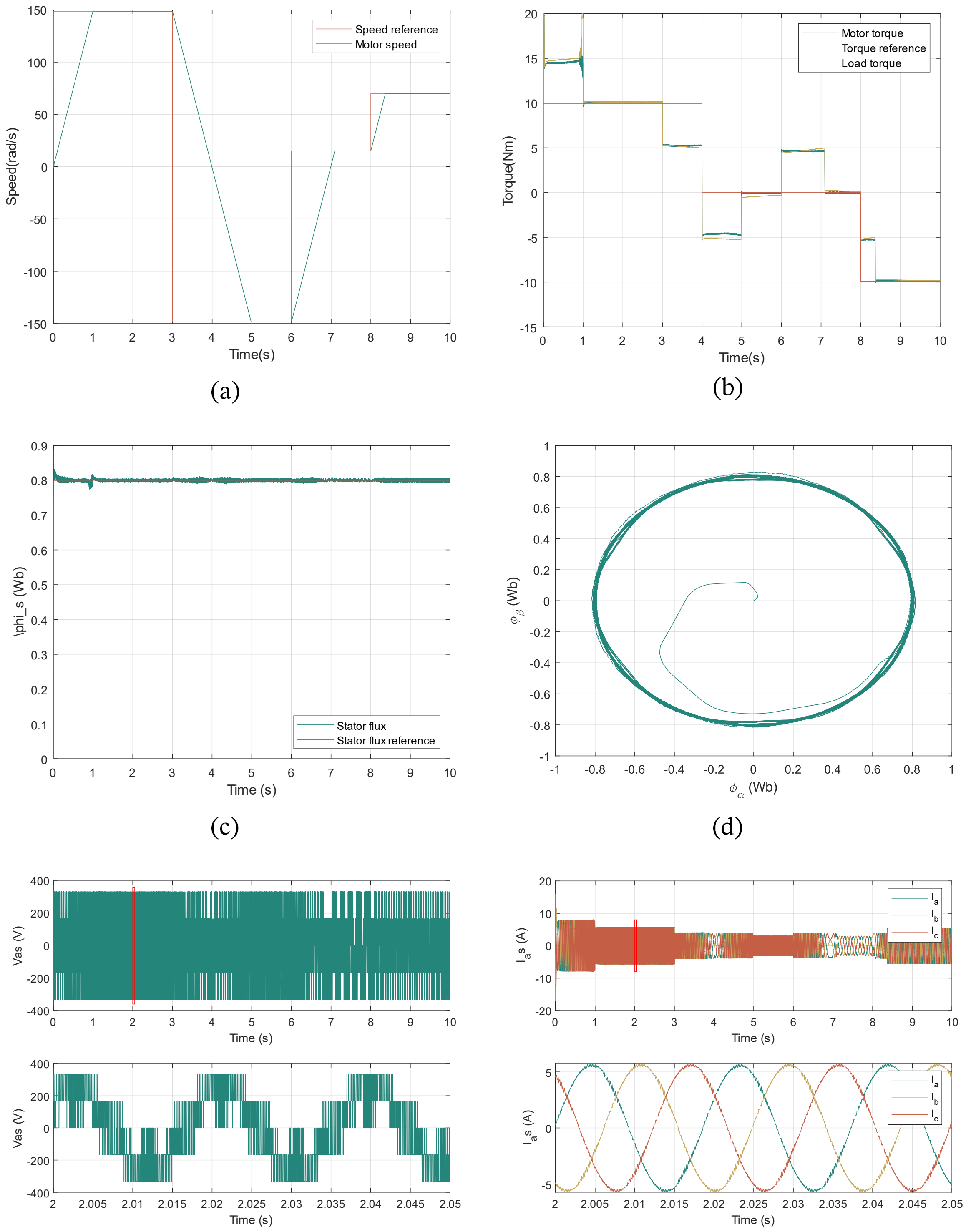
And torque by

$$C_{em} = \frac{1}{R_s} P\phi_s(U_{sy} - \Omega_{ss}\phi_s) \tag{II.35}$$

This shows that the flux error can be used to create the direct stator voltage reference through a PI controller, and torque to create the quadratic stator voltage reference. These reference voltages are then transformed to the αβ reference frame and fed to the SVM modulator to generate the appropriate switching states.

II.5.2 Simulation

The DTC-SVM scheme has been simulated using the same speed and flux references as with the classic DTC. The same speed control loop with same parameters has also been added. The three types of load have also been simulated. The acceleration/deceleration rate has been limited to 150rad/s^2 and maximum torque limited to +/- 20 N.m. The motor parameters used for the simulation are given in Appendix A: Induction motor parameters.


Speed reference
Motor speed
Speed(rad/s)
Time(s)
Motor torque
Torque reference
Load torque
Torque(Nm)
Time(s)
\phi_s (Wb)
Stator flux
Stator flux reference
Time (s)
ϕ_β (Wb)
ϕ_α (Wb)
Vas (V)
Time (s)
I_as (A)
I_a
I_b
I_c
Time (s)

(a)

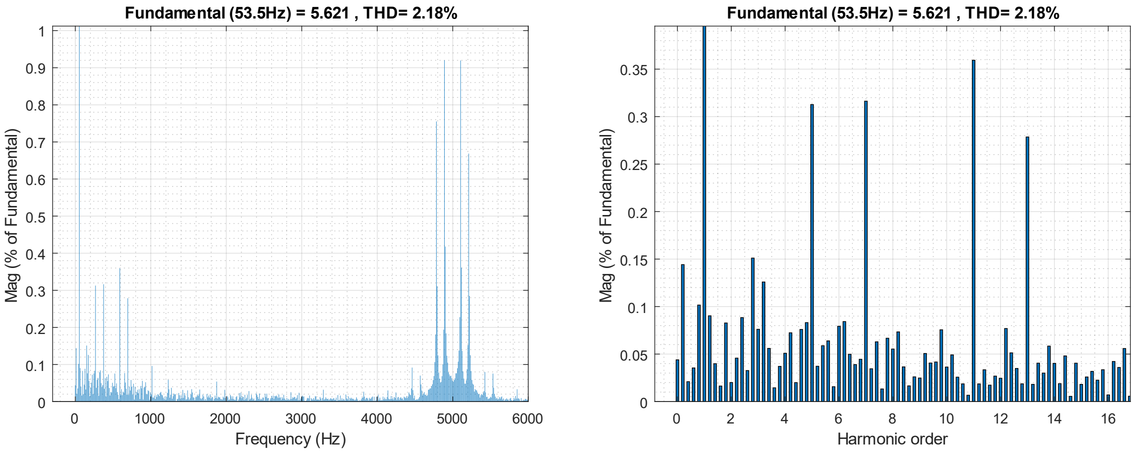
(b)

(c)

(d)

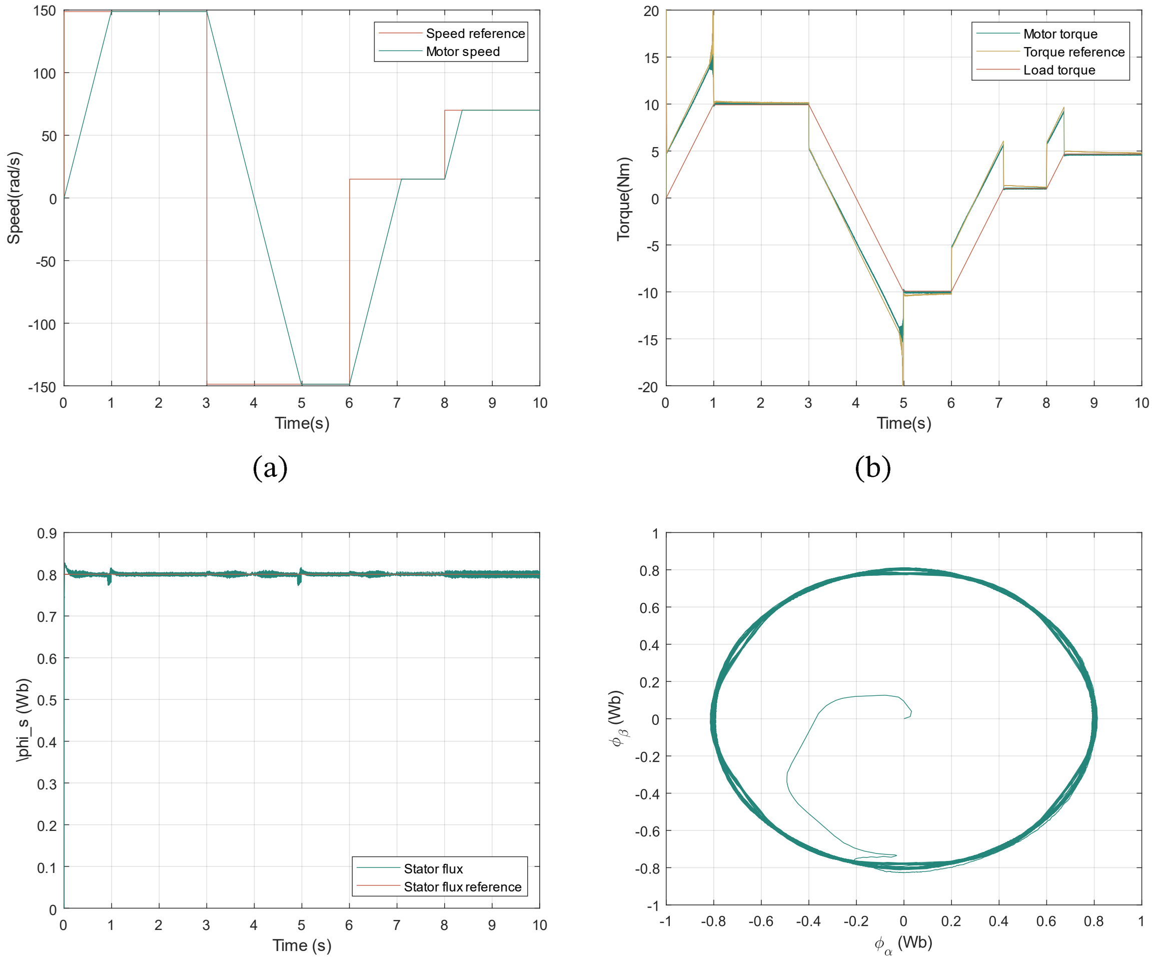
(e)

(f)



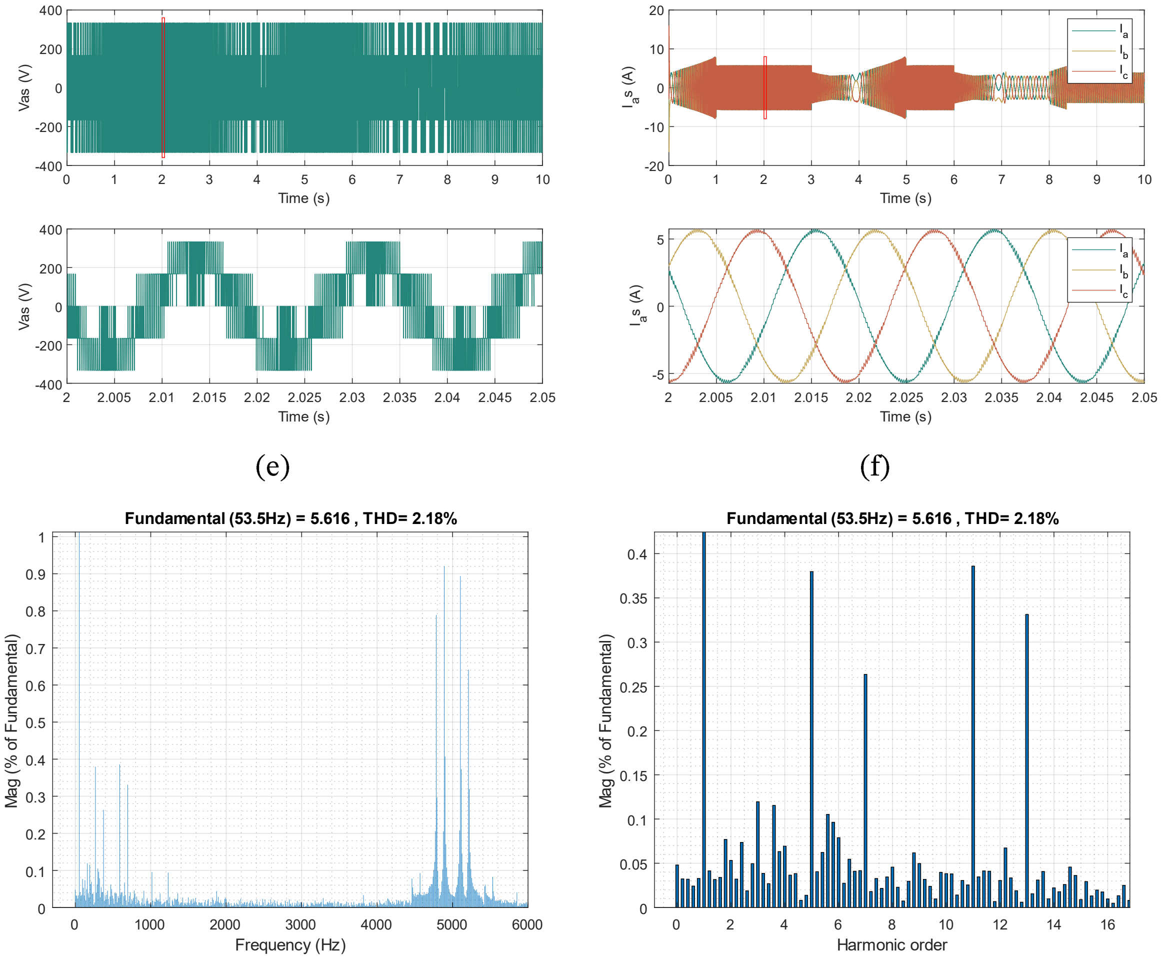
(g) (h)

Figure II.16 : Simulation results of DTC-SVM with a **constant load torque**;(a)motor speed and it's reference, (b) motor torque, it's reference and the load torque, (c) amplitude of the stator flux vector, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics



(a) (b)

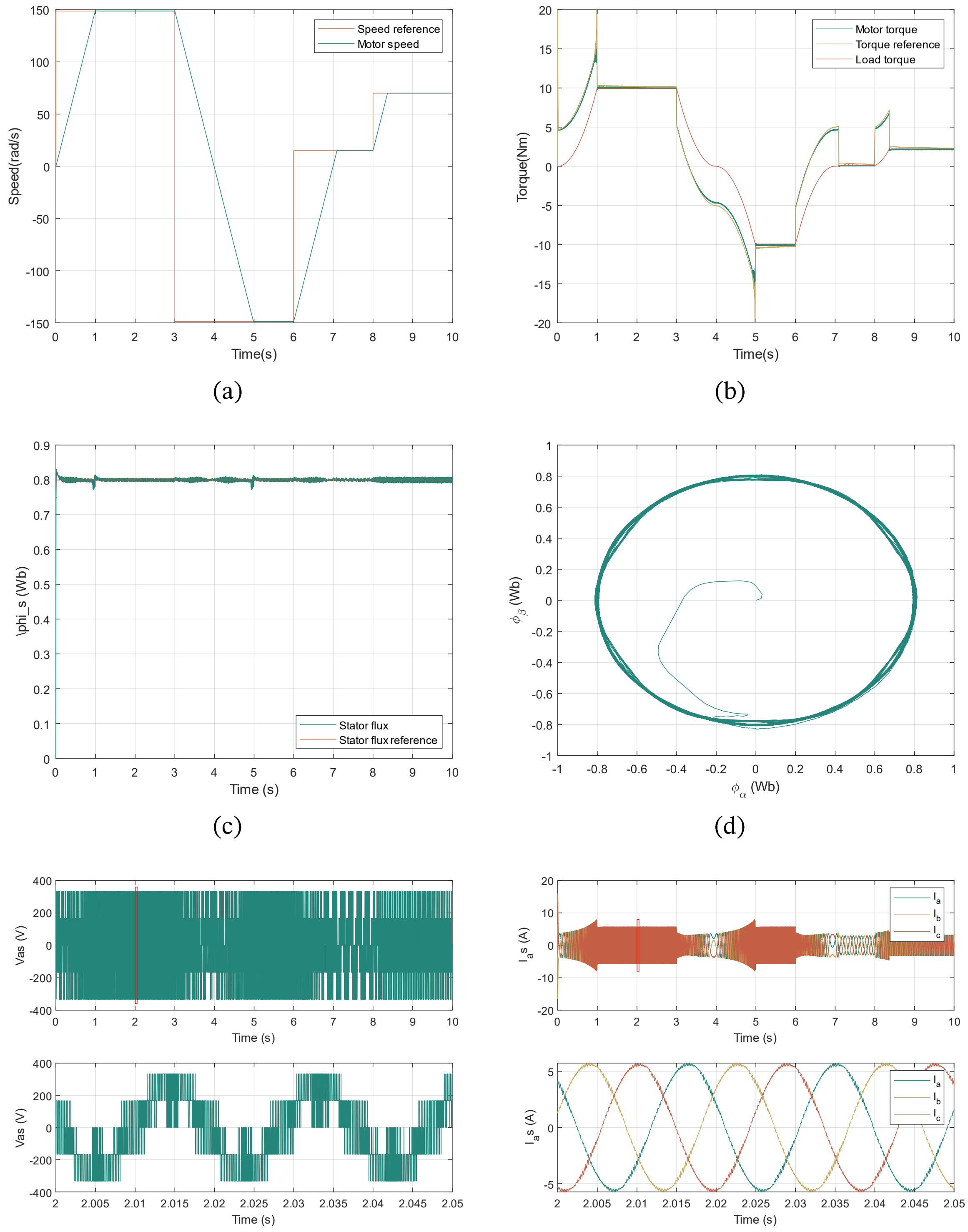
(c) (d)



