POLITECNICO DI TORINO

I Facoltà di Ingegneria

Corso di Laurea in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Sensorless Brushless DC Motors

Development and comparison of different
fault tolerant control algorithms

Relatore
prof. Paolo Maggiore

Candidato
Matas Gualtieri Lara

Marzo 2018

A nonno Leo e nonna Laila,

Abstract

The aviation industry pushes towards the More Electric Aircraft (MEA) concept, to meet existing market demands about reductions in gas emissions and operating and maintenance costs. The MEA concept is based on the utilization of electric power for all non-propulsive systems, which are traditionally powered by pneumatic, mechanical, hydraulic and electrical sources. The advancements in power electronics are favoring these radical changes. In particular the research is focused on the development of high reliable and fault tolerant electric systems; among them there are the electro-mechanical actuators (EMA).

This masters' thesis is part of a wider project, which has the purpose of design EMAs for secondary actuations in aircrafts, with the idea of extending the applications to primary flight controls in the near future. An essential aspect is the control of the speed and torque that the EMA has to guarantee during the actuation. Of course, in order not to endanger the operability of the aircraft, the control system must have high reliability. In this document the design of four fault tolerant algorithms devoted to control brushless DC motors is described and justified. These electric machines are the main components of an electro-mechanical actuator. In fact, thanks to their high energy density, controllability and reliability, they are one of the few kinds of motors implemented in aeronautic. Among all possible options available in the literature, the trapezoidal and field oriented controls have been thoroughly investigated for their simplicity in the implementation and high performance respectively. The research consists in understanding the detailed functioning of the algorithms, and the creation of a model in Matlab/Simulink® for the simulation testing. Both algorithms do not use encoders for the determination of the position and speed of the rotor. With proper use of the voltage and current signals, measured on the stator phases, it is possible to estimate these two essential parameters.

The results show that the FOC algorithm is feasible for sensorless applications which demand high performances, while the lack of hall sensors, in trapezoidal controller, complicates the design so much so that it could increase costs and unreliability. This is why the specification of the EMA requirements, as well as the testing of the algorithms on a real system, are required to finally determine which of those is the best solution. Since the importance of the reliability in the aeronautical field, an overview of the fault detection and diagnosis of the system is proposed as final topic of this work. In particular, the objective is the development of a fault tolerant control able to keep the rotor running, matching the speed and torque demand also after one phase failure. This system has been successfully integrated with both kinds of control, limiting as much as possible the degradation of performances and efficiency of the motor.

This thesis has been carried out in the department of power electronics at Skylife Engineering in Sevilla. It is part of the apprenticeship which concludes the double degree project in aerospace engineering, signed by the Polytechnic University of Turin and the Polytechnic University of Madrid.

After an overview of the state of the art brushless DC motors and their control methods and manufacturing, the FOC and trapezoidal control algorithms are thoroughly described in the second and third chapters. Then, a brief review of the PI tuning techniques is reported in chapter four, to conclude the theme of the speed and torque control. Finally, the phase failure problem and the reacting fault-tolerant algorithm are resumed in the fifth chapter, and all the results collected in the sixth chapter.

Contents

Chapter 1

Introduction to Brushless DC motors

Brushless DC (BLDC) motors are synchronous electrical machines supplied with direct current. They represent an evolution of the brushed DC motors which started being employed in the 1960s, during the revolution of the solid state electronics. However, the widespread of the DC motors began later, in the 1980s, when the discovery of the Neodymium-Iron-Boron (Nd-Fe-B) alloy allowed to decrease the market prices of the Permanent Magnet (PM) materials, encouraging the commercial interest in this technology [7].
The early DC motors had brushes for the commutation. Only when the mechanical commutation was replaced by electronic control, the brushes were eliminated. The lack of brushes and commutator increases the reliability and efficiency of the system, thanks to many fewer parts which are sources of friction, can break and need to be replaced. Furthermore, the brushless configuration can operate at higher speed in both loaded and unloaded conditions, reducing noise and electromagnetic interference. Finally, a motor without brushes has higher power to weight ratio, which represents an essential improvement for the aeronautical applications.

This chapter is about the state of art of BLDC motors, their operating principle, industrial applications and future trends.

1.1 BLDC manufacturing

A BLDC motor is manufactured with similar procedures used for AC motors, while its operating principle slightly changes from the brushed motor case. Stator and rotor are the fundamental parts of a motor: the rotor can spin both inside or outside the stator. The first configuration is called inner rotor, while, if the stator is placed inside, it is known as outer configuration (fig.1.1).
The stator is made out of laminated steel stacked up to carry the windings. It is fundamental to properly design the stator, because the final performances deeply depends on it. The sator windings can be arranged in two possible patterns: a star (Y) or delta pattern (Δ), represented in fig. 1.2. The function torque-speed

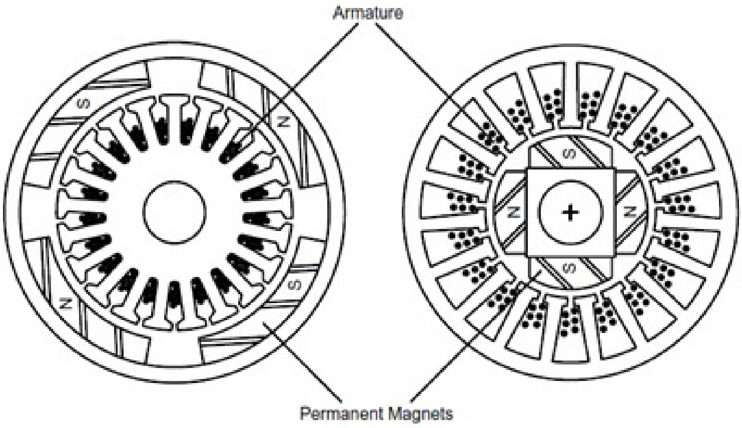


FIGURE 1.1: Comparison between inner (to the right) and outer (to the left) rotor configurations [4].

changes depending on which pattern has been chosen for the design. The Y pattern gives high torque at low speed; on the contrary, the Δ pattern gives high torque at high speed.
Besides the patterns, windings are usually organised in three phases placed in slots. However, the common configuration has exception, in fact the steel laminations can be either slotted or slotless. The slotless configuration results in larger air-gap between rotor and stator armature, which produces an unwelcome high reluctance in the magnetic circuit. This has to be compensated with more windings such that increasing costs. For all these reasons, slotting is usually preferred, even if it increases the frictional drag and acoustic noise.
The rotors can have different designs too. Two categories are usually proposed for the classification: rotors with permanent magnets (PM BLDC) or, in the case of induction motor, rotors with windings. Induction motors are preferred when higher performances are desired. In fact, even if the maximum efficiency is lower, its average value is higher over a wider speed and torque range [27].

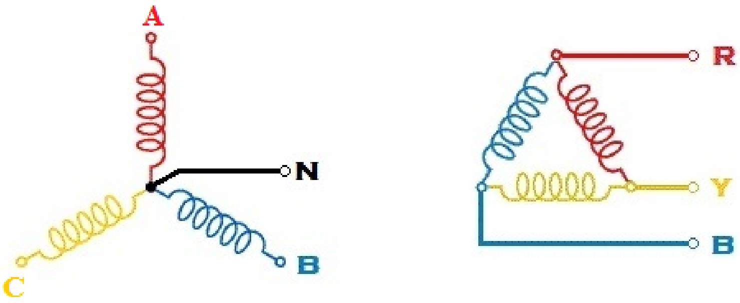


FIGURE 1.2: Comparison between star pattern (to the left) and Δ pattern (to the right).

This is because they can adjust the magnetic field, maximizing efficiency at each operating point. Conversely, induction machines are more difficult to control due to the higher complexity of the control laws. This implies higher costs in the implementation of an induction machine, that could hide the advantage of the lower cost for the manufacturing.
For the PM BLDC motors, the number of magnetic poles can vary. The higher the number of the poles, the smaller the ripple in the magnetic torque, but at the cost of reducing the maximum speed. In fact, for the same rotor speed, the higher the number of poles the higher is the electrical frequency, which is limited by the speed of the microprocessor. Usually, the rotor has an even number of poles: two or four in most of the applications.
The maximum torque is also affected by the material used for the manufacturing of permanent magnets, physically represented by the retenence magnetic flux density B_r (sec. 1.2).
The improvement in the manufacturing of magnetic materials is essential to make the PM motor as efficient as possible. Nowadays, typical PM are the Neodymium-Iron-Boron (Nd-Fe-B) magnets which provide the highest energy and residual flux density. They are also easier to produced if compared with the alnico magnets, but more expensive than the ceramic, that unfortunately have smaller retentivity. Typical B_r values for Nd-Fe-B magnets are in the range of $1T$ to $1.35T$, as shown in the NdFeB Magnets /Neodymium Iron Boron Magnets datasheet [5]. A complete comparison of magnetic, thermal ad mechanical characteristics of permanent magnet materials is proposed in the standard specification paper [1]. The performances of PM electrical motors have been improved also by the advancements in geometries and design innovations of the entire motor architecture.

1.2 Operating Principle of Electrical Machines

1.2.1 Torque Production

The torque produced in an electric motor is caused by the interaction of the stator and rotor magnetic forces. Accordingly with the Faraday's law, a wire that carries current, placed in a magnetic field, experiences a mechanical force determined by eq. (1.1). The force is the vectorial product of the current $\vec{i}[A]$ and the magnetic flux density $\vec{B}$ [T]

$$\vec{F} = l \cdot \vec{B} \times \vec{i} \tag{1.1}$$

where l $[m]$ is the length of the wire. Considering this vectorial expression, the electromagnetic force is maximum when the rotor and the stator fields are in quadrature, maximizing the efficiency of the machine. This is why the designer should always aim to achieve this condition. For DC motors the current must be constantly commuted to guarantee it.

$$\vec{T} = \vec{F} \times \vec{r} \tag{1.2}$$

The resultant electromagnetic torque ($\vec{T}$) depends on the motor size. ($\vec{r}$) is the lever arm of the force, and corresponds to the vector which goes form the centre of the rotor to the air gap surrounding it. In the case of PM BLDC, the rotating $\vec{B}$ has constant magnitude, the wire length and motor geometry do not change, then the only parameter that the torque depends on, and varies during operation, is the stator current. Hence the torque expression is simplified, assuming that the current is perpendicular to the field:

$$T = k_t I \tag{1.3}$$

where the $k_t[Nm \cdot A]$ is the motor torque constant, and for a given motor it represents how much torque is produced per unit armature current.

When currents flows through windings, the electromagnetic torque is produced, inducing the motor rotation. As soon as the motor starts rotating, the flux linkage in the coils changes. According to Faraday's law of induction, a time-variant flux linkage induces a back electromotive force (BEMF) in the coils.

$$BEMF = -\frac{d\varphi\left(\vec{B}\right)}{dt} \tag{1.4}$$

Developing the time derivative in the eq. (1.4) and considering that $\vec{B}$ is rotating at the rotor speed $\omega[rad/s]$ with constant magnitude, the BEMF can be easily expressed by:

$$BEMF = k_e \omega \tag{1.5}$$

The waveform of the BEMF voltage is determined by the distribution of the flux in the air gap, and the the e.m.f. constant is equal to the torque constant $k_t = k_e = k$ [37].

1.2.2 Equivalent Circuit

The equivalent electrical circuit is useful to understand the operating principle and for future considerations on sensorless control techniques. In the fig. 1.3, V is the voltage applied to the armature terminals and E is the BEMF. R and L are respectively the resistance and inductance of the entire armature. When the machine operates as motor and not as generator, the BEMF always opposes the applied voltage V [37]. Hence, V is greater than E, and the voltage equation is given by:

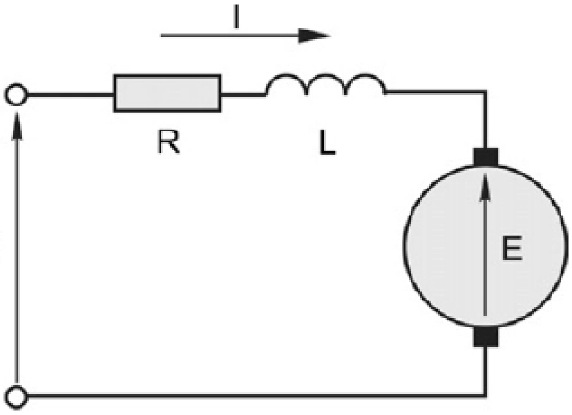


Figure 1.3: Equivalent electrical circuit of a PM BLDC motor.

$$V = E + I \cdot R + L \cdot \frac{dI}{dt} \tag{1.6}$$

The voltage drop on the inductive component is proportional to the rate of change of current, and disappears under steady-state conditions. This component smooths the current waveform when the motor is supplied with a Pulse Width Modulation (PWM) signal, but provokes an unwelcome effect under transient conditions. In the steady-state, the speed can be easily expressed as a function of the applied voltage, the load torque (T_L) and motor parameters. In fact, the armature current is constant when there are no transient, so the armature inductance term does not affect the result, and because there is no acceleration, the motor torque is equal to the load torque.

$$\omega = \frac{V}{k} - \frac{R}{k^2} T_L \tag{1.7}$$

Concluding, first term in eq. (1.7), called no-load speed, is directly proportional to the armature voltage, while the second term gives the drop in speed for a given load torque. The slope of the $\omega = f(T_L)$ curve is R/k^2; it demonstrates that the smaller the armature resistance, the smaller the drop in speed when load is applied [37].

1.3 Motors Control Techniques

In a BLDC motor the torque is produced by the attraction or repulsion between the net stator magnetic field and the one produced by the permanent magnet. By controlling the current flow in the three windings, it is possible to determine the direction and magnitude of the magnetic field produced by the stator. In particular, with commutation the direction is constantly changed. With brushes the commutation is mechanical, while the BLDC motors implement electronic to replace them.
To produce a steady torque, the current space vector should ideally be synchronous with the rotor, maintaining a quadrature direction, and have constant magnitude. To do so, various control algorithms are published in the literature, which also describes their development and the results of the validation tests. These techniques can be classified into three categories: Trapezoidal, Sinusoidal and Field Oriented Control (FOC).

1.3.1 Trapezoidal Control

The trapezoidal control is the easiest and the most common way to control a BLDC motor. This method exploit the hall effect sensors to determine the rotor position, from which the commutation instant is determined. Two windings of the stator are connected to the DC generator at the same time, while the third is floating. As the rotor moves, the phase voltage is electronically switched, obtaining a proper combination of active phases to keep on the rotor running. This produces a current space vector that steps between six distinct directions. The scheme is relatively simple and cheap to implement, this is why it is ideal for low-cost and low performance applications. But it has also several drawbacks. The most important is that the current space vector can only assume six directions, provoking an high torque ripple, at a frequency of six times the electrical rotational speed of the motor. This leads to efficiency losses and makes difficult to control at low speeds [40].

1.3.2 Sinusoidal Control

The Sinusoidal Control drives the stator windings with three currents that vary sinusoidally as the motor rotates. This requires a proper modulation, such that the resulting space vector is always in the quadrature direction with the rotor

magnetic field. Only an accurate measurement of the rotor position allows to achieve this. The position information from an encoder is then used to synthesize two sinusoids, that are phase shifted of 120^o and whose amplitude is proportional to the desired torque. The phase current is sensed back and provided to a pair of PI controllers, whose output controls the PWM generators. The PWM signal is then supplied to the transistor bridge, controlling the winding currents and producing a vector with the features discussed before.
The sinusoidal control allows to ideally eliminate the torque ripple, giving a smooth rotation with higher efficiency. Unfortunately, because of the limited gain and frequency response of the PI controllers, as the speed increases, the current command in feedback becomes more difficult to track. This provokes a reduction of electromagnetic torque until it becomes zero, stopping the motor or reaching a steady state at an undesired speed value. Then, the sinusoidal control scheme is used in low speed applications, but the FOC has to be adopted to extend the speed range.

Because of this reasons, the thesis is focused on the development of the FOC control and how achieve a low-cost design configuration. This is why the sinusoidal control is only briefly described above, while two chapters has been dedicated to FOC and trapezoidal control techniques.

1.4 Applications and future Trends

PM BLDC motors find applications in domestic appliances, automotive[1], aerospace equipments[2] and so on, managing an electric power which ranges from microwatt to megawatts. The advanced control algorithms introduced in this chapters, together with other techniques available in the literature, makes suitable this kind of motors also in very precise position and speed controllers, extending their implementation to robotics and others high precision servos. They are employed in various high-speed applications too, such as the hard disk drive of computers. The high speed induces a smaller torque, which provokes an undesired longer transient period [32]. Many other applications of PM BLDC motors are reported in the literature such as: tread mills, washers, dexterous robotic hands, wheelchairs, compressors of household air conditioners, commercial freezers, fans and pumps [32].
In the last years, the research has grown its interest for the use of this kind of motors in electric and hybrid vehicles. The research aims to study the effects of their application on polluting emissions. In fact, these motors are characterized by high power density and efficiency, that could contribute to achieve the objective [32]. Besides the technological advantages, the research is

[1] on the left of the fig. 1.4 the Micronas product [34]

[2] such as drones, on the right of the fig. 1.4 a detail of a propeller of a small drone [6]

Figure 1.4: Examples of applications of BLDC motors in dayly technologies.

focused on the possibilities to reduce the overall cost [32] with improvements in the motor architecture and control algorithm [20] [22] [42]. In the next future BLDC motors is expected to have sensorless configurations, which reduces the cost and increases the reliability of the overall system. The economic viability and better performances will encourage a further spread of these motors to an even wider range of applications [32].

Chapter 2

Trapezoidal Control

In the low-cost sensor-based applications, the most common control technique is the trapezoidal control, also known as six-step controller, which uses the hall-effect sensors to determine its position. The operating principle and the proposed simulation models are described in this chapter.

2.1 Control Using Sensors

2.1.1 Introduction

The architecture of the six-step controller consists in one main loop dedicated to the speed control. It acts directly on the electromagnetic torque. Inside this loop, a motor driver determines the commutation instants. In order to produce the maximum torque, the angle between the stator and the rotor flux is kept as close to 90^o as possible. Unfortunately this technique allows the stator flux to occupy six different orientations, without the possibility to change continuously between them. In fact, the stator flux can have six different orientations, and cannot changes continuously from one state to the next. Consequently, the commutation is repeated each electrical 60^o, inducing an high torque and speed ripples.

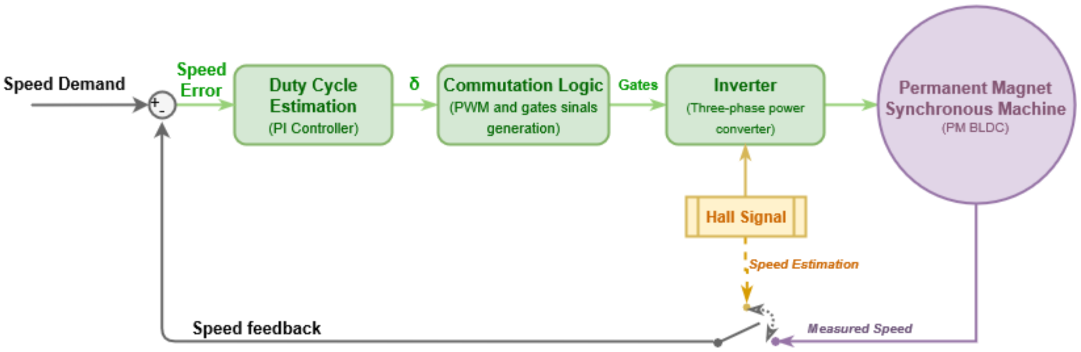


FIGURE 2.1: Six-step controller basic scheme.

In figure 2.1 is represented schematically the six-step controller whose main blocks are:

- ▷ **Three-phase power converter**. Dedicated to the stator phases commutation, based on the rotor position determination.
- ▷ **Hall sensors**. Generate three signals whose combination corresponds to one of the six states.
- ▷ **Speed feedback**. The rotor speed and the demanded one are compared generating an error. The speed could be directly measured or estimated from the hall signal frequency.
- ▷ **Duty Cycle Estimator**. The motor speed depends on the amplitude of the applied voltage. The voltage is supplied to the phases as a PWM signal. It allows setting the equivalent voltage amplitude by regulating the duty cycle (δ). Since the relation between speed, voltage and duty cycle is linear, then the motor speed is $\omega_m \propto \delta$.
- ▷ **Commutation logic**. It translates the hall signals into logical commands for the MOSFET transistors.

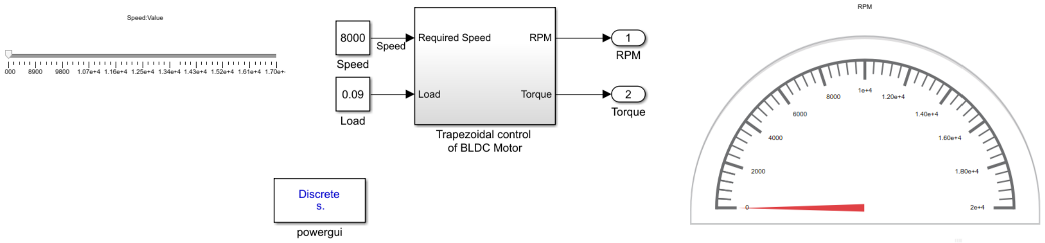


FIGURE 2.2: Simulink user interface of Trapezoidal Controller of a PM BLDC motor.

The functions operated by each block are thoroughly described in this section, giving further informations about the model implemented in Simulink.

Motor Model

Two types of BLDC motors can be distinguished according to the BEMF waveform [43]:

- ▷ Sinusoidal back-EMF waveform motor. This configuration uses a vector control scheme to obtain a ripple-free torque. It has strong limitations related to the difficulty to have exact sinusoidal BEMF. The motor has the structure the is more complicated, bigger and more expensive.

- ▷ Trapezoidal back-EMF waveform motor. In this case the constant torque is produced with the switching method. The constant torque is just theoretically possible, because the current cannot be established instantaneously in a motor phase, and a torque ripple is repeated each 60^o of phase commutation.

The modelled control algorithm has been tested on a reference motor, whose datasheet is reported in the appendix A. The motor is the EC-4pole 30 produced by Maxon Motor (in fig.2.3a and 2.3b). It has been chosen as reference case because it has aeronautic certification, 24DC supply voltage and 4 poles (2 pair poles) and permanent magnet rotor, which are typical characteristics for the aeronautical applications.
The dynamic model of the motor is already implemented in the block *Permanent Magnet Synchronous Machine (PMSM)* available in the Simulink library. All the details of the model are reported in the Mathworks help; here following just the useful parameters are summed up, accordingly with the motor parameters in the datasheet. In order to match the real characteristics of the motor, for a future implementation of the control algorithm, the model has a trapezoidal back-EMF waveform. The modelling equations are reported in the Mathworks help of the same block. Those are expressed in the phase reference frame assuming a constant inductance L_s with the rotor position.

$$\begin{aligned}
&\frac{d}{dt}i_a = \frac{1}{3L_s}\left(2v_{ab} + v_{bc} - 3R_s i_a + \lambda p\omega_m(-2\varphi'_a + \varphi'_b + \varphi'_c)\right)\\
&\frac{d}{dt}i_b = \frac{1}{3L_s}\left(-v_{ab} + v_{bc} - 3R_s i_b + \lambda p\omega_m(\varphi'_a - 2\varphi'_b + \varphi'_c)\right)\\
&i_c = -\left(i_a + i_b\right)\\
&T_e = p\lambda\left(\varphi'_a i_a + \varphi'_b i_b + \varphi'_c i_c\right)\\
&\frac{d}{dt}\omega_r = \frac{1}{J}\left(T_e - T_f - F\omega_m - T_m\right)\\
&\frac{d}{dt}\vartheta = \omega_m
\end{aligned} \tag{2.1}$$

where:

L_s Inductance of the stator windings.

R Resistance of the stator windings.

i_a, i_b, i_c a, b and c phase currents.

φ_a, φ_b, φ_c a, b and c phase electromotive forces.

v_a, v_b, v_c ab and bc phase to phase voltages.

ω_m Angular velocity of the rotor.

λ Amplitude of the flux induced by the permanent magnets in the stator phases.

p Number of pole pairs.

T_e Electromagnetic torque.

J Combined inertia of rotor and load.

F Combined viscous friction of rotor and load.

ϑ Rotor angular position.

T_m Shaft mechanical torque.

T_f Shaft static friction torque.

It is remarkable that the simulink model of a PMSM assumes a linear magnetic circuit with no saturation of the stator and rotor iron. This assumption is justified in the Mathworks help, observing that PMSM usually have large air gap that do not allow reaching the saturation of the materials. Furthermore, in the same model description, it is suggested to use a small parasitic resistive load[1] connected to the machine terminals, to avoid numerical oscillations when implementing discrete systems. The dynamic model has the following outputs:

i_a, i_b, i_c Stator phase currents.

e_a, e_b, e_c Phase back EMF.

h_a, h_b, h_c Hall effect signals.

ϑ, ω_m Rotor speed and angle.

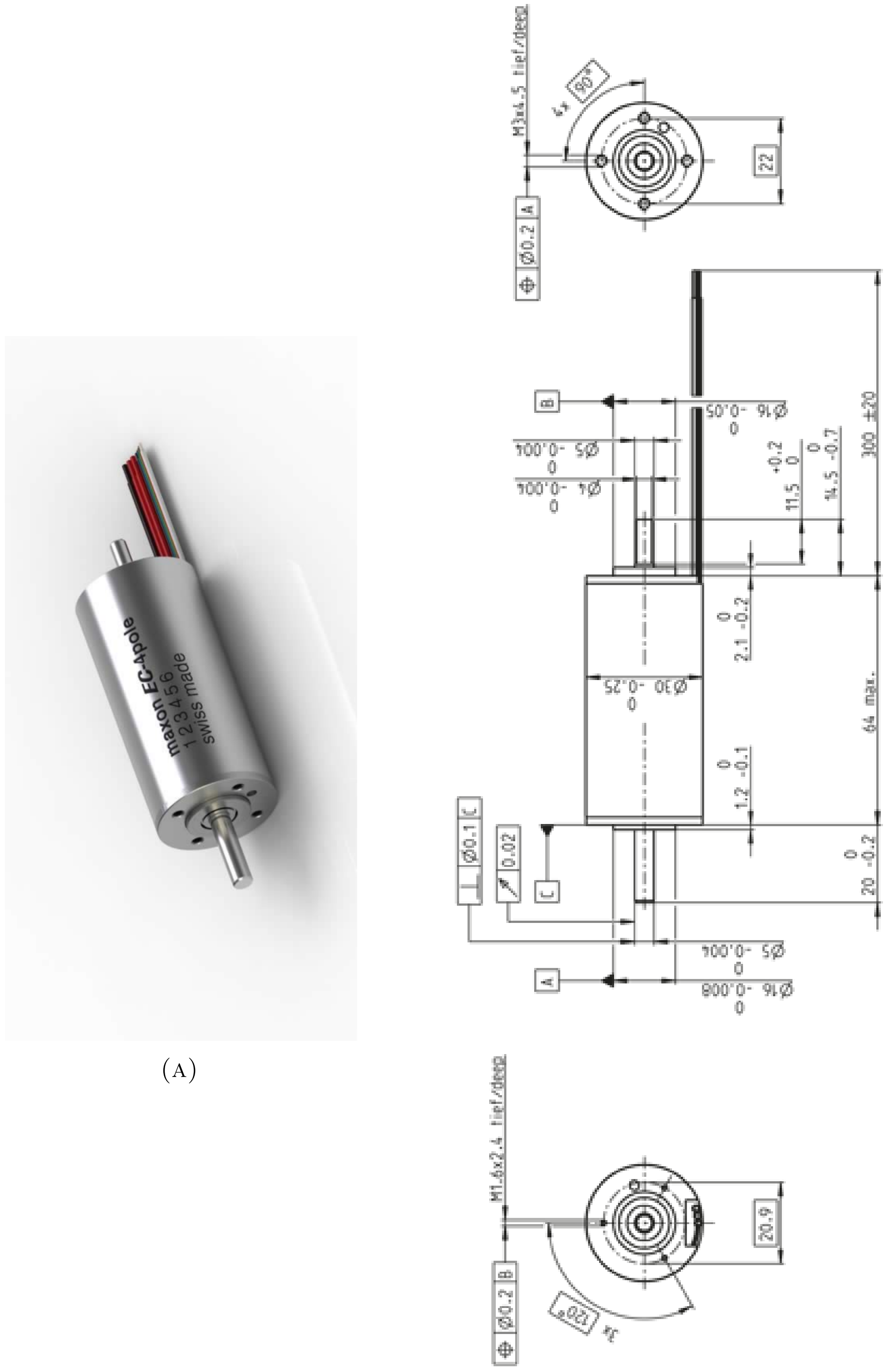
T_e Electromagnetic torque.

This measurements are useful for the next blocks to monitor the speed and allow its control.

[1] The resistive load is proportional to the sample time . As reference value in the help it is remarked that with a $25\mu s$ time step on a 60 Hz system, the minimum load is approximately 2.5% of the machine nominal power.

Parameter	**Value**	Notes
Number of phases (stator)	3	
Back EMF waveform	Trapezoidal	
Stator phase resistance	0.102Ω	
Stator phase inductance Ls	$16\mu H$	
Voltage constant	$1.429V_{peak}/kRPM$	Peak line to line voltage per 1000 rpm. Calculated as the ratio between the nominal voltage and speed on the datasheet.
Back EMF flat area	120^o	It is the width of the flat top for a half period of the electromotive force.
Inertia	$33.3e^{-7}kg \cdot m^2$	
viscous damping	$7e^{-6}N \cdot m \cdot s$	This value is not reported on the datasheet. It has been tuned by attempting to achieve the no load current of $723mA$.
pole pairs	2	-

Table 2.1: PM BLDC Motors Simulink model parameters.



(A)

(B)

Figure 2.3: EC-4pole 30, prod. n^o 305013, Maxon Motor

Motor PWM Driver

The *Motors PWM Driver* is the Simulink block which receives the hall sensors and the control signals (*ctrl_signal* fig. 2.5) as inputs, and gives the gates driver signals (*PWM_g*) as outputs. The hall signals are provided directly by the *Permanent Magnet Synchronous Machine* and allow to determine the rotor position. Usually, motors are equipped with three hall sensors, equally spaced of 120^o. Each sensor produces per each rotor round a number of squared signals equal to the number of pole pairs. In this specific case, hall sensors produce two rising edges per round and the phase between signals corresponds to the physical angle between sensors. It is possible to keep on the rotor running by commutating between states, correctly interpreting the halls signals.

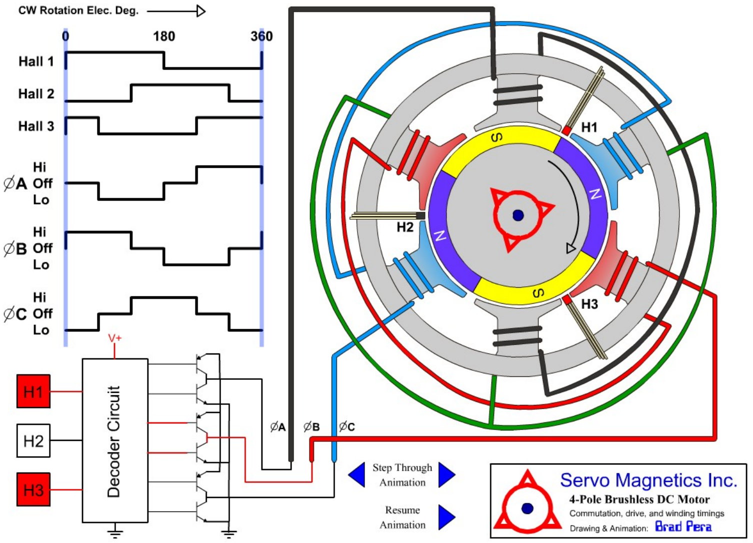


Figure 2.4: Commutation, drive and winding timings of a 4-Pole brushless DC Motor. [41]

In order to transform the hall signals into logical state for the MOSFET transistors, the driver block perform two main functions:

- ▷ First of all, the hall signals are interpreted as a rotor position information, represented with the *emf* three logical states (-1, 0 and 1) and finally transformed into a boolean signal able to drive the MOSFETs. The transistors conveys properly the current in the stator windings, by acting as electronic switches. The gate signal produced is able to drive the motor only in saturation conditions[2].