(e) (f)

(g) (h)

Figure II.17 : Simulation results of DTC-SVM with a **load torque proportional to speed**;(a)motor speed and it's reference, (b) motor torque, it's reference and the load torque, (c) amplitude of the stator flux vector, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics


Speed reference
Motor speed
Speed(rad/s)
Time(s)
150
100
50
0
-50
-100
-150
0 1 2 3 4 5 6 7 8 9 10
Motor torque
Torque reference
Load torque
Torque(Nm)
Time(s)
20
15
10
5
0
-5
-10
-15
-20
Stator flux
Stator flux reference
\phi_s (Wb)
Time (s)
0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0
ϕ_β (Wb)
ϕ_α (Wb)
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
Vas (V)
400 200 0 -200 -400
Time (s)
2 2.005 2.01 2.015 2.02 2.025 2.03 2.035 2.04 2.045 2.05
I_a^s (A)
I_a
I_b
I_c
20 10 0 -10 -20
5 0 -5

(a)

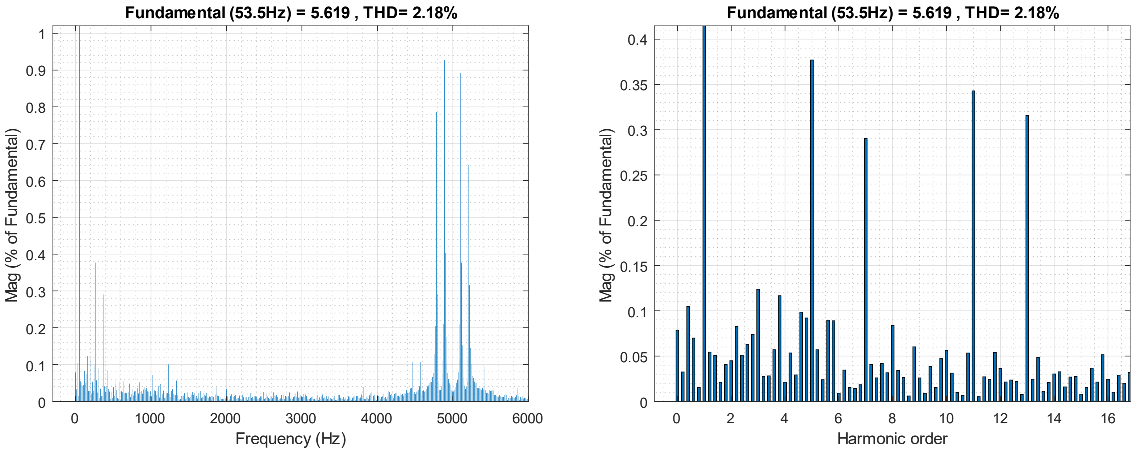
(b)

(c)

(d)

(e)

(f)



(g) (h)

Figure II.18 : Simulation results of DTC-SVM with **load torque proportional to the square of motor speed**; (a)motor speed and it's reference, (b) motor torque, it's reference and the load torque, (c) amplitude of the stator flux vector, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics

Figures II.16, II.17 and 11.18 show that motor speed closely follows the refence with a maximum overshoot of less than 0.1 rad/s and the imposed acceleration rate of 150 rad/s^2 is respected. Torque ripples are minimal and torque follows the reference. Flux remains close to its reference in a band smaller than 0.01Wb wide. The frequency and amplitude of the stator voltage varies with respect to speed, while current amplitude varies with the variation of torque with no large spikes in current, current harmonics are also low.

The simulation results for both load torque types are similar to the results for a constant load torque, the torque is able to adjust to a rapidly changing load, flux is kept tightly close to the reference. Current harmonics are the smallest with most harmonics present at the switching frequency of 5 kHz and a THD of 2.18% and low order harmonics lower than 0.5% of the fundamental.

In all three cases the motor speed follows the reference and the imposed acceleration rate. DTC-SVM shows and advantage over classic DTC in current distortion and a fixed switching frequency, but with a higher computational cost as it requires two more PI control loops.

II.6 Results discussion and comparison

Scalar control is the easiest control method to implement, it is computationally light with the SVM being optional as a sine-PWM switching method would be sufficient, it only requires a motor speed feedback through a speed encoder and one PI control loop to achieve a good speed regulation with no steady state error. It also presents very low current harmonics, less than the classic DTC control.

The classis DTC control method shows superior control features to Constant V/f, as it controls both torque and flux and speed through a speed control loop. It achieves a very precise speed regulation with negligible overshoot, the torque controller adjusts quickly to changes in load and the startup inrush current is minimal. This control method is also more robust than scalar control.

It however is dependent on a good knowledge of motor parameters namely the stator resistance which is required to estimate flux and torque.

The DTC-SVM has the best performance overall with all the advantages of scalar control in low current harmonics and DTC with good torque and flux regulation. It is however more costly as it requires two more PI controllers along the flux and torque estimator which is also dependent on motor parameters.

In all the control methods low speed operation is achieved for a brief period of time, though currents were high enough that they would lead to overheating of the motor. This requires external cooling if the motor is to be ran at low speed for extended periods of time

The comparison between the three control methods is summarized in Table II.3.

TABLE II.3 : COMPARISON OF SIMULATION RESULTS FOR SCALAR CONTROL, DTC AND DTC-SVM

Comparison criterion	Constant V/f	DTC	DTC-SVM
Speed dynamic response	Slow	Fast	Fast
Torque dynamic response	Slow	Fast	Fast
Flux dynamic response	Slow	Fast	Fast
Current THD	Low	Medium	Low
Switching frequency	Fixed	Variable	Fixed
Parameter sensitivity	Insensitive	sensitive	sensitive
Controller(s)	1xPI	2xhysteresis	2xPI
Algorithm complexity	Simple	Medium complexity	Medium complexity
Computation time	Low	Low	Medium

II.7 Conclusion

In this chapter an induction motor model was established and was used to model and simulate three control methods: Constant V/f, DTC and DTC-SVM using the MATLAB/Simulink environment. The controls were tested on various load types in different conditions and the results were discussed and compared.

Scalar control showed performances good enough for most applications as it was able to vary the motor speed effectively and ramp-up the speed progressively during startup and shutdown. Its ease of implementation and lack of need of current and voltage measurements and motor parameters make it suitable for pumps, fans, conveyors, HVAC systems… etc. where precision in speed/torque control isn't necessary and reducing costs is important.

DTC offers more robust and more performant control but with added complexity. DTC-SVM improves on DTC by having a more performant control with better current harmonics spectrum and an easier implementation on DSPs and Microcontroller as DTC's variable switching frequency make harder to implement on a digital system without modifications. These control methods are suited for applications requiring precision, robustness or operation in specific conditions like EVs and traction systems, high power industrial drives…etc

CHAPTER III

STM32F4 DISCOVERY

III STM32F4 DISCOVERY

III.1 Introduction

The STM32F4 Discovery kit is a versatile tool, giving users access to a powerful microcontroller and all the tools necessary for implementing, testing and debugging projects.

This card will be used for the implementation of the DTC control. It offers synchronizable timers, ADCs DAC's and an easy to use debugging tool, as well as the ability to add all sorts of peripherals.

In this chapter we will present the features of the STM32F4 and its main peripherals, as well as the STM32Cube IDE that will be used for its programming.

III.2 Microcontroller description and features

The STM32F4DISCOVERY Discovery kit allows users to easily develop applications with the STM32F407VG high performance microcontroller. It includes an ST-LINK/V2 or ST-LINK/V2-A embedded debug tool, two ST-MEMS digital accelerometers, a digital microphone, one audio DAC with integrated class D speaker driver, LEDs, push buttons and a USB OTG micro-AB connector[64].

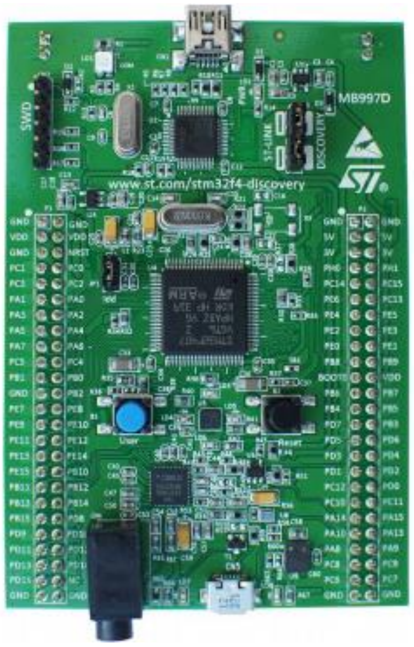


Figure III.1 : STM32F407VG card

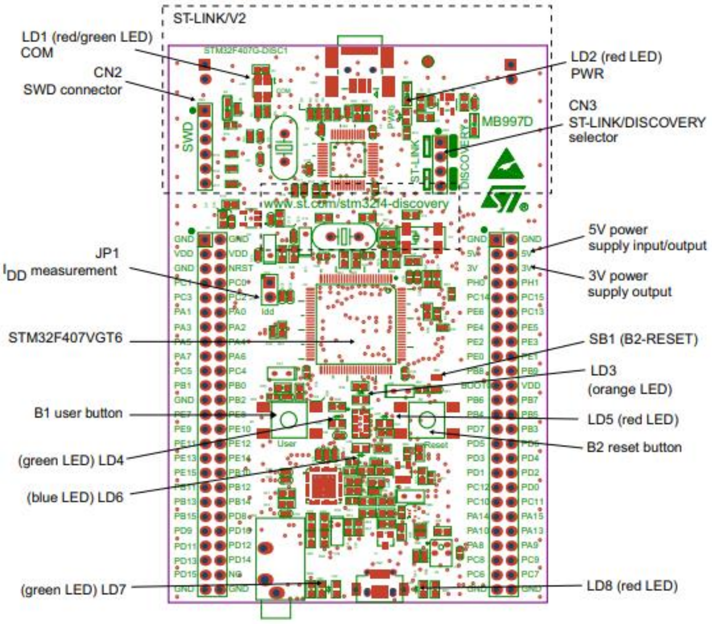


Figure III.2 : STM32F4DISCOVERY top layout

The STM32F407 family is based on the high-performance ARM Cortex-M4 32-bit RISC (Reduced Instruction Set Computer) core operating at a frequency of up to 168MHz. It implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security. The STM32F407xx family incorporates high-speed embedded memories (Flash memory up to 1 Mbyte, up to 192 Kbytes of SRAM), up to 4 Kbytes of backup SRAM, and an extensive range of enhanced I/Os and peripherals connected to two APB buses, three AHB buses and a 32-bit multi-AHB bus matrix.

All devices offer three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose 32-bit timers. a true random number generator (RNG). They also feature standard and advanced communication interfaces.

New advanced peripherals include an SDIO, an enhanced flexible static memory control (FSMC) interface (for devices offered in packages of 100 pins and more), a camera interface for CMOS sensors.

The STM32F407xx family operates in the –40 to +105 °C temperature range from a 1.8 to 3.6 V power supply. The supply voltage can drop to 1.7V when the device operates in the 0 to 70°C temperature range using an external power supply supervisor. A comprehensive set of power-saving mode allows the design of low-power applications.

The main features of the STM32F407VGT6U microcontroller are summarized in Table III.1 [65].

Table III.1 : Features of the STM32F407Vx microcontroller family

Peripherals		**STM32F407Vx**
Flash memory in Kbytes		1024
SRAM in Kbytes	**System**	192 (112+16+64)
	Backup	4
FSMC memory controller		Yes
Ethernet		Yes
Random number generator		Yes

Peripherals		**STM32F407Vx**
Timers	**General-purpose**	10
	Advanced control	2
	Basic	2
	IWDG	Yes
	WWDG	Yes
	RTC	Yes

Communication Interfaces	**SPI/I^{2}S**	3/2(full duplex)
	I^{2}C	3
	USART/UART	4/2
	USB OTG FS	Yes
	USB OTG HS	Yes
	CAN	2
	SDIO	Yes
Camera interface		Yes
Package		LQFP100

GPIOs	82
12-bit ADC	3
Number of Channels for ADC	24
12-bit DAC	Yes
Number of Channels for ADC	2
CPU frequency	168Mhz
Operating Voltage	1.8 to 3.6V
Operating Ambient Temperatures	-40 to 85°C / -40 to + 105°C
Operating Junction temperature	-40 to +125°C

These features make the STM32F407xx microcontroller family suitable for a wide range of applications:

- Motor drive and application control

- Medical equipment

- Industrial applications: PLC, inverters, circuit breakers

- Printers, and scanners

- Alarm systems, video intercom, and HVAC

- Home audio appliances

III.3 Interrupts, Timers and ADCs

III.3.1 Interrupts

Interrupts are used to get the attention of the microprocessor through an electrical signal called interrupt signal.

The interrupt controller is capable of receiving an interrupt request signal, interpreting it as a justified reason to stop the processor from executing the regular program and actually forcing the processor to jump to and execute a special piece of code called interrupt function. It is the task of the processor to return to the regular program after the execution of the interrupt function and continue its execution.

In STM32F4xx series microcontrollers, the interrupt controller is called Nested Vectored Interrupt Controller. The NVIC controller can handle interrupt requests caused by most hardware like DMA requests, interrupt requests from communication channels USART, CAN, I2C, and timers built into the microcontroller, and also receives interrupt requests from ports through an additional hardware called External interrupt/event Controller (EXTI).

Once the NVIC controller receives the interrupt request it validates it and notifies the processor. The processor then requests the code of the interrupt request source and uses it to find the pointer to the corresponding interrupt function from the interrupt vector table. The table is located in the memory of the microcontroller, and has more than 80 entries for vectors[66].

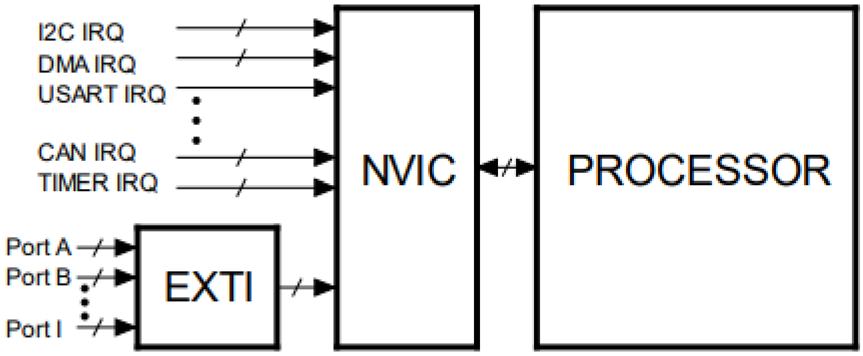


Figure III.3 : Simplified chain of interrupt request processing

III.3.2 Timers

A Timer Module in its most basic form is a digital logic circuit that counts up every clock cycle. More functionalities are implemented in hardware to support the timer module so it can count up or down. It can have a Prescaler to divide the input clock frequency by a selectable value. It can also have circuitry for input capture, PWM signal generation, and much more.

The basic timers consist of a 16-bit auto-reload counter driven by a programmable Prescaler. They may be used as generic timers for time-base generation but they are also specifically used to drive the digital-to-analog converter.

The general-purpose timers consist of a 16-bit auto-reload counter driven by a programmable Prescaler. They may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare and PWM). Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer Prescaler and the RCC clock controller Prescalers. The timers are completely independent and do not share any resources. They can be synchronized together as well.

The advanced-control timers consist of a 16-bit auto-reload counter driven by a programmable Prescaler. They have the same functionalities as general purpose timers but offer more features like complementary outputs, programmable dead time generation, break input and more.

PWM Mode

This mode allows the generation of PWM signals, the timer gets clocked from an internal source and counts up to the auto-reload register value (ARR), then the output channel pin is driven HIGH. It remains low until the timer counts (CNT) reach the CCRx register value, the match event causes the output channel pin to be driven LOW. And it remains until the timer counts up to the auto-reload register value. This is the up-counting PWM mode. The following diagram shows how the signal is generated.

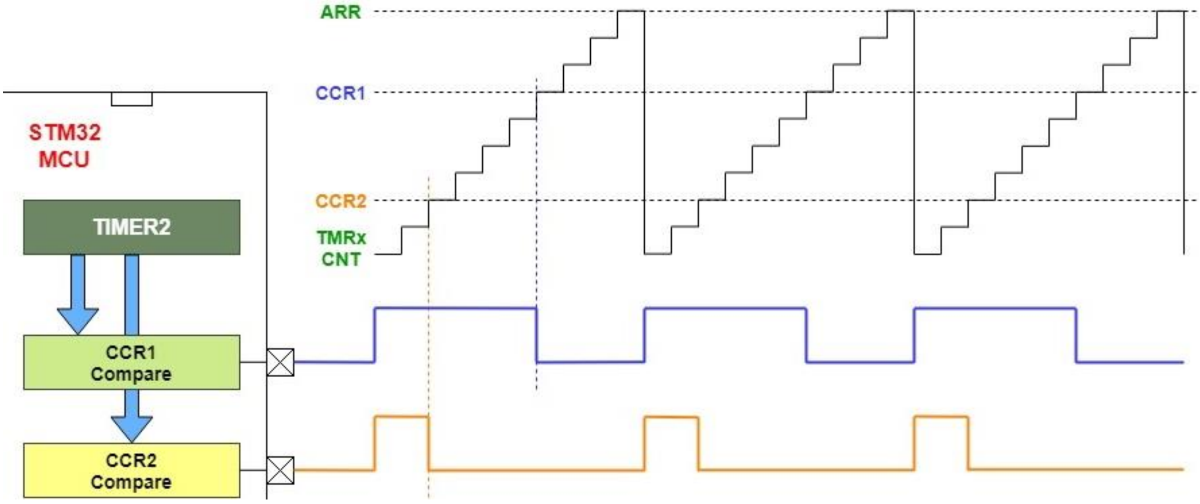


Figure III.4 : Diagram of the generation of a PWM signal in up-counting mode

The PWM period (1/F$_{PWM}$) is defined by the following parameters: ARR value, the Prescaler value, and the internal clock itself which drives the timer module FCLK. The formula down below is used for calculating the F$_{PWM}$ for the output.

$$F_{PWM} = \frac{F_{CLK}}{(ARR + 1) \times (PSC + 1)} \tag{III.1}$$

The duty cycle is controlled by changing the value of the CCRx register. The duty cycle is given by the formula:

$$DutyCycle_{PWM}(\%) = \frac{CCRx}{ARRx} \tag{III.2}$$

In Edge-aligned mode there are two possible configurations: up-counting, and down-counting. In the PWM Mode 1 the output PWM signal is high as long as TIMx_CNT <TIMx_CCRx, the opposite is true for PWM Mode 2.