- ▷ In order to generate intermediate supply states depending on the speed control loop, the primitive squared wave is mixed with the PWM signal. It

[2] maximum voltage/speed

allows intermediary voltages in the range of 0 to V_{DC} (24V), by varying the duty cycle.

This two functions are carried out by the simulink diagram in the fig. 2.5.

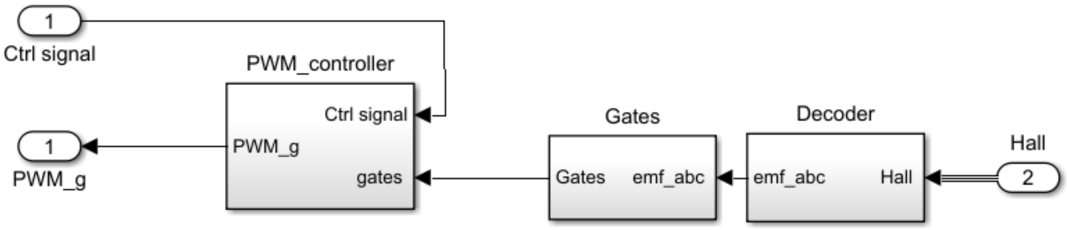


Figure 2.5: PWM motor driver block.

The *Decoder* and *Gates* blocks accomplish the first objective of the driver. These are digital circuits that implement the two truth tables in fig. 2.7. This logical relations can be verified on the stylized BLDC motor and its driver circuit in fig. 2.6. The second function of the driver block is achieved by implementing a PWM generator and mixing its signal with the gates commands. The PWM (Pulse-Width Modulation) is a modulation technique implemented also for the electronic control of the power. This technique is widely used to control voltages level due to its efficiency and easy implementation in controls, by setting the duty cycle [16].

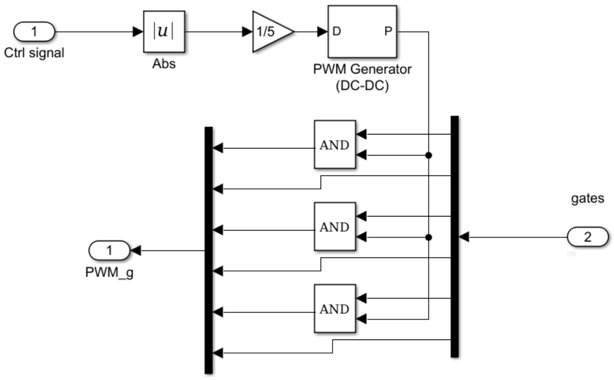
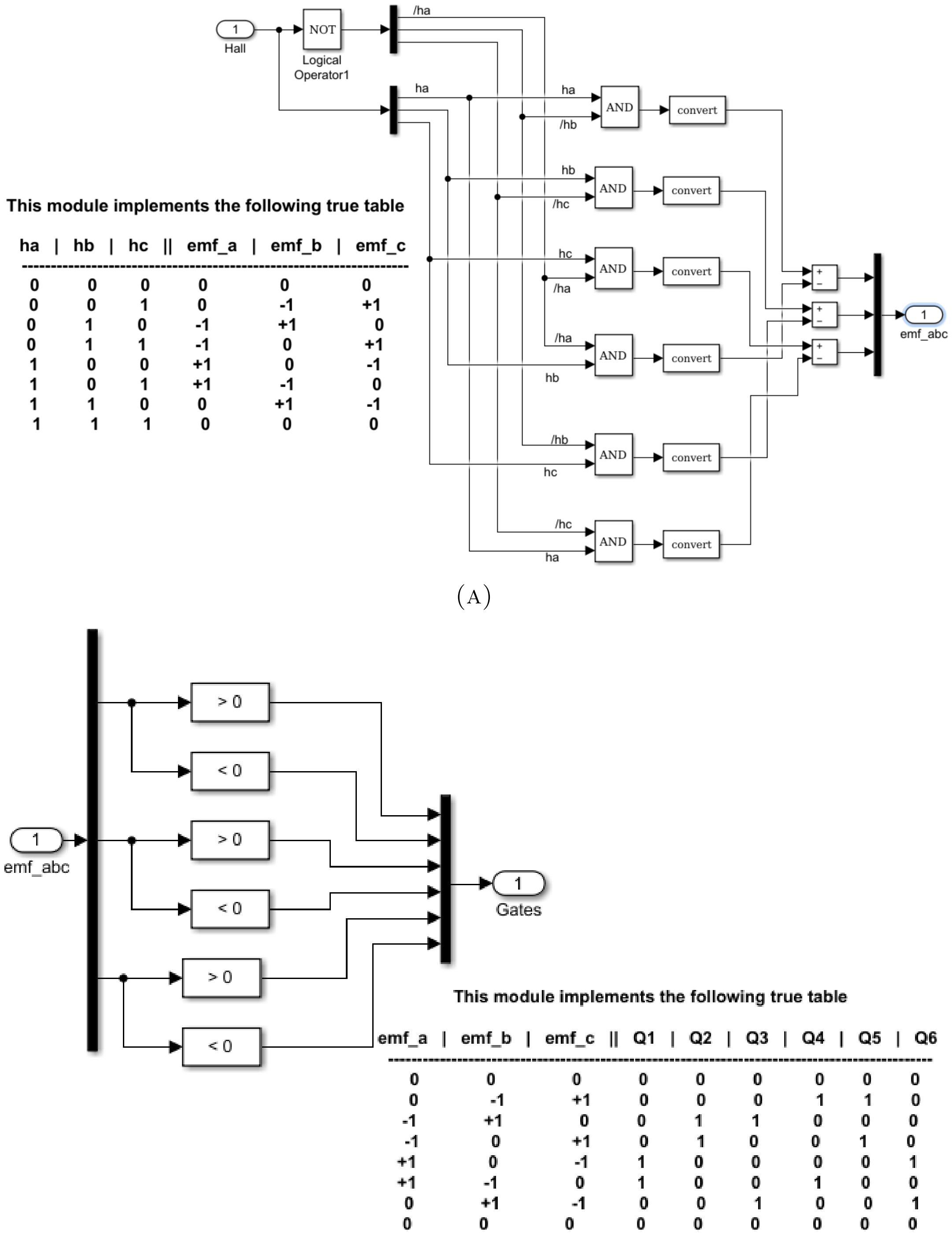


Figure 2.6: *PWM_controller*. Digital circuit for the mixing of gates driver commands with PWM signal controlled by the *ctrl_signal*.

The electrical power depends on the average value of the voltage and current fed to the load. The voltage average value in turn is determined by the relative duration of off and on states. The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percentage, 100% being fully on. The longer is the on period the higher are the duty cycle and the total power supplied to the load. In order to get the resultant waveform as smooth as possible, the PWM switching frequency has to be much higher than the characteristics system frequency, which in this case depends on the load characteristics and the rapidity of speed control loop. The maximum PWM frequency is limited by the capacity of the processor; two representative values have been chosen: $20kHz$ and $150kHz$.

The digital circuit in fig. 2.6 mixes the command of gates with the PWM signals. The *PWM Generator (DC-DC)* block, available in the simulink library, produces the squared wave whose duty cycle is proportional to the *ctrl_signal*, generated by the speed control loop[3]. The PWM signal is applied only on the upper transistor of each leg of the bridge, while the lower transistor is driven directly by the gate signal. This choice is necessary for further development of the system in sensorless applications, because it allows a better estimation of the back EMF. This observation has to be reconsidered during the implementation of the control for a real motor, because the filtering performances, used to determine the bemf, are strongly influenced by the overall system features.

[3] The *ctrl_signal* is in the range of 0 to $5V$, this why it has to be normalized in the range of 0 to 1



This module implements the following true table

ha	hb	hc	emf_a	emf_b	emf_c
0	0	0	0	0	0
0	0	1	0	-1	+1
0	1	0	-1	+1	0
0	1	1	-1	0	+1
1	0	0	+1	0	-1
1	0	1	+1	-1	0
1	1	0	0	+1	-1
1	1	1	0	0	0

(A)

This module implements the following true table

emf_a	emf_b	emf_c	Q1	Q2	Q3	Q4	Q5	Q6
0	0	0	0	0	0	0	0	0
0	-1	+1	0	0	0	1	1	0
-1	+1	0	0	1	1	0	0	0
-1	0	+1	0	1	0	0	1	0
+1	0	-1	1	0	0	0	0	1
+1	-1	0	1	0	0	1	0	0
0	+1	-1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0

(B)

Figure 2.7: Simulink model of the *Decoder* and *Gates* blocks.

Three-phase power stage

Once synthesized the gates commands, the logical signal is transformed into a power-carrying signal with a *three-phase power stage*, which is composed of six power MOSFET transistors. They energize only two motor phases at the same time. In order to smooth the high commutation frequency, the power stage is equipped with a snubber circuit.

"A snubber is an electrical device that prevents voltage spikes due to sudden changes in current. These voltage spikes, or transients, can damage the circuit and cause arcing and sparks. One type of electrical snubber is the RC snubber, which is composed of a resistor in parallel with a capacitor. Transients are usually caused by switches in the circuit" [45].

The RC snubber design depends on the type of switch and its switching frequency. Several methods are proposed at the websites of electronics manufacturer[4]. This methods might be considered during the implementation of the control on a real motor for an optimal design. For the model in matter, the R and C values are chosen reasonably with the following empirical formulas, without further analysis:

$$R_{snubber} = R_{statorphase}/2$$

$$C_{snubber} \sim L_{statorphase}$$

The RC snubber and transistors bridge are both implemented int the *Universal Bridge* simulink block.

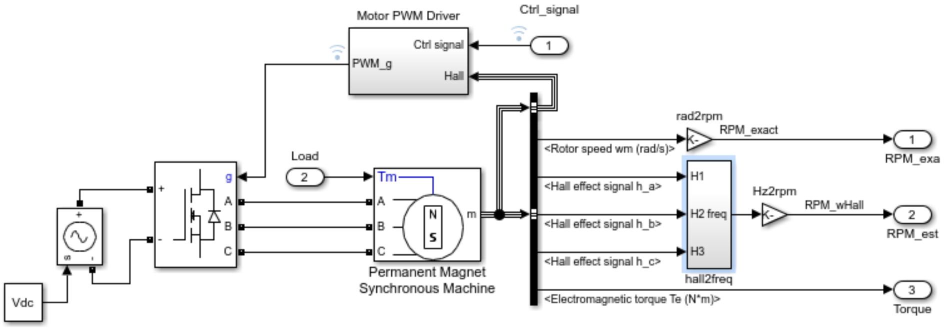


Figure 2.8: *BRUSHLESS DC & DRIVER* block. Overall inner loop model of a six-step controller

.

[4] An example can be found at daycounter.com

2.1.2 Speed estimator

The number of sensors has to be limited in order to reduce the costs of the system. For this reason instead of using a tachometer, the hall sensor signal can be exploit to estimate the speed. In order to achieve a sufficient sensibility of the system, the three hall signals are summed into a unique signal, which has the triple frequency of the sources. This solution makes system faster detecting variation of the rotor speed, essentially to do not limit the performances of the speed control during transient intervals. The reliability of the control is negatively affected by this design; then a fault detection routine have to be considered in order to exclude the damaged sensors. Once the three signals are summed into one, a digital frequency estimator provides a calculation of electrical frequency of the incoming squared signal, by measuring the period between two successive rising edges. Because of its binary logic, the summed signal has to attempt the 0 to 1 range; this is achieved by subtracting the constant 1 to the incoming signal. If the hall sensors generate a squared wave with $\delta = 50\%$, the rapidity of frequency estimator can be doubled by measuring the period between successive rising and falling edges. The frequency calculated is not the physical speed of the rotor. The following factors should be considered:

- The hall signal has an electrical frequency, that is $f_{electrical} = N_{polepairs} f_{physical}$ where $N_{polepairs}$ is the number of the pole pairs.
- The hall signals are summed, tripling the electrical frequency.
- If the period is calculated between a rising and consecutive falling edge, the frequency is doubled.

In the fig. 2.9 is shown the implementation of the speed estimator in simulink.

2.2 Trapezoidal control sensorless

"In recent years, many household electric appliances use brushless DC motors as mechanical power due to their good properties, such as high efficiency, easy maintenance, and low noise. However, the BLDCM has several shortcomings in the speed control application, for example: the additional sensors which cause undesired restrictions, such as the limitation of the high temperature environment and of the available space for the hall and speed devices. In industrial fields, the researchers invested a lot of time and money to find solution for these problems" [33].

The sensorless configuration is desirable for reasons of costs, reliability and mechanical packaging. The rotor speed and position are monitored by sensing back voltage and/or current in the motor phase winding. These measurements are polluted by the noise, this is why they require a narrow band pass filter in

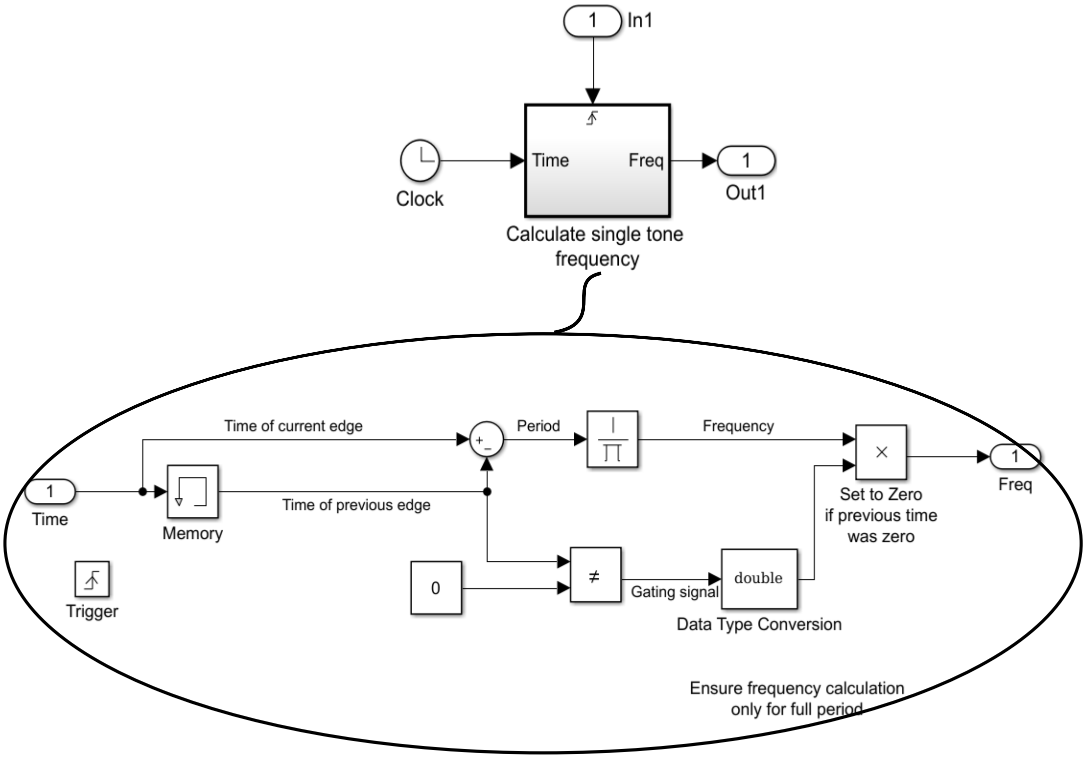


Figure 2.9: Digital circuit for the estimation of the rotational speed, exploiting the hall signals.

order to be used as useful information. Filtering properly the signals and the mathematical processes necessary to interpret them deeply increase the system complexity. Then, a detailed study of costs and performances has to be carried out to verify if the sensorless configuration is actually the most feasible solution. Two types of sensorless control technique can be found in the literature [9] [12]. One type determines the rotor position using the zero-cross detection of back EMF, while the second is a motor-model based technique. The easiest and cheapest solution is developed as primitive implementation. In fact, the second method estimates the motor-model parameters at each controller time step from its constant parameters, terminal voltages and currents, making really hard to adapt this solution to a low-cost control. With further studies, the model can be easily adapted to a more complex and precise solution, if an higher-performances control is required for a sensorless application. The overall model of the six-step sensorless controller is in figure 2.10. It consists in four main blocks:

- ▷ **Motor & Driver**. It is the same implemented in the basic trapezoidal controller.

- ▷ **Nominal Working Block**. It generates the virtual hall signals once the

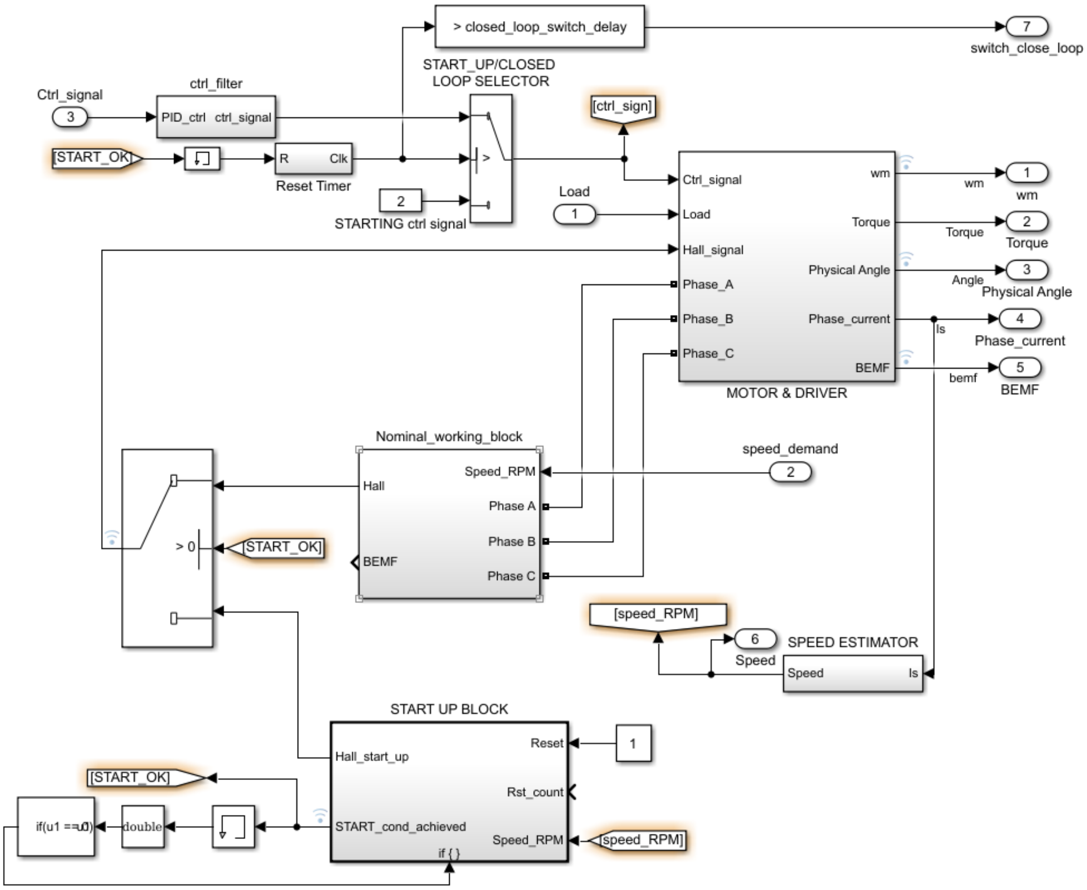


Figure 2.10: Overall model of the six-step sensorless controller in Simulink.

start up phase is completed[5].

- **Start-up block**. It produces the virtual hall signals during the start-up phase, based on a parabolic increasing rotor angle.

2.2.1 Virtual Hall sensors

A close relation exists between the hall signal and the back electromotive force produced by the rotational magnetic field, that induces a current flow in the stator windings. This relation can be visualize in the fig. 2.11 where are plotted the back EMF in the phase A and the signal of the hall A. The phase between the rising edge of the squared signal and the zero-crossing point of BEMF is 15^o, and depends on the relative position between stator winding of phase A and the hall sensor. Since the truth table used by the driver is maintained for the sensorless configuration, this delay has to be considered to build up correctly the virtual hall signal. The exact determination of the BEMF in each phase is

[5] *Start_OK* signal commutates from 0 to 1 value

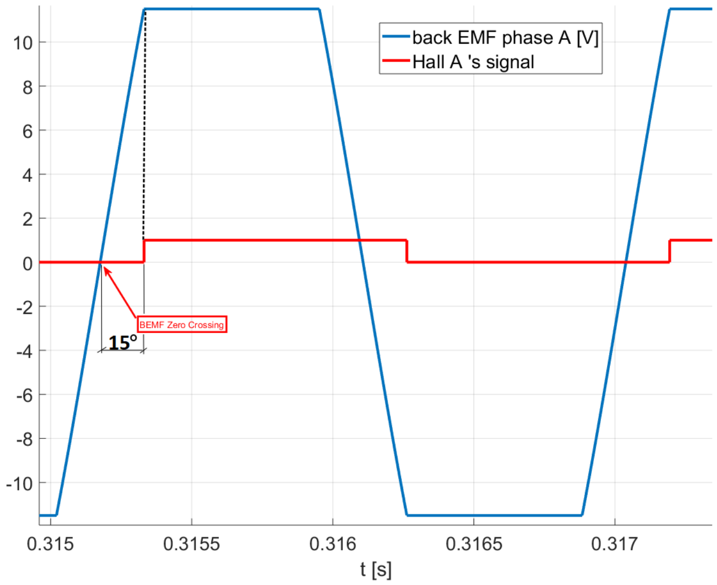


Figure 2.11: Comparison between back EMF in the phase A and the signal of the hall A, determined by the six-step controller with sensors model described in the previous section.

fundamental for a proper commutation. Due to the coexistence of induced and supplied current, the BEMF cannot be determined directly from the phase to phase voltage measurement. Further observations have to be considered to finally detect the zero-crossing point, and consequently the rising edge of the virtual hall signal.

The back EMF voltage can be measured using the method proposed by Uzuka [8] which consists in building a virtual neutral point, equipotent to the center of a Y wound motor, and then measuring the voltage difference between this point and the floating terminal. Unfortunately, the neutral voltage is constantly jumping from zero up to near DC bus voltage, because of the superimposition of the PWM signal on the neutral voltage. This large amount of electrical noise has to be filtered to sense the BEMF properly. The low pass filter causes undesired delay and attenuation in the signal. The poor signal to noise ratio introduces a minimum speed at which the BEMF can be measured, making impossible to use this technique for low speed application and at the start-up of the motor. The design of filters is optimized for the nominal speed (16100 RPM), but as the rotor speed increases the delay do the same, accordingly with the low pass filter characteristics. This delay disturbs current alignment with the BEMF, causing severe problems for commutation at high speed. Consequently, this method tends to have a narrow speed range.

Two alternatives to the classical zero-crossing detection techniques are proposed in literature to reduce the switching noise: the BEMF integration [10] and the third harmonic voltage integration [11]. Regrettably, these methods do not solve the low speed operation problem, while decreasing simplicity and low cost, which are the main advantages of the BEMF zero-crossing detection technique.

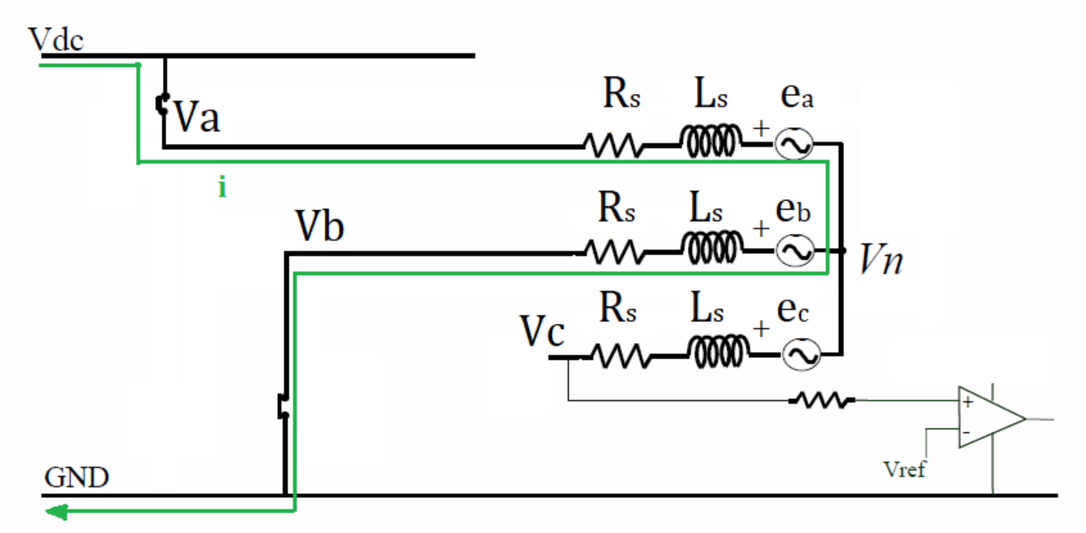


FIGURE 2.12: Model of BEMF detection circuit during the PWM off time period of phase C (floating phase).

Since only two phases of the motor are energized at the same time, the BEMF in the third phase can be measured from the voltage at this latter point, during the period it is floating. Considering for example the PWM off time period of phase C, no current flows in the C-phase winding. Hence, from the circuit structure in fig. 2.12, the back EMF in C is $e_c = V_c - V_n$. As already said, the neutral point is not available, than the voltage drop is considered on a fictitious neutral point. The Simulink model for the back sensing and generation of hall signal is shown in fig. 2.13. It consists in 4 fundamental blocks which has the rotor speed and phase terminals as inputs and the estimated BEMF/hall signal as output.

- The *zero_cross_detection* block is repeated per each phase terminal. It consists in a first order low pass filter block [6] tuned with a cut off frequency of about $3183Hz$[7] which is the best compromise between a good attenuation of noise[8] and a minimum delay at higher speed rotor[9]. After the filtering, the hall signals are generated detecting the sign change of the filtered

[6] *bemf_filter* block.

[7] $R = 5k\Omega$ and $C = 10nF$

[8] Some harmonics with lower frequency affects the zero-crossing determination.

[9] The maximum electrical speed admitted by the motor manufacturer is $50'000$ RPM, but in nominal condition the electrical frequency is $537Hz$ which corresponds to $16'100$ RPM of the rotor speed.

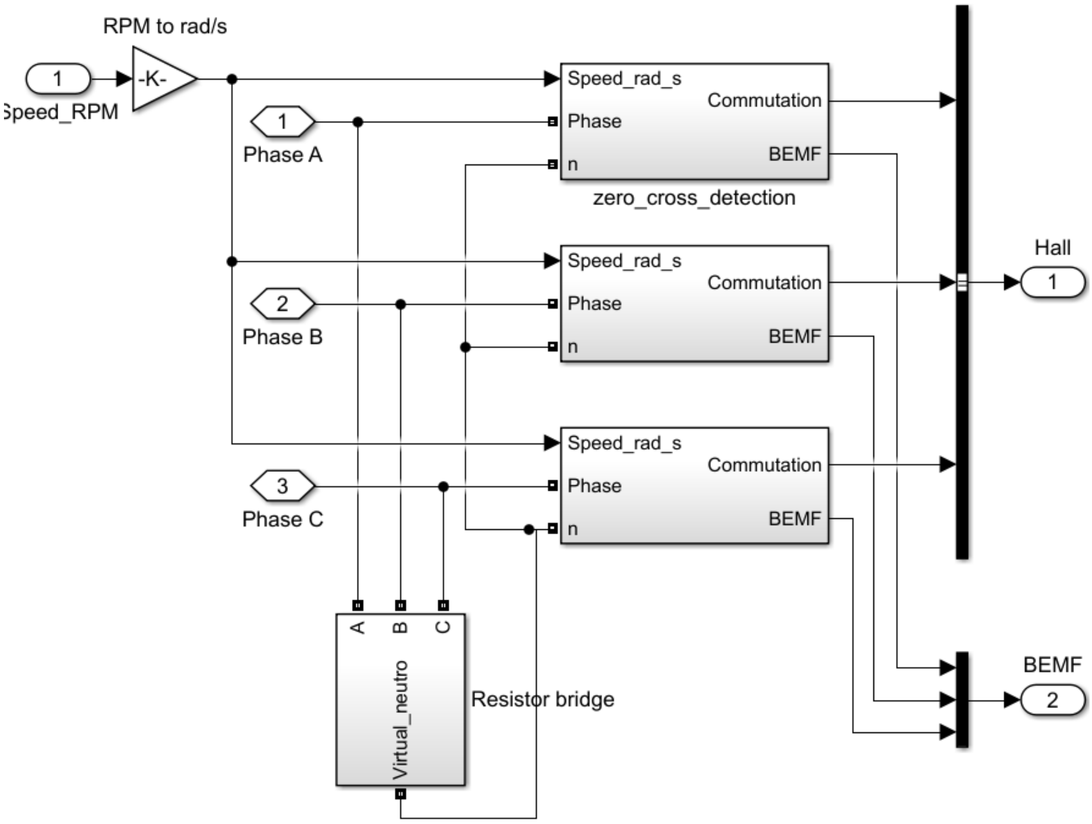


Figure 2.13: Overall Simulink model of back EMF sensing circuit.

phase-to-neutral voltage. A comparator with hysteresis is used to generate the squared signal in order to avoid undesired commutations caused by the residual noise. The thresholds are $0V$ for the switch on point and $-0.6V$ for the switch off point. These values assure a minimum tolerance to the noise peaks at low speed, but at the cost of a longer on-state period of the virtual hall sensors. This fact has to be considered if the speed is evaluated with the *speed_estimator* described in the previous section (2.1.2), excluding the possibility to use either rising and falling edges for a more rapid speed estimation. Finally, to overlap the virtual hall signals to those generated by the sensors, the squared signal is delayed of 15^o with a *Variable Time Delay* block.

- The virtual neutral point is reproduced by three resistors connected to each phase line. The resistance value[10] has to be sufficiently high to do not reduce the efficiency of the system.

[10] $R = 1M\Omega$

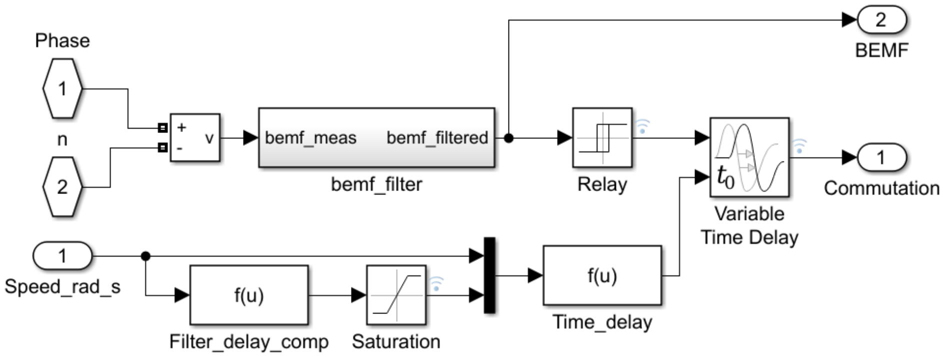


FIGURE 2.14: Simulink scheme of the *zero_cross_detection*, generating the BEMF and the virtual hall signals.

The virtual hall signals estimated with this algorithm have a maximum phase error of around 5^o, in a reasonable speed range of 10000 RPM to 17000 RPM [11]. For both speed range limits, a visual comparison between virtual hall signal and hall sensors signal is reported in fig. 2.16. The speed can be also estimated using the phase current signals, but it gives similar results than using the BEMF wave. Thus, the speed estimation becomes a big limitations for the sensorless closed loop.

2.2.2 Start-up sensorless

The sensorless schemes are not self-starting. In fact, to detect such voltage with a signal to noise ratio acceptable, the motor has to be brought up to a certain speed value in an open-loop configuration. Since the application of the motor is not specified, we imagine to have a constant torque during the start-up. That lets assume a constant acceleration of the stator magnetic field [28]. An external potentiometer can be used to set experimentally the desired torque. The start-up procedure can be considered finished once the speed threshold is achieved. At the same time, a minimum number of stator magnetic field turns has to be assured. This *AND* condition acts like a filter of eventual misleading noises in the speed estimation. In some applications, this start-up strategy could not be satisfactory; in fact there are cases in which the torque load requires a minimum delay before stabilizing its value.
The fig. 2.17 shows a comparison between the assumed rotor angle[12] and the measured rotor angle. The trend obtained demonstrates that this technique can be implemented successfully for the start-up of the motor, integrated with a six-step algorithm. The blue solid line refers to the *Start_OK* signal which commutes to 1 when the difference between the assumed and real rotor angle is

[11] The error is 3.9^o at 10000 RPM and 5.1^o at 17000 RPM

[12] A constant acceleration of $5000 rad/s^2$ is suggested.

less than 4^o. The trend of the error during the entire start-up is shown in the fig. 2.18. Once the start condition has been achieved, the system switches to the nominal functioning control driver. The start-up block is organized in three main blocks:

- **Start up angle**. This block function generates the suggested angle in radians, assuming a constant acceleration of the rotor.
- **Virtual Hall start up**. To generate the virtual hall signals, the physical angle is compared with proper threshold, determined by the correlation between position and hall signal. Such virtual sensors are imaginary placed with a 30^o angle from the three stator windings, in counter-clockwise. The model structure is shown in the fig. 2.15 with a detail of the *Hall_start_up* logic circuit.

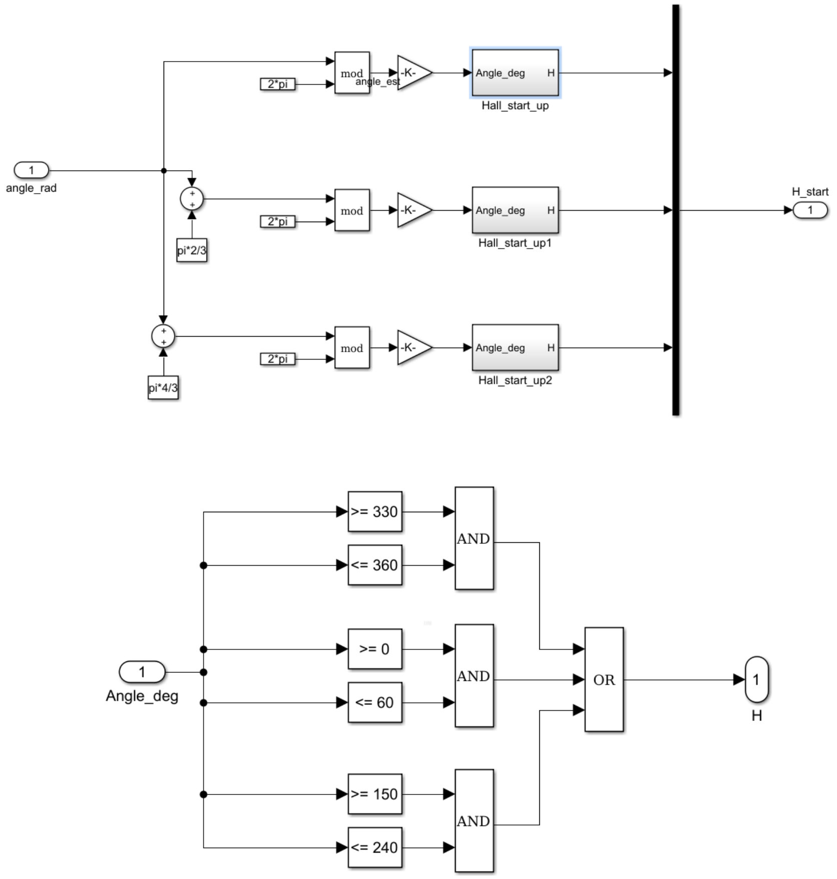
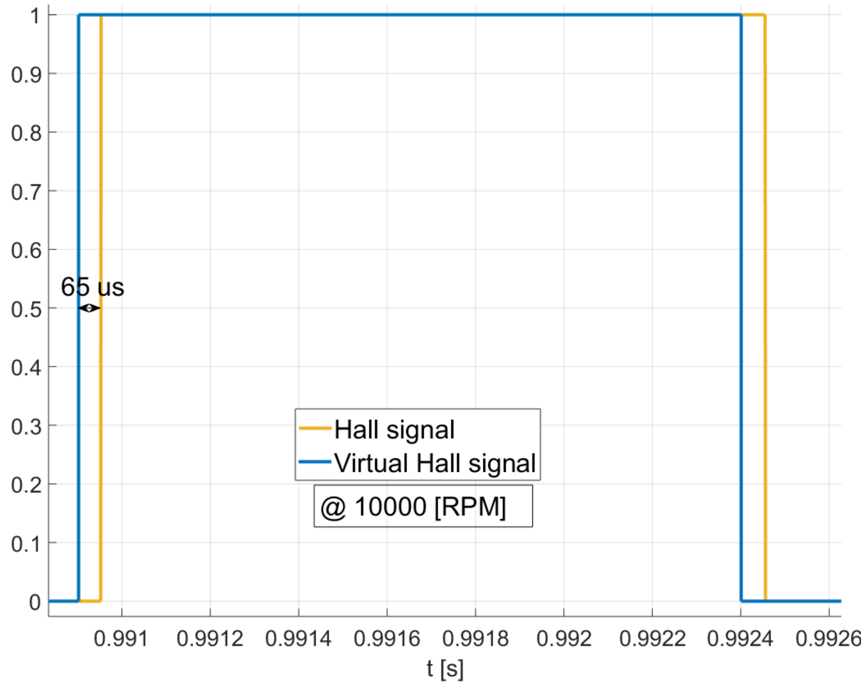


Figure 2.15: Digital circuit which reproduces virtual hall signal during the start-up.

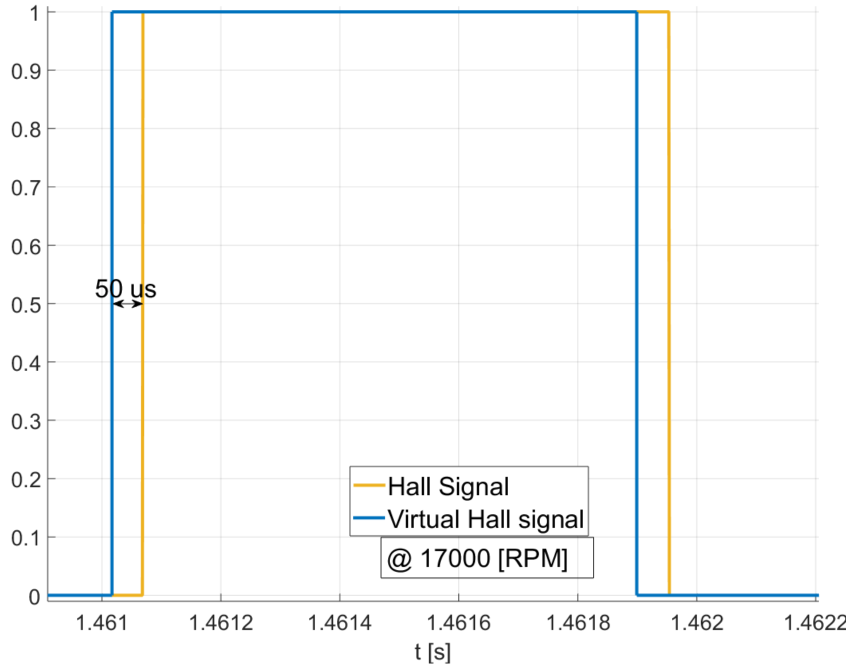
- **Start condition**. This block is dedicated to count the number of turns that a rotor, with the assumed angle, has accomplished. At the same time, it is verifying the speed threshold. The output is 1 once the minimum speed and number of turns are achieved.

2.3 Final Remarks

The six-step controller is relatively easy to implement. The physical laws which regulate its operating principle are well defined and implemented with a logic circuit, and only one control loop is necessary. The structure of the controller becomes more complicated when using only voltage and current sensors for the feedback. This solution should reduce costs related to the hall sensors and increase the reliability of the system. Unfortunately, the system functioning is stronger dependent on the filtering of the feedback signals. That limits the operational speed range. As consequence, an independent start-up routines becomes necessary, limiting even more the range of applications of the same design.



(A)



(B)

Figure 2.16: (A) Virtual hall signal compared to the hall sensor signal at 10000 RPM. (B) Virtual hall signal compared to the hall sensor signal at 17000 RPM.

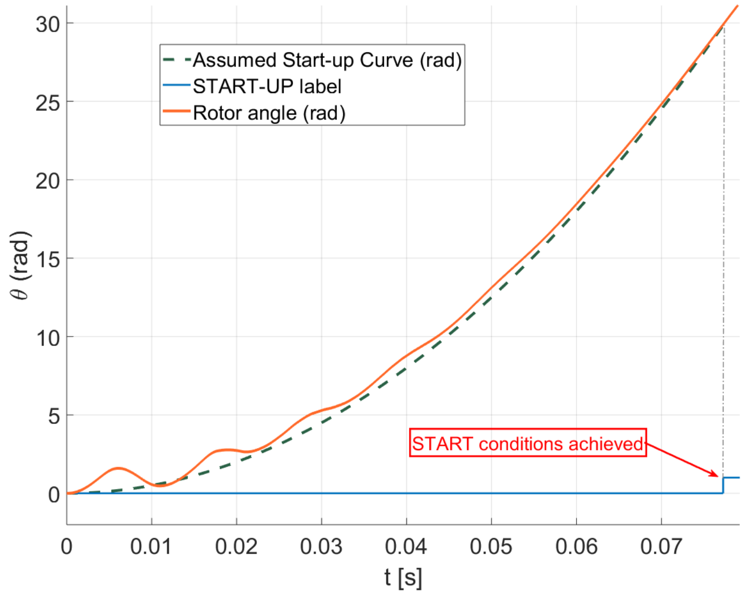


FIGURE 2.17: Comparison of real rotor angle (orange solid line) with the one generated with a constant acceleration of $5000 rad/s^2$ (green dash line) at the start-up. The blue line refers to the *Start_OK* signal.

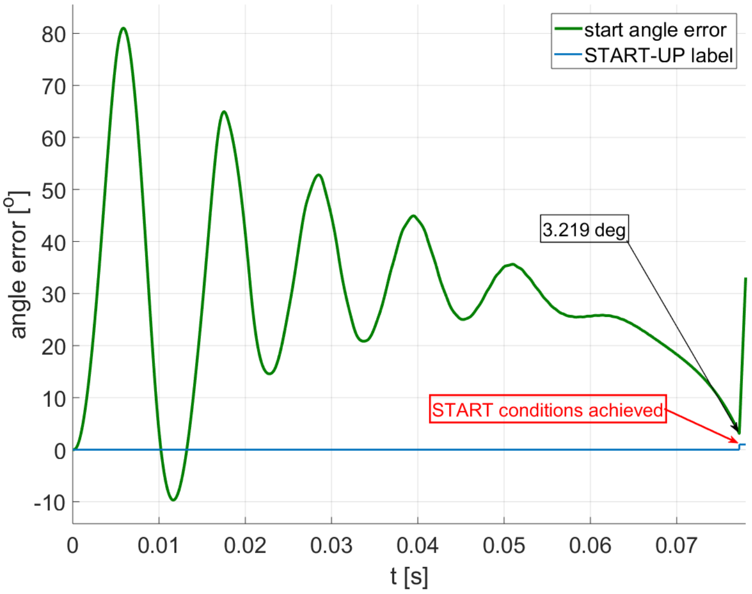


FIGURE 2.18: Mismatch between the estimated and measured rotor angle during the start-up.

Chapter 3
Field Oriented Control

The *Field Oriented Control (FOC)* is used for a more precise control of BLDC motors. This technique reduces drastically the torque ripple and eliminates the current spikes at the commutation. At the same time, it represents an improvement of the sinusoidal control, because it allows good performances also at higher speed [2].
In this chapter the FOC operating principle is explained. It is based on space vector modulation, this is why an entire section is focused on how a three-phase inverter works and which are its advantages. The motor used for the simulation is the same adopted in the chapter of the trapezoidal control. Finally, taking into account the low cost requirement, the sensorless version of the FOC is the final goal. The description of the supplementary blocks, required to replace the position and speed sensors, is reported at the end of the chapter.

3.1 Principle of working

The precise control performed by FOC is granted by the continuous monitoring of the current space vector. The current space vector shares the same direction and a proportional magnitude of the stator magnetic field. The use of a single rotative current vector is useful to simplify the representation of the three real currents in the stator windings. This latter makes easier to design the control, because it is possible to give a physical meaning to the two components of the current vector. In particular, the quadrature component is proportional to the electromagnetic torque, while the direct component represent the magnetizing flux.
In steady state conditions, the current space vector should be constant in magnitude and turn synchronously with the rotor. The efficiency of the system is closely related to the maximization of the torque. As already said, the torque is maximized if the space vector is in quadrature with the rotating magnetic. While no torque will be produced if the two vectors are parallels. Three sinusoids, phase shifted of 120^o, results from the projection of the steady current vector

along the windings directions[1]. The amplitude of sinusoids is proportional to the desired torque.
As consequence, the FOC has to be composed of two feedback loop. The outer loop regulates the speed, and demands a certain torque/current level to the inner loop, which in turn regulates the torque magnitude.

Inner loop

This loop consists in two PI controllers which determine the voltage at the motor phase terminals. The voltage amplitude depends on the current necessary to produce the torque demanded.

- In steady state the torque demanded has to balance the losses produced in the circuit and by the friction.

- In transient states the torque has to give a plus/minus component to produce the acceleration/deceleration necessary to get to the demanded speed.

The phase currents are sensed back giving ideally three sinusoidal signals. These signals are represented by three vectors in a stator-fixed reference system. They are combined into one vector whose components are the direct and quadrature currents, if expressed in a rotor-fixed reference *(d-q reference frame)*. These components give information about torque intensity and magnetizing flux respectively. Hence, they are used as inputs to the pair of PI controllers.
Only two current sensors are required, because the current in the third winding is the negative sum of the other two, accordingly with the Kirchhoff's principle.
The output from each PI controller is a voltage in the range 0 to $5V$. Before providing the voltages to the PWM generator, they are transformed back to the stator-fixed reference system. The inverter will finally drive the transistors bridge.
Managing the current vector directly into the d-q reference frame allows to overcome the problem at high speed, which affects instead the sinusoidal control [2]. In fact, PI controller, with limited gain and frequency response, cannot maintain the quadrature direction with high frequency perturbations. While, using the d-q reference, PI controllers operate on DC rather than sinusoidal signals, since the current vector is static or slowly variable during transient. To manipulate currents and voltages in d-q reference system, a mathematical transformation is required. Then, higher performance of the processor, with consequent higher costs and complexity of the system, will be unavoidable.

[1] The windings are oriented 120^o apart from each other

Outer loop

The outer loop is dedicated to monitor the rotor speed, comparing it to the value demanded, and trying to achieve a zero error by regulating the quadrature current. Once more, a PI controller is used, and it has the speed error as input and the torque demand as output. It is directly connected to the PI controller which regulates the quadrature voltage component. Finally, the direct component regulation is independent of the outer loop, since the d-current has to be zero[2]. The fig. 3.1 shows the conceptual scheme of a FOC control, represented by the following steps [28]:

1. The stator current i_a and i_b are measured, while i_c is calculated applying the Kirchhoff principle $ia + ib + ic = 0$.
2. The 3-phase current is transformed to a d-q reference frame with two sequential transformations:
   - Clark transformation to a two-axis system, providing two sinusoidal currents: i_α and i_β. These, viewed from the perspective of the stator, are sinusoids 90^o phase shifted.
   - Park transformation aligns the $\alpha\beta$ axis with the rotor-fixed reference frame, getting the I_d and I_q currents. The angle calculated at the last iteration is needed for this transformation. At steady state I_d and I_q are constant.
3. The error that I_d and I_q forms with the reference values are inputs of two PIs. V_d and V_q are the output of the controllers provided to the inverter.
4. The new transformation angle is estimated/measured and used for the inverse Clark-Park transformation. The voltages V_d and V_q are transformed back to the stationary reference frame, obtaining a three-phase voltage. Depending on the way SVM is carried out, the inverter algorithm may require only the transformation to $\alpha\beta$ frame.
5. Thanks to the voltage signal obtained at point 3, the algorithm knows the position and the magnitude of the space vector at the next step. Then, the new PWM duty cycle is calculated with the SVM algorithm.

[2] Permanent magnet motors does not need magnetization.

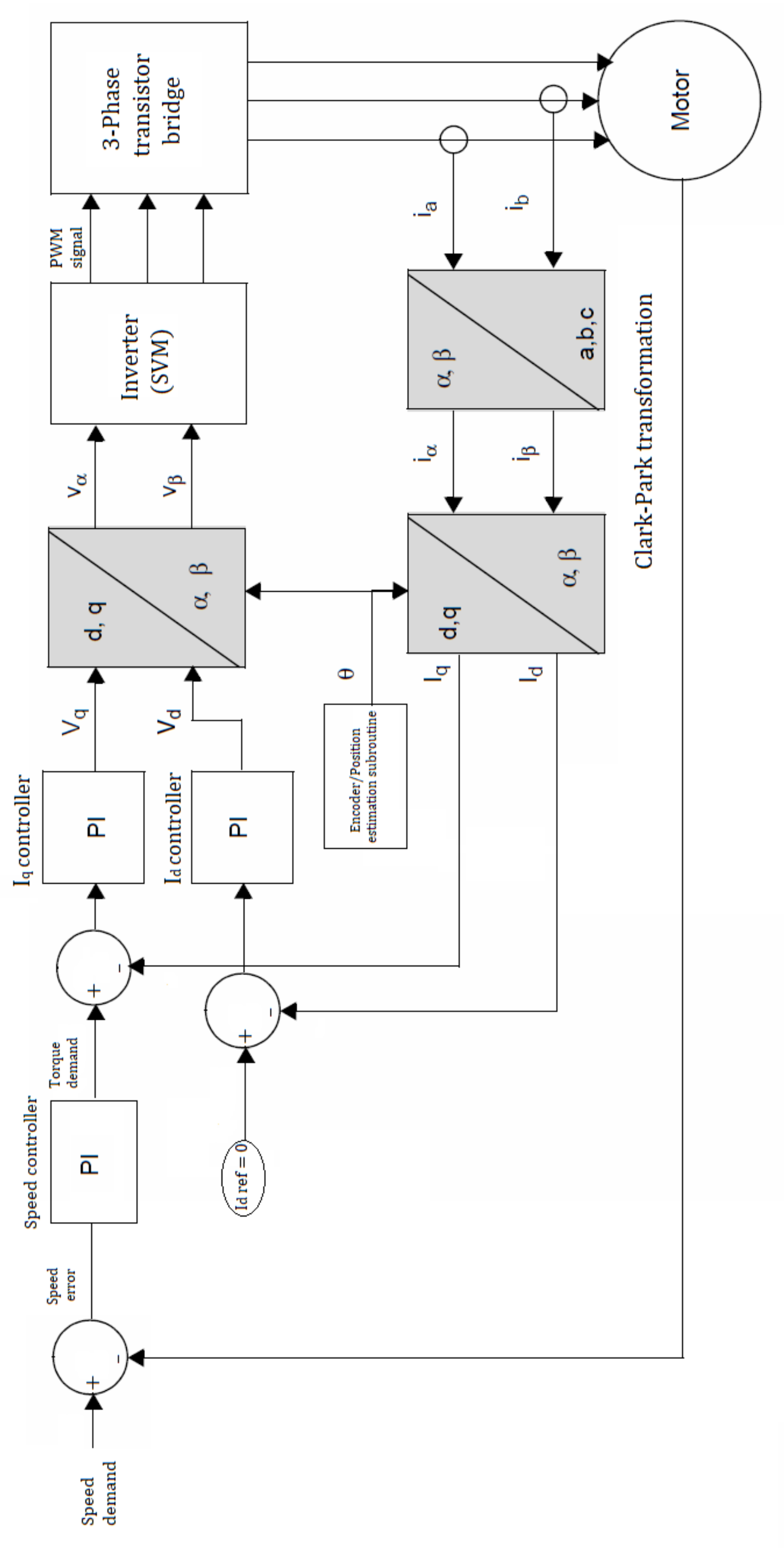


Figure 3.1: FOC block diagram.

3.2 Space Vector Modulation

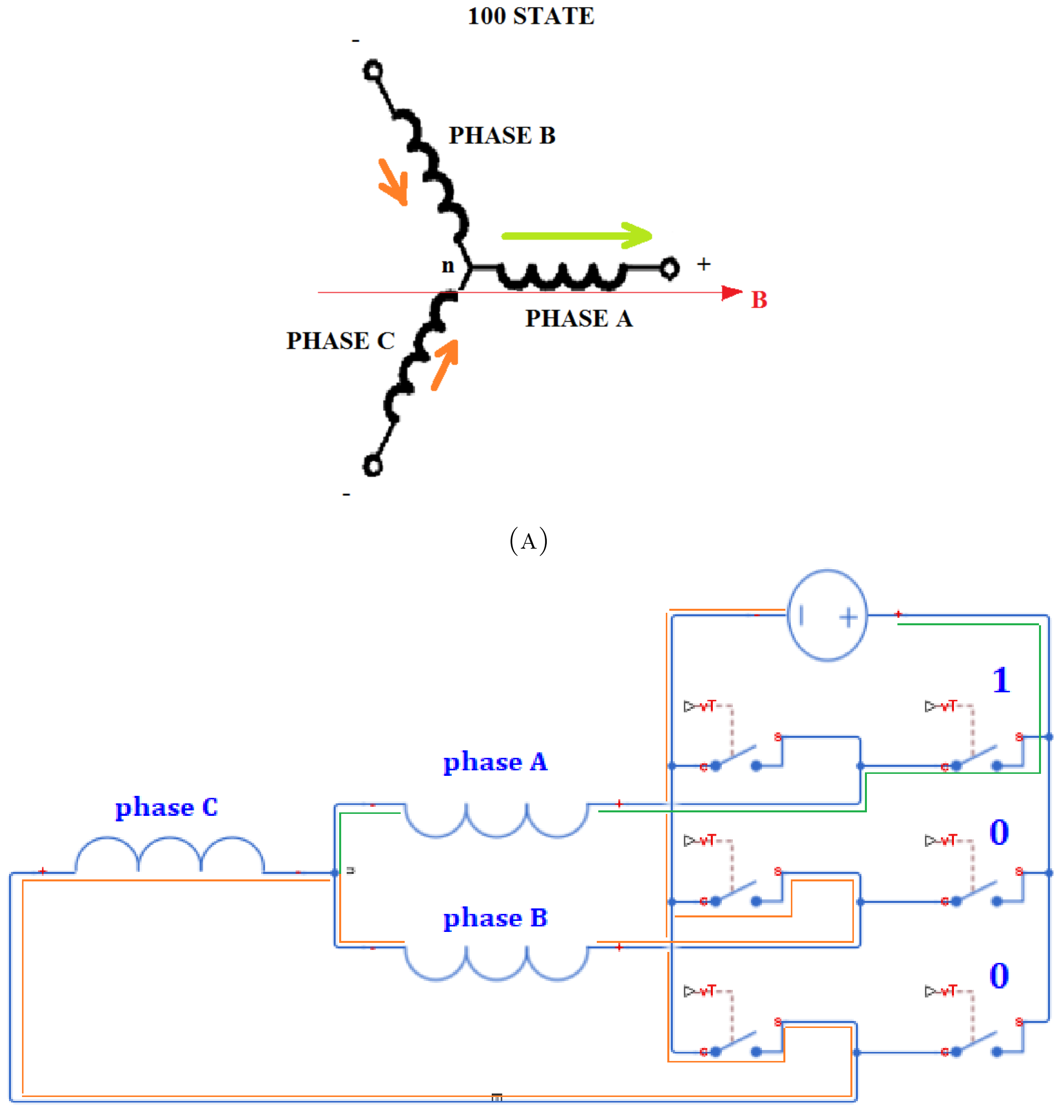
The *Space Vector Modulation (SVM)* is an advanced PWM method. Because of its superior performances, in recent years it has been implemented in several applications. The major benefit is the reduction of the total harmonic distortion created by the rapid switching. DC-AC inverters use this modulation to drive a three-phase transistors bridge. Despite its apparent complexity, it simplifies the control algorithm and allows to obtain a sinusoidal current. $\vec{B}$ traces a perfect circle if the three-phase current is sinusoidal. Then, a sinusoidal waveform is desirable because it reduces the torque ripple. This symmetry is impossible to obtain in practice, but it represents the ideal condition that the design should achieve.

3.2.1 Three-phase transistor bridge

A three-phase transistor bridge is composed of six transistors, two per each phase, and the phases are connected to the intermediate point. These transistors are driven by the inverter with logical commands. Before considering the overall operating principle, it is essential to study what each combination of states causes. The top switch ON corresponds to the sate 1. The bottom and top switches cannot be ON at the same time to avoid a short-circuit. The possible states per each phase are two, that means 8 combinations.

In the fig. 3.2a is represented the *1-0-0* state which corresponds to connect only the phase A to the positive of the DC generator. In the figure 3.2b it is shown that current follows the green path until the neutral point. There, it is equally divided[3] into phases B and C. Considering the left-hand rules and that the current flow in the phases B and C has opposite direction than in phase A, the resultant flux $\vec{B}$ is the one represented in the fig.3.2a. As we can see, the state *1-0-0* corresponds to a $\vec{B}$ parallel to the direction of winding A. States *0-0-0* and *1-1-1* are called inactive or zero states; the other five active states can be solved with the same technique. For this reason FOC method is also called six-step control. The tab. 3.1 is a resume of the $\vec{B}$ orientations related to the logical state, graphically represented in the fig. 3.3. The six active states divide the diagram into five sectors of 60^o each. The deep difference between FOC and trapezoidal control is related to how current/flux vector commutes from one state to the other. In the trapezoidal control the commutation between two next states is instantaneous. It is not efficient at all, in fact the current will not be sinusoidal, high current peaks occur and the stator magnetic field is not in quadrature with the current vector. While the space vector modulation allows a smooth commutation from one state to the next. For a two level inverter, all the six vectors lie along the radii of a hexagon. Inside the hexagon,

[3] Taking into account the hypothesis of perfect symmetry



(A)

(B)

Figure 3.2: A)Instantaneous position of current vector corresponding to the logical state 100 in the three-phase transistor bridge. B) Current path in the three-phase transistors bridge, corresponding to the logical state 100.

various circle of operation can exist. The rotating voltage vector is indicated as Vs, and for the SVM algorithm it is sampled faster than the frequency of its fundamental harmonic. Unfortunately, the frequency cannot be as high as we want, because losses increase with frequency. The sampling frequency of an inverter corresponds to the frequency of a triangle signal, that is compared with the three phase voltage wave, generating a PWM signal with a variable duty

Truth Table	
State	$\vec{B}$ **orientation***
000	Zero-state
001	240^o
010	120^o
011	180^o
100	0^o
101	300^o
110	60^o
111	Zero-state

Table 3.1: *The orientation of instantaneous $\vec{B}$ is the angle referred to phase-A coil position.

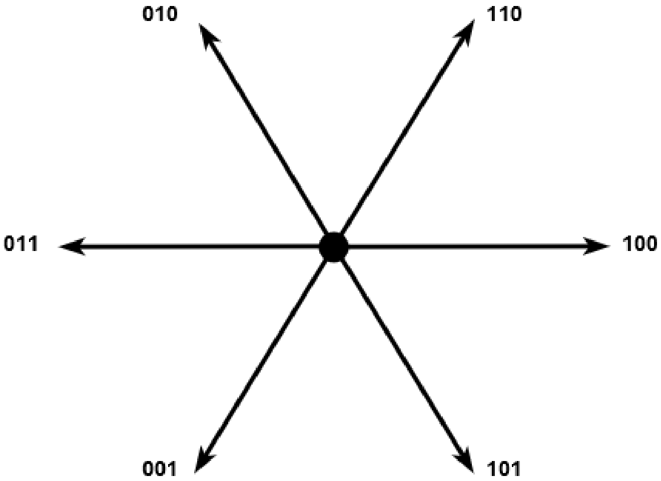


Figure 3.3: Truth table of three-phase transistors bridge.

cycle. The sampling frequency determines how smooth is the commutation. The rotor angle and velocity are assumed stationary during a sampling period.

With the SVM, an arbitrary voltage vector can be produced combining the eight standard states. It is obtained by switching continuously between the two states which delimits the sector where the Vs is (fig. 3.4). An example is proposed to explain it. We assume that the vector to be realized $\vec{V_s}$ is located in the first sector. The first sector is defined as the area between the standard vectors $\vec{V_1}$ (*1-0-0* state) and $\vec{V_2}$ (*1-1-0* state) represented in the fig. 3.4. $\vec{V_s}$ can be obtained

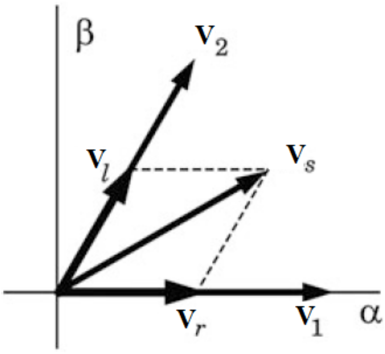


FIGURE 3.4: Realization of an arbitrary voltage vector from two boundary vectors.

from the combination of the two boundary vectors $\vec{V_r}$ and $\vec{V_l}$ in the directions of the two standard vectors [30]. Then, vectors $\vec{V_r}$ and $\vec{V_l}$ corresponds to $\vec{V_1}$ and $\vec{V_2}$ if the electrical power is supplied for the entire available period T_p. In fact $\vec{V_1}$ and $\vec{V_2}$ have the maximum voltage amplitude, which can be calculated with eq. (3.1).

$$|\vec{V_s}|_{max} = \frac{2}{3}V_{DC} \tag{3.1}$$

The switching times of the two states in matter, T_r and T_l, must be calculated as function of the entire available period, supply voltage and desired amplitude of the boundary vectors.

$$T_r = T_p\frac{|\vec{V_r}|}{|\vec{V_s}|_{max}}; \qquad T_l = T_p\frac{|\vec{V_l}|}{|\vec{V_s}|_{max}}; \tag{3.2}$$

In the remaining period $T_p - (T_r + T_l)$, one of the two zero states, $\vec{V_0}$ or $\vec{V_7}$, is issued, fulfilling the following equation:

$$\vec{V_s} = \vec{V_r} + \vec{V_l} + \vec{V_{0/7}} = \frac{T_r}{T_p}\vec{V_1} + \frac{T_l}{T_p}\vec{V_2} + \frac{T_p - (T_r + T_l)}{T_p}\vec{V_{0/7}} \tag{3.3}$$

Even if theoretically the sequence of switching is not relevant, in practice is preferred a sequence that requires to switch every transistor pair only once within a pulse period, to reduce the number of switching as much as possible. If the last switching state was u_0, this would be the sequence:

$$\vec{V_0} \rightarrow \vec{V_1} \rightarrow \vec{V_2} \rightarrow \vec{V_7}$$

but if the last switching state was u_7, this other would be the sequence:

$$\vec{V_7} \rightarrow \vec{V_2} \rightarrow \vec{V_1} \rightarrow \vec{V_0}$$

Then, two vectors are actually realized in one pulse period T_p. With this strategy the switching losses of the inverter become minimal [30]. The same process explained for the first sector can be repeated with the others, changing $\vec{V_1}$ and $\vec{V_2}$ with the new border states. The relation between the pulse frequency $f_p = 1/T_p$ and the sampling frequency $1/T$ is based on the principle that the same voltage vector $\vec{V_s}$, has to be realized within at least one or several pulse periods. Thereby, it is possible to find a suitable ratio between pulse frequency and sampling frequency. This latter is upper limited by the computing power of the microcontroller.

3.2.2 SVM practical application

Once the modulus and phase angle of the voltage vector are known, the inverter has to decide how and how long the transistor pairs have to be switched. The way phases have to be activated depends on the position of the voltage vector, i.e. its phase angle. While the calculus of the on-period is a bit more complex. From eq. (3.2) it is obvious that the calculation of the switching times T_r, T_l depends only on the information about the moduli of the two boundary vectors u_r, u_l. The voltage vector can be expressed in the two reference systems:

- ▷ **dq coordinates**. In this case, the total phase angle is the sum of the rotor angle ϑ_s and the angle that $\vec{V_s}$ forms with the d-axis.

$$\vartheta_u = \vartheta_s + arctan\left(\frac{V_{sq}}{V_{sd}}\right) \tag{3.4}$$

- ▷ $\alpha\beta$ **coordinates**. This representation contains the information about the phase angle, implicitly in the components.