In center aligned mode, the counter counts up and down cyclically, producing a symmetric PWM signal.

The possible Counter modes for the timer and their description are given in Table III.2.

TABLE III.2 : AVAILABLE COUNTER MODE FOR A TIMER

Counter Mode	Description
Up-counting Mode	The timer counts from zero up to the Period value (which cannot be higher than the timer resolution - 16/32-bit) and then generates an overflow event
Down-counting Mode	The timer counts down from the Period value to zero and then generates an underflow event
Center-aligned 1 Mode	In center-aligned mode, the counter counts from 0 to the Period value – 1, generates an overflow event, then counts from the Period value down to 1 and generates a counter underflow event. Then it restarts counting from 0. The Output compare interrupt flag of channels configured in output mode is set when the counter counts down.
Center-aligned 2 Mode	Same as Center-aligned 1 Mode, but the Output compare interrupt flag of channels configured in output mode is set when the counter counts up.
Center-aligned 3 Mode	Same as Center-aligned 1 Mode, but the Output compare interrupt flag of channels configured in output mode is set when the counter counts up and down

A description of the Timers available on the STM32F4 is given in Appendix D: STM32F4 Peripherals.

III.3.3 ADCs

An ADC (Analog-To-Digital) converter is an electronic circuit that takes an analog voltage as input and converts it into digital data, a value that represents the voltage level in binary code. The ADC samples the analog input whenever it gets trigger to start conversion. And it performs a process called quantization so as to decide on the voltage level and its binary code that gets pushed in the output register.

The ADC needs a stabilization time of t_{STAB} before it starts converting accurately. After the start of ADC conversion and after 14 clock cycles, the EOC flag is set and the 16-bit ADC Data register contains the result of the conversion[67].

Figure III.5 shows the timing of an ADC conversion in respect to the ADC clock.

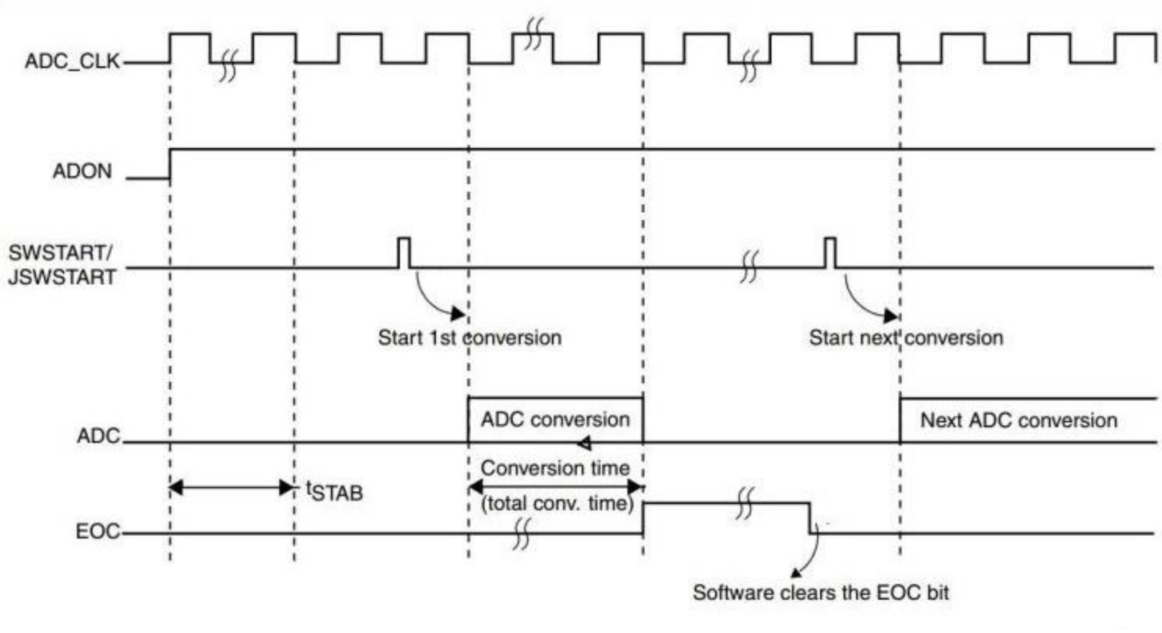


Figure III.5 : Timing diagram of an ADC

The Resolution of an ADC indicates the number of discrete values it can produce over the range of analog values. The Resolution also determines the magnitude of the quantization error and therefore determines the maximum possible average signal-to-noise ratio (SNR) for a typical ADC. The resolution Q depends on both the number of bits n generated by the quantizer and also the FSR (full-scale range) for the analog reference voltage line. Here is the formula to calculate the resolution of quantization Q which is basically a division for the FSR voltage by the number of levels 2^n:

$$Resolution(Q) = \frac{FSR}{2^n} \qquad \text{(III.3)}$$

The quantizer circuit is the working horse of an ADC. Its function is to convert the analog voltage captured by the Sample/Hold circuitry to the digital corresponding value. There are many schemes for creating a quantization circuit which also determines the type of an ADC.

The quantization error is due to the fact that analog signals will always lie between two distinct Q levels, this leads to "quantization errors" that can't be avoided. However, the overall accuracy can be improved by using different techniques.

The Signal to Quantization Noise Ratio (SQNR) is calculated using the following formula:

$$SQNR \cong 1.764 + 6.02n\ [dB] \qquad \text{(III.4)}$$

ADC Modes:

In Single Conversion mode, the ADC does one conversion. This mode is started either by setting the ADON bit in the ADC_CR2 register (for a regular channel only) or by an external trigger (for a regular or injected channel), while the CONT bit is 0.

In continuous conversion mode, ADC starts another conversion as soon as it finishes one. This mode is started either by an external trigger or by setting the ADON bit in the ADC_CR2 register, while the CONT bit is 1.

Scan mode is used to scan a group of analog channels. A single conversion is performed for each channel of the group. After each end of conversion, the next channel of the group is converted automatically. If the CONT bit is set, conversion does not stop at the last selected group channel but continues again from the first selected group channel. When using scan mode, DMA bit must be set and the direct memory access controller is used to transfer the converted data of regular group channels to SRAM after each update of the ADC_DR register. The injected channel converted data is always stored in the ADC_JDRx registers.

Discontinuous mode is enabled by setting the DISCEN bit in the ADC_CR1 register. It can be used to convert a short sequence of n conversions (n <=8) which is a part of the sequence of conversions selected in the ADC_SQRx registers. The value of n is specified by writing to the DISCNUM[2:0] bits in the ADC_CR1 register. When an external trigger occurs, it starts the next n conversions selected in the ADC_SQRx registers until all the conversions in the sequence are done. The total sequence length is defined by the L[3:0] bits in the ADC_SQR1 register.

III.4 IDE (STM32 Cube IDE)

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. It is based on the ECLIPSE/CDT framework and GCC toolchain for the development, and GDB for the debugging. It integrates all STM32CubeMX functionalities

After the selection of an empty STM32 MCU or MPU, or preconfigured microcontroller or microprocessor from the selection of a board, the project is created and initialization code generated. At any time during the development, the user can return to the initialization and configuration of the peripherals or middleware and regenerate the initialization code with no impact on the user code [68].

III.4.1STM32CubeIDE Interface

Information Center

The information center is displayed the first time the software is started (Figure III.6). It enables the user to quickly reach information about the product and how to use it. By clicking on the corresponding hypertext links, manuals are opened and latest information is obtained from the STMicroelectronics website. It can be opened at any time via the [Help]>[Information Center] command. When the Information Center tab is closed, the standard views in the C/C++ perspective is opened. [69]

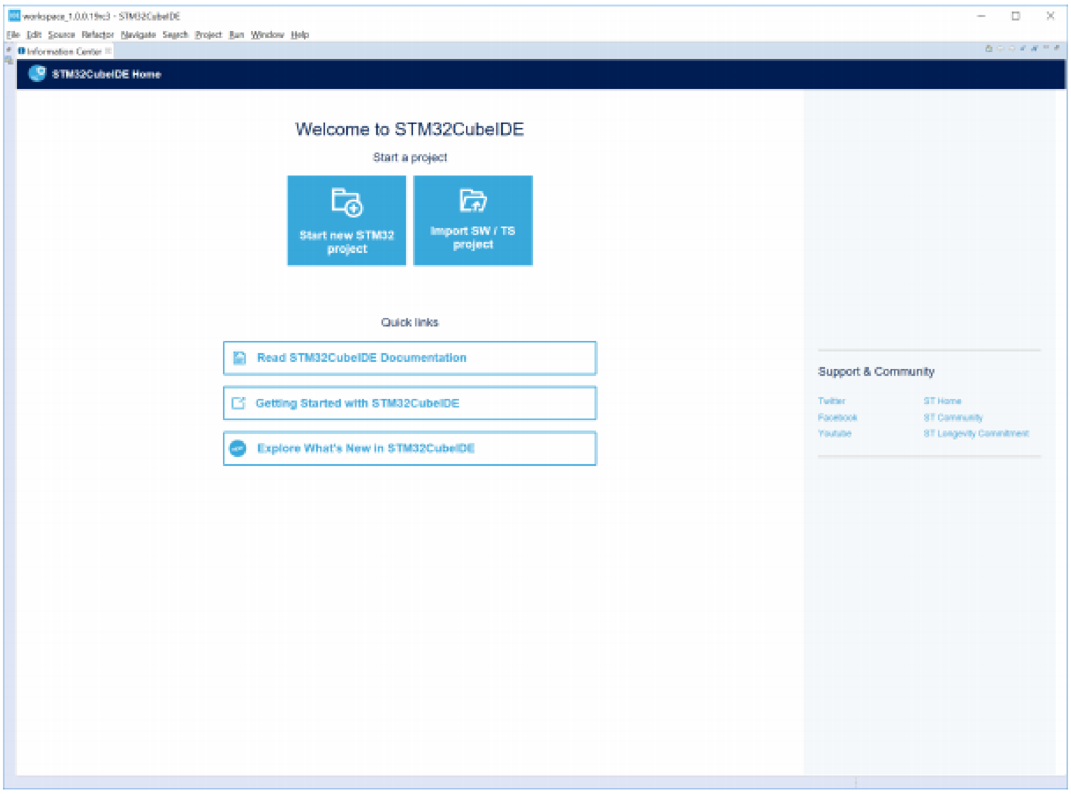
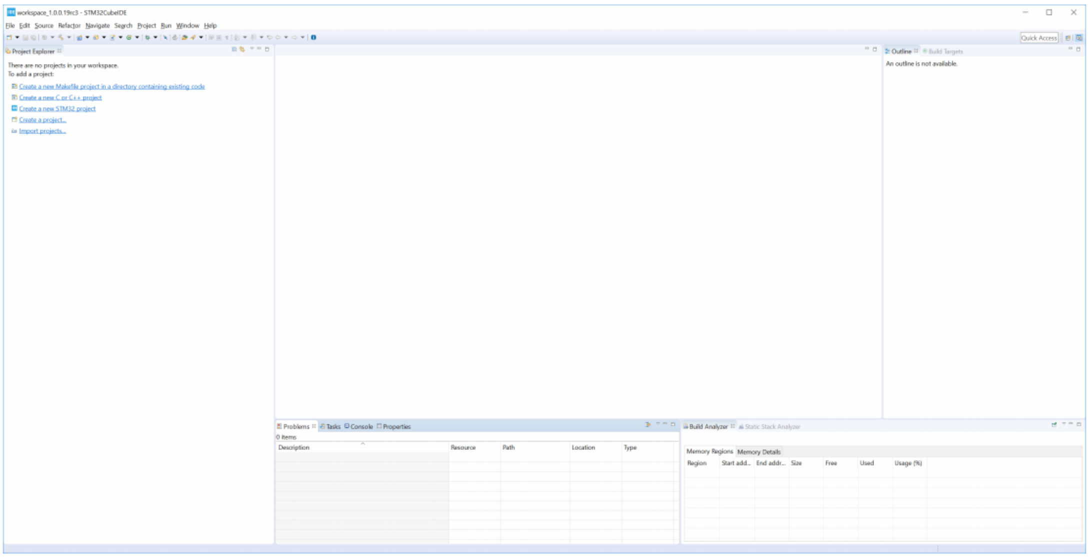


Figure III.6 : STM32CubeIDE Information Center

Workspace and Projects

STM32CubeIDE is built using the ECLIPS framework and inherits therefore many of its characteristics. The ECLIPSE editing environment uses perspectives. A perspective is a set of

windows dedicated to a purpose. The C/C++ perspective (as shown in Figure III.7) is dedicated to writing and editing code, and navigating through projects.



There are no projects in your workspace.
To add a project:

- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a new STM32 project
- Create a project...
- Import projects...

An outline is not available.

0 items

Figure III.7 : STM32CubeIDE with thee C/C++ perspective open with no project opened

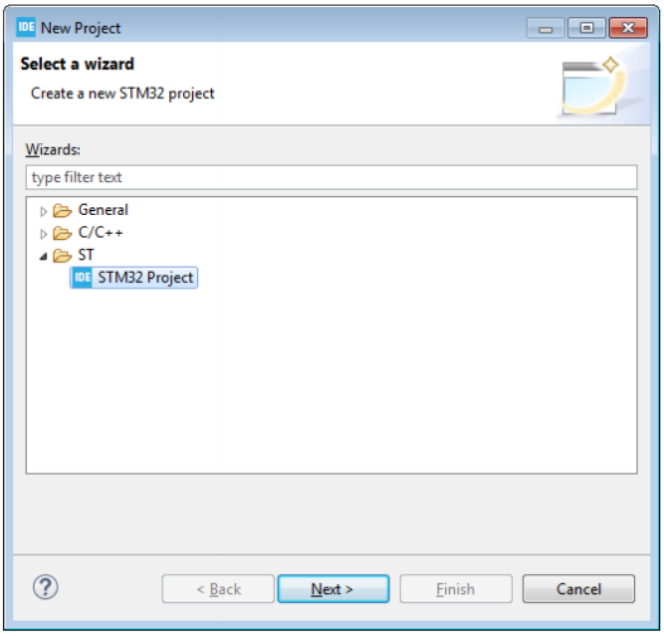
A workspace is a container that includes project folders or information about project folders, and a metadata folder that contains information about the projects. A workspace is simply a folder on a hard drive, which can be located anywhere in the storage media. When STM32CubeIDE starts up, it asks which workspace must be used. This may be changed at any time by selecting [File]>[Switch Workspace] and navigating to another folder.

Creating a new project: One way to create a new project is to use the STM32 Project wizard (Figure III.8). It is selected through the [File]>[New]>[STM32 Project] menu command, and launches the embedded MCUFinder. These steps must then be followed:

1- Select the target MCU or board and go to the next page

2- Enter a project name and select the setting wanted for the project in the dialogue boxes

3- Click on [Finish]. This brings up the STM32CubeMX window for configuring the peripherals, clock, middleware, and the power consumption.



New Project

Select a wizard

Create a new STM32 project

Wizards:

type filter text

Figure III.8 : STM32 Project Wizard

Figure III.9 shows the main icons on the top ribbon that are used to run and manage projects. Icon 1 is used to create new C source code module, header files, or new object such as project, library, or repository.

Icon 2 is used to build the project.

Icon 3 is used to launch a specific debug configuration, or configure a debug configuration by clicking on the arrow.

Icon 4 launches various search utilities.

The arrows in 5 navigate among recently visited places in the project.

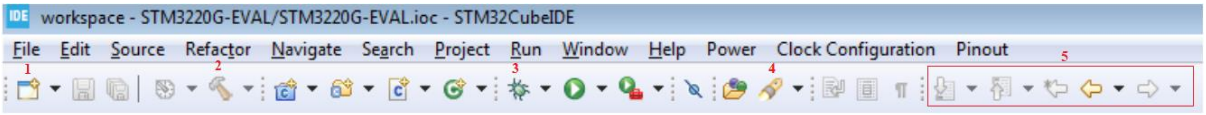


Figure III.9 : Main icons for editing, running, debugging and project management.

Debugging

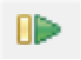
Once the project is built without errors, use the icon 3 from Figure III.9 to launch the program code into the part together with the debugger. The first time a debug session is launched, STM32CubeIDE builds the project and then shows the debug launch configuration menu. This gives the user the opportunity to verify settings and make changes if desired.

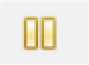
When ready to launch the debug session, click on [OK]. Several things happen including the launch of the debugger driver and GDB server, as well as the programming of the application into the part, and any other actions called for in the startup script. At this point, the application is normally halted at the first line of main().

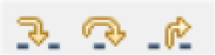
To set a breakpoint, click on the blue horizontal bar next to the statement line number.

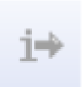
To view memory locations, use the Variables (variables in focus or global variables), Memory, or Expressions window

These are the main debugging icons:

 Resumes full speed execution of the application on the target.

 Halts execution.

 Step into a function, over a function, or out of a function.