Therefore, for the calculation of the boundary components two strategies can be used:

- ▷ The phase angle ϑ_u is calculated from eq. (3.4), after that the angle γ (fig. 3.5) is deduced. γ represents the angle ϑ_u reduced to the first sector.

Then the calculation of the boundary components is performed by using the following formula:

$$|\vec{V_r}| = \frac{2}{\sqrt{3}} \sin(60^o - \gamma); \qquad |\vec{V_l}| = \frac{2}{\sqrt{3}} \sin(\gamma) \tag{3.5}$$

with

$$|\vec{V_s}| = \sqrt{V_{sd}^2 + V_{sq}^2} \tag{3.6}$$

▷ After the coordinate transformation to the stator-fixed components, $\vec{V_{s\alpha}}$ and $\vec{V_{s\beta}}$ are known. Per each sector $\vec{V_r}$ and $\vec{V_l}$ can be calculated using the formula resumed in the table 2.3 in the Quang's book [31]. The application of this strategy seems to be more complicated because of the several formulas involved, but with some simple consideration, it is demonstrated how it can be simplified.

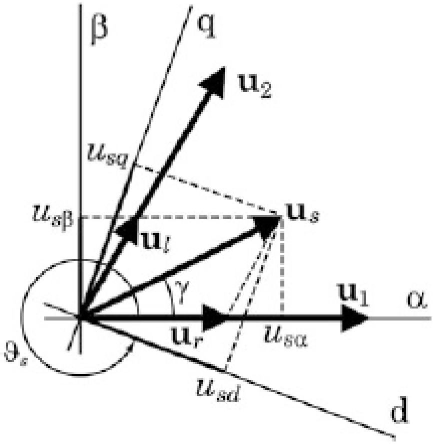


FIGURE 3.5: Possible representation of the desired voltage vector in the two different reference systems.

The SVPWM algorithm can be easly implemented using only the "sampled reference phase amplitudes". To do so, the angle γ is substituted with angle α for a generalization to all the sectors, and not only for the first one. In fact, α is defined as the angle that voltage space vector forms with the initial boundary [4] of the sector in matter. Then, from eq. (3.1), eq. (3.2) and eq. (3.5) the periods T_r and T_l can be rewritten as:

$$\begin{aligned} T_r &= \frac{2|V_S|}{\sqrt{3}V_{DC}} T_S \sin(60^o - \alpha) \\ T_l &= \frac{2|V_S|}{\sqrt{3}V_{DC}} T_S \sin(\alpha) \end{aligned} \tag{3.7}$$

The voltage vectors in the $\alpha\beta$ reference frame can be expressed as a function of phase vectors:

$$\begin{aligned} V_\alpha &= \frac{3}{2} V_A \\ V_\beta &= \frac{\sqrt{3}}{2}(V_B - V_C) \\ V_A + V_B &+ V_C = 0 \end{aligned} \tag{3.8}$$

V_α and V_β are projection of the V_S vector on $\alpha\beta$ axes, then also these identities are valid:

$$\begin{aligned} V_\alpha &= |V_S| \cos(\alpha) \\ V_\beta &= |V_S| \sin(\alpha) \end{aligned} \tag{3.9}$$

Combining equations eq. (3.9) and eq. (3.8), using trigonometric formulas and substituting, T_r and T_l can be rewritten as a pure function of the sampled reference phase amplitudes (T_{AS}, T_{BS} and T_{CS}):

$$\begin{aligned} T_1 &= T_S \frac{V_A}{V_{DC}} - T_S \frac{V_B}{V_{DC}} = T_{AS} - T_{BS} \\ T_2 &= T_S \frac{V_B}{V_{DC}} - T_S \frac{V_C}{V_{DC}} = T_{BS} - T_{CS} \end{aligned} \tag{3.10}$$

The digital implementation of this algorithm is really fast, and T_r and T_l periods can be easily adapted per each sector simply redefining the α angle.

[4] It is necessary to remark that the initial and final boundaries changes in function of the rotation sense

Sector	T_r	T_l
1	$T_{AS} - T_{BS}$	$T_{BS} - T_{CS}$
2	$T_{AS} - T_{CS}$	$T_{BS} - T_{AS}$
3	$T_{BS} - T_{CS}$	$T_{CS} - T_{AS}$
4	$T_{BS} - T_{AS}$	$T_{CS} - T_{BS}$
5	$T_{CS} - T_{AS}$	$T_{AS} - T_{BS}$
6	$T_{CS} - T_{BS}$	$T_{AS} - T_{CS}$

TABLE 3.2: SVPWM sampled reference phase in terms of the sampled reference and phase amplitude.

In the table 3.2, the magnitudes which refers to the standard state, that is the initial bound of the sector in matter, are indicated with the subindex 1, while referring to the other bound with subindex 2. Finally, the algorithm for the space vector PWM can be resumed in these steps [28]:

1. From the control loop, at the output of the inverse Park transformation, the V_α and V_β voltages are known;

2. From the equation eq. (3.8), it is possible to determine the three phase voltages V_A, V_B and V_C;

3. From the equation eq. (3.11) the characteristics times T_{AS}, T_{BS} and T_{CS} can be determined;

4. The maximum T_{max} and minimum T_{min} periods have to be determined, using for example the three element sorting algorithm. It has to be noticed that, for how they are defined, T_{max} and T_{min} are respectively positive and negative.

5. The effective time (time which corresponds to active states) is equal to $T_{eff} = T_{max} - T_{min}$and consequently the $T_{zero} = T_S - T_{eff}$ (period of the zero states).

6. Finally, it is possible to calculate the gating signals during which the top switch is turned on:

$$
\begin{aligned}
T_{ga} &= T_{AS} + T_{offset} \\
T_{gb} &= T_{BS} + T_{offset} \\
T_{gc} &= T_{CS} + T_{offset}
\end{aligned}
\tag{3.11}
$$

where $T_{offset} = T_{zero}/2 - T_{min}$.

3.2.3 Remarks

SVM has become the most popular PWM technique for three-phase inverters implemented in the control of AC induction motors, BLDC motors and switched reluctance [17]. The reasons of this widespread can be related to the following advantages:

- ▷ No look up table is needed;
- ▷ Sector identification is not required;
- ▷ The angle α of the reference space vector, measured from the start of the sector, is not needed.
- ▷ Voltage space vector amplitude is not needed.

Only phase amplitudes (tab.3.2) in a sampling period are needed. This makes the algorithm very easy to implement. Furthermore, the advantage of using the SVPWM instead of the sinusoidal PWM consists in reducing the superior harmonics [17].

3.3 Simulation Model

In the fig. 3.6 is reported the implementation of the BLDC control with FOC using a motor fully equipped with sensors. Then, the position and speed rotor are instantaneously measured very precisely with encoders. The extended model for the sensorless solution is described in the next section.

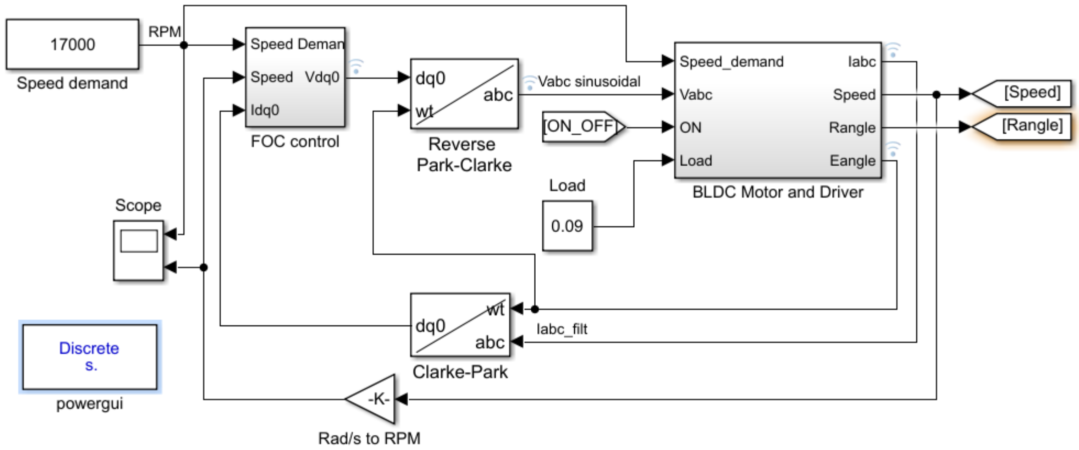


Figure 3.6: Simulink model of implemented FOC technique.

The solutions that uses encoders consists in five main parts:

FOC control block consists in three PI controller, dedicated to generate the quadrature and direct voltage at each time step.

Clark-Park blocks transform directly and inversely between stator fixed and rotor fixed reference frame. Those ideally split the model into a time-dependent (to the right) and time-invariant (to the left) parts.

BLDC Motor & Driver contains the start-up driver and three-phase inverter, besides the motor model and sensors.

Start condition is dedicated to to verify the start conditions[5] and gives the command to switch the system from open-loop to closed-loop.

3.3.1 FOC controller block

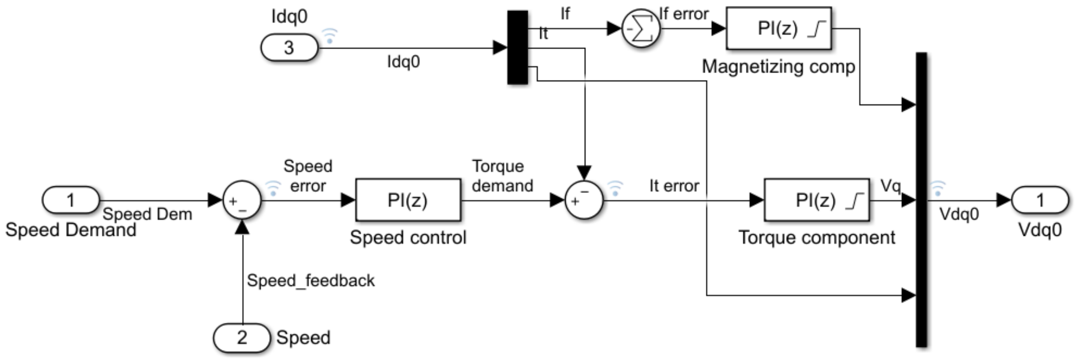


FIGURE 3.7: PI controller used in the FOC Simulink model.

Three discrete P-I controllers are used. One for the direct current component, one for quadrature current, and one for the speed-outer loop control. The input to the *Magnetizing component* controller is the direct current, changed in sign. It attempts to drive the direct current component to zero, This maximizes the efficiency of the system, forcing the current space vector to be exclusively in the quadrature direction. The *Torque component* controller has the requested torque as input. The output of the two PI controllers is a voltage command which represent the space vector in $d - q$ reference frame. The *Speed control* has the speed error as input, and depending on that, it produces the torque command which assures the necessary acceleration or deceleration.

3.3.2 Clark-Park Transformation

This transformation converts balanced three-phase quantities into balanced two-phase quadrature quantities. In literature [3] Clark-Park transformation is

[5] Speed constantly bigger than a certain threshold.

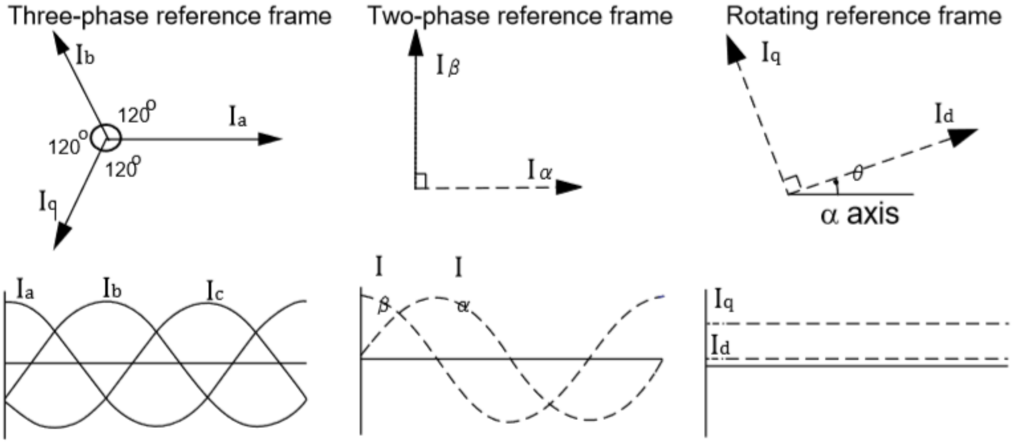


Figure 3.8: Clark-Park transformation graphically represented.

represented in two steps fig. 3.8.
The Clarke transformation translates the three-phase quantities to the stationary reference $\alpha\beta$, using the following equations:

$$\begin{aligned} I_\alpha &= \frac{2}{3}(I_a) - \frac{1}{3}(I_b - I_c); \\ I_\beta &= \frac{2}{\sqrt{3}}(I_b - I_c); \\ I_c + I_b &+ I_a = 0 \end{aligned} \tag{3.12}$$

Park transformation converts the quantities from the $\alpha\beta$ frame into the rotating reference, through the following equations:

$$\begin{aligned} I_d &= I_\alpha \cos(\vartheta) + I_\beta \sin(\vartheta); \\ I_q &= I_\beta \cos(\vartheta) - I_\alpha \sin(\vartheta) \end{aligned} \tag{3.13}$$

Simplifying the equations, the implemented transformation is the following:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} \cos(\vartheta) & \sin(\vartheta) \\ -\sin(\vartheta) & \cos(\vartheta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1\Big/\sqrt{3} & 2\Big/\sqrt{3} \end{bmatrix} \begin{bmatrix} I_a \\ I_b \end{bmatrix} \tag{3.14}$$

3.3.3 BLDC Motor & Driver

The motor model is the same as described in the Trapezoidal Control chapter, while the driver architecture changes. The driver consists in a hall-to-gate digital

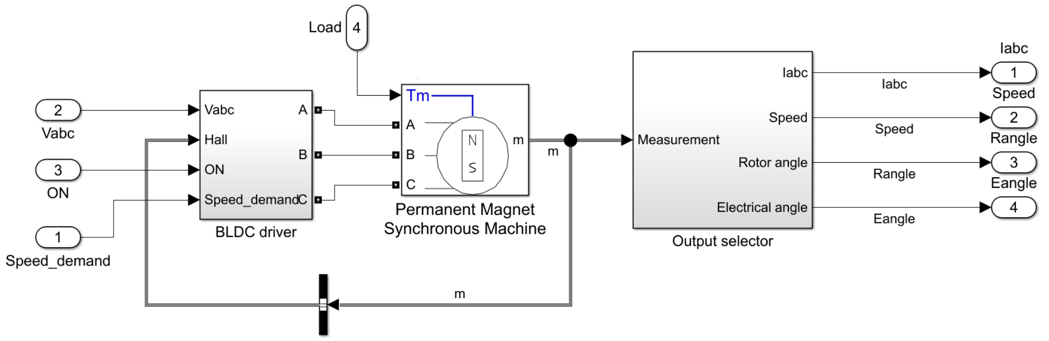


FIGURE 3.9: Motor & Driver in FOC controller.

circuit and a three-phase inverter. The hall-to-gate circuit has been described in the previous chapter; in this case it is only used to start the motor. Once the rotor has a minimum speed acceptable for the FOC control, the driver switches to the inverter. This block is represented in fig. 3.10, already implemented in the Simulink library. It has the three-phase voltage signal and the reference voltage value as inputs, providing the PWM and the modulation signals as outputs. The three-phase voltage comes from the Clark-Park transformation of V_{sd} and V_{sq}. These are compared with a triangular wave, which has the reference value as amplitude ($5V_{DC}$ in this case). It represents an alternative technique to the one described in the section 3.2.2. This algorithm has been used due to the reliability guaranteed by Mathworks. To use properly this block, the switching frequency has to be defined. It corresponds to the triangle signal frequency and it is upper limited by the performances of the microprocessor. The gate signals drive the six-transistors bridge.

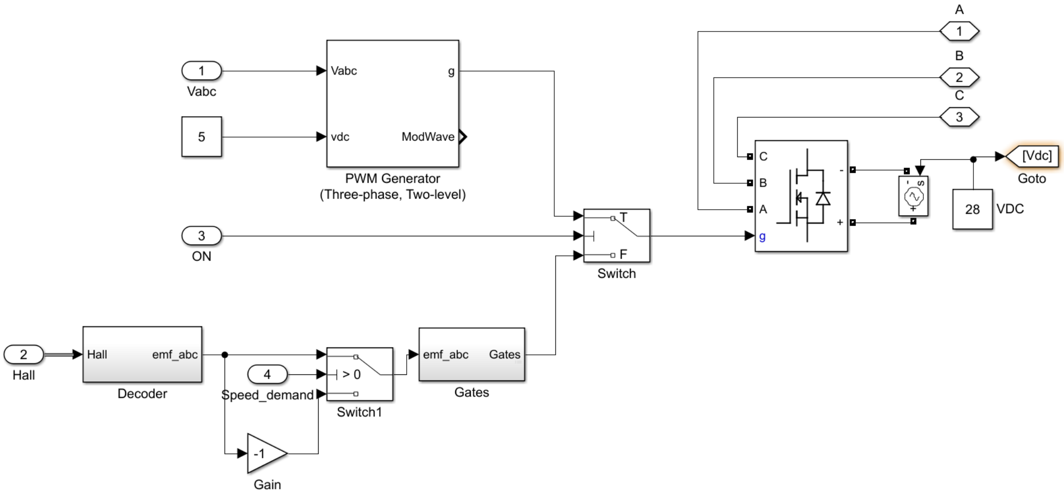


FIGURE 3.10: FOC driver detail.

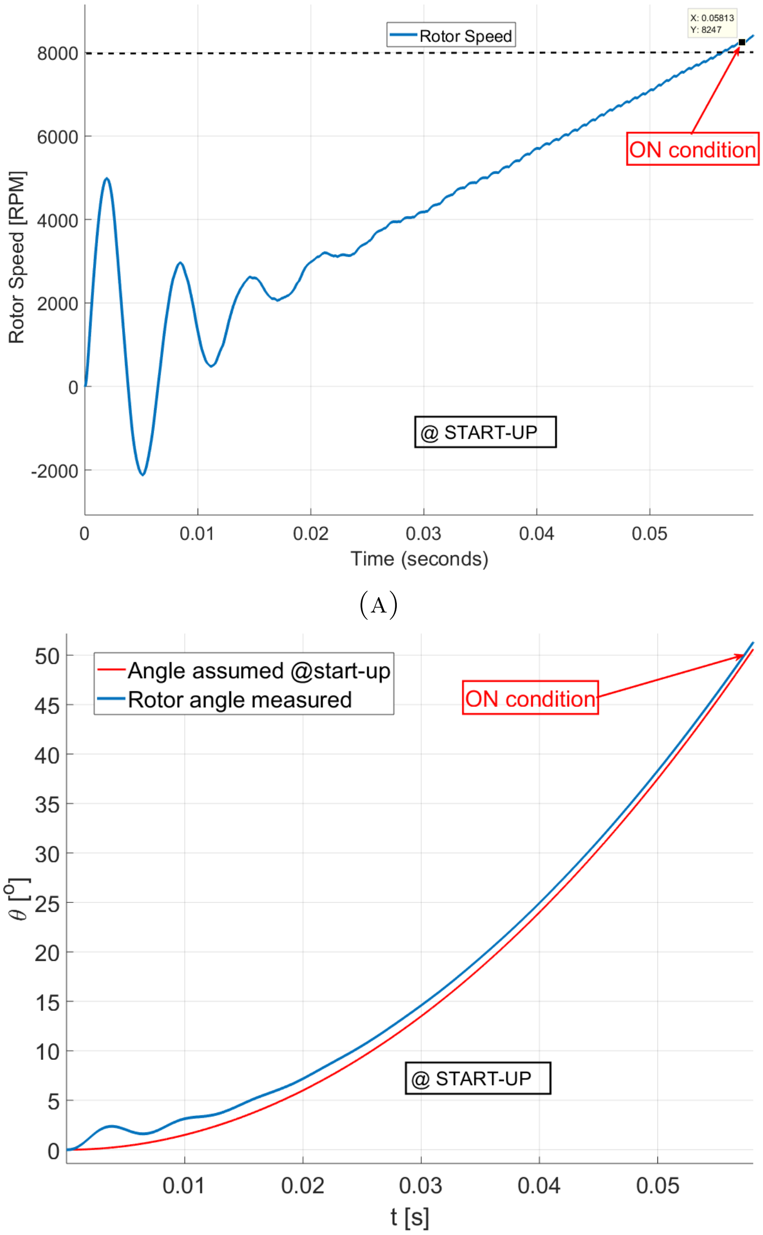
3.4 Simulation Model Sensorless

As already done for the six-step control, a sensorless FOC algorithm has been designed. Determine the virtual hall signals is not enough in this case. In fact the FOC requires the knowledge of rotor position instantaneously. Once again, the sensorless algorithm is based on the BEMF estimation. To do so, a minimum speed is required to sense back the BEMF, demanding an independent start-up subroutine.

The start-up routine is based on the Microchip AN1078 algorithm [28]. Assuming a fixed acceleration rate, a parabolic trend of the rotor angle ϑ is provided to the inverse Clark-Park blocks. Even if the start-up is carried out in an open loop state, the I_d and I_q currents are sensed back to the PI controllers, in order to automatically initialize the integral term for the closed loop configuration. Once the motor speed overcome the defined threshold, the control system switches over to closed-loop configuration, estimating ϑ from the BEMF. The resulting start-up behaviour is represented in the fig. 3.11, where it is possible to evaluate the speed trend and the comparison between the assumed and the measured electrical angle.
For the closed-loop state, the rotor position is calculated employing the BEMF voltage. This latter is estimated in the same way as described in the section *Trapezoidal control sensorless* (sec. 2.2). To do so, a filtering process[6] is carried out, trying to smooth the high frequency harmonics introduced by the PWM signal and the response of the motor circuit. The cut-off frequency is set to be equal to the electrical revolutions per second, because it corresponds to the fundamental frequency of the drive current and the motor voltage. Then, the filter and the *filter delay correction* are optimised for the nominal speed with a corresponding phase delay of -45^o [28]. Obviously, the relation between the phase delay and speed has to be considered for ϑ compensation in all speed ranges. The *filter delay correction* implements a basic linear relation between delay and speed, with saturation at 0 and -90 degrees. This latter block has the speed and the filter parameters as inputs, giving the delay angle as output. The phase delay is then added to the rotor angle calculated with the BEMF.
The magnet axis position, which lets determine the rotor angle, could be obtained by the reverse trigonometric formula of the BEMF expressed in the stator-fixed ($\alpha\beta$) reference frame eq. (3.15) [24]:

$$\begin{aligned} e_{i\alpha} &= -k_e \omega_r \sin(\vartheta_r); \\ e_{i\beta} &= k_e \omega_r \cos(\vartheta_r) \end{aligned} \tag{3.15}$$

[6] A filter of the first order is considered sufficient [28].



(A)

(B)

FIGURE 3.11: The speed trend (A) and the comparison between the assumed and the measured electrical angle (B) during the start-up phase.

where k_e is the back-EMF constant, which depends on the motor parameters, ω_r is the rotor electrical speed, and ϑ_r is the magnet axis position. Then, the rotor

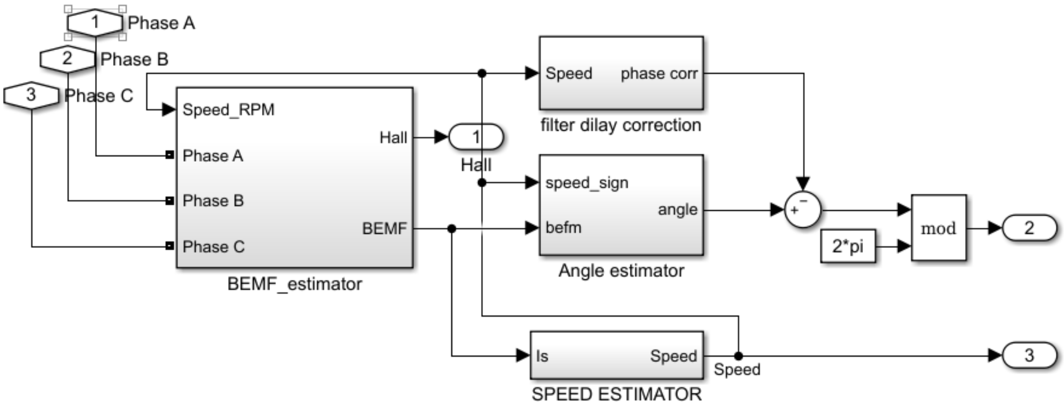


Figure 3.12: *Virtual sensor* block developed for the FOC sensorless.

angle can be calculated as:

$$\vartheta_r = \arctan\left(\frac{e_\alpha}{e_\beta}\right) \tag{3.16}$$

The simulink algorithm which implements the eq. (3.16) is shown in the fig. 3.13.

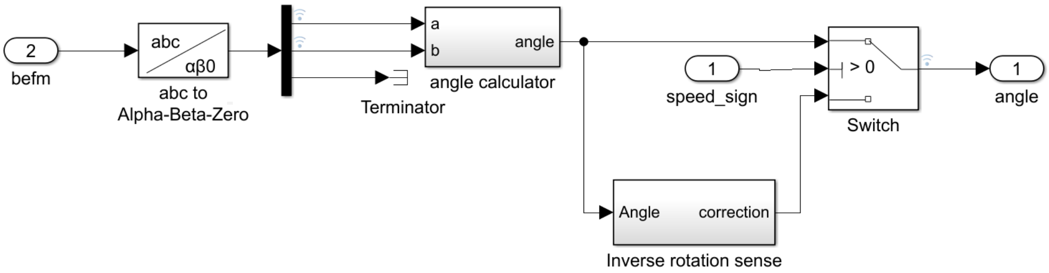


Figure 3.13: *Angle estimator* block developed for the FOC sensorless.

The *angle calculator block*, shown in the fig. 3.14, is dedicated to the detection of the angular quadrant, based on the BEMF sign analysis.

3.5 Final Remarks

Trapezoidal commutation is inadequate to provide smooth and precise motor control of BLDC motors, particularly at low speeds. Sinusoidal commutation solves this problem attempting to drive the three motor windings with three sinusoidal currents. This eliminates the torque ripple and commutation spikes

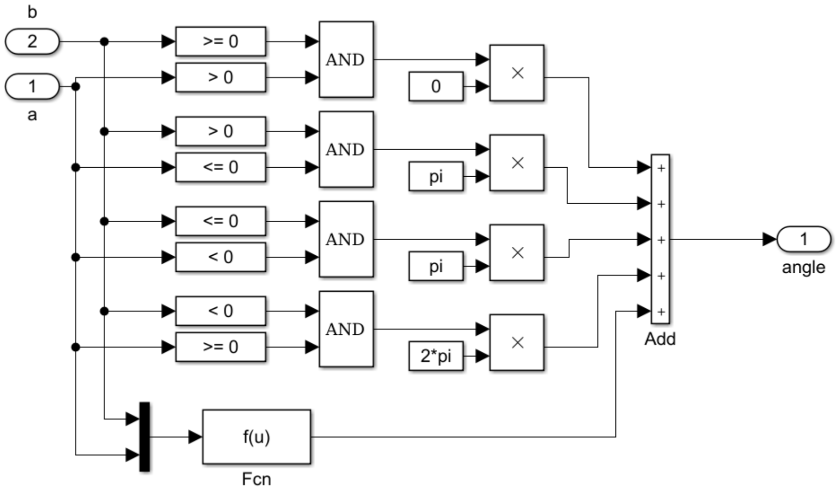


FIGURE 3.14: *Angle calculator* which extends the trigonometric calculation of the angle, to the 360^o domain.

associated with trapezoidal commutation. An accurate measurement of the rotor position is required, in order to generate smooth sinusoidal modulation of the motor currents. The hall devices are inadequate because they provide only a coarse measure of the rotor position, making necessary an encoder which increases the costs. However, the sinusoidal control is very effective at low motor speeds, but it is inadequate at high motor speeds. This is because of the limitation in the tracking of the sinusoidal signal, that has an increasing frequency proportional to the rotor speed.
The FOC overcome the limitation of the sinusoidal algorithm by controlling the current in the d-q reference. that makes the error signal less time-dependent. Hence, it introduces several advantages, such as [40]:

- Wide range of operational speed.
- Simplification of a AC model into a simple linear system.
- Faster dynamic response.
- High efficiency.

Unfortunately it requires an higher computation power and it is more difficult to implement as well.

Chapter 4
PID Controller

The acronym PID stands for Proportional-Integral-Derivative. These controllers are implemented in the vast majority of automatic controls for industrial processes applications[1]. Their spread is due to the effectiveness in a wide range of operation conditions, and the simplicity of their operating principle. Despite their long history and know-how gained from years of experience, the cost/benefit ratio increased only when microprocessors and software tools started being available at reduced cost. The simplicity of these controllers is also their weakness. In fact, it limits the range of plants that they can control satisfactorily with any member of the PID family [39].

4.1 PID Structure

The PID operation is described by a transfer functions which relates the error $E(s) = R(s) - Y(s)$ and controller output $U(s)$. Two possible configurations exist: series and parallel form, respectively described by eq. (4.1) and eq. (4.2) [15].

$$C_{PID}(s) = K_p \left(1 + \frac{1}{T_r s} + \frac{T_d s}{\tau_D s + 1} \right) \tag{4.1}$$

$$C_{PID}(s) = K_p + \frac{I_p}{s} + \frac{D_p s}{\gamma_p D_p s + 1} \tag{4.2}$$

where T_r and T_d are known as the reset time and derivative time; τ_D is the additional time constant; Kp, I_p and D_p are respectively the Proportional, Integrator and Derivative gains, set on with the tuning.
The fig. 4.1 shows that members of PID family have three control actions in different combinations: proportional (P), integral (I) and derivative (D). Although

[1] Around 90% of industrial applications.

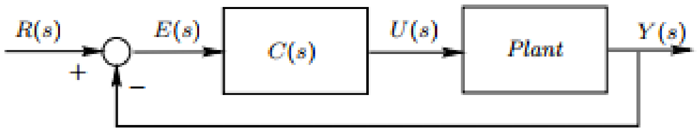


FIGURE 4.1: Block diagram of a parallel PID controller [15].

the effect of each action is not independent of each other, their roles can be described as follows [15]:

- **The Proportional** component affects the instantaneous value of the control error. It has proportional effect that is sufficient to control any stable plant, but provides limited performances and the error does not become zero at steady state. The P-gain is usually represented by the proportional band (PB) defined as

$$PB[\%] = \frac{100[\%]}{K_p} \tag{4.3}$$

  which is the error in percentage required to yield a 100% change in the controller output.

- **The Integral** component takes into account the accumulated error, giving a control mode with slow reaction. It forces the steady state error to zero in the presence of a step reference and constant disturbance. The integral action by itself determines the loop stability and it is responsible of the wind-up effect.
  This latter occurs when the actuator is saturated; the integrator will continue to integrate whilst the input is constrained by the saturation limit. This may lead the state of the integrator to unacceptable high values, causing a very poor transient response. The wind-up effect can be avoided by implementing several methods, available in the literature.

- **The Derivative** component acts on the rate of change of the control error, this is why it is also known as predictive mode. Its contribution disappears when the error is constant. Its main limitation is the instability in response to high frequency control errors, such as in case of setpoint changes or measurements affected by noise. The additional time constant is chosen in the range $0.1T_d \leq \tau_D \leq 0.2T_d$ and is called derivative time constant. It acts like the time constant of a RC filter, then the smaller it is, the larger the pass frequency band.

The derivative block is not used because the feedback error is affected by high noise. This is why, only a PI is implemented for the control loop.

4.2 PID Tuning

The PID tuning has to accomplish three conflictive objectives:

Stability. The closed-loop system output remains bounded for a bounded input.

Rapidity. The closed-loop system tracks reference changes and suppresses disturbances as rapidly as possible. The larger the loop bandwidth, the faster the controller responds to changes in the setpoint or disturbances in the loop.

Robustness. The loop design has enough gain and phase margin to tolerate modeling errors or variations in the system dynamics.

To do so, two tuning methods have been used for the control system design and explained below. These are the empirical method and PID tuner function available in Simulink.

4.2.1 Empirical Tuning Technique

One of the most common ways to design a PID controller was to use empirical tuning rules, based on measurements on the real plant. Nowadays, it is preferable for the PID designer to employ model based techniques. In any case, PID empirical tuning can be adopted as first approximation in the PID design [15]. Among empirical techniques, this research has so far focused on two of those: Ziegler-Nichols (Z-N) Oscillation Method and the Reaction Curve Based Methods. The Z-N procedure is only valid for open loop stable plants and it requires forcing the plant to oscillate. This operation can be dangerous and expensive. In the reference literature it is shown how execute this tuning. To avoid these limitations, Cohen and Coon carried out further studies based on the same model eq. (4.4). Their method, known as Reaction Curve Based Method, offers a more homogeneous response for the same range of x values[2], compared to the Z-N technique. This fact encourages to adopt such technique in the research in matter, for the preliminary design of the PI. Many plants, particularly those used in the process industries, can be satisfactorily described by the model in the eq. (4.4), which lets easily obtain the equations for the tuning, resumed in the tab. 4.1.

$$G_0(s) = \frac{K_0 e^{(-s\tau_0)}}{\nu_0 s + 1} \tag{4.4}$$

[2] x is the ratio between the delay τ_0 and ν_0 (eq. 4.5).

Hence, the first step is modeling the real plants. The linearized parameters of the model can be obtained with an open loop experiment, using a specific procedure which consists in the following steps [15]:

1. With the plant in open loop, lead the plant manually to a normal operating point. It is equivalent to set the plant output, that can be the speed or the phase current in the case in matter, at $y(t) = y_0$ for a constant plant input $u(t) = u_0$.
2. At an initial time t_0, apply a step change to the plant input, from u_0 to u_f, in the range of 10% to 20% of the full scale.
3. Record the plant output until it settles to the new operating point. This curve is known as the process reaction curve, it has to have the same shape in fig.4.2, in order to apply the Cohen's method.

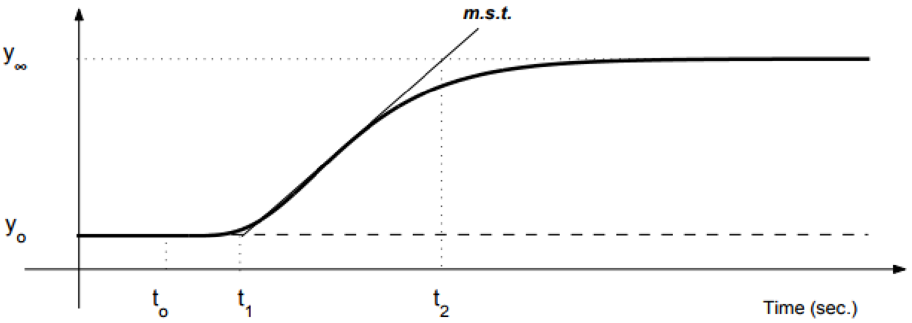


FIGURE 4.2: Example of a process reaction curve [15].

4. Compute the parameters of the model as follows:

$$K_0 = \frac{y_f - y_0}{u_f - u_0}; \quad \tau_0 = t_1 - t_0; \quad \nu_0 = t_2 - t_1 \tag{4.5}$$

The PID coefficients parameter are calculated applying the setting rules proposed in tab.4.1, which depends on the linearized model parameter.

Results

The results in tab. 4.2 are obtained by adopting the empirical method described before. The output value has to remain in the range of 10% to 20% of the full scale. For the trapezoidal control, the only output magnitude is the control signal voltage, whose full scale is $5V$. The open loop system is assumed

Cohen-Coon Tuning Method			
	K_p	T_r	T_d
P	$\frac{\nu_0}{K_0\tau_0}[1+\frac{\tau_0}{3\nu_0}]$		
PI	$\frac{\nu_0}{K_0\tau_0}[0.9+\frac{\tau_0}{12\nu_0}]$	$\frac{\tau_0[30\nu_0+3\tau_0]}{9\nu_0+20\tau_0}$	
PID	$\frac{\nu_0}{K_0\tau_0}[4/3+\frac{\tau_0}{4\nu_0}]$	$\frac{\tau_0[32\nu_0+6\tau_0]}{13\nu_0+8\tau_0}$	$\frac{4\tau_0\nu_0}{11\nu_0+2\tau_0}$

Table 4.1: Cohen-Coon tuning using the reaction curve.

to have a linear response. It has to be remarked that t_1 and t_2 are determined by the intersection between the tangent with maximum slope and the two horizontal lines $y = y_0$ and $y = y_\infty$. Such tests are carried out using the sensors provided with the motor model block.

	Test 1	**Test 2**
y_f	12431 RPM	14356 RPM
y_0	10545 RPM	10545 RPM
u_f	3.5V	4V
u_0	3V	3V
t_1	0.2001s	0.2003s
t_2	0.2050s	0.2050s

Table 4.2: Cohen-Coon tuning measurement on simulating model

Using the average values of the measured parameters and refining manually the K_p and T_r calculated for the trapezoidal control, the Cohen's method gives as result:

$$K_p = 5.8414e - 04$$

$$T_r = 0.0175$$

The refinement of the gains is unavoidable, due to the high uncertainty in the determination of the tangent with maximum slope.

4.2.2 Model-Based PID Tuning in Simulink

It is common to use a model based techniques for the refinement of PI in industrial processes. The PID Tuner function in simulink is a support for this complex operation. This tool allows a good balance between performance and robustness for PID with either one- or two-degree-of-freedom. It automatically computes a linear model of the plant. The linear plant includes all the blocks which are between the PID output and input. The linearizing algorithm is based on the open-loop response of the modeled plant. Sometimes this operation is not automatic, thus the alternative consists in estimating the plant model from measured or simulated response data, with a method similar to the Cohen's one. The models implemented for FOC and Trapezoidal control, are too complex to be automatically linearized, this is why for the tuning the plant is modeled using this two tools:

1. The System Identification Toolbox ^{TM} estimates the parameters of the linear plant model analysing the response in time-domain;
2. PID Tuner is dedicated to the tuning, considering the plant model generated by the first tool.

The response data required by the first tool can be either measured from the real plant, or obtained by simulating its simulink model.

Plant Transfer Function

The procedure is described in detail in the Mathworks help [46], while only the useful information to achieve the tuning are summarized below. As already said, PID Tuner interface gives two alternative input sources to determine the parameters of the transfer function:

- ▷ **Simulate Data**. The system response is obtained by simulating the Simulink model. This option is used in the preliminary phases of the design, when the real motor cannot be tested. In this case, the software temporarily removes the PID Controller block from the model, and injects a signal where at the PID output. The response, instead, is measured at the input of the PID block. The disturbance signal can be either a step or a pulse, and both cases have to have such properties:
  - **Sample Time** ΔT . Controller sample rate. It has the same value used as time step for the discrete solver.
  - **Offset** u_0 . Controller output offset value.
  - **Onset Time** T_Δ . It is the delay necessary for the plant to reach the required steady state, before applying the step change.
  - **Step Amplitude A** . Step size of the controller output (plant input) that has to be applied to the model. This value is added to the offset value.
- ▷ **Import measured system response data**. Used to adjust the PID parameters to the real dynamic of the plant.

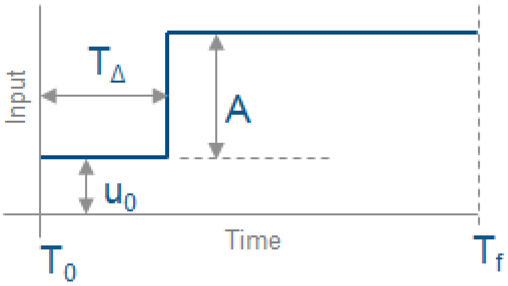


Figure 4.3: Input signal properties for the open loop model simulation.

In both cases, the input signal has to be chosen carefully to carry out a useful plant model. Once the simulation is completed, the system plots three curves. One represent the offset response, i.e. the plant response to a constant input of u_0. The second shows the complete plant response. The last curve is the difference between the first (input response) and second (offset response) curves taking into account the negative feedback sign. This latter will be used for the identification of plant parameters.
The plant identification follows the response simulation. The PID Tuner finally identifies a plant model using the response data with a graphical matching of the linearized modeled plant response with the response curve previously obtained. The overlapping can be achieved manually, clicking on the plant curve and changing the zero or pole location, or using the Auto Estimate option. The plant structure available in the tuner tool are: one pole, two real poles and underdamped pair.

PID Tuner Algorithm

The PID tuner algorithm chooses a crossover frequency based on the plant dynamics and designed for a target phase margin of 60^o [3]. From the PID Tuner interface the algorithm computes new PID gains, by acting on the knobs of the response time bandwidth and transient response (or phase margin). For a given robustness, the tuning algorithm chooses P and I gains as compromise between the two performances: reference tracking and disturbance rejection. The design focus can be changed to favor one of those two. Using a controller with more tunable parameters (PID, PI or P), it is more likely that the PID algorithm can achieve the desired focus without sacrificing robustness [46].

[3] It represents the robustness of the system

Chapter 5
Fault Tolerant System

Since the 1960s the automation of the industrial processes has increased progressively to comply the demand of higher performances and quality. The automation requires a constant supervision of the technical processes, in order to avoid abnormal events which reduce performances, or even worse, creates unsafe situations. Then, the research has carried out plenty of fault detection techniques, which are daily applied in the industrial technology.
This topic is especially important for the aerospace field. In fact, the on-board equipment has to be light, stiff, easily maintainable and above all reliable. These latter two features have to be considered since the beginning of the design. We must also consider that forcing an equipment to its limit capabilities reduces its lifetime. Hence, the best compromise among the optimization of performances, robustness and reliability has to be achieved. For a future implementation of FOC algorithm to an electrical motor, intended for aeronautic use, it becomes essential to provide the system of fault tolerant solutions, which ensure that the motor continues operating until the aircraft can land safely. This is why the thesis is supplemented by an entire chapter dedicated to the fault detection and diagnosis problems.

5.1 Useful definitions

Before starting with the description of the fault detection techniques, it is essential to clarify the meanings of basics terminology, in order to avoid misunderstanding while reading the document. All the definitions are reported in the literature [25], and accepted as well by the scientific community.

▷ **Fault**
"A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual standard condition."

▷ **Failure**
"A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions."

▷ **Reliability**
"Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time."

▷ **Safety**
"Ability of a system not to cause danger to person, equipment or environment."

5.2 Background review

An accurate and efficient monitoring together with a good fault diagnosis can drastically improve reliability and stability of the plant, besides reducing the overall costs. Several studies has been carried out to statistically determines the reliability, giving as results an expected improvement up to 5-6 percentage points [23]. The only detection of the fault or failure is not enough to increase the reliability of the system. All the following tasks have to be considered:

1. **Supervision**. The supervision task essentially consist in detect an unacceptable deviation of the system behavior from the standard conditions. This is carried out by monitoring certain parameters with dedicated sensors. In the case of a dangerous process state, the supervision has to automatically initiates an appropriate counteraction.

2. **Diagnosis**. The diagnosis is complementary to the supervision, and it is the process of determining the nature and circumstances of a diseased condition, by examining the informations provided by on-line sensors [36]. Most of the time, the diagnosis of faults in electrical machines is based on the current sensors. The time and frequency analysis of the current signal gives useful information about the health condition of the equipment. An ideal diagnostic procedure should use a minimum number of measurements, necessary to clearly identify the incipient failure modes in a minimum time.

3. **Prognosis**. The prognosis is a forecasting of the outcome of a diseased condition. It is usually a difficult task, because it requires a detailed knowledge of the physic process in analysis. Thus, past experiences on similar devices play an important role for the correct prognosis. It becomes essential to carry out an intense laboratory experimentation campaign [25].

4. **Fault management**. The fault management task is the combination of all the operations that, once detected the fault and determined its features, allow to restore the nominal operating of the system, as economical as possible and in the safest way.

TASK 1: Supervision

The supervision task is aimed at showing the present state. It also takes appropriate actions to avoid damage or accidents, when undesired or unpermitted states are detected. It is essential in order to avoid a malfunctions or failures that will occur, sooner or later, after a fault [35]. It therefore consists in two fundamental tasks:

1. **Monitoring:** the measured variables, which are remarkable for the fault detection purpose, are checked with regard to tolerances. Pressures, forces, liquid levels, temperatures, speeds, and oscillations are just some examples of the checked variables. Normally the supervision can produce two severity levels of alarm. The first one is triggered when the quantities are within the tolerance zone. It is useful for the operator to take appropriate counteractions before the fault severity rises. If also the tolerance zone is exceeded, the alarm is raised to spur an immediate solution. Another drawback arises when a severe fault or failure occurs. It is known as alarm-shower and it consists in several alarms that are triggered in a short time.

2. **Automatic protection:** the counteractions of the system is automatized when the exceeding of a threshold implies a dangerous process state. For this reason, the reliable systems are usually designed in a fail-safe configuration, providing them, for example, an emergency shutdown [25].

The greatest challenge of the supervision task is to find a compromise between the detection of abnormal deviations and unmeasurable or wrong alarms. In fact, the normal fluctuations of the variables may cause temporary states in which the values are within or even exceed the tolerance zone. If not properly designed, this could cause several false alarms which overload the work of the operators and/or useless maintenance procedures which rises costs. The classical limit-value-based supervision method is suitable for the overall supervision of the processes, when the process stays in a steady state or if the monitored variable does not depend on the operating point. The method is very simple and reliable, but it cannot be implemented in systems with variables changing dynamically, like forces in rotating machines, or pressures and temperatures in chemical reactions, besides deep limitations in the fault diagnosis [25].

TASK 2 and 3: Diagnosis and Prognosis

Diagnosis and Prognosis usually accompany the basic supervision task. They allow:

- ▷ To satisfy more strict requirements for the fault-tolerant system.
- ▷ To early detect small faults with abrupt or incipient time behavior.

- ▷ To characterize the nature of the faults in the processes and their actuators and sensors, useful for the future maintenance and improvement in the design.
- ▷ To detect faults in closed loops.
- ▷ To supervise the process in transient states.

Depending on the performances, applications and nature of the system, fault diagnosis and prognosis tasks can be designed in several ways. The information processed during these tasks can be estimated or measured. The goal is to determine the kind, size and location of the fault by relating the heuristic symptoms to the faults, using classification or reasoning methods via fault symptom trees. The diagnosis can be carried out either automatically or by the operator [25].

TASK 4: Fault management

The actions provided for the fault management deeply depend on the severity of the fault occurred. This is why it is proposed a definition of the "hazard classes", defined according to the counteraction that has to be taken. The classification is reported below [25]:

- ▷ **Safe operation**. In case of an imminent danger for the process or the environment, the system is shut down.
- ▷ **Reliable operation**. It consists in hinder a further fault expansion, through changes in the operational state. For example, the solution could be to reduce the load, speed, pressure, temperature and so on.
- ▷ **Reconfiguration**. The process is reconfigured to keep the operation under control. The reconfiguration can consists in using other sensors, actuators or redundant components.
- ▷ **Inspection**. The inspection is carried out when the fault requires a detailed diagnosis.
- ▷ **Extraordinary maintenance**. It can be execute at the moment or in the next opportunity depending on the type and severity of the fault. It usually consists in a tuning of the process parameters, or exchanging the worn parts.
- ▷ **Repair**. It is the procedure which removes a fault or failure. Two possible levels of repair can be carried out: overhaul or revision.

The maintenance has remarkable costs, which resemble the 20% of the total. This is why, nowadays, advanced supervision and diagnosis techniques are implemented more and more in the technology. The maintenance dependent on process condition, known as maintenance on demand, and the tele-diagnosis are just few example of modern solutions, which may help to reduce the costs and improve the life time of the processes.
The appearance of certain faults and failures cannot be avoided totally, even if reliability and safety analysis has been considered in the design, as well as quality control methods have been applied during manufacturing. Therefore, high-integrity systems must tolerate these unavoidable faults by additional efforts, trying to achieve a fault-tolerant design. This means that faults are compensated in order to do not lead to system failures. The fault tolerant system decreases the performances proportionally to the severity of the failure, while for a naively designed system even a small failure can cause a total breakdown [13].

All sort of fault tolerant techniques exist, but besides the obvious improvement of the components quality, they all can be resumed as redundant systems. The redundant modules are usually in a parallel configuration and can be hardware or software, either diverse or identical to the nominal-operating module. In general, these modules are supervised with fault-detection capability and, with a reconfiguration mechanism, the failed modules are switched off, substituted by on spare modules [35]. This latter technique is known as dynamic redundancy, and it is not the only possible. Also the static redundancy can be implemented, with redundant modules which are continuously used, regardless whether faults are present or not. This latter redundancy does not require a fail-over time and provides higher short term reliability [38].

For the thesis in matter, the goal is to develop a BLDC motor which tolerates a phase disconnection. Therefore, the rotor has to continue rotating properly in the event of a single phase failure. At the same time, the system has to continue assuring the speed control, even if the degradations in the efficiency, torque and speed ripple have to be unavoidably assumed.

5.2.1 Fault detection and diagnosis techniques

The fault detection and diagnosis methods are numerous and deeply varied. They are typically classified in three categories:

- ▷ **Quantitative model-based methods**. This category ranges from analytical methods to artificial intelligence and statistical approaches.

- ▷ **Qualitative model-based methods**. These techniques use less complex mathematical models.

- ▷ **Process history based methods**. These methods are based only on the process history data.

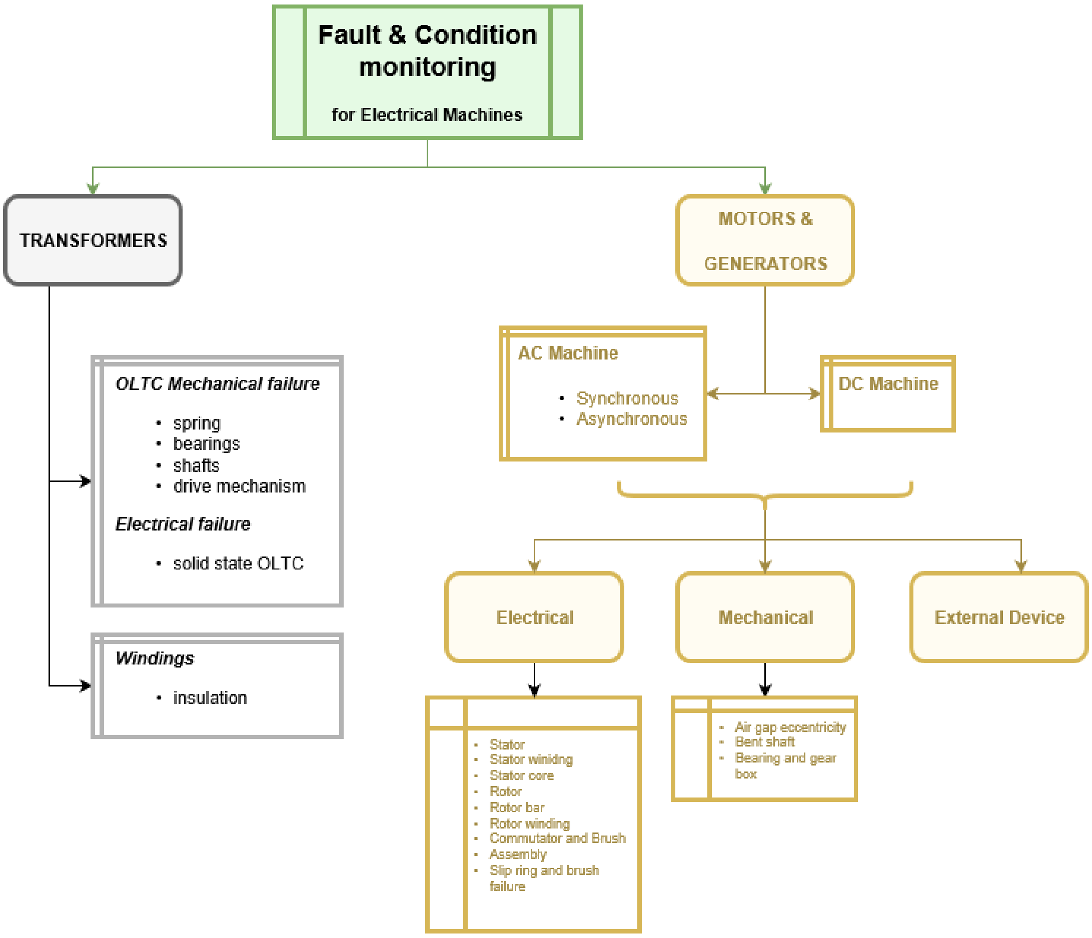


Figure 5.1: Fault Tree of a generic electric machine.

Such a wide selection always allows the designer to find one feasible solution for his or her own case. Unfortunately, the same variety represent a big challenge for the design, to someone that is not a specialist in these techniques. The experience about fault detection and diagnosis problems, besides the deep knowledge of the system operating principle, are essential to find the most suitable technique.
As first step for a proper fault detection and diagnosis, it is necessary to understand how the mechanism can fail. Several tests and studies have been carried out to cover all the possibilities. Then, it is essential to classify the kind of failures that, for the case in matter, may affect the electrical machines. The failure in these equipments can be distinguished into electrical or mechanical, depending upon the root cause of failures (fig. 5.1 [36]). The fault diagnosis of electrical machines requires strong knowledge of their operating principle and production

process, involving several technical fields. Then, it is not easy either to detect and diagnose a fault in a BLDC motor, and to evaluate the safety of the whole system due to crossed interactions among all the components [23].
When a fault occurs, the internal process changes from nominal to a diseased operating state. Such changes appears as a deviation of the value of internal parameters such as: capacitance, stiffness, electrical and thermal resistance, temperatures, currents and so on. Most of the time, this parameters cannot be measured due to the lack of dedicated sensors, or, more likely, sensors are not sufficiently sensitive to detect such changes with a proper signal to noise ratio [25]. The use of additional sensors, over certain limit, is not desirable to increase the detection capability of the system. This is because more sensors require further components, like cables, transmitters, plugs and so on, increasing the costs and deteriorating the overall reliability. In fact, the bigger is the number of components used, the higher is the probability of faults.
Figure 5.2 gives a survey of analytical fault-detection methods, according to the classification proposed in the R. Isermann's book [25]. The same reference reports a thorough description of those methods. The "Limit checking of absolute value" is the one implemented for the goal of the thesis; that is why we propose a brief review of this method in the next section[1].

Limit checking of absolute value

It is one of the fault detection methods based on single signal measurements. Two thresholds represents the admitted maximum value (Y_{max}) and minimum value (Y_{min}). No fault has occurred if the monitored variable $Y(t)$ stays within the defined tolerance zone.

$$Y_{min} < Y(t) < Y_{max} \tag{5.1}$$

This is an undoubtedly simple method; nevertheless it is applied in almost all automatic systems [35]. The thresholds are mostly selected based on experience, trying to avoid false alarms. The category of single-signal fault detection method includes other techniques; an example is the "trend checking" which uses the first derivative of the monitored parameter. But, because of phase-failure is an unpredictable event, this method cannot be implemented.
A possible refinement of the basic limit checking is the *"adaptive thresholds"* method. It is suitable in the cases where residuals may contain a static part proportional to the input, while the dynamic part depends on the amplitude and frequencies of the input excitation. To cope with this problem, first-order

[1] The detection method implemented for the phase failure is the *"Limit checking of absolute value"*. The justifying reasons are explained later in the chapter.

high-pass and low-pass filters are applied to the residuals. Their time constants are selected according to the dominating time constant of the process, and the input excitation harmonics [35].
The limit checking can be further improved with a primitive model-based fault-detection method. It is known as "plausibility checks" technique and consists in evaluate the values of the monitored parameters with regard to credible values. Each single measurement is examined with a basic limit check with usually wide tolerances. Then, the measurements can be related to each other by using logic rules like AND or OR, and if they are noy satisfied either the process or the measurements are faulty. This lets relate the tolerance zone to the operating conditions, avoiding false alarms in transient phases like the start-up.
Once detected, the fault has to be analyzed in order to determine its size, location, time of detection and as much details as possible. In the fig. 5.3 is reported the classification proposed by R. Isermann in his book [35]. Two main groups of fault diagnosis can be distinguished:

- ▷ **Classification methods**. One of these has to be implemented if the relation between the faults and symptoms is learned experimentally, and stored as knowledge base [35]. Then, once the relation has been stored, the faults can be concluded by comparison of the observed symptoms and the normal reference.

- ▷ **Inference methods**. Sometimes the basic relation between faults and symptoms are at least partially known. Inference methods can be implemented when this a-priori knowledge is represented in causal relations: fault⇌events⇌ symptoms [35]. The establishment of these causalities can follow two opposite paths: the fault-tree analysis or event-tree analysis. The first one, known also as physical causalities method, determines the relation starting from faults and going back to symptoms, through intermediate events. While the second proceeds from the symptoms, to get to the faults, and it is also known as the diagnostic forward-chaining causalities.

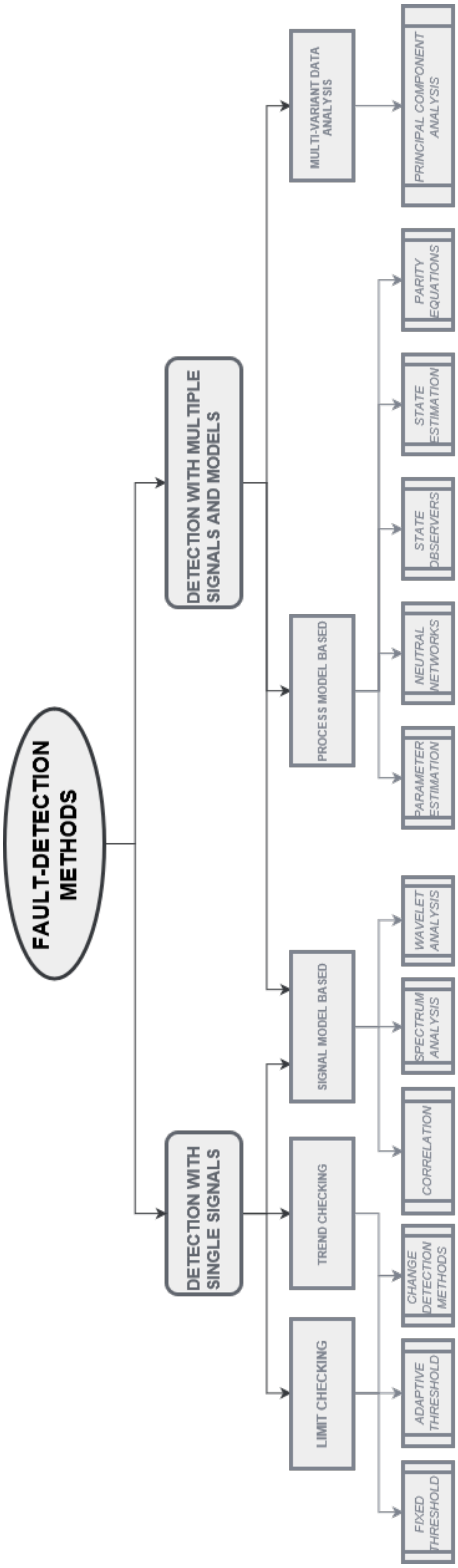


Figure 5.2: Fault detection methods tree .

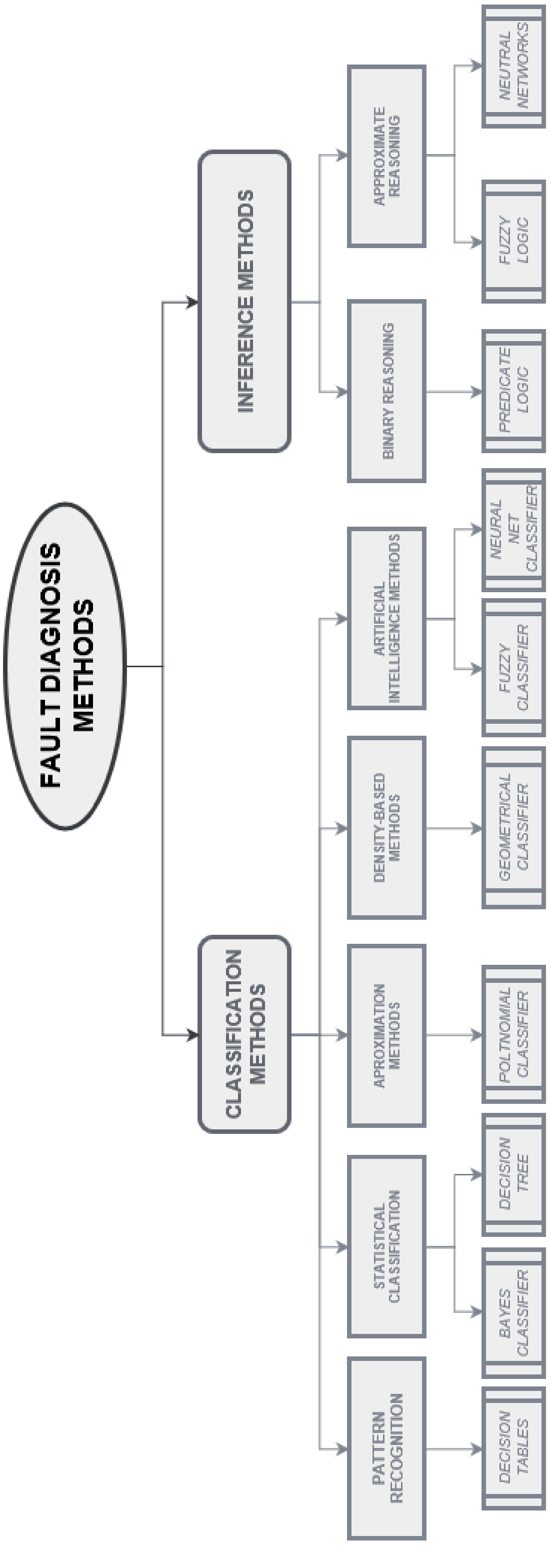


Figure 5.3: Fault diagnosis methods tree .

The operating process, and how faults affect it, is deeply different comparing the closed loop with the open loop configuration. That is why an effective monitoring strategy has also to consider the architecture of the control.

- ▷ **Open loop**. In the case of a remaining fault, it generally provokes a permanent offset of the output of the system. Then, a direct or indirect measure of the output magnitude may be enough to detect the fault. Obviously, further information are probably necessaries for the diagnosis and prognosis.

- ▷ **Closed loop**. In this case the behavior of the system is more complicated. It naturally tends to react to the fault due to the feedback. Depending on the time history of the parameters changes, the output may shows only a somewhat shorter and vanishing small deviation. It occurs if the control has an integral behavior [25]. Therefore, not only the output has to be supervised. In fact, the internal variables shows a permanent offset, proportional to the acting process of the feedback. If other variables were not checked, only if the fault grows in size and causes the variable to reach the saturation, the output would finally show a permanent deviation.

5.3 Fault Diagnosis in BLDC motors

An overview of typical detection and diagnosis methods for industrial technology has been reported in the previous section (sec. 5.2). To implement the fault tolerant system for the failure in matter, we have to analyze the specific case of a BLDC motor. The results of the experimental investigations summarized in the R. Isermann's book [35] help focusing the research on the most interesting and suitable techniques for the phase failure. It has been demonstrated that in a DC motor, 14 different faults can be detected by measuring only three signals, together with the combination of parity equation and parameter estimation. The faults can be classified in additive and multiplicative. The additive faults, which include sensor-offset, can be easily detected by parity equations in normal operation. In this case the diagnosis does not require additional input excitations, which are instead indispensable for multiplicative faults, like deviation of motor parameters. In this latter case, the fault detection is better achieved by parameter estimation. The fig. 5.4 shows the diagnosis algorithm implemented in the above mentioned experiments [35]. Each block comprises a meta-class of faults, and the individual fault is recognized with a classification tree. After performing 140 experiments with different faults, the hierarchical decision tree proved to be highly suitable for the diagnosis, achieving a 98% classification rate in a cross-validation scheme.