 Toggle between C and instruction stepping.

↺ Reset the chip and restart execution.

■ Terminate the debug session

III.5 Conclusion

In this chapter we have introduced the primary peripherals that the STM32F4 features, namely the interrupts, Timers and ADCs. These peripherals will be used to implement the motor control algorithms.

The STM32F4 Discovery is a good development board for embedded projects in motor control as it features an FPU, advanced timers with a maximum frequency of 168MHz, motor control features such as dead-time insertion and complementary PWM outputs, 12bit ADCs... etc.

CHAPTER IV

CONTROL IMPLEMENTATION ON THE STM32F4

IV CONTROL IMPLEMENTATION ON THE STM32F4

IV.1 Introduction

In this chapter the constant V/f and DTC control algorithms will be implemented on the STM32f4 MCU. A description of the programs will be given and the configuration of the peripherals used explained.

A Hardware-in-the-loop simulation is used to test the control algorithms while running on the microcontroller, the motor and converter models run on Simulink. This allows testing the controller with different types of loads easily and to monitor all the motor signals (flux, torque ...etc.) without the need for expensive hardware.

IV.2 Structure of the program

IV.2.1 Programming tools

Before we can start developing applications for the STM32 platform, we need a complete tool-chain. A tool-chain is a set of programs, compilers and tools that allow us to write our code and to navigate inside the source files of our application, to navigate inside the application code, to compile the source code using a cross-platform compiler and to upload and debug our application on the target development board.

ST's IDE: STM32CubeIDE includes all these tools, it is based an Eclipse/GCC tool-chain. This IDE abstracts from us most of these steps, but all the configurations and configuration files are accessible and modifiable making it suitable for beginners and advanced programmers. This tool-chain is described in Figure IV.1.

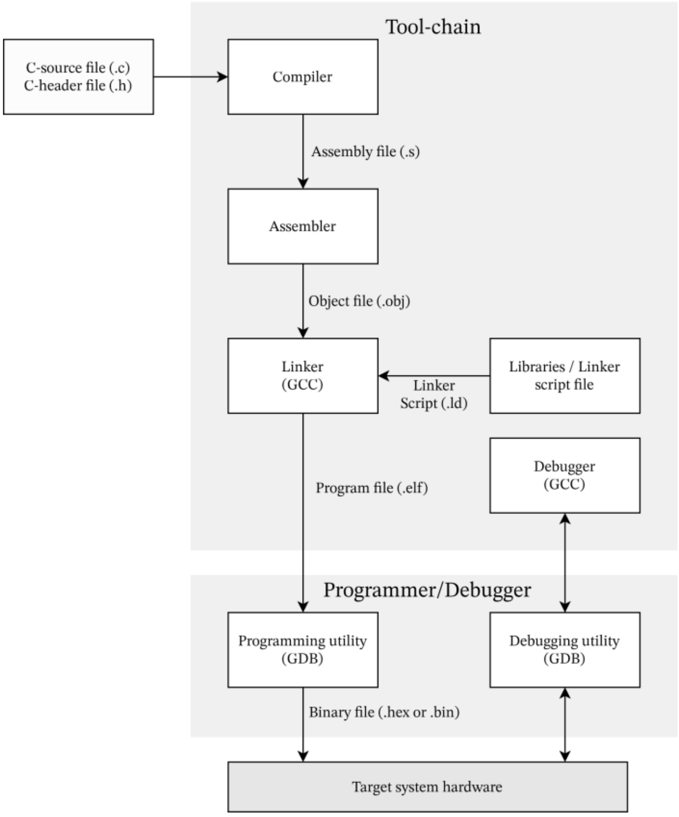


Figure IV.1 : Tool-chain diagram for programming the STM32 using GCC

Eclipse is an Open Source and a free Java based IDE. It is one of the most widespread and complete development environments. GCC (GNU Compiler Collection) is a complete compiler suite. It is the only development tool able to compile several programming languages (front-end) to tens of hardware architectures that come in several variants. It includes in addition to the compiler a linker, a debugger known as GDB (GNU Debugger), several tools for binary files inspection, disassembly and optimization.

The onboard ST-link on the STM32F4-Discovery board allows us to upload and debug our code.

IV.2.2 Structure of the implemented control algorithms

The measured signals (Speed, voltage, currents) are fed to our program after being converted through ADCs and a timer with two channels configured as encoder input. These measures are used by the main program to calculate the duty cycles to be written on the output compare registers of the timer to generate three phase PWM signals with their complementary outputs. This program is called after the acquisition of the measurements by an interrupt.

The ADCs should have the same sampling frequency as the switching frequency.

IV.2.2.1 Implementation of Constant V/f

Figure IV.2 shows a flow chart of the algorithm that was implemented with the C language on the STM32F4.

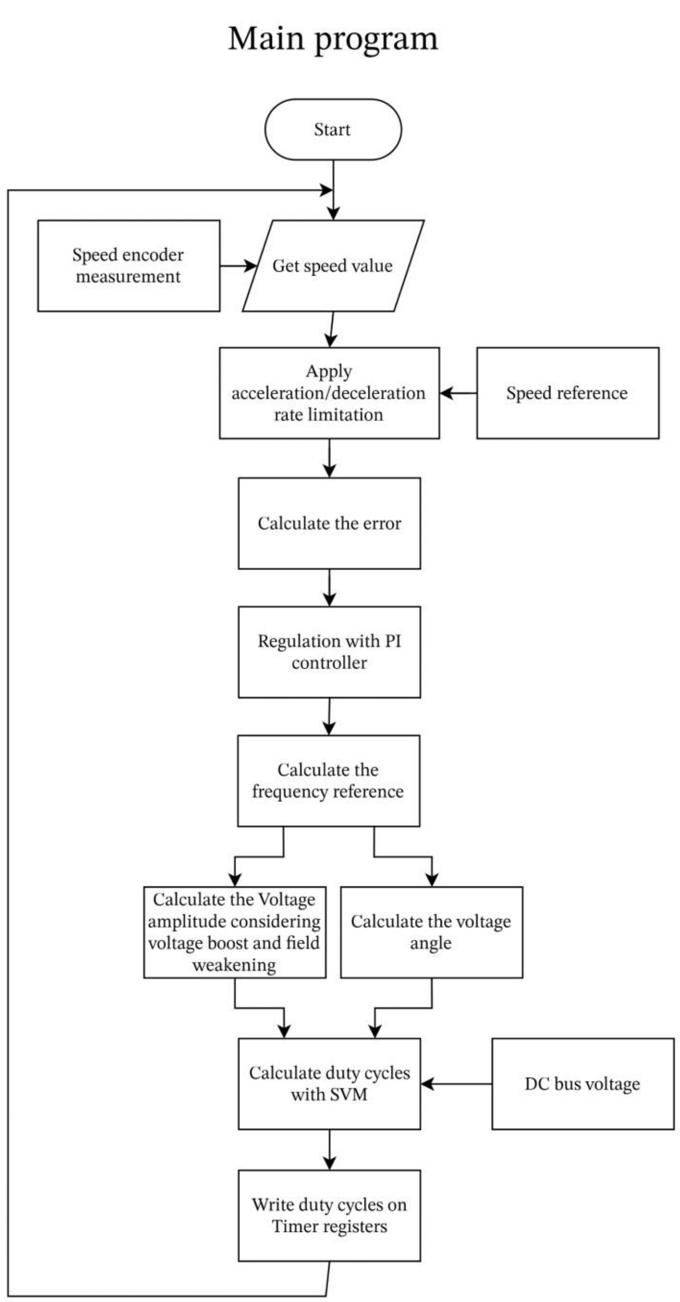

Main program
Start
Speed encoder measurement
Get speed value
Apply acceleration/deceleration rate limitation
Speed reference
Calculate the error
Regulation with PI controller
Calculate the frequency reference
Calculate the Voltage amplitude considering voltage boost and field weakening
Calculate the voltage angle
Calculate duty cycles with SVM
DC bus voltage
Write duty cycles on Timer registers
Measurement subroutine
Start
Calculate measured speed through a timer configured as encoder input
Measure DC bus voltage
Return

Figure IV.2 : Flow chart of the implemented constant V/f algorithm

The main program gets the measures speed and dc voltage from the ADC and encoder, it then applies an acceleration/deceleration rate limitation that considers the current speed and the speed reference to generate a new limited speed reference. This helps ramp up the motor more smoothly and controllably and helps avoiding motor stall. The PI controller than uses trapezoidal integration of the error on speed and stores the parameters and values on a struct which is then added to the motor speed to perform the regulation. The voltage angle is a simple integration of the frequency regulated by the PI controllers and the voltage amplitude is the frequency multiplied by a gain (V/f = constant) increased by a voltage boost in low frequencies and limited for high speed operation. The duty cycles are then calculated by an SVM algorithm. The duty cycles calculated by the SVM are directly written to the timer registers.

IV.2.2.2 Implementation of DTC

Figure IV.3 shows a flow chart of the DTC algorithm implemented on the STM32.

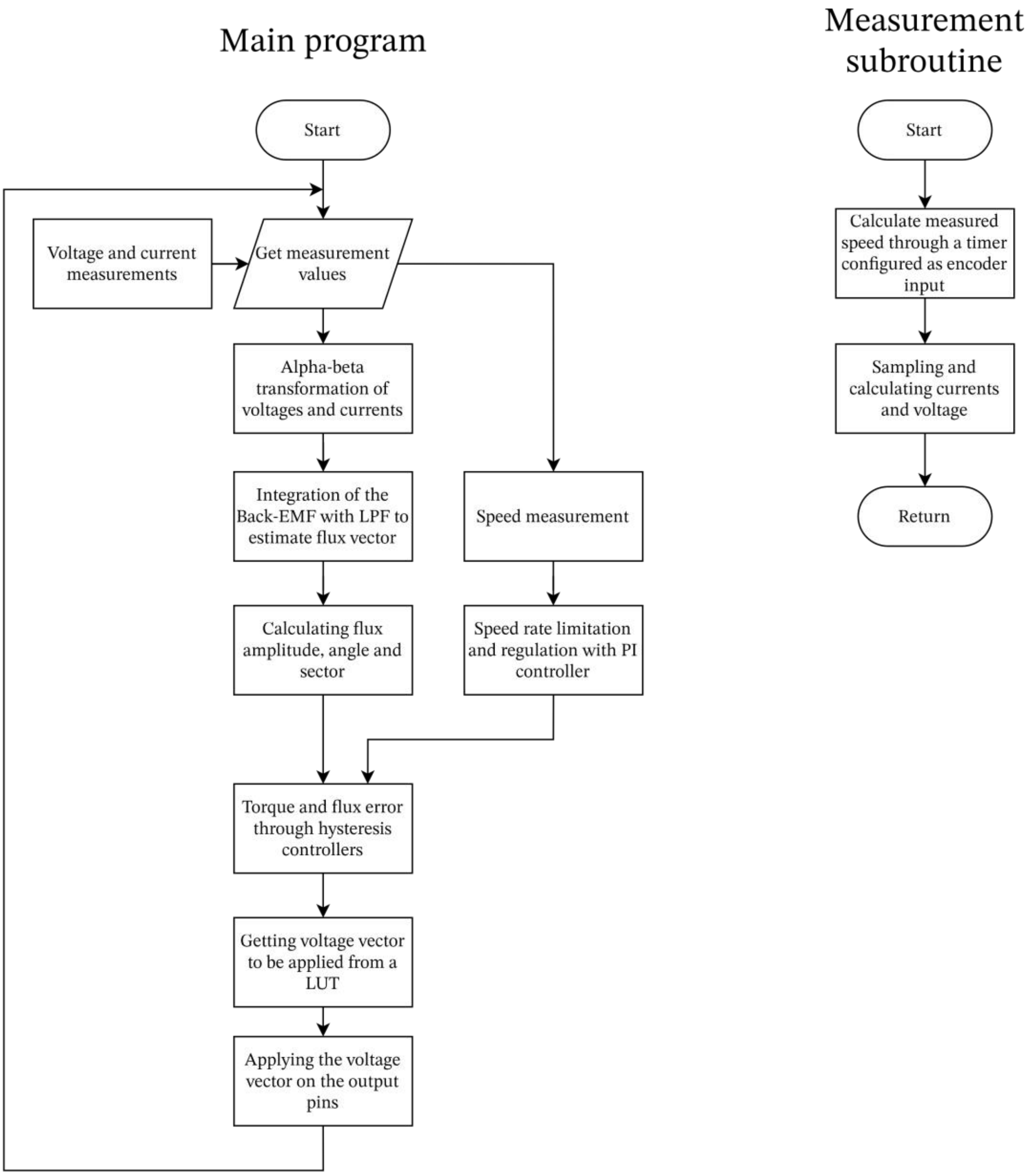

Main program
Measurement subroutine
Start
Voltage and current measurements
Get measurement values
Alpha-beta transformation of voltages and currents
Integration of the Back-EMF with LPF to estimate flux vector
Speed measurement
Calculating flux amplitude, angle and sector
Speed rate limitation and regulation with PI controller
Torque and flux error through hysteresis controllers
Getting voltage vector to be applied from a LUT
Applying the voltage vector on the output pins
Start
Calculate measured speed through a timer configured as encoder input
Sampling and calculating currents and voltage
Return

Figure IV.3 : Flow chart of the implemented DTC algorithm

The main program gets the measured values of currents and DC voltage from the ADCs and the speed, the latter is regulated through a PI controller to get a torque reference. The speed PI controller avoids the saturation of the integrator by limiting the integration interval on a limited band of error, operating only when the error is small, this leads to the integrator functioning solely as a steady-state error compensator. This is the simplest yet effective anti-windup method. Currents and DC voltage are used to calculate the voltage and current vector in the α,β reference frame, the back-EMF is then calculated using equation (II.24II.24. A low pass filter for calculating flux is necessary as noise in the sampled voltage and current signals makes the estimated flux drifts from the real value over time. The version implemented also features a compensation of the flux estimate since a good estimation of flux leads to a better and more roust control of both flux and torque. The hysteresis controllers and look-up table are then used to get the voltage vector to apply. The voltage vector application works by applying a 0% or 100% duty cycle to the timer. This limits to switching frequency of the DTC to the timer frequency of 10 kHz, leading to the voltage vector being applied for 100μs before the next cycle begins.

IV.3 Initialization and configuration of the peripherals

IV.3.1 Clock tree

The clock configuration is left at the default values. It is configured in CubeMX from the clock window, as shown in Figure IV.4.

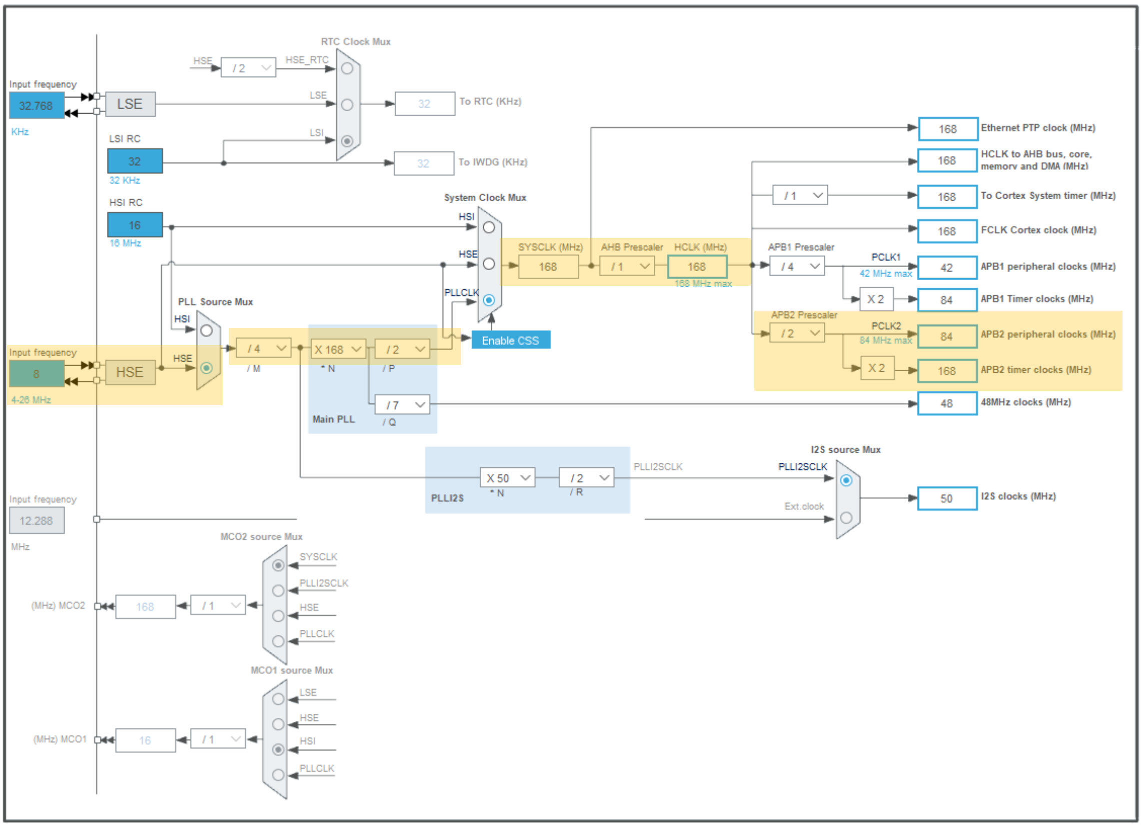


Figure IV.4 : Clock configuration tree for the project

It is imperative to have a good understanding of how the clocking works, as only some values are permissible for each field. CubeMX generates an appropriate clock configuration whenever a field is modified, which greatly simplifies our work. The HSE (High Speed External) is the crystal oscillator available on the discovery board. It is used to generate the more precise 168MHz clock frequency, instead of using the internal (HIS) RC oscillator. By configuring the appropriate multipliers and prescalers we can get the maximum frequency for our timer TIM1 and ADC of 168MHz on APB2. These setting are highlighted on Figure IV.4.