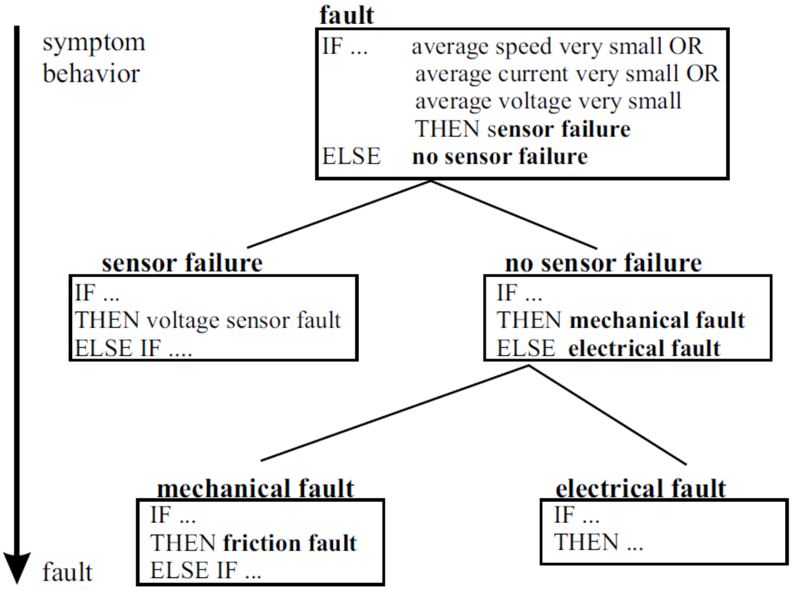


FIGURE 5.4: Hierarchical fault-diagnosis system. Each block comprises a fuzzy classification tree.

5.3.1 Phase disconnection failure

The disconnection of one stator winding of a three-phases PM BLDC motor is investigated. The challenge is to keep the rotor running, even if one stator phase is not connected to the generator. The phase disconnection may occur both during the nominal operating of the motor or at the start-up.
A similar study has been carried out by Bianchi, Bolognani and Dai Pré [29] considering a five-phase PM Motor. This configuration lets design a controlled BLDC motor ables to tolerate until two phase disconnections. The system positively reacts to three fault types: the open circuit condition of a single phase, the open circuit condition of two nonadjacent phases, and the open circuit condition of two adjacent phases. In normal operating conditions, the electric power supply of a five-phase motor is split into five inverter legs. In the event of failure of one or more phases, the remaining healthy phases allow to operate maintaining a fairly smooth torque profile. Anyway, the average torque amplitude decreases due to the reduced number of active phases. Of course, this reduction is not always acceptable; for more electric aircrafts, compressors and pumps are two examples where the torque and speed have to be maintained at nominal value, even after the phase failure [14] [18] [26]. The key of this solution is to consider the significant higher harmonics besides the fundamental. It is demonstrated that these higher harmonics cause high torque oscillations. The post-fault current waveform remain sinusoidal, simplifying the current control. This strategy takes into account that:

- ▷ The second order torque harmonic can be canceled if the current waveforms are sinusoidal. This is possible with a proper choice of the current phase angles of the healthy phases.
- ▷ Once the fault occurs, the remaining healthy phases are selected in order to achieve symmetry with respect to the fault.

After the fault event, besides smooth and suitably high torque, this strategy produces a current amplitudes with 30% of maximum overrating. The adaptation of the control system to the five-phase motor requires a minimum additional cost, with respect to the conventional three-phase configuration. Despite algorithm and structure are more complex and expensive, this solution may be attractive for applications where a smooth torque ripple is a requirement for the fault-tolerant system [19] [21].

5.4 Proposed Fault Tolerant System

The novelty of this thesis is the design of a three-phase BLDC motor able to tolerate the disconnection of a single stator phase. Compared to the five-phases motor, the design of a control system with these features has to overcome some backwards, that cannot be avoided in the post-fault configuration, unless changing the motor structure [29].

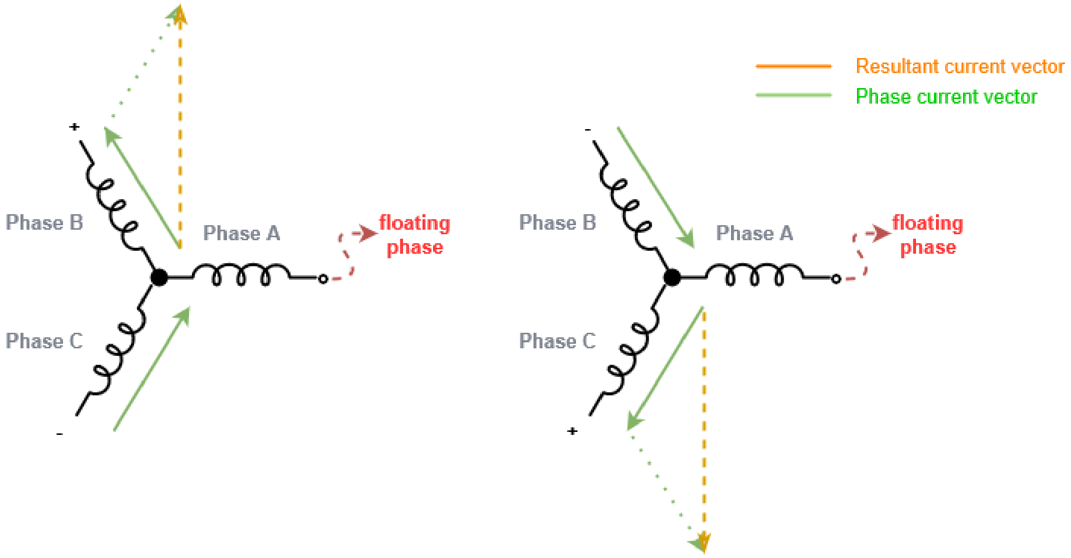


Figure 5.5: Representation of the two resultant stator current vectors in the case of the disconnection of the phase A.

First of all, the asymmetric configuration of the remaining healthy phases, automatically provokes an higher torque ripple. By supplying two stator windings, only two of the six active states can be generated. This is due to no current can flow through the floating phase, then the remaining four states cannot be physically reproduced. Imagining the case in which the failure affects the phase A (fig. 5.5), the remaining active states are represented in the fig. 5.6. Then, the state has to commute each 180^o, provoking an unavoidable high torque and speed ripple.

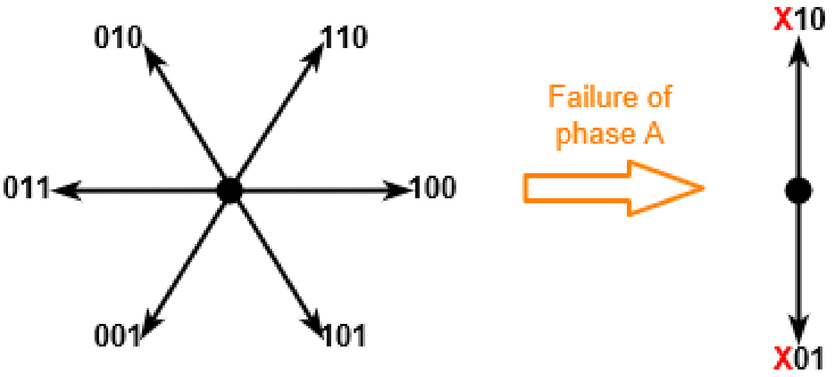


FIGURE 5.6: Changes in the available active states brought by the failure of the phase A.

5.4.1 Phase disconnection symptoms

The phase-failure problem, as presented in this research, actually corresponds to two damages:

- ▷ Physical separation of the cable which join a stator winding to the transistors bridge.
- ▷ Failure of both transistors which form a leg of the transistors bridge.
- ▷ Failure of one stator winding.

All these cases are equivalent in terms of fault diagnosis, and a unique fault tolerant algorithm can be effective for all of those. They correspond to the most sever fault that the system can physically tolerate. We have to remember that it is impossible to avoid higher values of torque and speed ripples, besides higher current peaks[2]. The failure of just one transistor is not investigated because it is a less severe case than the above mentioned failures. This subcategory can be solved with the same algorithm, eventually optimized[3].

[2] The tolerant solution for the failure of two phases cannot be realized because it would be impossible to close the circuit, since the access to the neutral point is not usually available.

[3] Another transistor available allows to generate other two active states.

It is necessary to analyze how system behaves once the phase failure occurs. The algorithm developed will be much more clear if the consequences of this event are known. The failure may occur before, during or after the start-up. The torque and speed trends are characteristic of this failure and will help the diagnosis. That is why the FOC model, explained in the previous chapters (sec. 2.1.1), is exploited to simulate the motor behavior, after and induced phase disconnection. The failure is reproduced through the *Breaker* block. The same investigation has been done with the trapezoidal model too, obtaining similar results to those of the FOC controller. From now on, all the considerations are referred to the FOC case.

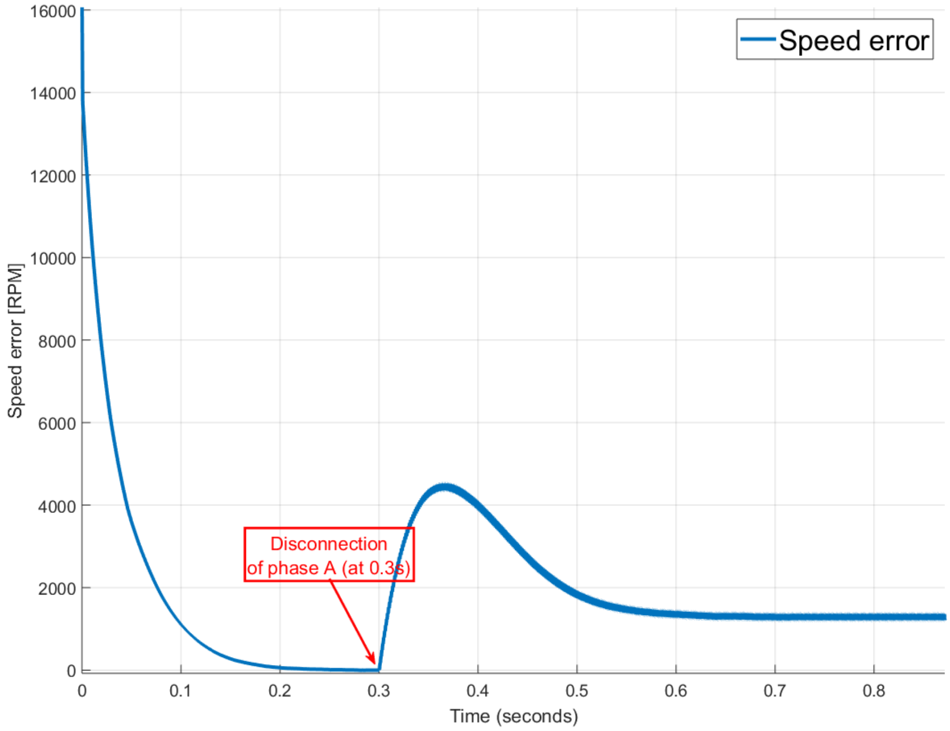


Figure 5.7: Speed error trend with a single-phase failure after the start-up.

Considering the motor running at nominal speed[4], the failure of phase A is induced at the instant $0.3s$. The figure 5.7 shows the trend of the speed error calculated as the difference between the demanded speed, which constantly remain at the nominal value, and the measured rotor speed. After a transition period of almost $0.3s$, the speed error is stable at $1285RPM$, with a ripple which

[4] 16100 RPM

ranges between $1200 \div 1370 RPM$. It is remarkable that the failure does not provoke the motor stalled. This allows to avoid strict restriction to the minimum period for the fault detection. After the failure, the steady state of the speed error is caused by the saturation of the electromagnetic torque. The figure 5.8 shows the trends of the electromagnetic torque and how much it exceeds the load demanded. The surplus of the torque[5] after the failure, expressed in percentage over the load demanded, increases of 22%. This growth is unsustainable; in fact as consequence the motor saturated cannot reach the nominal rotor speed.

Torque values	
Load Demanded	$0.09 Nm$
Before failure	
Torque produced	$0.108 Nm_{rms}$
Surplus of Torque	$0.018 Nm_{rms}$
Max Torque Ripple	5.5%
After failure	
Torque produced	$0.158 Nm_{rms}$
Surplus Torque	$0.068 Nm_{rms}$
Max Torque Ripple	120%
Increasing Losses	+22%

TABLE 5.1: Comparison of torque levels before and after phase A disconnection.

It is dutiful a final analysis of the current waveform and the rms value. As already said, with the FOC algorithm the phase current is sinusoidal[6]. The sinusoid has the electrical frequency and, in nominal conditions, an rms value[7] of almost $7A$. When the failure occurs, the rms and current peaks per each phase rises. This growth is shown in fig. 5.9 and fig. 5.10. The current measured in the disconnected phase, downstream the break point, is not exactly zero due to the RC filter linked between the stator winding and ground. This filter is introduced

[5] rms value

[6] except for higher harmonics induced by the pwm

[7] All the root mean square values of this chapter has been calculated with the *RMS* available in the library of Simulink, with a fundamental frequency of $10Hz$ and a sample time a hundred times bigger than the sample time used by the solver ($10^{-4}s$).

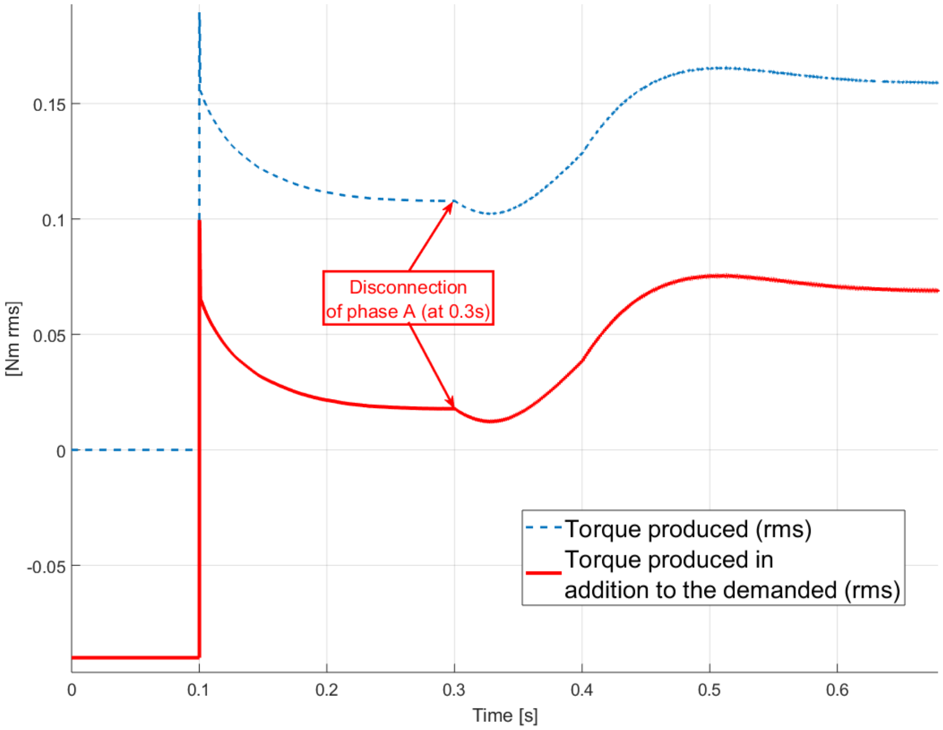


Figure 5.8: RMS value of torque before and after the phase failure at nominal speed. The figure shows both the surplus of torque and the total torque.

to balance the inductive effects when the phase is instantaneously disconnected. However, it has a maximum value of $0.7A_{rms}$ in the entire speed operational range[8].

5.4.2 Phase failure diagnosis

The symptoms analysis, resumed in the previous section, leads to an obvious easy-implementing techniques based on the *"limit checking of absolute values"*, for the detection of the phase failure. The limit checking may be applied on a current and/or voltage measurements. Several advantages and backwards are introduced by each solution.

- ▷ **Voltage sensors**. The voltage sensors can be placed across the output of the transistors bridge and the input of the stator phases. Usually, in the circuit no others useful points are accessible [9]. This solution cannot detect all the fault variants explained before (sec. 5.4.1). In fact, the sensors can detect a variation in voltage only if the open-break circuit occurs between its two terminals. Then, no transistor or stator winding failures can be

[8] $5000RPM$ to $16800RPM$

[9] for example it would be very useful have access to the neutral point.

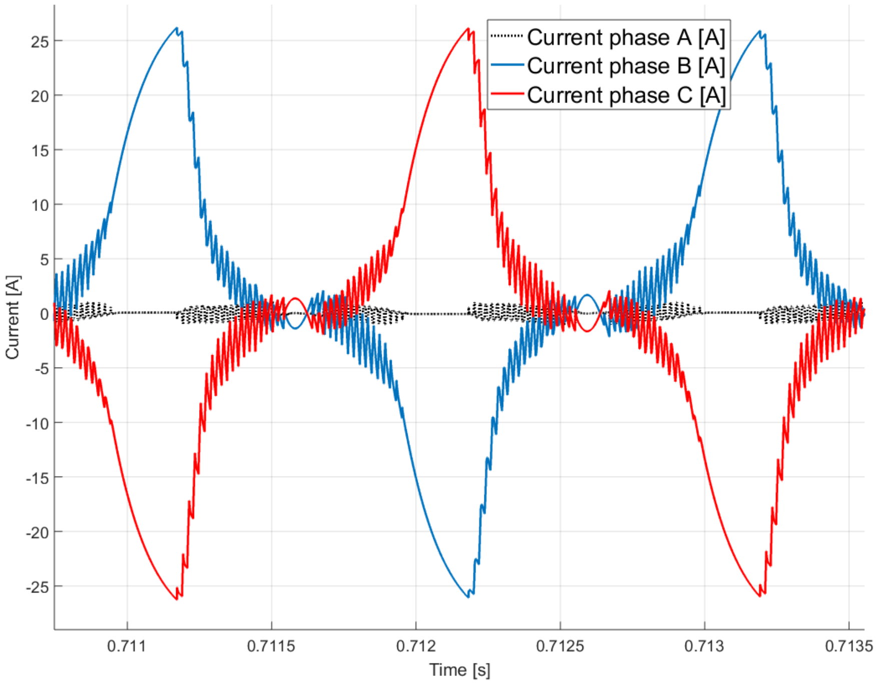


FIGURE 5.9: Current waveform in the stator windings after the failure, at nominal speed.

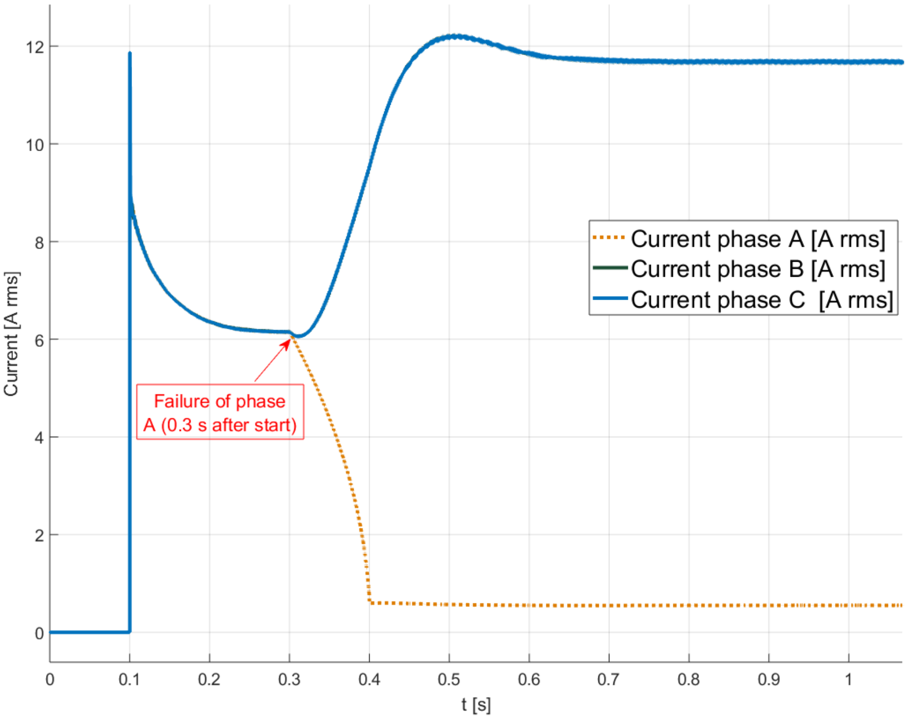


FIGURE 5.10: RMS value of currents in the stator windings after the failure, at nominal speed.

detected. On the other hand, tuning properly the thresholds and adopting an highly sensitive sensor, this technique could also detect a growth of the resistance, induced by a partial damage of the cable and/or an increasing temperature. Another disadvantage is related to the sensor positioning problem. In fact, voltmeter's terminals has to be constantly connected to the control board and the motor winding. This disconnection may fail leading to false alarms. In fact, as well as the mechanical stresses may cause the phase failure, also the voltmeter terminals can be affected.

▷ **Current sensors**. The current sensors has to be placed in series with the stator phase. Then, a failure of the same sensor could be the cause of the phase disconnection. Nevertheless, this solution seems to be much more feasible for the detection of failures, in the stator windings and transistors bridge too. The use of high-reliable current sensors is dutiful. Furthermore, the current sensors can be placed either upstream or downstream the cable which connects the transistors to the stator windings. The second case corresponds to the current sensors used for the feedback in the control loop. These same sensors can be exploited for the fault-tolerant system too, reducing costs and making the algorithm directly applicable to an existing control.

Due to the general purpose of the fault tolerance system, both solutions are implemented and tested in an ideal condition, in which sensors are perfectly working. Besides the detection of the phase disconnection event, the identification of which is/are the phase/phases damaged is essential for the fault tolerant algorithm. In fact, once the fault has been detected, the tolerant algorithm needs to know which are the remaining stator windings, in order to adapt the commutation sequence to the new configuration. This is the main goal of the detection and diagnostic block, introduced in series with the fault-tolerant driver. The *fault_detection* block consists in two subroutines: one dedicated to the continuous phase monitoring, and the second which analyses the alarms triggered by the current and voltage sensors. The fig. 5.11 shows the subroutine model dedicated to the continuous monitoring of the three phases. In the same image the reader can find a detail of the current and voltage alarms blocks. These blocks basically consists in a sensor in series with a comparator, repeated per each phase. In particular, the current alarms are triggered only if the phase current stays under $1A$ for at least $0.05s$. This time delay allows to apply the limit checking directly on the AC signal. Regarding to the voltage, if a disconnection occurs, the voltage between the output of the transistors bridge and the input of the stator winding is not zero, and corresponds to the PWM signal with V_{DC} amplitude. To avoid fluctuating signals and false alarms during the transition phase, the limits is set to 2 volts instead of zero.

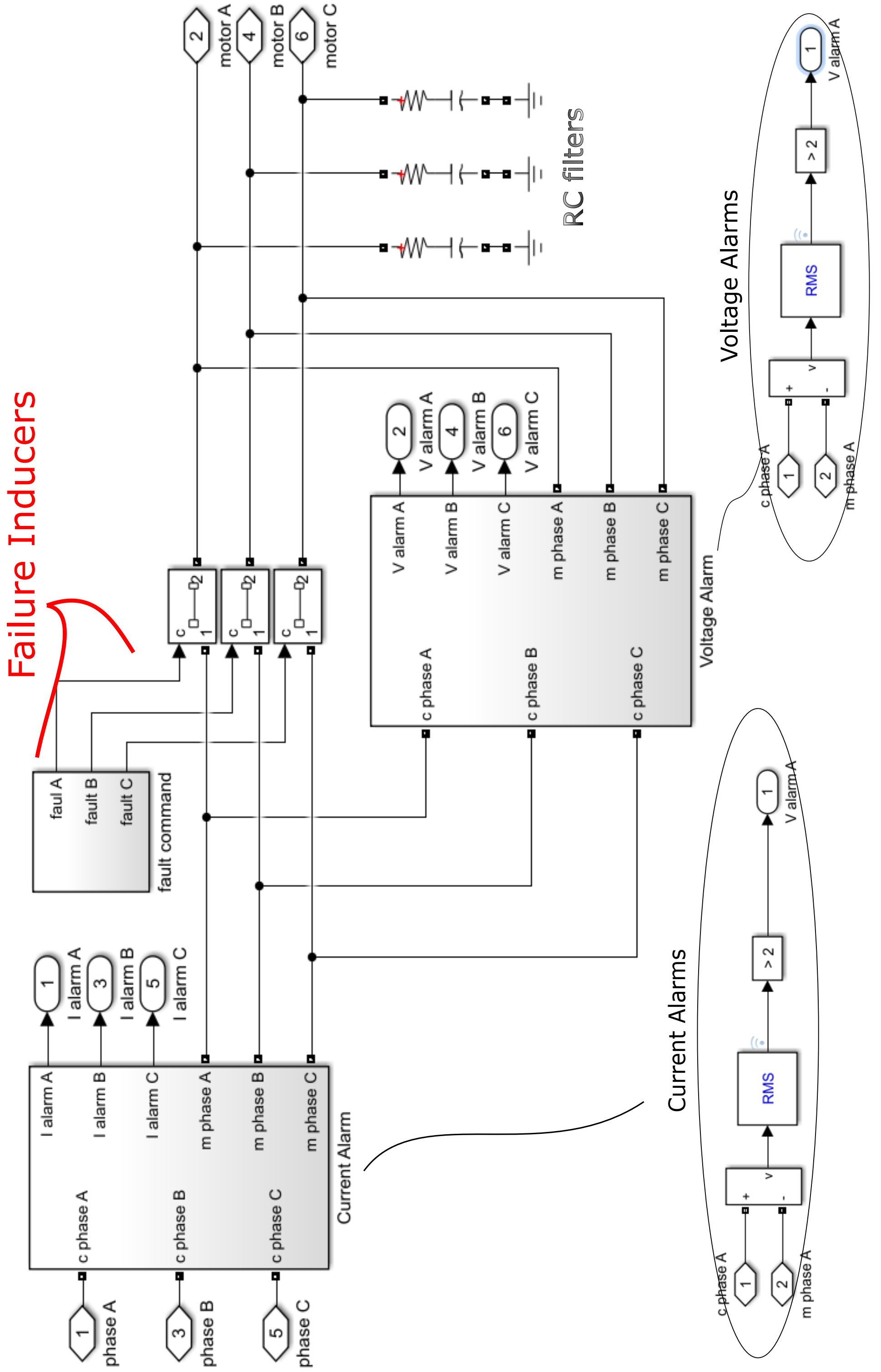
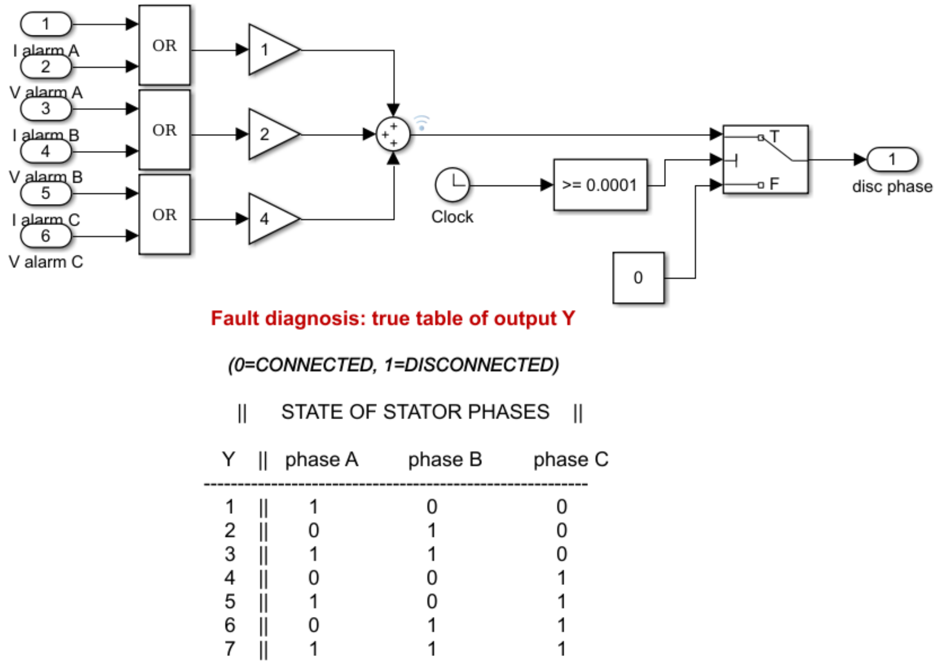


FIGURE 5.11: Overall scheme of the fault detection block.

The alarms are two per each phase and have a boolean interpretation: 0 means that no failure has occurred, while 1 corresponds to the failure condition. In order to detect the phase affected, the signals are multiplied for a specific weight, whose interpretation is resumed in the truth table shown in the scheme of the diagnosis block (fig. 5.12).



Fault diagnosis: true table of output Y

(0=CONNECTED, 1=DISCONNECTED)

|| STATE OF STATOR PHASES ||

Y	||	phase A	phase B	phase C
1	||	1	0	0
2	||	0	1	0
3	||	1	1	0
4	||	0	0	1
5	||	1	0	1
6	||	0	1	1
7	||	1	1	1

Figure 5.12: Fault diagnosis algorithm.

During the start-up with one phase disconnected, the system behaves as in normal conditions until it reaches the saturation. The motor saturates with a speed error stables at $1285RPM$ (fig. 5.13). As already said, the detection of the phase failure is based on the current measurement, verifying that the current is bigger than $1A$. The fault variable is forced to zero for the first $100\mu s$, to do not interpreter as a phase disconnection the no-current condition before the start-up. After this period, if one phase were actually disconnected, the current alarm will remain active switching the driver to the fault tolerant algorithm.

5.4.3 Fault tolerant algorithm

The three-phase inverter becomes inappropriate to drive a BLDC motor with only two phases available [10]. In order to avoid the saturation of the system

[10] As anticipated in the introduction, the three-phase motor can tolerate maximum one phase disconnection. For more than one failure, the system simply detect and diagnose where the failure occurs.

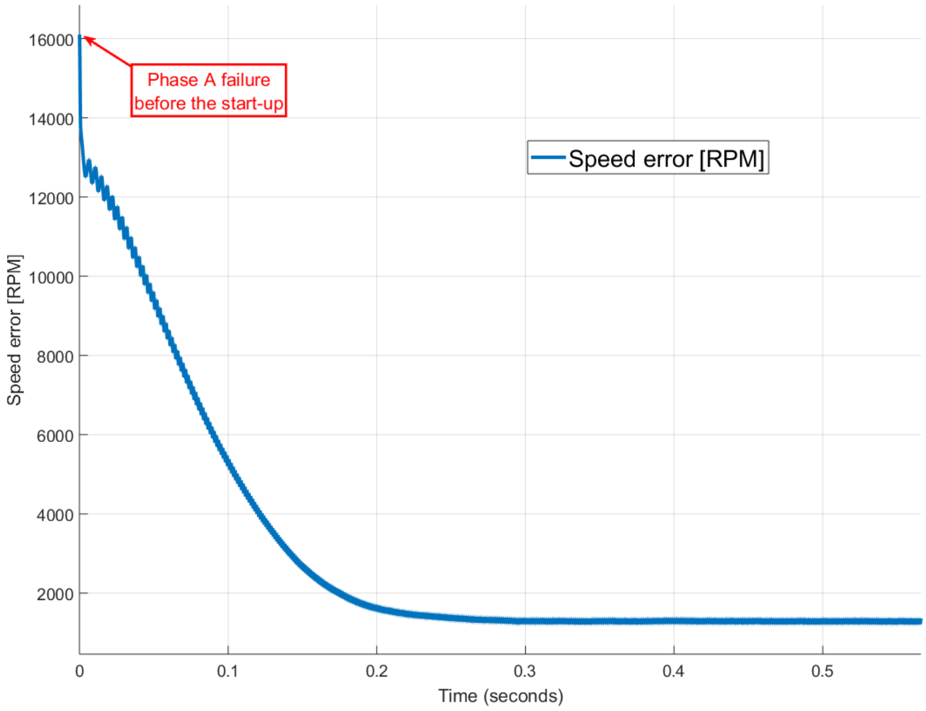


FIGURE 5.13: Speed error trend with a single phase failure before the start-up.

at rotor speeds lower than the nominal value, the remaining stator phases has to be supplied more efficiently. The algorithm is quite intuitive looking at the remaining active states that can be exploited (fig. 5.6). To produce a positive torque[11] the commutation has to occur each 180^o, instead of 60^o.

Once the fault is diagnosed, the available active states are known, as well as their distribution in the space. Through an encoder, hall signals or an estimation of the angle based on the BEMF, it is possible to determine the position of the magnetic field of the rotor. Then, the tolerant system maintains the angle between the stator current vector and the magnetic field in the range $180^o \rightarrow 0^o$. Once the two vectors are parallels, the commutation to the dual state keeps the rotor running. Referring to eq. (1.1) in the introduction chapter, it is possible to demonstrate that maintaining a constant current flow, for the entire period between one commutation and the following, the torque amplitude has a sinusoidal distribution with time (or angle). For this reason, periodical acceleration and deceleration phases will occur, making impossible to achieve a constant speed. Hence, big speed ripple has to be expected. This undesirable behavior and high phase current peaks require a change in the basic algorithm. As said

[11] Accordingly with the rotation sense.

in the FOC chapter, the sinusoidal PWM allows to obtain higher efficiency, thanks to the sinusoidal wave-shape of the phase current. Hence, the idea of implementing an hybrid trapezoidal-sinusoidal driver. Basically, the remaining active transistors are driven by a PWM signal, modulated with an half-sinusoid (fig. 5.14). This wave has a frequency equal to the electric frequency and the amplitude proportional to the PI output. The PWM signal generated with the process above mentioned is shown in the figure 5.15.

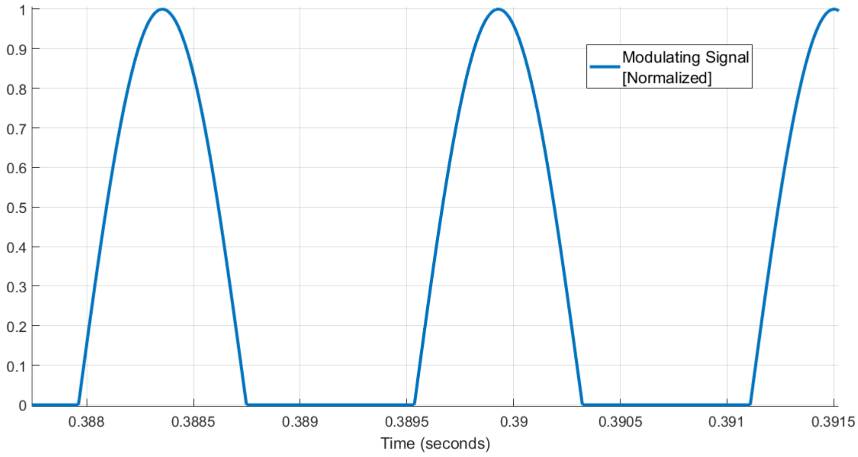


Figure 5.14: Modulating wave in fault-tolerant driver.

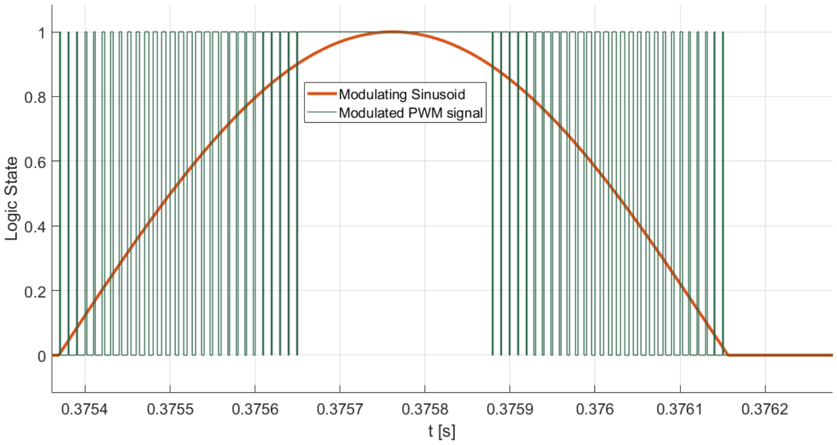


Figure 5.15: Modulating wave and PWM signal used by the fault-tolerant driver.

The fault-tolerant controller developed in Simulink is shown in the figure 5.16. It consists in three main blocks:

- **Post-fault hall**. This is the block indicated by the label 4 in red. It has two inputs: the electrical angle which determines the rotor position in each instant, and the *fault* signal. This latter has already been defined in the fig. 5.12 as the *disc phase* signal, referring to the truth table represented in the same figure. It determines both the failure event and which phases are floating. Through this two inputs, the block generates the post-fault halls which are fictitious hall signals able to properly drives the decoder and gates blocks[12]. Instead of providing all the six active state, the *post-fault halls* corresponds to the remaining active states, commutating between them in the way explained at the beginning of this section.

- ▷ **Phase Current Smoother.** The transformation from the basic gates to an half-sinusoidal PWM signals is achieved with this block. Its structure is represented in the detail circled in the same figure. It basically consists in three elements:

  1. PWM generator, whose duty cycle is determined by the the amplitude of the input signal[13].
  2. Sinusoidal function, whose frequency is determined by the *e speed* signal[14], while the amplitude is determined by the normalized control signal.
  3. The time variable, used to determine the argument of the sine, is provided by a clock. The timer has a reset input to enable the counter only when the correspondent gate is active.

- ▷ **PI**. The control coefficients has to be adapted for the post-fault configuration. Since the control structure is the same of the six-step controller with sensors, the PI coefficients are reused in this case[15].

The chapter 7 resumes and compares the results obtained with the fault tolerant system with the performances of the un-failed case.

[12] Decoder and Gates blocks has been described in the Motor PWM Driver section (sec. 2.1.1)

[13] The amplitude of the input signal is within the range $0 \div 1$.

[14] Instantaneous electrical speed measured with a tachometer.

[15] P = 0.00050581023378322, I = 0.016188855441962. The initial conditions of the system is forced to I = 3.5, optimized for the failure at the nominal speed condition.

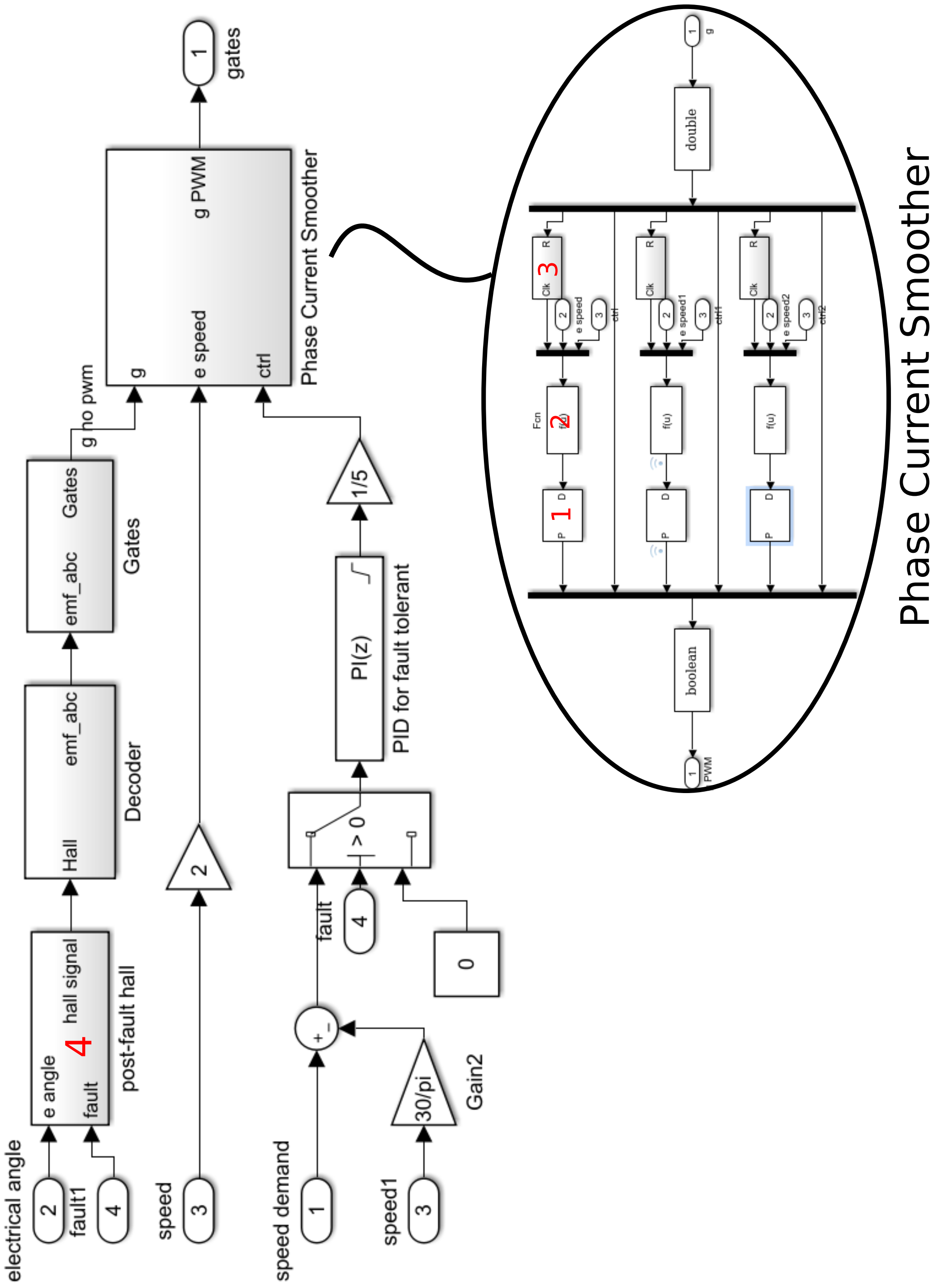


Figure 5.16: Fault-tolerant controller scheme.

Chapter 6

Final Results

This chapter resumes the remarkable results collected during the simulations. They refer both to the simulation of the control algorithms and fault tolerant system. The objectives are: to verify if the fault tolerant system is working correctly and to point out which is the most suitable control, depending on the application, by comparing the overall performances of each technique.
The simulations are several and featured by different parameters. Hence, in an unique paragraph these parameters are resumed and tagged. The labels are then reported in the caption of graphs and tables, to give an intuitive correlation between the result and the corresponding simulation conditions.

6.1 Simulation parameters

In order to correctly interpreter the results reported in this chapter, it is mandatory to associate them to the conditions in which they have been obtained. The features of the solver are always the same. What changes is the time step, indicated with the variable T_s.

Solver Features	
Type	Fixed step
Time Step	T_s
Algorithm	Automatic option
Periodic sample time constrain	Unconstrained
Simulation mode	Accelerator

Table 6.1: Solver parameters.

Other fixed parameters are:

- Nominal Load: $0.09Nm$

- Nominal Speed: 16100 RPM

While the simulation variables are:

Case N^o**1**	
Ts	$1e-7s$
PWM$_{freq}$	$20kHz$
Case N^o**2**	
Ts	$1e-7s$
PWM$_{freq}$	$150kHz$
Case N^o**3**	
Ts	$1e-6s$
PWM$_{freq}$	$20kHz$
Case N^o**4**	
Ts	$1e-6s$
PWM$_{freq}$	$150kHz$

TABLE 6.2: Simulation variables.

In order to obtain results for a period of $2s$ with an Intel processor i5-7600, 4 cores CPU @ 3.8GHz, the simulation requires:

- Trapezoidal: $\leq$ 1 minute

- FOC: $\leq$ 1 minute

- Trapezoidal Sensorless: $\leq$ 5 minutes

- FOC Sensorless: $\leq$ 15 minutes

- FOC with fault: $\leq$ 2 minutes

NOTE: Any transition period is considered as being concluded when the relative error of the magnitude, compared to the final value, is smaller than 1%.

6.2 Trapezoidal Control Performances

The trapezoidal control, despite its simple logic, becomes much more complicated in the sensorless configuration. In fact, its operation deeply depends on the filtering characteristics of the feedback signals. In particular, the rotor speed can be estimated by the fundamental frequency of the phase current, or of the BEMF signal. Both of them have a wide frequency spectrum, which makes difficult to design a filter feasible for a wide speed range. Admitting also big jumps in the speed demand, the optimum design becomes even more difficult. All this complications, together with worse performances, make wonder if the sensorless version is indeed preferable to a six-step controller with hall sensors, to cut costs and increase the reliability.
As anticipated in the *Trapezoidal Control* chapter (cp. 2), in this case the phase current is not sinusoidal. Figure 6.1 shows the waveform obtained with this control technique. In particular, the image refers to the current in phase A, at nominal speed conditions and obtained in the sensorless configuration. The commutation event corresponds to a rising or falling edge. Unfortunately, undesired current peaks are added to theses edges, due to the delay that virtual signals have, compared to the actual hall waves. These peaks can be reduced by optimizing the design of the snubber circuit and of the BEMF signal filter. The key is suppressing higher harmonics generated by the PWM signal.

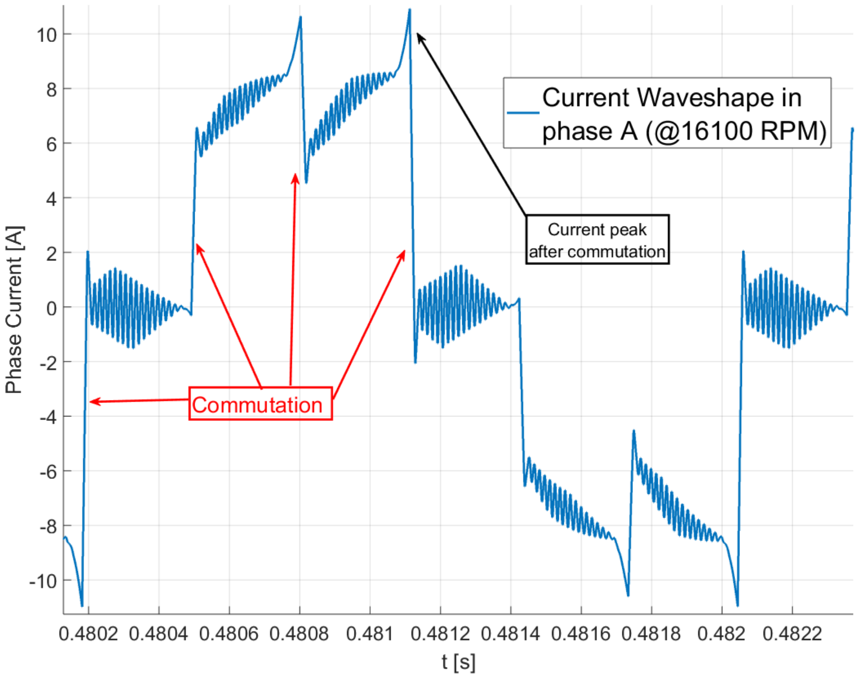


Figure 6.1: Waveform of the stator phase current obtained with the six-step control algorithm (2).

Besides increasing the efficiency, the improvement of the virtual hall signals may favour a better rotor speed estimation. In fact, the filtering of the fundamental component get worse as much as the speed value moves aways from the nominal condition. Then, the speed estimator fails due to the incorrect determination of the signal period. The same behaviour is obtained both by filtering the BEMF or current wave. The fig. 6.2 shows the effect of an unfeasible filtering at very low speed, by overlapping the trends of the estimated and measured speed. In particular, such drawbacks arise during a deceleration transition. That is why a smooth deceleration test, from 8000 to 2000 RPM, has been carried out with a double objective: determine the minimum rotor speed that system can control; and verify if this limit is caused by the speed estimation.

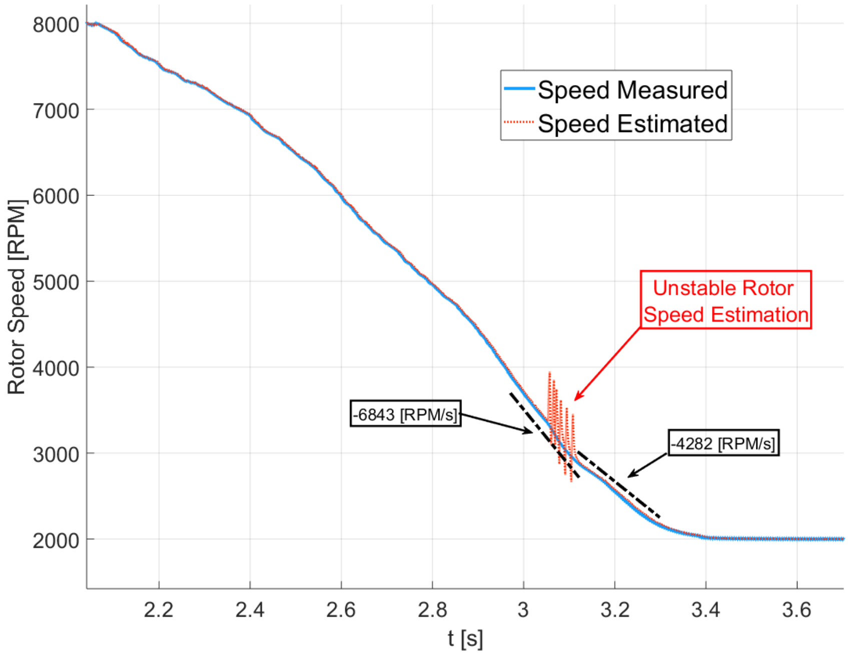


FIGURE 6.2: Minimum speed limit which the sensorless trapezoidal algorithm can control. The measured speed trend is compared with the estimated one (2).

In this specific case, the speed has been estimated by using the current fundamental wave. The speed estimation matches the real speed as much as the speed changes smoothly. The slower is the speed, the smaller is the deceleration rate admitted to avoid errors in the estimation. An example of temporary unstable estimation occurs at around 3000 RPM. The fault is recovered with a reduction of the deceleration rate, letting the motor to finally achieve the minimum admitted speed. The start-up is shown in the fig. 6.3 where the trends of the estimated and measured speed are compared. This initial phase consists in three steps, that will not be the same as for the sensorless FOC:

1. The trend of the rotor angle is assumed to be parabolic. From the angle

information, the three virtual hall signals are generated and provided to the driver.

2. Once the rotor speed gets to 5000 RPM, the feedback is connected to the PI input while achieving the 7000 RPM[1].

3. After $0.8s$ the control loop is finally closed, reacting to achieve the nominal speed.

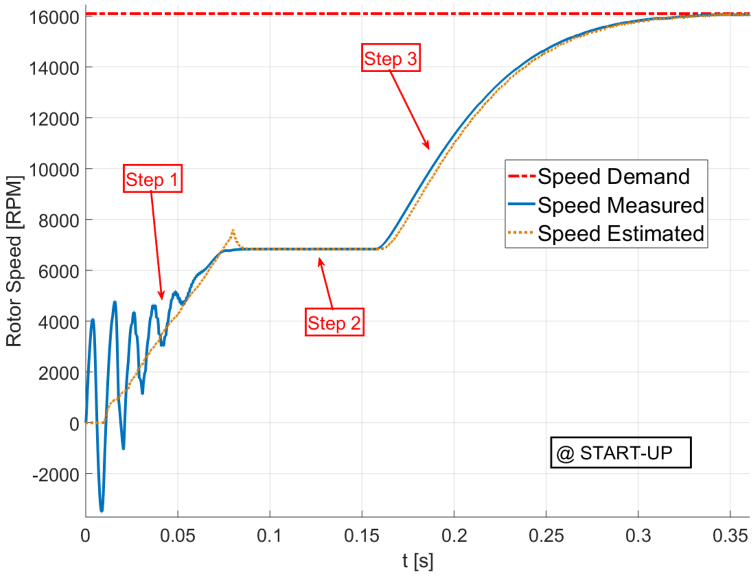


Figure 6.3: Comparison between the estimated and the measured rotor speed during the start-up phase of the six-step controller (2).

Finally, the figures 6.4, 6.5 and 6.6 compare the behaviours of the six-step control with or without hall sensors. The same comparison is carried out in three representative situations: start-up, acceleration and deceleration. The sensorless configuration needs $0.3184s$ for the start-up, has an avarage[2] acceleration rate of $41176[RPM/s]$, while the deceleration is much more smooth with a rate of $18128[RPM/s]$. This deep difference is due to the *speed demand filter* introduced to avoid a deceleration response too quick, that causes intolerable instabilities of the estimated speed. On the other hand, the trapezoidal control which uses hall sensors has much better performances: $0.1655s$ for the start-up, acceleration rate of $42424[RPM/s]$ and deceleration rate of $60039[RPM/s]$.

[1] This intermediate phase is necessary for the PI settling. Whithout it, the speed demand 5000 RPM to 16100 RPM would cause strong instabilities in the control loop, making impossible to conclude the start-up phase.

[2] Calculated as $\Delta SpeedDemand/\Delta t$, where $\Delta SpeedDemand$ is the jump in the speed demanded and Δt is the time necessary to achieve the speed demanded with a maximum relative error of 1%.

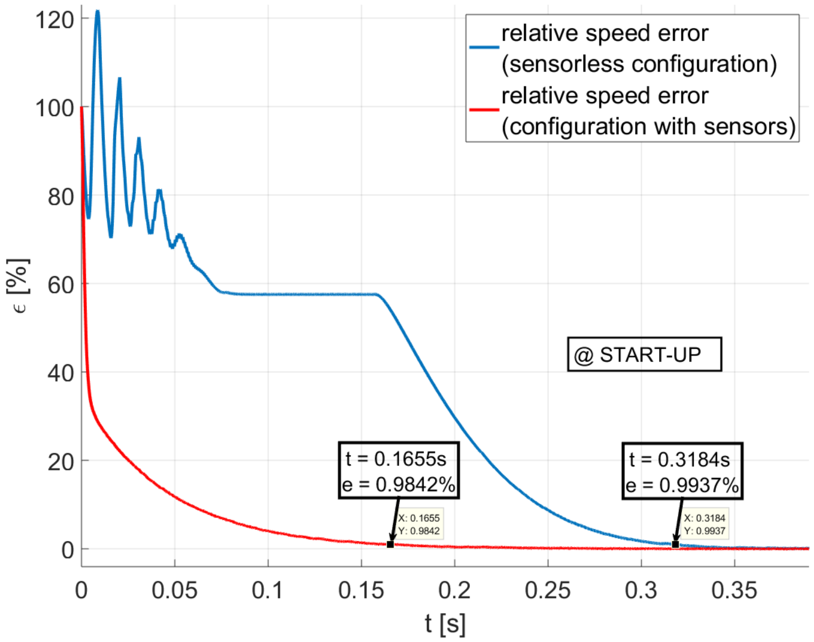


FIGURE 6.4: Trend of the speed's relative error during the start-up, for a six-step controller w or w/o speed and position sensors (2).

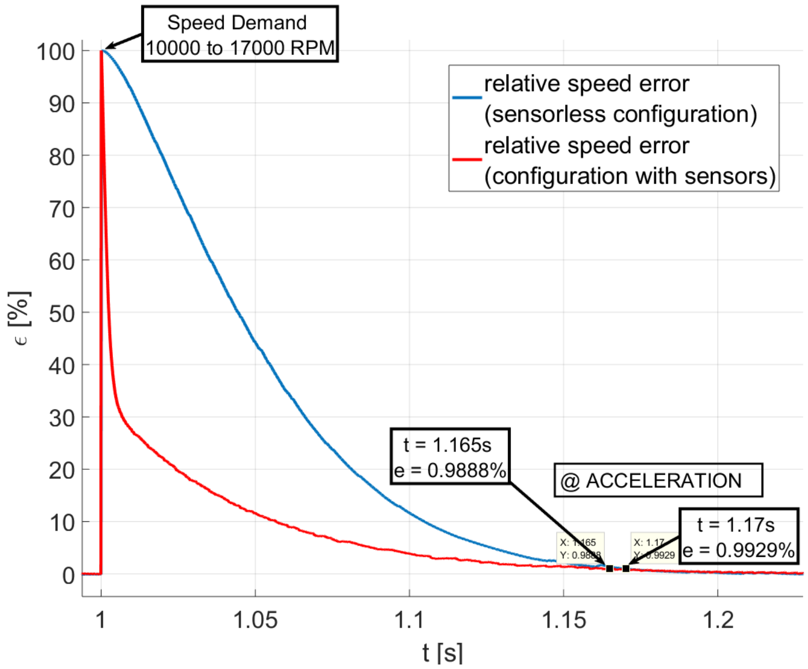


FIGURE 6.5: Trend of the speed's relative error during the acceleration phase, for a six-step controller w or w/o speed and position sensors (2).

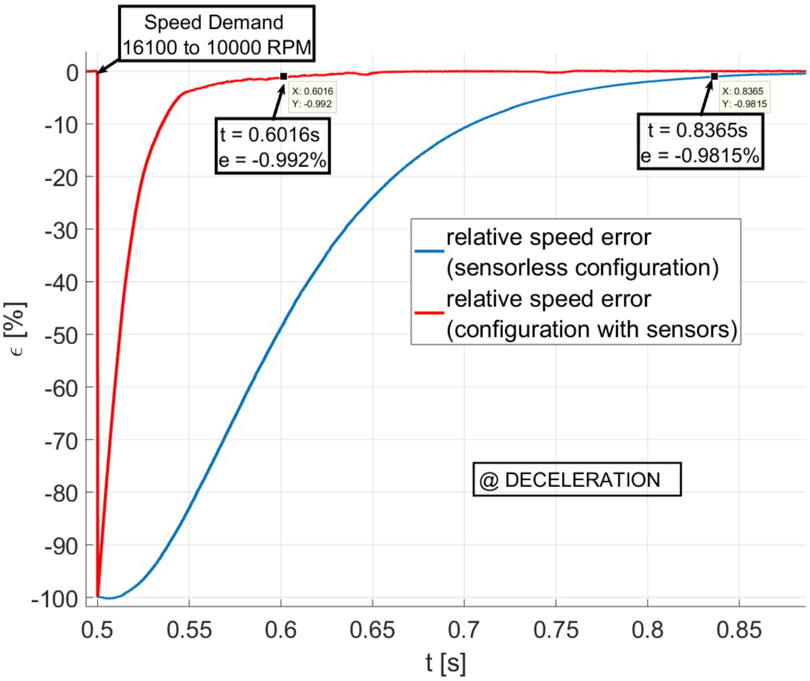


Figure 6.6: Trend of the speed's relative error during the deceleration phase, for a six-step controller w or w/o speed and position sensors (2).

6.3 FOC Performances

The FOC system has better performances than six-step controller, either using or not position and speed sensors. In fact, it is faster and more efficient, thanks to the direct and precise control of the torque and magnetizing flux. Furthermore, it is easier to adapt the algorithm to a sensorless configuration, still maintaining its applicability to a wide operational speed range. The figure 6.7 compares the current waveforms obtained at the rotor nominal speed with an SVM that has a switching frequency of $150kHz$ or $20kHz$[3]. Its frequency is proportional to the rotor speed, while the amplitude increases with the electromagnetic torque. The sinusoid is disturbed by higher frequency harmonics, generated by the continuous torque control and the PWM signal. Increasing the pwm frequency allows to reduce the maximum ripple from 30% to 5.5%, at nominal speed conditions (fig. 6.8).

NOTE: the percentage refers to the ratio between ripple amplitude and the rms value of the torque.

As already said, the FOC algorithm needs to know the rotor angle at every instant. In the sensorless configuration the angle is estimated, introducing a delay of $12.7\mu s$ and a maximum ripple amplitude of 3.577^o, at nominal rotor speed condition (fig. 6.9 and fig. 6.10). Once again, the digital *speed estimator* block determined the rotor speed from the frequency of the fundamental harmonic of the current spectrum. The speed estimation is much more reliable and stable compared to the six-step control case, throughout the whole operational speed range. Consequently, transitions during start-up, deceleration or acceleration are much faster than the previous case, as thoroughly described in the next section.

[3] The PI controller's sample frequency can be increased obtaining a similar improvement in the speed and torque ripples. The PI speed is quite smaller than the real capability of a microprocessor [44]. This is because the solver time step would be too small to simulate higher speeds of the PI.

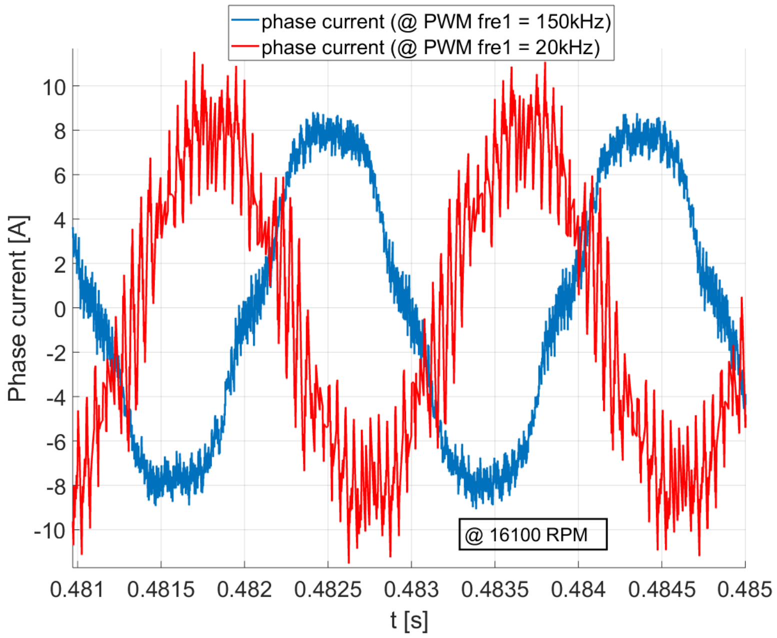


Figure 6.7: Waveform of the stator phase current obtained with the FOC algorithm.

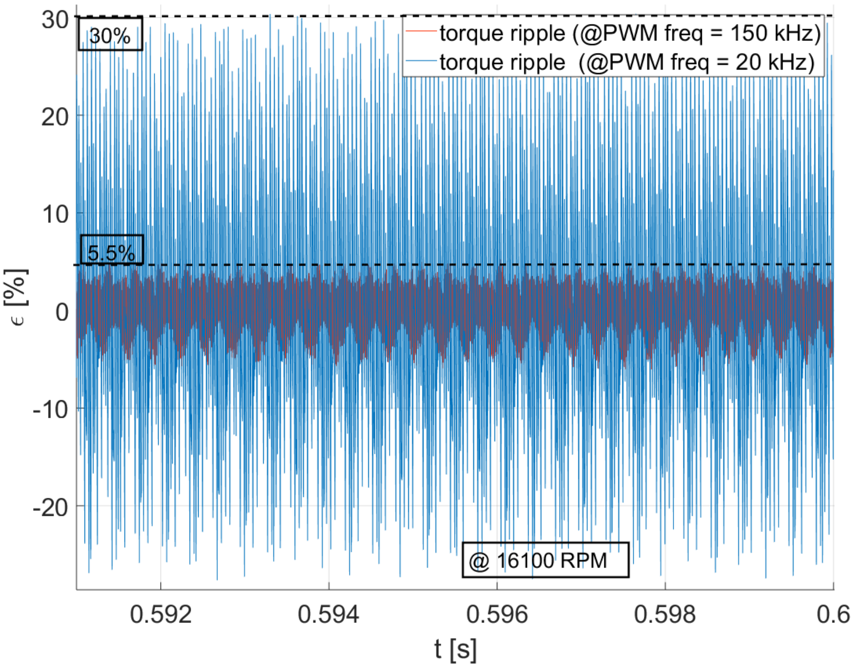


Figure 6.8: Torque ripple obtained with an inverter switching frequency of $20kHz$ and $150kHz$ (1 and 2).