IV.3.2 Timers

The TIM1 timer of the stm32f4 is an advanced timer that features three PWM channels with complementary output, dead time and break input.

The default clock configuration for the STM32F4-discovery board in cubeIDE sets the clock speed for TIM1 timer to 168 MHz (APB1 clock) when the clock source is set to Internal, it can be changed by going to the clock configuration panel.

We set the counter period (Auto-reload register) to 16799, and the Prescaler to 0, therefore the switching frequency will be equal to:

$$F_s = \frac{168\ 000\ 000}{(16799 + 1) \times (0 + 1)} = 10\ 000\ Hz \tag{IV.1}$$

The counter Mode is set to Center Aligned mode3 so that the output compare interrupt flag is set when the counter counts up and down, and PWM mode is set to PWM mode 1.

An interrupt is enabled to trigger the ADC conversion.

The break input allows us to stop the timer when a fault is detected.

To configure the deadtime we need to set a value of 8 bits (between 0 and 255). Calculating the dead time according to this settings value is as follows:

DTG is the value in 8 bits, with DTG[7] being the most significant bit and DTG[0] the least significant bit. TCK_INT is the internal clock timer, in our case APB1 is 168 MHz, therefore, TCK_INT = 1/168M = 5.952 ns.

If DTG[7] = 0 : Deadtime = DTG * TCK_INT

Else If DTG[6] = 0 : Deadtime = (64 + DTG[5:0]) * 2 * TCK_INT

Else If DTG[5] = 0 : Deadtime = (32 + DTG[4:0]) * 8 * TCK_INT

Else If DTG[7:5] = 111 : Deadtime = (32 + DTG[4:0]) * 16 * TCK_IN

For a deadtime of 2 μs the setting should give a deadtime of 336*TCK_IN. This would fall in third option. (32+DTG[4:0])* 8 = 336 means that DTG[4:0] = $(10)_{10}$, in binary DTG[4:0] = $(1010)_{2}$. By adding the first 3 bits DTG=$(11001010)_{2}$, giving us a value of 202 for the setting.

To start the PWM output we need to add a few more lines of code:

```
HAL_TIM_Base_Start_IT(&htim1);
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
HAL_TIMEx_PWMN_Start(&htim1, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2);
HAL_TIMEx_PWMN_Start(&htim1, TIM_CHANNEL_2);
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3);
HAL_TIMEx_PWMN_Start(&htim1, TIM_CHANNEL_3);
```

IV.3.3 ADCs

Configuring the ADC is as straightforward as configuring the timer in CubeIDE.

For our project we will use three channels of ADC1 offering us a 12bit resolution.

The clock prescaler is set to the maximum admissible value of 4 and the resolution to 12bits.

The conversion will be triggered by the TIM1 timer to set its conversion frequency to 10kHz, the conversion will in turn trigger our main program to start the calculations by generating an interrupt.

For scalar control one ADC channel is used for the DC voltage, for DTC three channels are used, one for DC voltage and two for the currents of two phases.

IV.3.4 Encoder

The TIM3 timer is set to encoder mode on channel 3 and channel 4, prescaler is set to 0 and counter period to 16799. The encoder configuration depends on the encoder used.

IV.4 Test results

IV.4.1 HIL Simulation

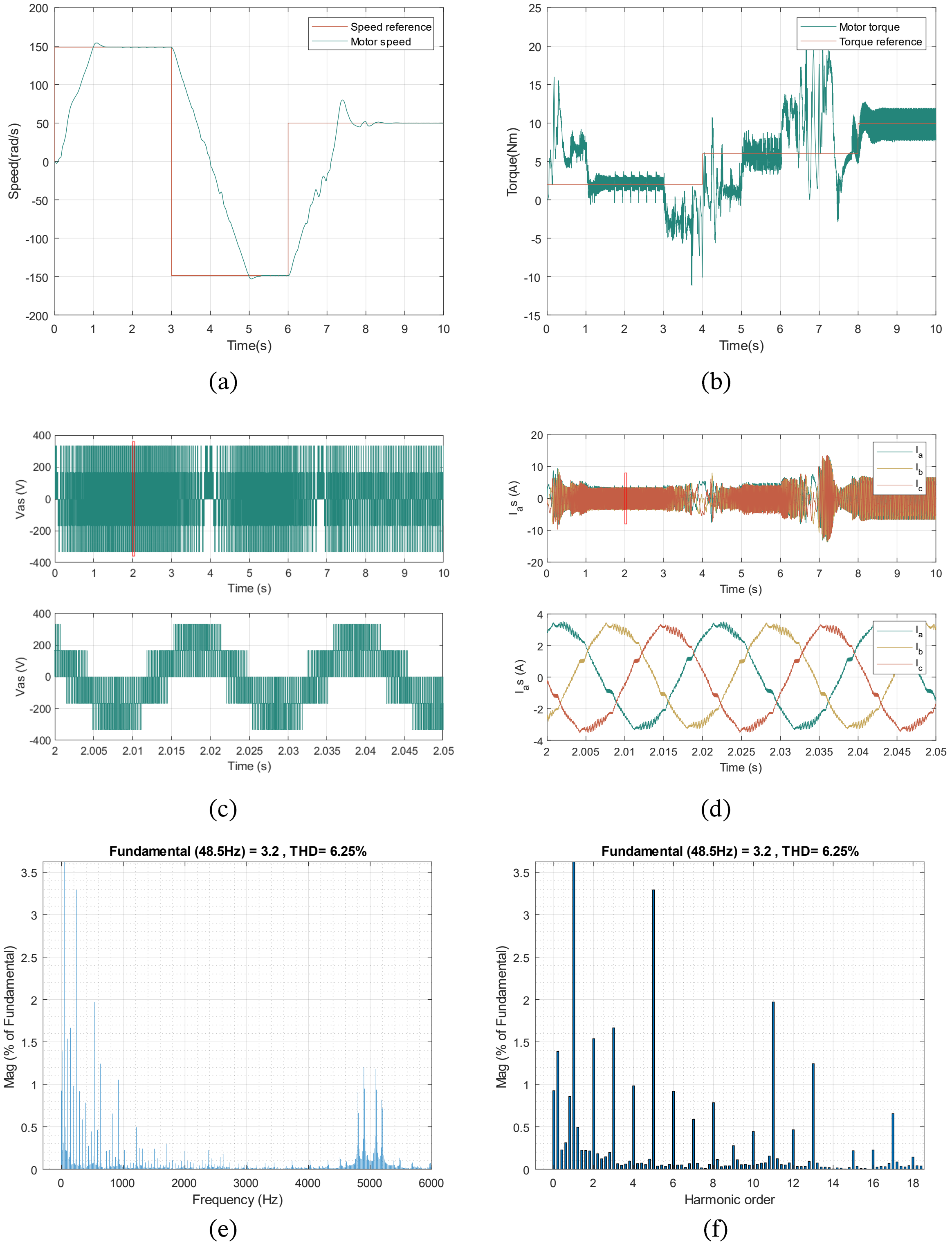
Hardware-in-the-loop (HIL) simulation is a technique that is used in the development and test of complex real-time embedded systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform. The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems. In this work the plant model includes the model of the DC voltage source, the voltage-source-inverter and the induction motor. The controller is the V/f constant and DTC algorithms that were implemented. The Simulation should be running in real time, but due to limitations in the speed of the USB interface, the simulation had to be slowed down.

The implemented algorithms were tested with MATLAB/Simulink by using the USB-CDC interface, the procedure is described in Appendix C: Testing STM32 Program with MATLAB/Simulink.

The motor parameters used for the simulation are given in Appendix A: Induction motor parameters. The acceleration rate is set to 150 rad/s.

Three types of load where simulated : screw compressors conveyors and feeders with a constant load, mixers with a load torque propotional the motor speed and fans and pumps with a load torque proportional to the square of the speed of the motor.

IV.4.2 Constant V/f control

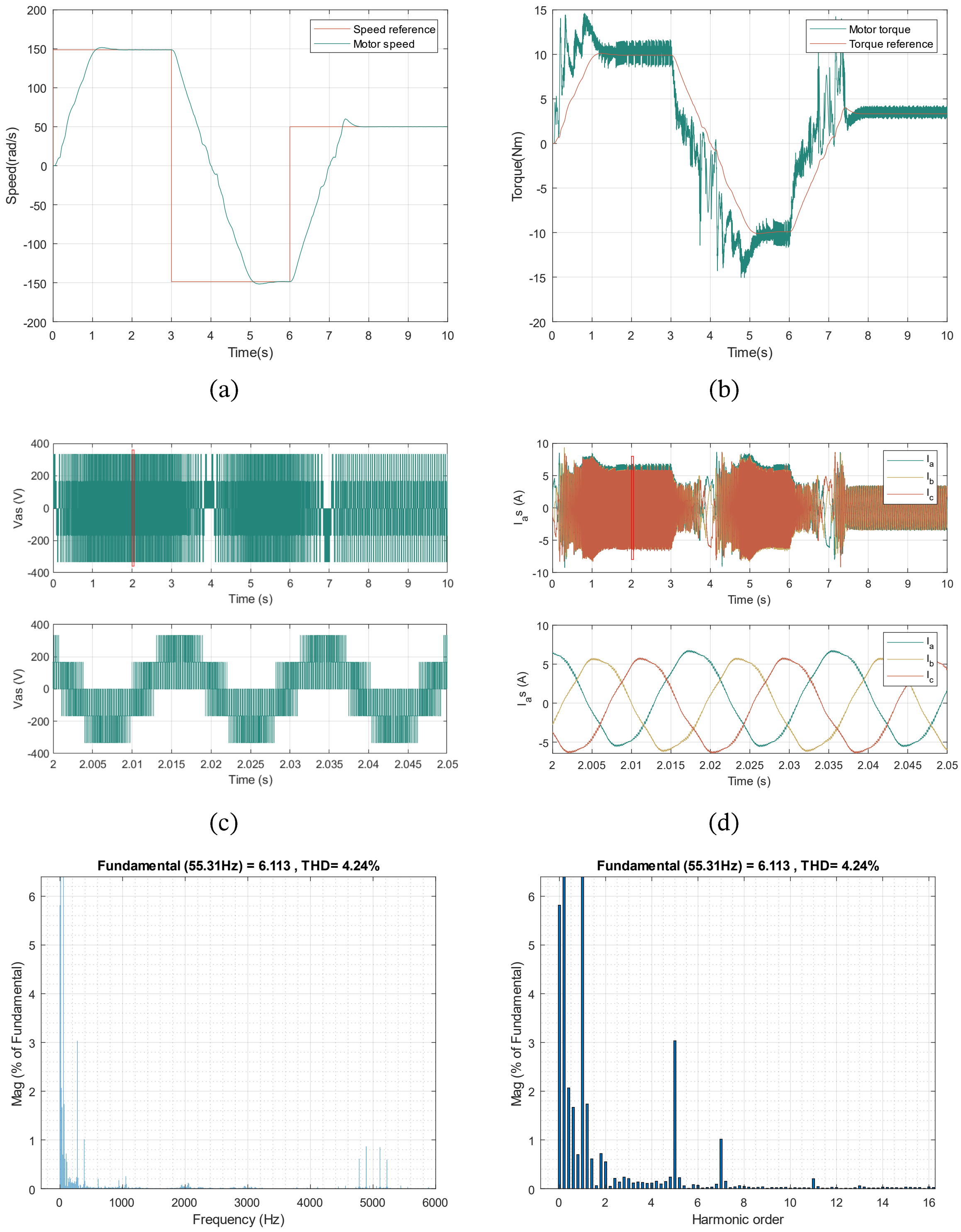


(a) (b)

(c) (d)

(e) (f)

Figure IV.5 : Test results for the implemented V/f constant control for a **constant load**; (a) Speed reference, motor speed, (b) Electromagnetic torque developed by the motor and load torque, (c) stator voltage for one phase, (d) stator current for all three phases, (e) current harmonics spectrum, (f) low order current harmonics

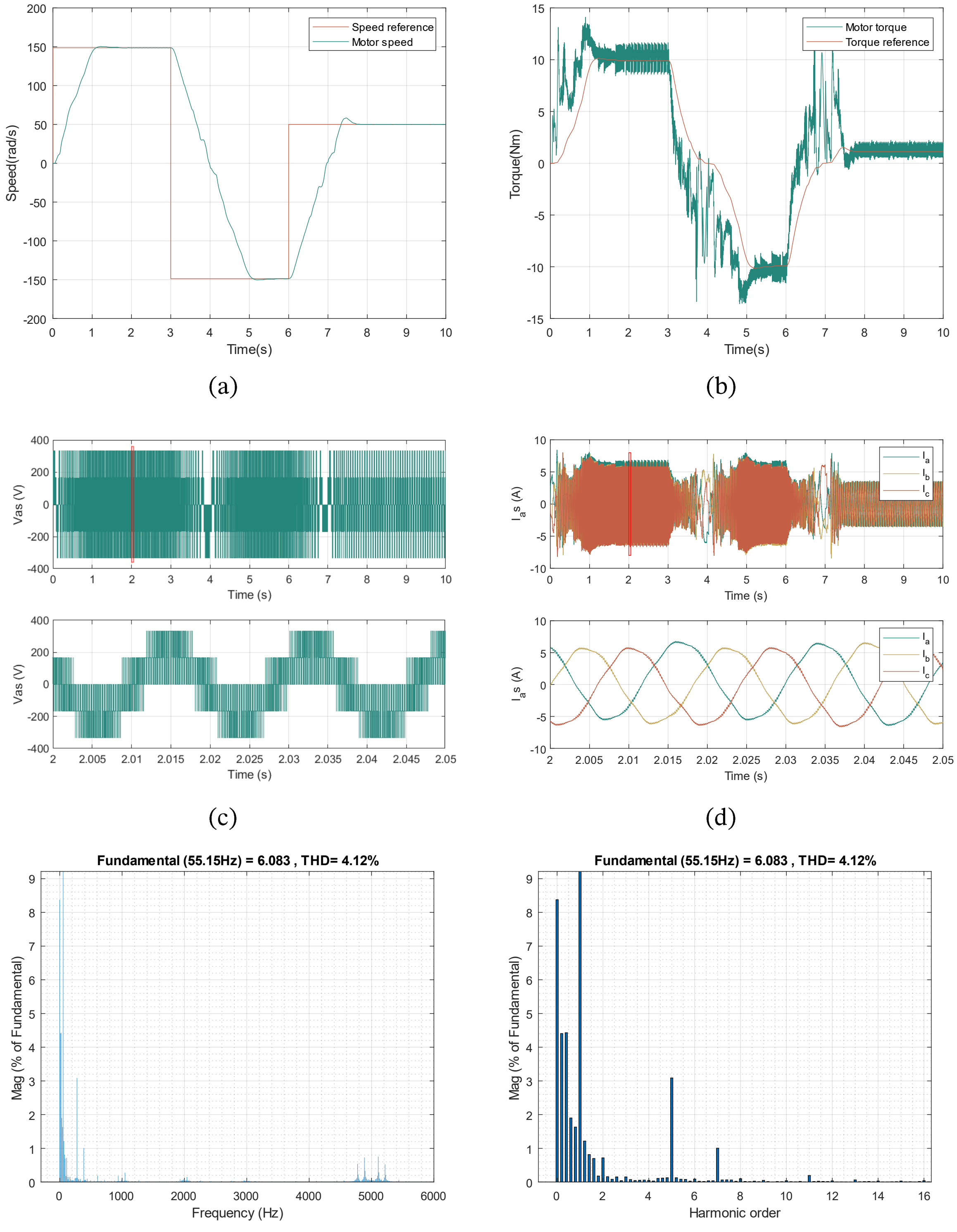


(a) (b)

(c) (d)

(e) (f)

Figure IV.6 : Test results for the implemented V/f constant control for a **load torque proportional to speed**; (a) Speed reference, motor speed, (b) Electromagnetic torque developed by the motor and load torque, (c) stator voltage for one phase, (d) stator current for all three phases, (e) current harmonics spectrum, (f) low order current harmonics



(a)

(b)

(c)

(d)

(e)

(f)

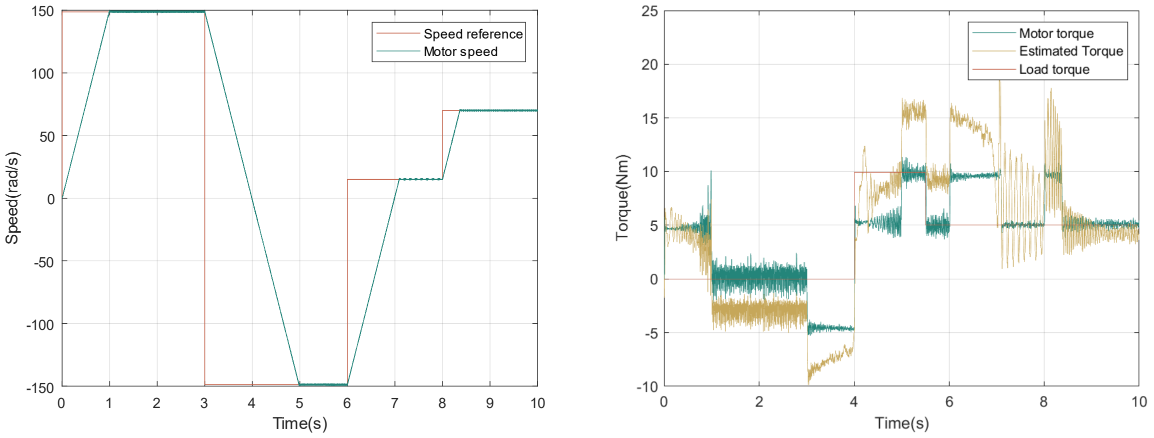
Figure IV.7 : Test results for the implemented V/f constant control for a **load propotional to the square of the motor speed**; (a) Speed reference, motor speed, (b) Electromagnetic torque developed by the motor and load torque, (c) stator voltage for one phase, (d) stator current for all three phases, (e) current harmonics spectrum, (f) low order current harmonics

Figures IV.5, IV.6 and IV.7 show that he speed is well adjusted to the reference in all cases, and the acceleration rate is limited to 150rad/s^2. We notice however more torque ripple with the implemented system comapred to the simulation, as well as more current distortion with a slightly higher THD of 4% to 6% and a higher DC component in currents, this is due to the errors and lag introduced by the HIL simulation method when the signals are being sampled, quantized and sent to the controller. The implementation shows overall similar results to the simulation with a small startup inrush current, speed following the voltage amplitude and frequency and torque following currents.

IV.4.3 DTC

The same parameters used for testing constant V/f control were used for DTC. The flux reference was set to 0.8 Wb, the cutoff frequency for the LPF filter set to 70Hz, and Butterworth low-pass filters were added for the input signals (currents and voltage). A PI controller for the mechanical speed regulation was added with an integrator wrap-up as anti-windup method.

Figure IV.8 shows the test results for a constant load, Figure IV.9 for a load proportional to speed and Figure IV.10 for a load proportional to the square of speed.



(a) (b)

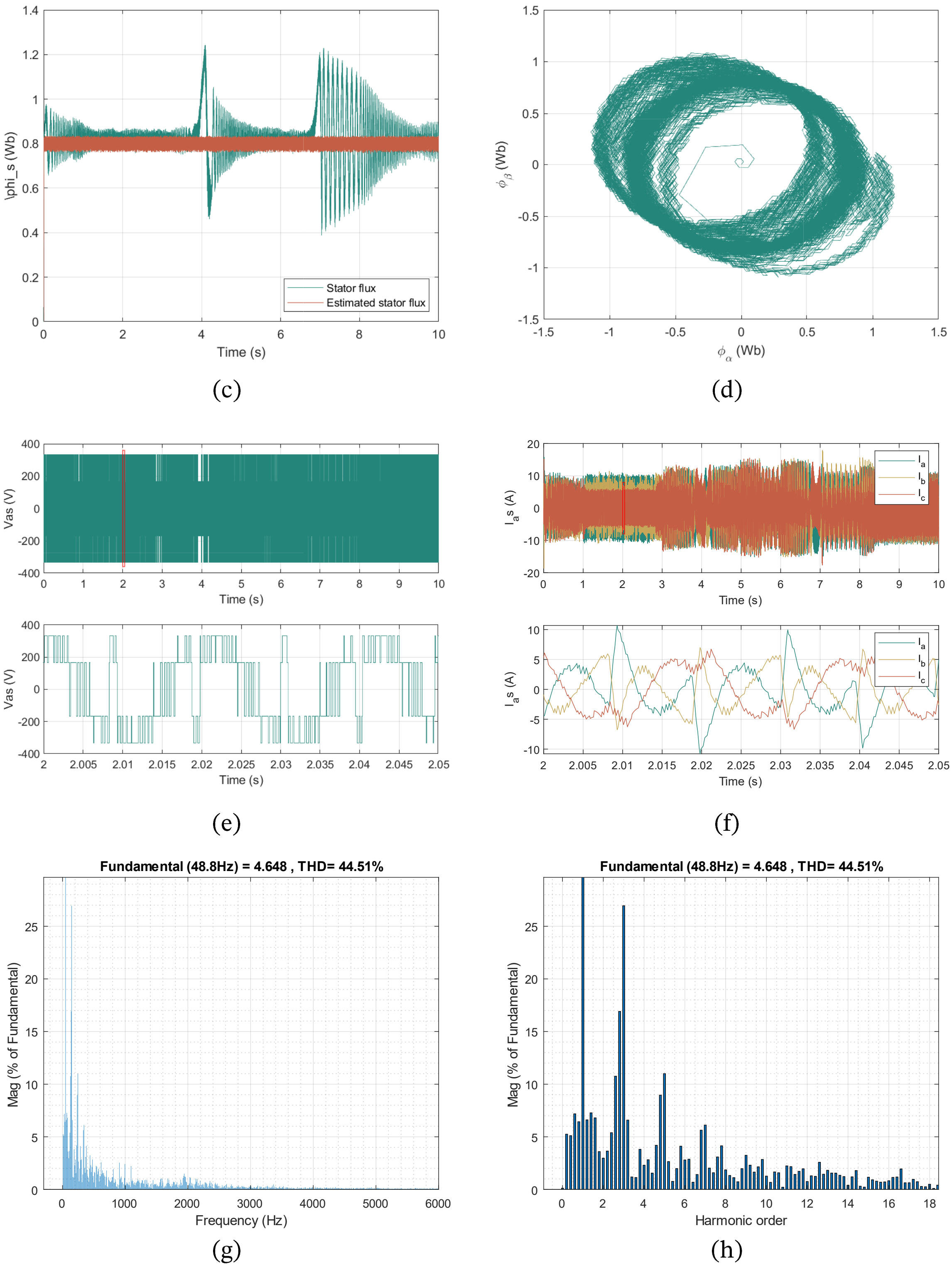
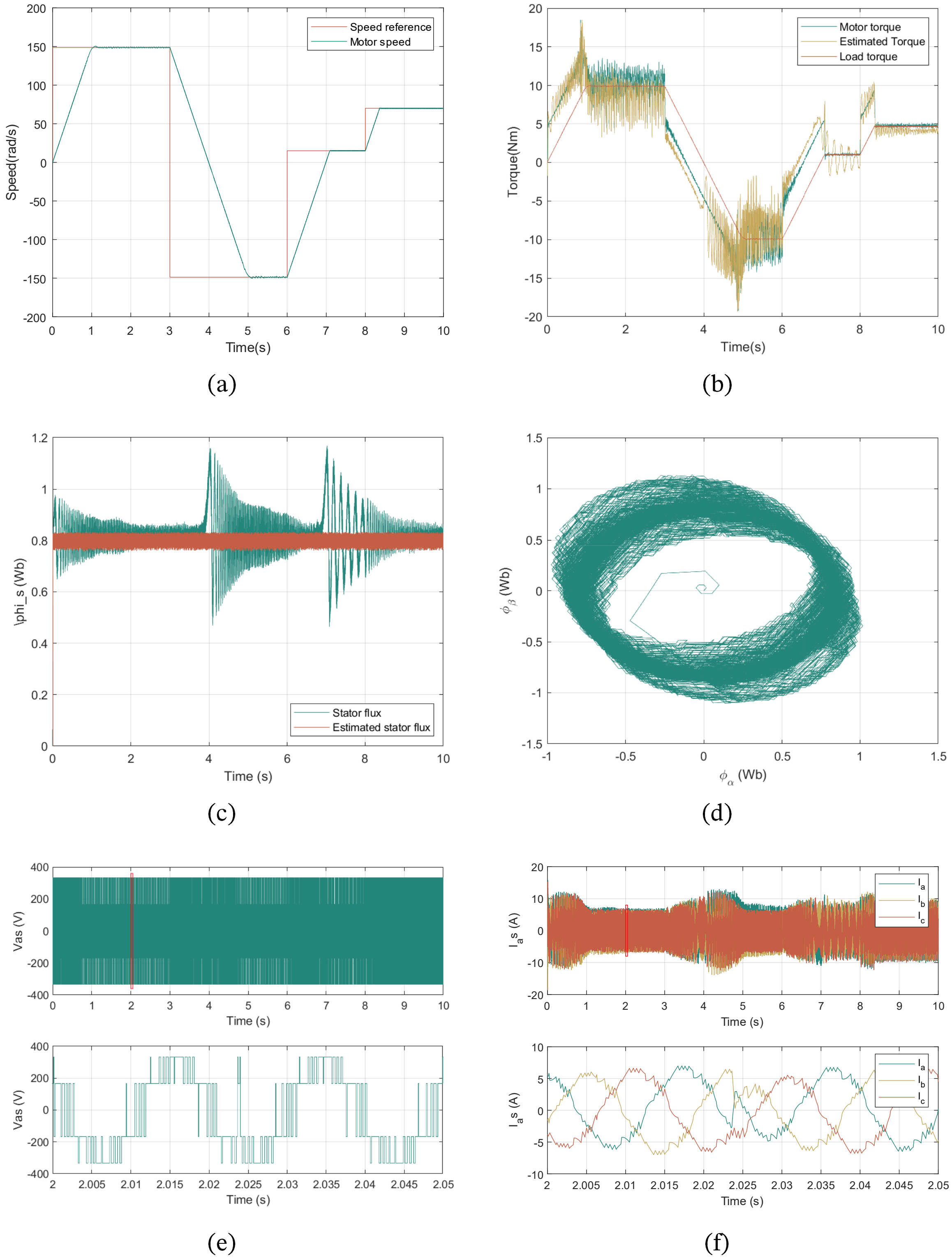


Figure IV.8 : Test results for the implemented DTC with a **constant load torque**;(a) motor speed and it's reference, (b) motor torque, the estimated torque and the load torque, (c) amplitude of the stator flux vector and the estimated flux, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics


Speed(rad/s)
Time(s)
Speed reference
Motor speed
200
150
100
50
0
-50
-100
-150
-200
0 1 2 3 4 5 6 7 8 9 10
Torque(Nm)
Time(s)
Motor torque
Estimated Torque
Load torque
20
15
10
5
0
-5
-10
-15
-20
0 2 4 6 8 10
\phi_s (Wb)
Time (s)
Stator flux
Estimated stator flux
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10
ϕ_β (Wb)
ϕ_α (Wb)
1.5
1
0.5
0
-0.5
-1
-1.5
-1 -0.5 0 0.5 1 1.5
Vas (V)
Time (s)
400
200
0
-200
-400
0 1 2 3 4 5 6 7 8 9 10
Vas (V)
Time (s)
2 2.005 2.01 2.015 2.02 2.025 2.03 2.035 2.04 2.045 2.05
$I_a s$ (A)
Time (s)
I_a
I_b
I_c
20
10
0
-10
-20
0 1 2 3 4 5 6 7 8 9 10
$I_a s$ (A)
Time (s)
10
5
0
-5
-10
2 2.005 2.01 2.015 2.02 2.025 2.03 2.035 2.04 2.045 2.05

(a)

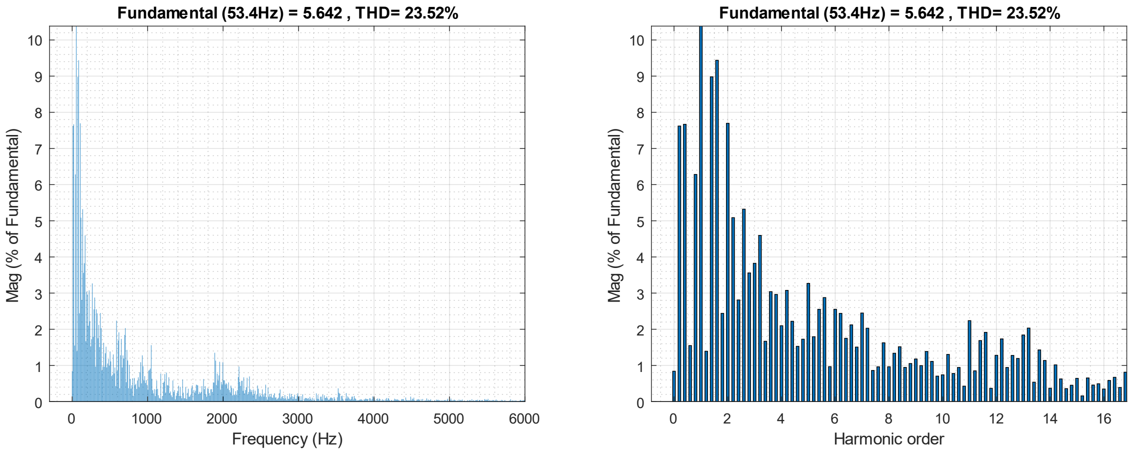
(b)

(c)

(d)

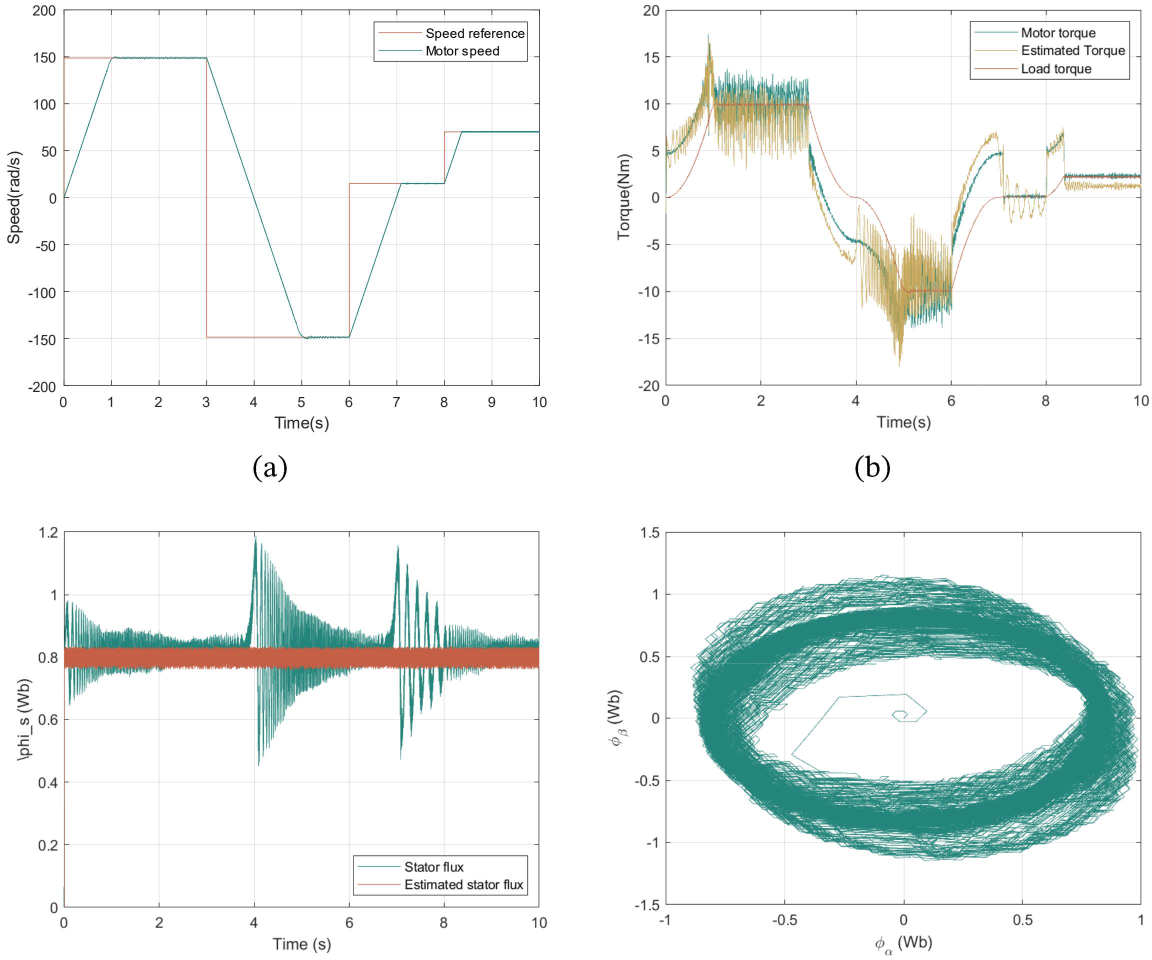
(e)

(f)



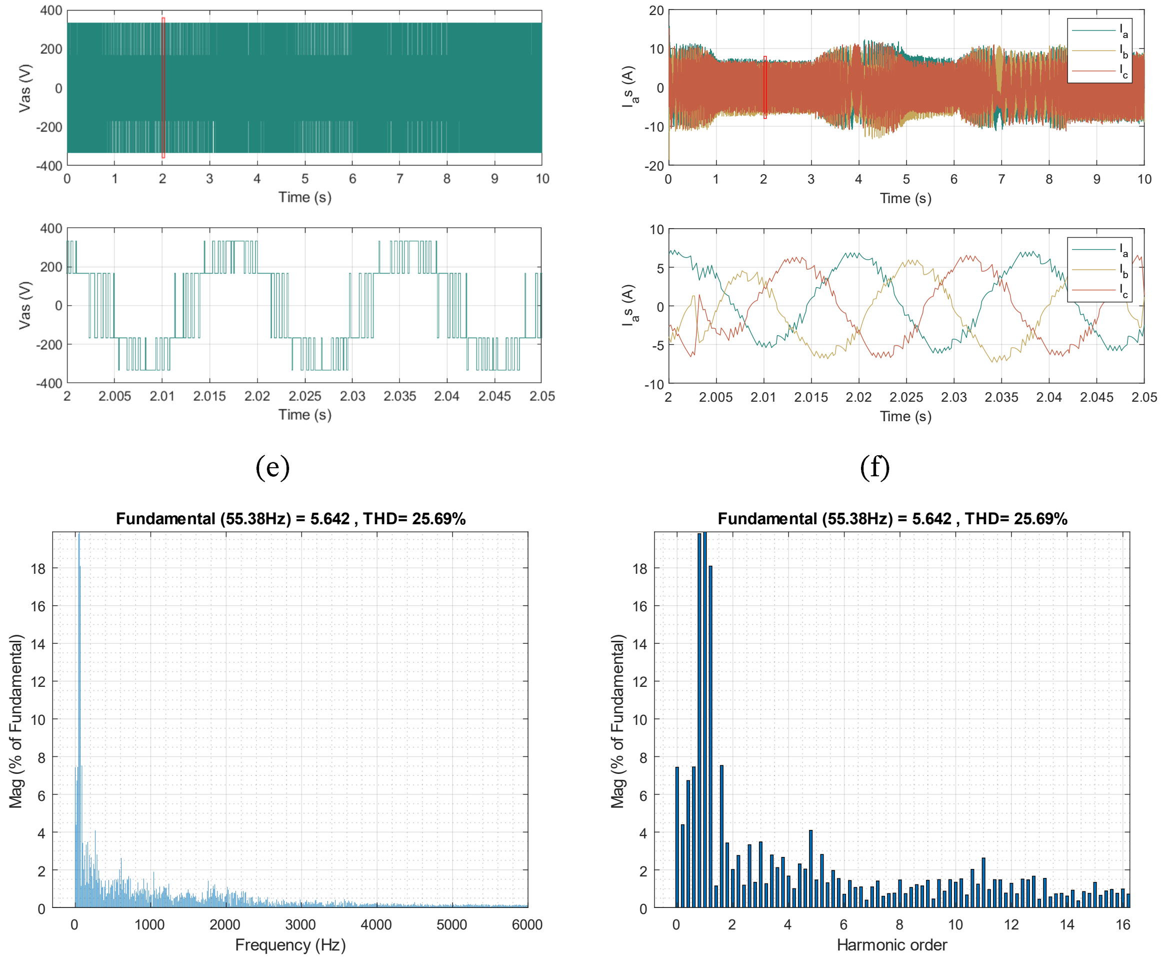
(g) (h)

Figure IV.9 : Test results for the implemented DTC with a **load torque proportional** to speed;(a)motor speed and it’s reference, (b) motor torque, the estimated torque and the load torque, (c) amplitude of the stator flux vector and the estimated flux, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics



(a) (b)

(c) (d)



(e) (f)

(g) (h)

Figure IV.10 : Test results for the implemented DTC with **load torque proportional to the square of motor speed**; (a) motor speed and it's reference, (b) motor torque, the estimated torque and the load torque, (c) amplitude of the stator flux vector and the estimated flux, (d) the stator flux vector in αβ frame, (e) the stator voltage for phase a, (f) the stator currents for all three phases, (g) current harmonics spectrum, (h) low order current harmonics

Figures IV.8, IV.9 and IV.10 show that the mechanical speed dynamic is excellent even in low speeds, speed follows the reference and the ramp-up rate is respected. The flux and torque estimations are accurate enough to allow a good regulation of both. Currents have a lot of distortion with a DC component present. The current THD is high and low order harmonics are important. This is due to multiple causes, first the fact that the classical DTC control method applies the appropriate voltage vector instantaneously when torque and flux deviate from the reference, by fixing the sampling frequency we have limited the calculation of this voltage vector to the next switching period, thus creating a delay of 100 µs, this led to an increase in torque ripple, a less precise flux regulation and current distortion, it is also due to the noise in the flux and torque estimation that trickles down to currents. The HIL simulation also introduces errors by the sampling and quantization of the signals before being sent to the controller.

The control algorithm for DTC shows however a good regulation of speed, and adequate regulation of torque and flux, on the contrary scalar control showed an acceptable regulation of speed and very little current harmonics, but torque ripples are more important. The test results for the implemented system confirm the results of the simulations.

The number of Core cycles for one period and the corresponding execution time for each control method is summarized in Table IV.1, the estimation was done using the counter of a timer with a counter period equal to the internal clock period. The Core usage is based on a 10 kHz switching frequency i.e. a period of 100μs.

TABLE IV.1 : COMPUTATION PERFORMANCES OF THE IMPLEMENTED ALGORITHMS

Control method	Number Cycles	Execution time (μs)	Core usage (%)
V/f constant with SVM	3385	20.15	20.15%
DTC	3669	21.84	21.84%

IV.5 Conclusion

In this chapter the control algorithms for constant V/f and DTC control were implemented on an STM32f4. The programs were tested using Simulink and the results simulations against different scenarios were analyzed.

Scalar control was easier to implement, required less measurement, was faster and had a better current harmonic spectrum. It was able to regulate the speed of the motor relatively well.

DTC control had a perfect adjustment of speed and was able to control torque and speed of the motor. The flux and torque estimator implemented using a compensated LPF was able to correctly estimate the parameters for the control and led to a successful regulation. DTC was however more complex to implement and required the knowledge of stator resistance.

Both control methods would be suitable for different use cases as both showed different advantages and drawbacks.

The STM32F4 Discovery board proved to be a very good tool for prototyping and testing motor control algorithms, it was powerful and had a plethora of peripherals and tools that made going from simulation to implementation on an embedded system seamless.

General conclusion

Motors consume approximately 25% of the world's electricity. Induction motors despite being less efficient than other motor types and harder to control, are the most widely used motors thanks to their ruggedness, ease of maintenance and low cost. VFDs offer induction motors the ability to operate with the performances as DC motors.

VFDs offer many advantages, they allow induction motors to operate in variable speed applications, protect the motor against startup current rush and increase power consumption efficiency. Control techniques allow this by either changing the supply frequency/voltage or by emulating the functioning of a DC motor. This thesis aims at studying and implementing two of the control methods: constant V/f and DTC.

First, an overview of converter topologies was given, then multiple control methods were presented as well as improvement techniques. Constant V/f, DTC and DTC-SVM were then simulated using MATLAB/Simulink and their performances analyzed and compared. DTC-SVM showed the best overall performances in controlling torque, flux, and speed, it compensated all the drawbacks of DTC. DTC had good performances, but the variable switching frequency make it hard to implement correctly on a digital signal processor without compromising on its ability to vary voltage without a modulator by fixing the switching frequency.

The STM32F4 Discovery board was presented along with the different peripherals later used to implement the aforementioned control algorithms.

A Hardware-in-the-loop simulation was used to test the controller running on the STM32. This method allowed the testing of the controller on various loads, without the need for physical equipment. This approach made debugging and testing faster and easier by giving access to all the motor variables and concentrating the effort on the controller design rather than on the hardware allowing a gradual prototyping of the VSD. The method used did not require expensive proprietary software, other than Simulink which was already used for the simulations.

The HIL simulation results showed that scalar control successfully controlled the speed of the motor in all the tested conditions, its cheap implementation is perfect for use in pumps, fans and other low precision applications, which represent the majority of industrial motor applications.

DTC showed a great advantage over constant V/f control by not only having a great regulation of speed but also the ability to control torque and flux. DTC requires more computation power and more measurements than scalar control making it more expensive.

Limiting the switching frequency of DTC led to more torque ripple and current noise, but it didn’t influence the control performances of the method.

As perspectives to this work we propose:

- Experimental test with a VSD prototype using the studied control methods.
- The implementation of speed-sensorless DTC-SVM.
- Making a PCB with the STM32F4 microcontroller for motor control.
- Using the STM32Cube.AI extension to implement a neural based DTC algorithm.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] ABB, T. International, and E. Agency, “Energy efficiency makes a difference,” 2008.

[2] “Technical guide No. 1 Direct torque control-the world’s most advanced AC drive technology ABB drives,” 2011.

[3] T. Friedli and J. W. Kolar, “Milestones in matrix converter research,” *IEEJ J. Ind. Appl.*, 2012, doi: 10.1541/ieejjia.1.2.

[4] J. Zhang, L. Li, and D. G. Dorrell, “Control and applications of direct matrix converters: A review,” *Chinese J. Electr. Eng.*, vol. 4, no. 2, pp. 18–27, 2019, doi: 10.23919/cjee.2018.8409346.

[5] J. W. Kolar, T. Friedli, J. Rodriguez, and P. W. Wheeler, “Review of three-phase PWM AC-AC converter topologies,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 11, pp. 4988–5006, 2011, doi: 10.1109/TIE.2011.2159353.

[6] M. P. Kazmierkowski, R. Krishnan, and F. Blaabjerg, *Control in Power Electronics: Selected Problems*. 2003.

[7] J. W. A. Wilson, “The Forced-Commutated Inverter as a Regenerative Rectifier,” *IEEE Trans. Ind. Appl.*, 1978, doi: 10.1109/TIA.1978.4503547.

[8] D. G. Holmes and T. A. Lipo, *Pulse Width Modulation for Power Converters*. 2010.

[9] D. Casadei, G. Serra, A. Tani, and L. Zarri, “Direct Torque Control for induction machines: A technology status review,” *Proc. - 2013 IEEE Work. Electr. Mach. Des. Control Diagnosis, WEMDCD 2013*, pp. 117–129, 2013, doi: 10.1109/WEMDCD.2013.6525172.

[10] F. Z. Peng, X. Yuan, X. Fang, and Z. Qian, “Z-source inverter for adjustable speed drives,” *IEEE Power Electron. Lett.*, 2003, doi: 10.1109/LPEL.2003.820935.

[11] G. S. Buja and M. P. Kazmierkowski, “Direct torque control of PWM inverter-fed AC motors - A survey,” *IEEE Trans. Ind. Electron.*, vol. 51, no. 4, pp. 744–757, 2004, doi: 10.1109/TIE.2004.831717.

[12] A. Kumar and T. Ramesh, “Direct Field Oriented Control of Induction Motor Drive,” *Proc. - 2015 2nd IEEE Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2015*, pp. 219–223, 2015, doi: 10.1109/ICACCE.2015.55.

[13] J. Yu, T. Zhang, and J. Qian, “Modern control methods for the induction motor,” in *Electrical Motor Products*, Elsevier, 2011, pp. 147–172.

[14] F. Blaschke, “The Principle of Field Orientation Applied to the New Transvector Closed Loop Control System for Rot.” 1972.

[15] K. Hasse, *Zur Dynamik drehzahlgeregelter Antriebe mit stromrichtergespeisten Asynchron-Kurzschlußläufermaschinen*. na, 1969.

[16] S. Hussain and M. A. Bazaz, “Review of vector control strategies for three phase induction motor drive,” *2015 Int. Conf. Recent Dev. Control. Autom. Power Eng. RDCAPE 2015*, pp. 96–101, 2015, doi: 10.1109/RDCAPE.2015.7281376.

[17] B. K. Bose, *Modern Power Electronics And Ac Drives*. Prentice-Hall, 2002.

[18] T. Matsuo and T. A. Lipo, “A Rotor Parameter Identification Scheme for Vector-Controlled

Induction Motor Drives," *IEEE Trans. Ind. Appl.*, 1985, doi: 10.1109/TIA.1985.349719.

[19] M. Koyama, M. Yano, I. Kamiyama, and S. Yano, "MICROPROCESSOR-BASED VECTOR CONTROL SYSTEM FOR INDUCTION MOTOR DRIVES WITH ROTOR TIME CONSTANT IDENTIFICATION FUNCTION.," *IEEE Trans. Ind. Appl.*, 1986, doi: 10.1109/TIA.1986.4504742.

[20] T. Y. Chang and C. T. Pan, "A Practical Vector Control Algorithm for μ-Based Induction Motor Drives Using a New Space Vector Current Controller," *IEEE Trans. Ind. Electron.*, 1994, doi: 10.1109/41.281614.

[21] E. Cerruto, A. Consoli, A. Raciti, and A. Testa, "Fuzzy adaptive vector control of induction motor drives," *IEEE Trans. Power Electron.*, 1997, doi: 10.1109/63.641501.

[22] S. Tadakuma, S. Tanaka, H. Naitoh, and K. Shimane, "Improvement of robustness of vector-controlled induction motors using feedforward and feedback control," *IEEE Trans. Power Electron.*, 1997, doi: 10.1109/63.558731.

[23] M. Sreejeth, M. Singh, and P. Kumar, "Efficiency optimization of vector controlled induction motor drive," in *IECON Proceedings (Industrial Electronics Conference)*, 2012, doi: 10.1109/IECON.2012.6388935.

[24] M. G. SimõTes and B. K. Bose, "Neural Network Based Estimation of Feedback Signals for a Vector Controlled Induction Motor Drive," *IEEE Trans. Ind. Appl.*, 1995, doi: 10.1109/28.382124.

[25] B. K. Bose, N. R. Patel, and K. Rajashekara, "A neuro-fuzzy-based on-line efficiency optimization control of a stator flux-oriented direct vector-controlled induction motor drive," *IEEE Trans. Ind. Electron.*, 1997, doi: 10.1109/41.564168.

[26] "DE3438504A1 - A method and device for controlling a polyphase machine - Google Patents." [Online]. Available: https://patents.google.com/patent/DE3438504A1/en. [Accessed: 21-Feb-2020].

[27] M. Depenbrock, "Direct Self-Control (DSC) of Inverter-Fed Induction Machine," *IEEE Trans. Power Electron.*, 1988, doi: 10.1109/63.17963.

[28] T. Noguchi and I. Takahashi, "Quick torque response control of an induction motor based on a new concept," *IEEJ Tech. Meet. Rotating Mach*, vol. RM84-76, pp. 61–70, 1984.

[29] I. Takahashi and T. Noguchi, "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 5, pp. 820–827, 1986, doi: 10.1109/TIA.1986.4504799.

[30] J. R. G. Schofield, "Direct torque control - DTC," in *IEE Colloquium (Digest)*, 1995, no. 181, doi: 10.1049/ic:19951108.

[31] A. Parthan, L. P. Suresh, and J. R. A. Raj, "A brief review on torque control of induction motor," *Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2017*, 2017, doi: 10.1109/ICCPCT.2017.8074348.

[32] N. El Ouanjli *et al.*, "Modern improvement techniques of direct torque control for induction motor drives-A review," *Prot. Control Mod. Power Syst.*, vol. 4, no. 1, 2019, doi: 10.1186/s41601-019-0125-5.

[33] X. del T. Garcia, B. Zigmund, A. A. Terlizzi, R. Pavlanin, and L. Salvatore, "Comparison between FOC and DTC Strategies for Permanent Magnet Synchronous Motors," *Adv. Electr.*

Electron. Eng., vol. 5, no. 1, pp. 76–81, 2006, doi: 10.15598.

[34] M. P. Kazmierkowski, L. G. Franquelo, J. Rodriguez, M. A. Perez, and J. I. Leon, “High-performance motor drives,” *IEEE Industrial Electronics Magazine*, vol. 5, no. 3, pp. 6–26, Sep-2011.

[35] T. G. Habetler, F. Profumo, M. Pastorelli, and L. M. Tolbert, “Direct Torque Control of Induction Machines Using Space Vector Modulation,” *IEEE Trans. Ind. Appl.*, vol. 28, no. 5, pp. 1045–1053, 1992, doi: 10.1109/28.158828.

[36] J. Maes and J. Melkebeek, “Discrete time direct torque control of induction motors using back-EMF measurement,” in *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)*, 1998, vol. 1, pp. 407–414, doi: 10.1109/ias.1998.732335.

[37] V. I. Utkin, “Sliding Mode Control Design Principles and Applications to Electric Drives,” *IEEE Trans. Ind. Electron.*, vol. 40, no. 1, pp. 23–36, Feb. 1993, doi: 10.1109/41.184818.

[38] M. Żelechowski, “Space Vector Modulated–Direct Torque Controlled (DTC – SVM) Inverter–Fed Induction Motor Drive,” *Ph.D. Thesis*, 2005.

[39] D. Casadei, F. Milanesi, G. Serra, A. Tani, and L. Zarri, “Control of induction motors for wide speed range for electric vehicle drives,” in *Proceedings of the 2008 International Conference on Electrical Machines, ICEM’08*, 2008, doi: 10.1109/ICELMACH.2008.4800158.

[40] E. Daryabeigi, G. R. A. Markadeh, and C. Lucas, “Emotional controller (BELBIC) for electric drives- A review,” *IECON Proc. (Industrial Electron. Conf.*, no. v, pp. 2901–2907, 2010, doi: 10.1109/IECON.2010.5674934.

[41] L. A. Zadeh, “Fuzzy sets,” *Inf. Control*, 1965, doi: 10.1016/S0019-9958(65)90241-X.

[42] D. Xu, B. Wang, G. Zhang, G. Wang, and Y. Yu, “A review of sensorless control methods for AC motor drives,” *Trans. Electr. Mach. Syst.*, vol. 2, no. 1, pp. 104–115, 2019, doi: 10.23919/TEMS.2018.8326456.

[43] D. Basic, F. Malrait, and P. Rouchon, “Current controller for low-frequency signal injection and rotor flux position tracking at low speeds,” *IEEE Trans. Ind. Electron.*, 2011, doi: 10.1109/TIE.2010.2100336.

[44] J. Holtz, “Sensorless position control of induction motors - an emerging technology,” in *International Workshop on Advanced Motion Control, AMC*, 1998, doi: 10.1109/iecon.1998.723873.

[45] M. Cirrincione, M. Pucci, G. Cirrincione, and G. A. Capolino, “A new TLS-based MRAS speed estimation with adaptive integration for high-performance induction machine drives,” *IEEE Trans. Ind. Appl.*, 2004, doi: 10.1109/TIA.2004.830779.

[46] S. Suwankawin and S. Sangwongwanich, “A speed-sensorless IM drive with decoupling control and stability analysis of speed estimation,” *IEEE Trans. Ind. Electron.*, 2002, doi: 10.1109/41.993278.

[47] M. Hinkkanen, L. Harnefors, and J. Luomi, “Reduced-order flux observers with stator-resistance adaptation for speed-sensorless induction motor drives,” *IEEE Trans. Power Electron.*, 2010, doi: 10.1109/TPEL.2009.2039650.

[48] R. P. Vieira, C. C. Gastaldini, R. Z. Azzolin, and H. A. Gründling, “Sensorless sliding-mode rotor speed observer of induction machines based on magnetizing current estimation,” *IEEE Trans. Ind. Electron.*, 2014, doi: 10.1109/TIE.2013.2290759.

[49] A. Accetta, M. Cirrincione, M. Pucci, and G. Vitale, “Neural sensorless control of linear induction motors by a full-order luenberger observer considering the end effects,” *IEEE Trans. Ind. Appl.*, 2014, doi: 10.1109/TIA.2013.2288429.

[50] G. Garcia Soto, E. Mendes, and A. Razek, “Reduced-order observers for rotor flux, rotor resistance and speed estimation for vector controlled induction motor drives using the extended Kaiman filter technique,” *IEE Proc. Electr. Power Appl.*, 1999, doi: 10.1049/ip-epa:19990293.

[51] “Variable Frequency Drives Market | Growth, Trends, and Forecast (2020-2025).” [Online]. Available: https://www.mordorintelligence.com/industry-reports/variable-frequency-drive-market-industry. [Accessed: 03-Mar-2020].

[52] “Ecodesign | ABB.” [Online]. Available: https://new.abb.com/drives/ecodesign. [Accessed: 09-Aug-2020].

[53] “Variable Frequency Drive Market Type, Power Rating | VFD Report, 2023.” [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/variable-frequency-drive-market-878.html. [Accessed: 03-Mar-2020].

[54] F. Blaabjerg, F. Iannuzzo, and L. Ceccarelli, “Power Electronics and Drive Systems,” in *Multiphysics Simulation by Design for Electrical Machines, Power Electronics, and Drives*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2017, pp. 251–281.

[55] A. Khedher and M. Faouzi Mimouni, “Sensorless-adaptive DTC of double star induction motor,” *Energy Convers. Manag.*, vol. 51, no. 12, pp. 2878–2892, Dec. 2010, doi: 10.1016/j.enconman.2010.06.028.

[56] E. Jalali, “Combined vector control and direct torque control method for high performance induction motor drives,” vol. 48, pp. 3095–3101, 2007, doi: 10.1016/j.enconman.2007.05.010.

[57] I. Ludtke and M. G. Jayne, “Direct torque control of induction motors,” in *IEE Colloquium on Vector Control and Direct Torque Control of Induction Motors*, 1995, pp. 6/1-6/6, doi: 10.1049/ic:19951113.

[58] P. Vas, *Sensorless vector and direct torque control.* Oxford University Press, 1998.

[59] A. Lokriti, I. Salhi, and S. Doubabi, “IM Direct Torque Control with no flux distortion and no static torque error,” *ISA Trans.*, vol. 59, pp. 256–267, 2015, doi: 10.1016/j.isatra.2015.08.014.

[60] T. Sutikno, N. R. N. Idris, and A. Jidin, “A review of direct torque control of induction motors for sustainable reliability and energy efficient drives,” *Renew. Sustain. Energy Rev.*, vol. 32, pp. 548–558, 2014, doi: 10.1016/j.rser.2014.01.040.

[61] A. K. Singh, C. U. Reddy, K. K. Prabhakar, and P. Kumar, “FPGA implementation of direct torque control of induction motor with reduced ripples in torque and flux,” *2015 IEEE Int. Transp. Electrif. Conf. ITEC-India 2015*, 2016, doi: 10.1109/ITEC-India.2015.7386917.

[62] R. P. Aguilera, P. Acuna, G. Konstantinou, S. Vazquez, and J. I. Leon, “Basic control principles in power electronics,” in *Control of Power Electronic Converters and Systems*, Elsevier, 2018, pp. 31–68.

[63] A. Benachour, “Commande sans Capteur basée sur DTC d’une Machine Asynchrone alimenté par Convertisseur Matriciel,” Ecole Nationale Polytechnique d’Alger, 2017.

[64] “UM1472 User manual Discovery kit with STM32F407VG MCU,” 2017.

[65] A. R. M. C. M. C. U. Fpu and F. Ram, “STM32F405xx, STM32F407xx Datasheet,” vol. 176, no. September, 2013.

[66] D. Ponikvar, “Interrupts & ports,” *STM32F407 project*. [Online]. Available: https://www.fmf.uni-lj.si/~ponikvar/STM32F407 project/Interrupts and ports.pdf. [Accessed: 07-May-2020].

[67] “STM32 ADC Tutorial - Complete Guide With Examples - DMA / Interrupt.” [Online]. Available: https://deepbluembedded.com/stm32-adc-tutorial-complete-guide-with-examples/. [Accessed: 22-Aug-2020].

[68] STMicroelectronics, “Data Brief - Integrated development environment for STM32 products.” 2019.

[69] STMicroelectronics, “STM32CubeIDE quick start guide.” STMicroelectronics, 2019.

APPENDIX

APPENDIX

Appendix A: Induction motor parameters

Properties	Value	Properties	Value
Nominal Power	1.5 kW	Operating frequency	50 Hz
Nominal current	6.7/3.7 A	Mutual inductance	0.258 H
Stator resistance	4.85 Ω	Nominal voltage	220/380 V
Rotor resistance	6.3 Ω	Nominal speed	1420 RPM
Stator inductance	0.274 H	Moment of inertia	0.031 Kg.m^2
Rotor inductance	0.274 H	Friction coefficient	0.001136 N.m.s/rad

Appendix B: PI controllers

Speed PI controller

The mechanical speed of an induction motor is given by the equation:

$$J\frac{d\Omega}{dx} = T_e - T_l - f_v\Omega \tag{1}$$

This equation gives the following transfer function:

$$J\Omega s = T_e - Tl\ - f_v\Omega \Leftrightarrow \ \Omega = (T_e - T_l)\ \frac{1}{Js+f_v} \tag{2}$$

The speed regulation with a PI controller is therefore described in Figure B.1.

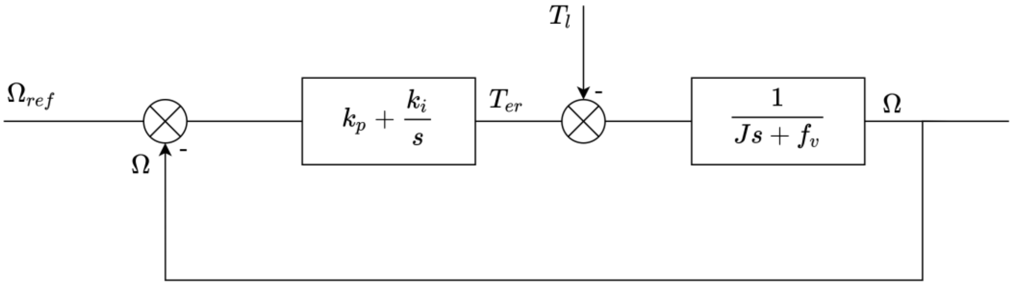


Figure B.0.1 Block diagram of mechanical speed PI regulation

From this block diagram the closed-loop transfer function can be written as:

$$\Omega = \left(\left(k_p + \frac{k_i}{s}\right)\left(\Omega_{ref} - \Omega\right) - T_l\right)\frac{1}{Js+f_v} \tag{3}$$

By considering the load torque a disturbance the mechanical speed transfer function becomes:

$$\Omega = \frac{k_p}{J}\frac{s+\dfrac{k_i}{k_p}}{s^2+\dfrac{k_p+f_v}{J}s+\dfrac{k_i}{J}}\Omega_{ref} \tag{4}$$

The denominator of this transfer function is of the form: $P(s) = s^2 + 2\xi\omega_n s + \omega_n^2$

And thus:

$$\begin{cases} \omega_n^2 = \dfrac{k_i}{J} \\ 2\xi\omega_n = \dfrac{k_p+f_v}{J} \end{cases} \tag{5}$$

Therefore:

$$\begin{cases} k_i = J\omega_n^2 \\ k_p = 2\xi\omega_n J - f_v \end{cases} \tag{6}$$

A typical value for ζ is ζ =0.7. The natural frequency ω_n can be calculating by choosing a settling time T_s for which the response remains within 2% of the final value. ω_n is then given by:

$$\omega_n \cong \frac{4}{\xi T_s} \tag{7}$$

Appendix C: Testing STM32 Program with MATLAB/Simulink

Exchanging data between the ST32F4 Discovery board can be done using the UART protocol, but since the ST-LINK on this board is not wired to the UART pins, they either have to be physically soldered or accessed with a USB-TTL adapter. Another option would be to use the USB-CDC interface of the STM32F4 Discovery board to send and receive data to a MATLAB/Simulink simulation, or any other serial monitor.

Here, it was used to receive measurements and send control signals from the simulation of an induction motor and inverter in order to test control algorithms without the need for real hardware, as described in the following figure.

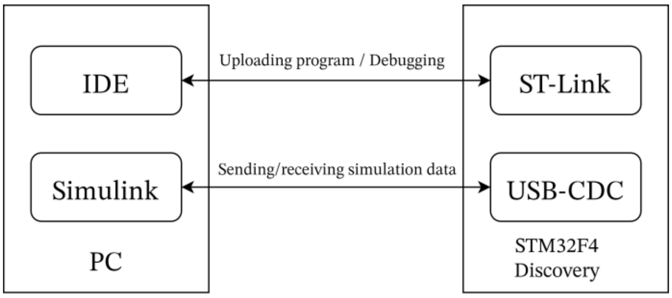

IDE
Simulink
PC
Uploading program / Debugging
Sending/receiving simulation data
ST-Link
USB-CDC
STM32F4
Discovery

Figure C.1 : Diagram of the communication interface between the PC and the STM32

To configure the USB for the STM32 we open the device configuration tool for the project, in the USB_OTG_FS category we set the mode to “Device_Only”, then go to the Middleware category, USB_DEVICE, in the drop-down menu for Class for FS IP select “Communication Device Class (Virtual Port Com). This will configure the USB to be used as a a virtual com port on windows.

It is also important to change the Minimum Heap size in the Project Manager, we can set it to 0x600 for instance. We then generate the code.

The file that needs to be modified is the “usb_cdc_if.c” file present under “USB_DEVICE/App” folder.

To send data we use the CDC_Transmit_FS() function, it takes to arguments : a pointer to a buffer containing the data to be sent and the length of data to be sent.

When data is received it can be accessed in the CDC_Receive_FS() function via a buffer.

This function is modified to record our data inside global variable, to convert the data from an array of bytes to any other type we declare a union containing arrays of any data type, and the an array of bytes on which the buffer will be recorded. Example:

```
union floatUnion {
  float f[5];
  uint8_t buffArray[20];
} floatBuffer;
```

The floatBuffer union is used to convert a buffer of 20 bytes into an array of 5 floats, the size of a float is 4 bytes, therefore the size of this union is 20 bytes.

Accessing the variables of a union is similar to accessing variables in a struct.

To record the values, we record each byte individually:

```
floatBuffer.buffArray[0] = *Buf;
floatBuffer.buffArray[1] = *(Buf+1);
floatBuffer.buffArray[2] = *(Buf+2);
```

. . .

We can then access the float variables with floatBuffer.f[0], floatBuffer.f[1] ...

We can create a buffer for sending in a similar fashion, and then pass a pointer to the bytes array of the buffer to the CDC_Transmit_FS() function.

On the Simulink side, we use the “Serial send”, “Serial Receive” and “Serial Configuration” blocks of the Instrument Control Toolbox.

In the Serial Configuration block, we set the communication port to the Virtual com port of the STM, the Baud rate to 115200 and leave all the rest to default.

In the Serial receive block we can specify the data size for example a data size of [1 5] will output an array of five values, whose type will be selected in the Data type field. We make sure to enable blocking mode so that the simulation is halted until data is received, the block sample time is where we set the sample time of the received data, the data will be output at this sampling rate.

The data to be sent can be sampled with a zero-order hold block, type cast, multiplexed, and sent to the “Serial send” block configured in blocking mode.

The resulting Simulink model is described in the following diagram:

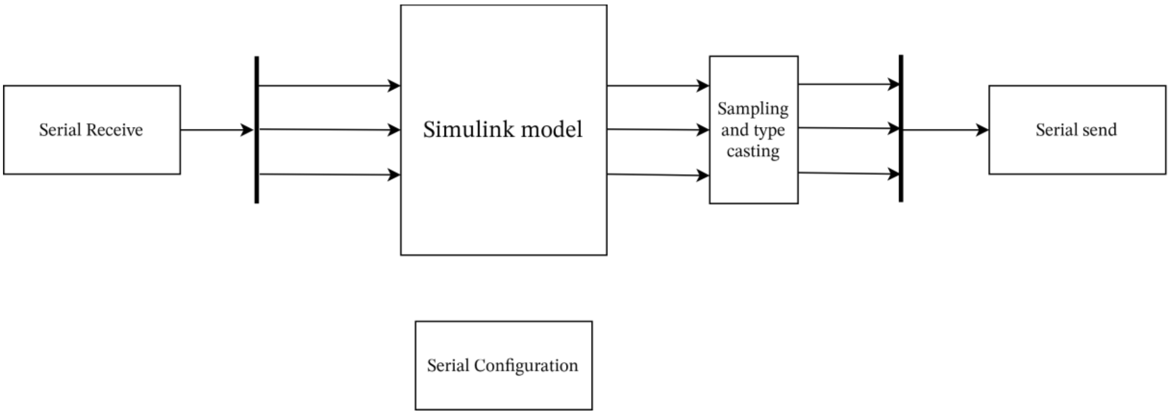


Figure C.0.2 : Diagram of the Simulink model for simulating with data acquired from the STM32

Appendix D: STM32F4 Peripherals

Timers

The STM32F407 devices include two advanced-control timers, eight general-purpose timers, two basic timers and two watchdog timers. All timer counters can be frozen in debug mode.

Features of the different timers available on the STM32F407 devices are compared in Table [65].

TABLE D.1: FEATURES OF THE ADVANCED-CONTROL, GENERAL-PURPOSE AND BASIC TIMERS ON THE STM32F407

Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary output	Max interface clock (MHz)	Max timer clock (MHz)
Advanced-Control	TIM1, TIM8	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	Yes	84	168
General purpose	TIM2, TIM5	32-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM3, TIM4	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM9	16-bit	Up	Any integer between 1 and 65536	No	2	No	84	168
	TIM10, TIM11	16-bit	Up	Any integer between 1 and 65536	No	1	No	84	168
	TIM12	16-bit	Up	Any integer between 1 and 65536	No	2	No	42	84

	TIM13, TIM14	16-bit	Up	Any integer between 1 and 65536	No	1	No	42	84
Basic	TIM6, TIM7	16-bit	Up	Any integer between 1 and 65536	Yes	0	No	42	84

Advanced-control timers (TIM1 and TIM8)

The advanced-control timers (TIM1, TIM8) can be seen as three-phase PWM generators multiplexed on 6 channels. They have complementary PWM outputs with programmable inserted dead times. They can also be considered as complete general-purpose timers. Their 4 independent channels can be used for:

- Input capture
- Output compare
- PWM generation (edge- or center-aligned modes)
- One-pulse mode output

If configured as standard 16-bit timers, they have the same features as the general-purpose TIMx timers. If configured as 16-bit PWM generators, they have full modulation capability (0-100%).

The advanced-control timer can work together with the TIMx timers via the Timer Link feature for synchronization or event chaining.

General-purpose timers (TIMx)

There are ten synchronizable general-purpose timers available on the STM32F407:

- **TIM2, TIM3, TIM4, TIM5**: The TIM2 and TIM5 timers are based on a 32-bit auto-reload up/downcounter and a 16-bit prescaler. The TIM3 and TIM4 timers are based on a 16-bit auto-reload up/downcounter and a 16-bit prescaler. They all feature 4 independent channels for input capture/output compare, PWM or one-pulse mode output. This gives up to 16 input capture/output compare/PWMs on the largest packages. The TIM2, TIM3, TIM4, TIM5 general-purpose timers can work together, or with the other general-purpose timers and the advanced-control timers TIM1 and TIM8 via the Timer Link feature for synchronization or event chaining. Any of these general-purpose timers can be used to generate PWM outputs. They are capable of handling quadrature (incremental) encoder signals and the digital outputs from 1 to 4 hall-effect sensors.

- **TIM9, TIM10, TIM11, TIM12, TIM13, TIM14**: These timers are based on a 16-bit auto-reload upcounter and a 16-bit prescaler. TIM9 and TIM12 have two independent channels for input capture/output compare, PWM or one-pulse mode output, whereas TIM10, TIM11, TIM13, and TIM14 feature one independent channel. They can be synchronized with the other TIM2, TIM3, TIM4, TIM5 general-purpose timers. They can also be used as simple time bases. These are the only timers that do not support DMA request generation.

Basic Timers (TIM6, TIM7)

These timers are mainly used for DAC trigger and waveform generation. They can also be used as a generic 16-bit time base.

ADCs

Three 12-bit analog-to-digital converters are embedded and each ADC shares up to 16 external channels, performing conversions in the single-shot or scan mode. In scan mode, automatic conversion is performed on a selected group of analog inputs.

Additional logic functions embedded in the ADC interface allow: Simultaneous sample and hold, Interleaved sample and hold.

An analog watchdog feature allows very precise monitoring of the converted voltage of one, some or all selected channels. The ADC can be served by the DMA controller. An interrupt is generated when the converted voltage is outside the programmed thresholds.

To synchronize analog to digital conversion and timers, the ADCs could be triggered by any of the TIM1, TIM2, TIM3, TIM4, TIM5, or TIM8 timers.[65]

DACs

The two 12-bit buffered DAC channels on the STM32F407 can be used to convert two digital signals into two analog voltage signal outputs.

It supports the following features:

- two DAC converters: one for each output channel
- 8-bit or 12-bit monotonic output
- left or right data alignment in 12-bit mode
- synchronized update capability
- noise-wave generation
- triangular-wave generation
- dual DAC channel independent or simultaneous conversions
- DMA capability for each channel
- external triggers for conversion
- input voltage reference VREF+

Eight DAC trigger inputs are used in the device. The DAC channels are triggered through the timer update outputs that are connected to different DMA streams.