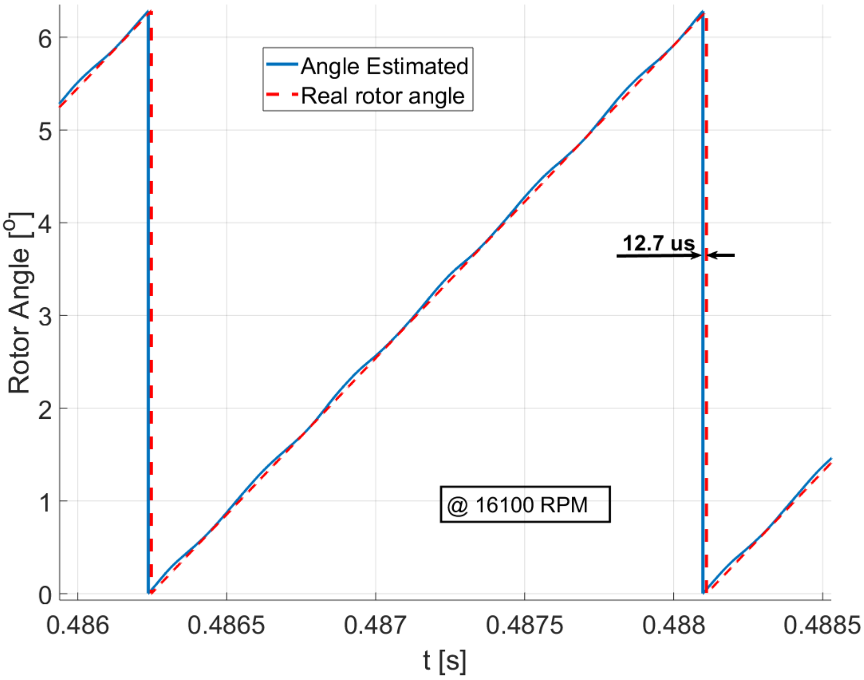


FIGURE 6.9: Comparison between the rotor angle estimated by BEMF signal and the encoder measurement (4).

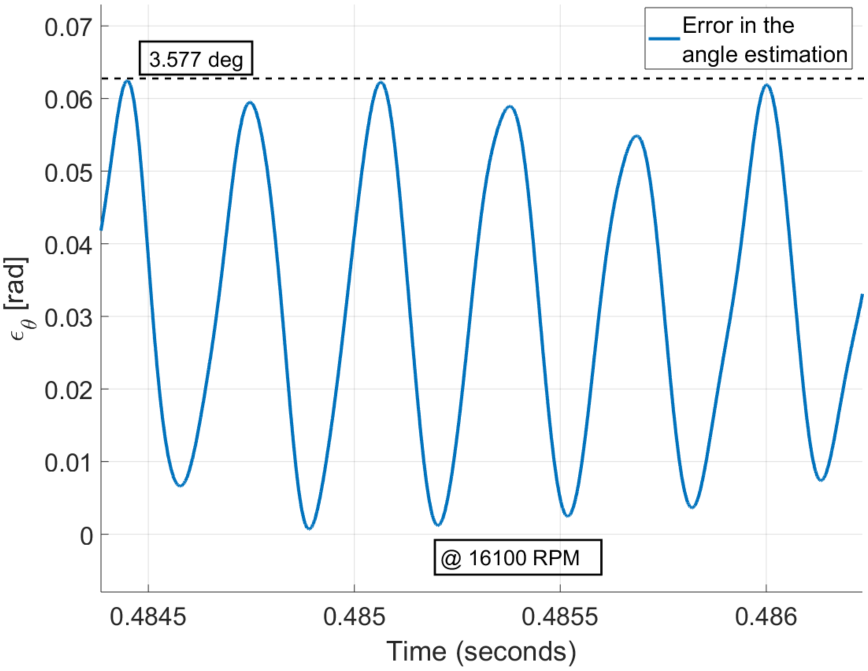


FIGURE 6.10: Relative error of the estimated rotor angle compared to the encoder measurement (4).

6.4 Comparative of the Control Techniques

FOC and trapezoidal controls are finally compared in order to determine, depending on the application, which is the most feasible algorithm. The following five performances have been considered useful for a first evaluation: trends and rates of the acceleration and deceleration phases, torque and speed ripple values and sensorless start-up features. The main goal of the thesis is the design a control without encoders, that is why the graphs proposed below shows to the performances that models have if equipped exclusively with voltage and current sensors. Also the estimated and measured speed trends are compared to demonstrates how easier is the speed estimation in the FOC compared to the six-step control case.

The start-up is shown in fig. 6.11 in terms of the speed relative error. The three curves are shown which refer to: measured and estimated speed in the FOC sensorless model and the measured speed in the sensorless six-step controller. The sensorless start-up in the FOC is more homogeneous than in the six-step controller case. This advantages is due to the better speed estimation and the control of d and q currents since the beginning. Hence, during the start-up only the inner loop is closed. While the outer loop is closed once achieved the start conditions, by switching the inputs of the Clark-Park's blocks from the assumed start-up angle to the estimated angle. The switching instant is indicated in fig. 6.11, where the red curve has a sudden change of slope. The start-up ends after less than $200ms$, saving almost $125ms$ than the trapezoidal control case, with reduced speed fluctuations and smaller peaks in the starting current.

The figures 6.12 and 6.13 compare the performances during acceleration or deceleration of the motor, demanded by the user with a speed step. The acceleration rate reaches the $63063[RPM/s]$, while the deceleration has an avarage rate of $45865[RPM/s]$. The difference between the two rates is not so high as in the case of the trapezoidal control, since the problems arise because of the speed estimation.

Finally, the sensorless FOC algorithm reduces the maximum amplitude of the speed ripple form 0.068% to 0.036%, and of the torque ripple form 32.79% to 5.5%, in favor of substantial improvements in the overall system efficiency, as well as less mechanical and electrical stresses.

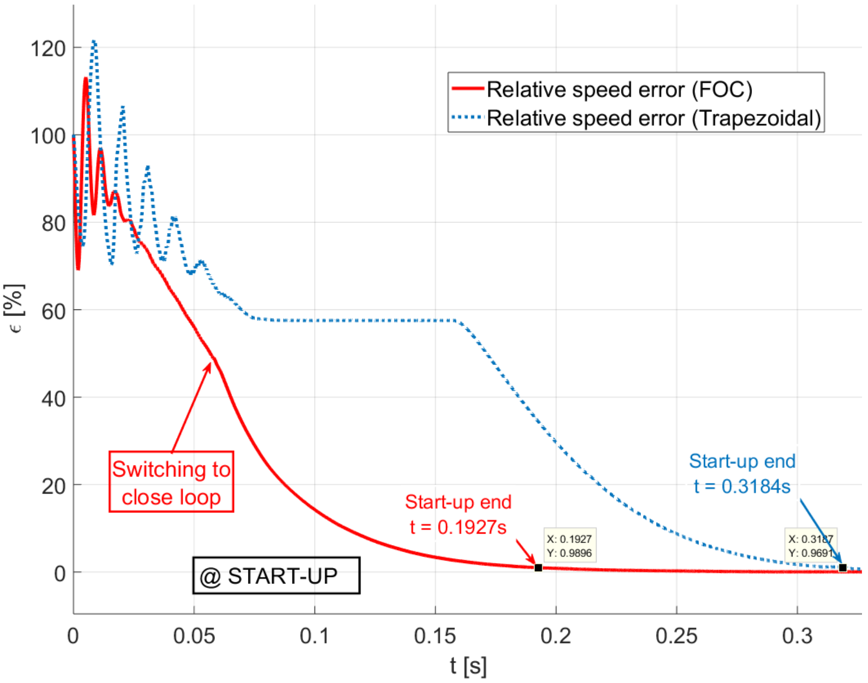


FIGURE 6.11: Comparison of the trends followed by the speed relative error during the start-up, both for FOC and trapezoidal control (2).

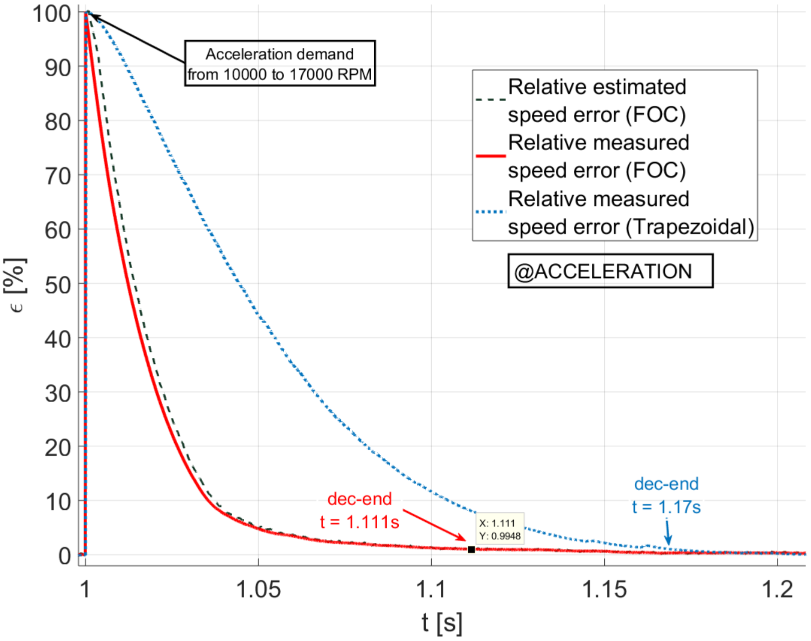


FIGURE 6.12: Comparison of the trends followed by the speed relative error during the acceleration phase, both for FOC and trapezoidal control (2).

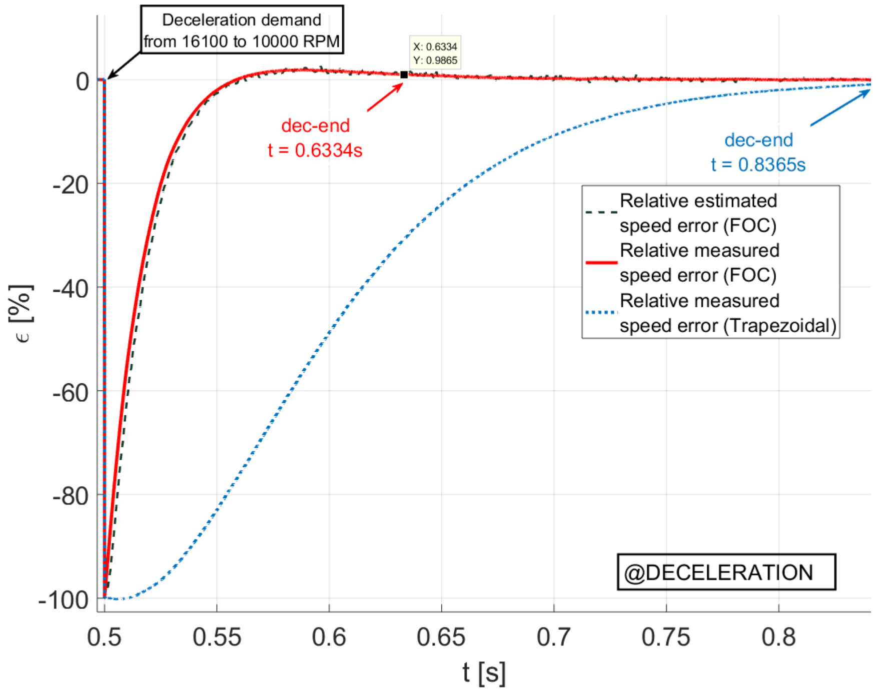


Figure 6.13: Comparison of the trends followed by the speed relative error during the deceleration phase, both for FOC and trapezoidal control (2).

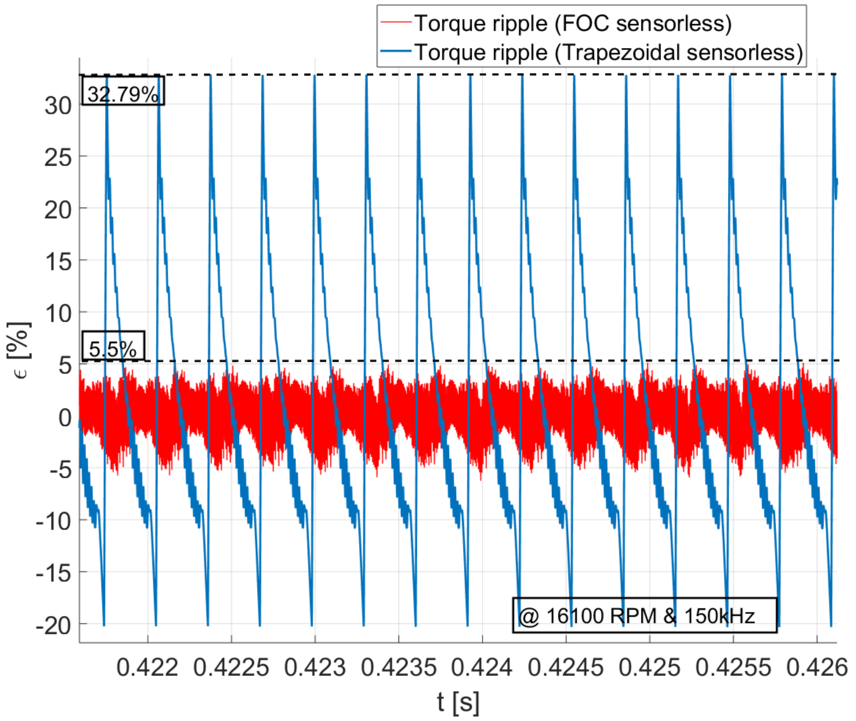


Figure 6.14: Comparison of the relative torque ripples obtained with FOC and Trapezoidal sensorless control, at nominal speed (2).

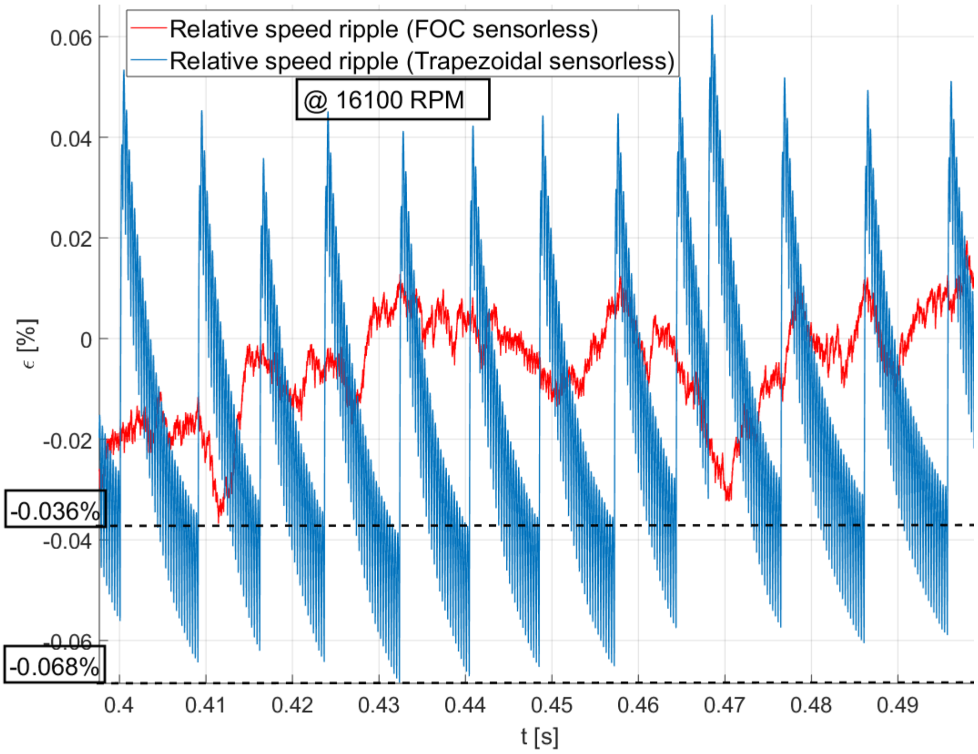


FIGURE 6.15: Comparison of the relative speed ripples obtained with FOC and Trapezoidal sensorless control, at nominal speed (2).

6.5 Fault Tolerant Driver

The problem of the phase failure has been introduced in the chapter 5. Failure of phase A is the case analyzed, and the results reported below refer to this specific event. Obviously, the same observation can be extended in case of failure of phase B or C. The fault tolerant algorithm is implemented on a FOC model equipped with an encoder for the rotor speed and the position measurement. The same simulation has been carried out on a six-step controller too, obtaining similar results. The tolerant algorithm designed can be easily integrated with sensorless control, since it requires only few changes of the PI coefficients.
The fault diagnosis system needs less than $1ms$ to detect the phase failure and distinguishes which is the phase disconnected. Assuming that the failure occurs at nominal speed, the rotor experiences a transitional phase before recovering its nominal speed. This transition is almost $0.6s$ long, a period that can be reduced to $0.15s$ by increasing the PI velocity by a factor of ten. In fact the PI velocity proportionally depends on the sampling frequency of the solver[4]. The figure 6.16 shows how the speed error evolves after the phase failure, comparing the results obtained with a sample period of the solver that changes from $1e-6s$ to $1e-7s$. It is evident that, if the absolute speed error cannot reach values bigger than $200RPM$[5] during the transitional phase, the design will need a microprocessor with high speed. Those undesired fluctuations can be observed also after a deceleration transition, as shown in the fig. 6.17. In both cases, just after the failure event or the deceleration, the fluctuations can be eliminated by increasing ten times the PI sample frequency (fig. 6.17). The speed ripple that has a relative amplitude of 0.03% in the nominal operating case , increases to 0.168% once finished the transitional phase with the fault-tolerant driver.
To avoid those fluctuations, since now on all the results has been obtained with a solver period of $1e-7s$. The same deceleration and acceleration transitions observed for the FOC without phase failure, are simulated again with the phase A disconnected. The performances are just slightly degraded in both cases: deceleration with a rate of $37888RPM/s$ instead of $47656RPM/s$[6] (fig. 6.19), and acceleration with a rate of $40229RPM/s$ instead of $48951RPM/s$ (fig. 6.18). Finally, it is necessary to say a few words about the stator current waveform and the electromagnetic torque obtained with the tolerant system. In the fig. 6.20 the current in phase B and the sinusoidal shape obtained in the healthy motor are compared. Even if the sinusoidal shape disappears in the faulty motor, the superior harmonics are limited thanks to the sinusoidal-pwm voltage signal. The current peak increases of +100%, from $8.5A$ to $17A$, which is negligible if compared to the +606% that is obtained in case of implementing a simple

[4] The sample frequency of the fault-tolerant PI is 10 times smaller than the sample frequency of the solver ($f_s = 1/T_s$)

[5] Relative error of 1.24%.

[6] -20.5%

commutation strategy. As shown in fig. 5.16, the tolerant system implements a single control loop with one PI controller, whose sample frequency is ten times smaller than the solver's, and a *phase current smoother* with a PWM switching frequency of $100kHz$. Also in the post-fault, increasing either the PI or PWM frequency increases the overall efficieny, and at the same time reduces the torque and speed ripple. This is shown in the fig. 6.21 where are compared the surplus of torque[7] that the stator has to produce to keep the rotor running at nominal speed. This excess is a sympthoms of how efficient the motor transform the power supply into mechanical energy. Hence, the higher is the surplus, the smaller is the efficiency. In case of nominal operating conditions only almost $0.011Nms$ more than $0.09Nms$ are required, while if phase A fails the surplus required reach almost $0.0165Nms$. This value further degrades to $0.021Nms$ if the PI speed is ten time smaller. The primitive cause of reduction in efficency is the availability of only two state vectors, which arise the torque ripple from 5.5% to 114% (fig. 6.22).

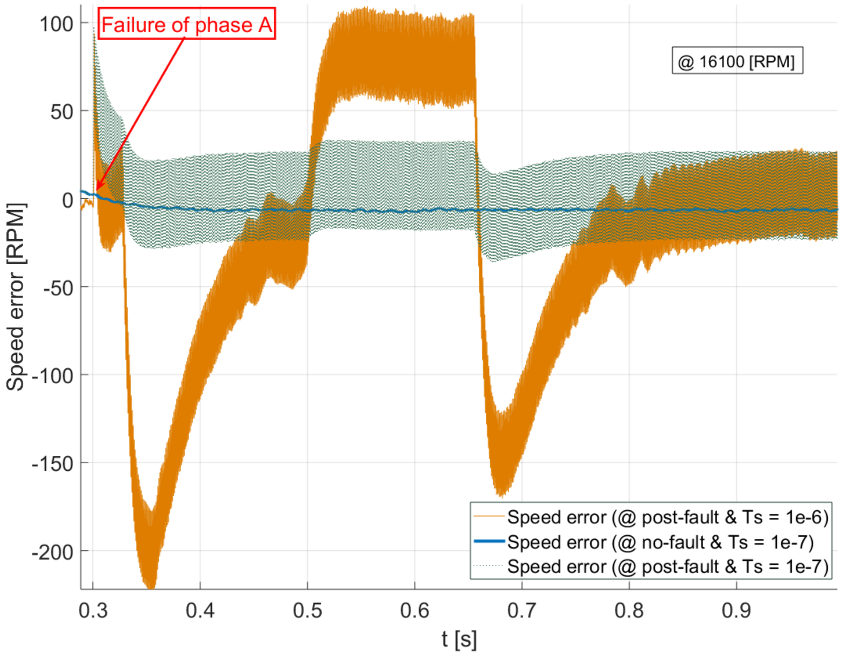


FIGURE 6.16: Speed error trend just after switching the driver to the fault tolerant system. The phase transition is compared in case of $Ts = 1e-6$ and $Ts = 1e-7$ (1 and 2).

[7] $T_{surplus} = T_{rms} - T_{load}$ where T_{rms} is the root mean square of the electromagnetic torque produced, and T_{load} is the nominal load $0.09Nm$

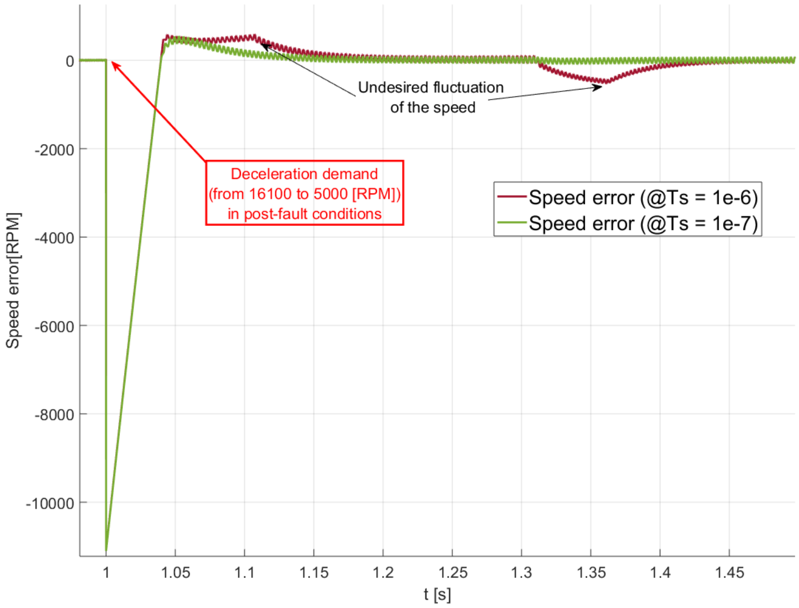


Figure 6.17: Speed error trend during the deceleration of faulty motor. The phase transition is compared in case of $Ts = 1e - 6$ and $Ts = 1e - 7$ $Ts = 1e - 6$ (1 and 2).

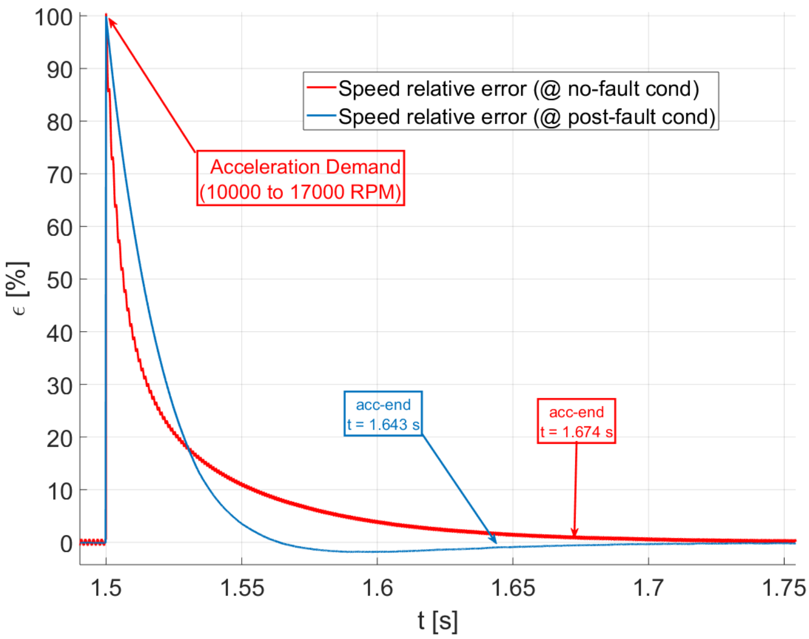


Figure 6.18: Comparison of the speed relative error during the acceleration phase obtained in the nominal operating and the faulty case (2).

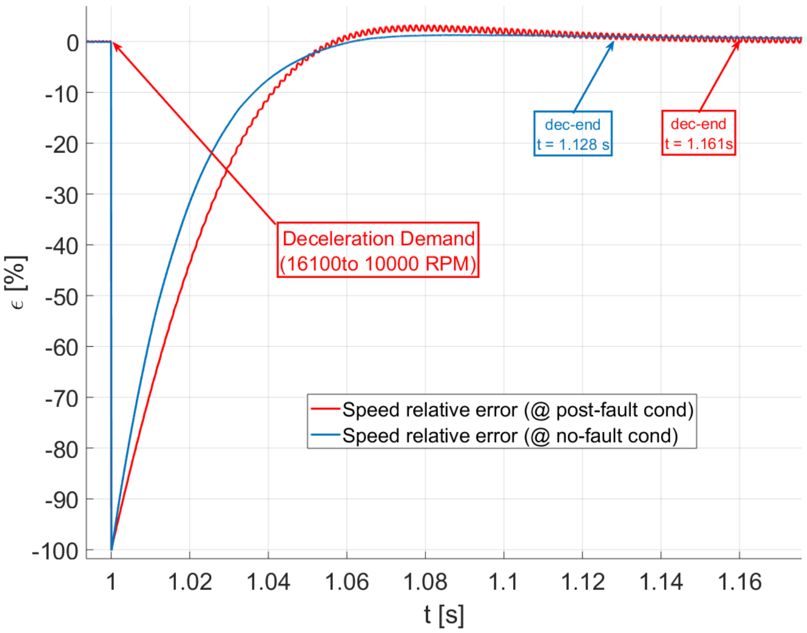


FIGURE 6.19: Comparison of the speed relative error during the deceleration phase obtained in the nominal operating and the faulty case (2).

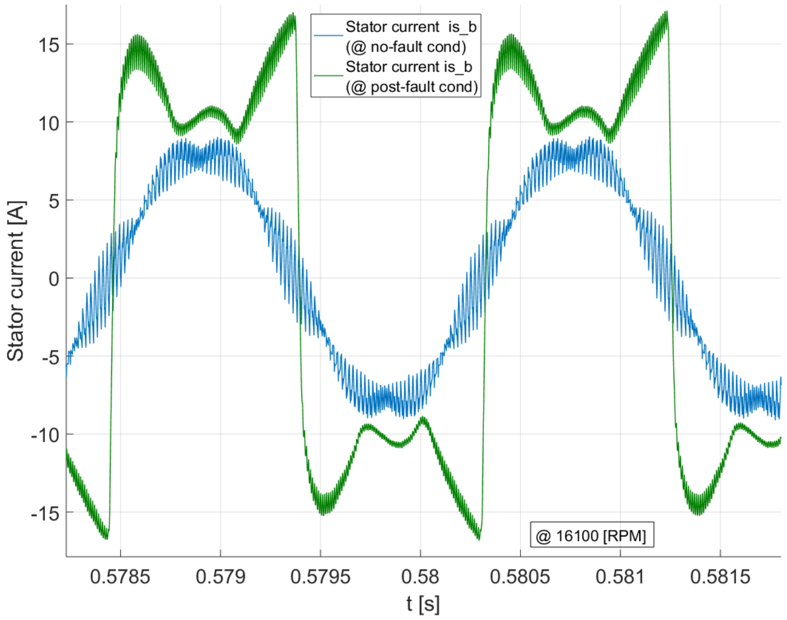


FIGURE 6.20: Comparison of the current waveform obtained in the nominal operating and the faulty case, at nominal speed (2).

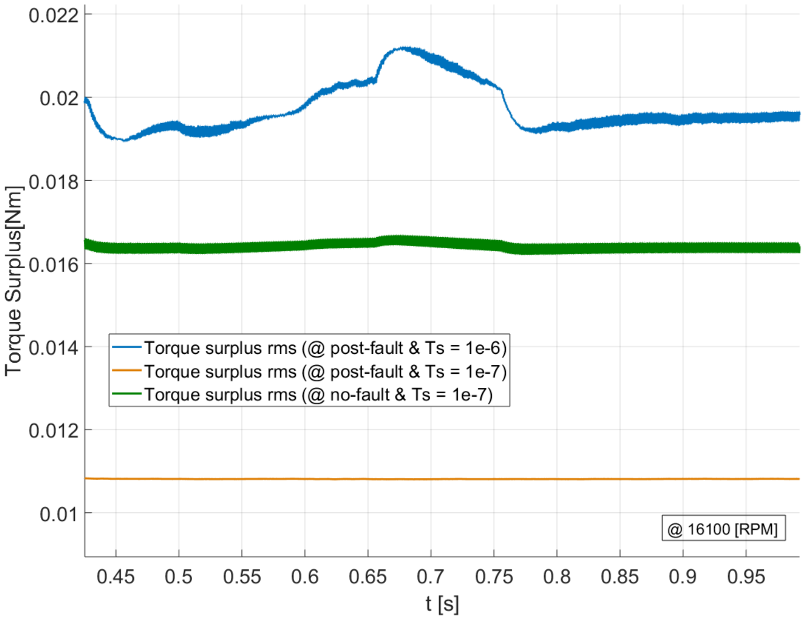


Figure 6.21: Comparison of the surplus of torque required at nominal speed obtained in the nominal operating and faulty case with different Ts (1 and 2).

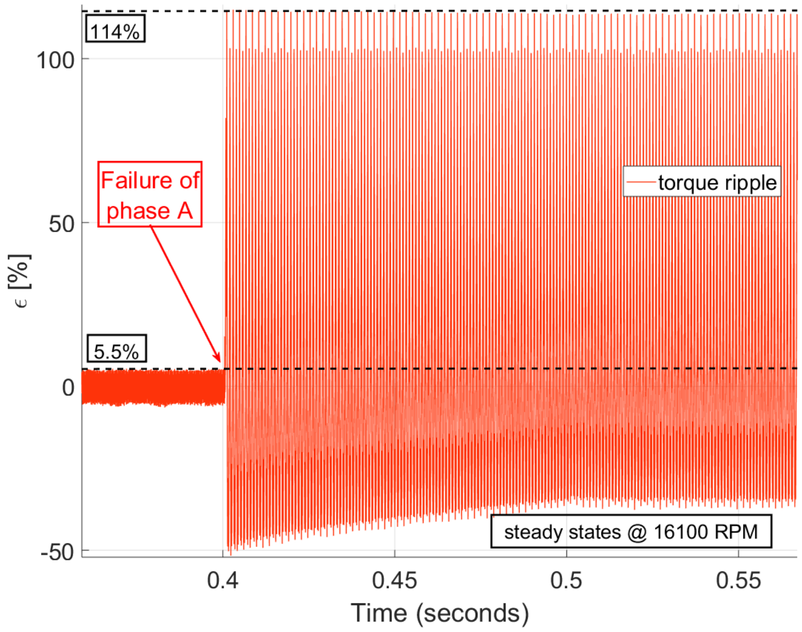


Figure 6.22: Torque ripple in percentage before and after the phase failure (2).

Summary of Performances

	TRAPEZOIDAL[8]	FOC[8]	FAULT-TOLERANT FOC
START-UP PERIOD	0.3184s	0.1927s	0.2335s
ACCELERATION RATE	41176 RPM/s	48951 RPM/s	40229 RPM/s
DECELERATION RATE	18128 RPM/s	47656 RPM/s	37888 RPM/s
MINIMUM SPEED	2000 RPM	1500 RPM	5000 RPM
TORQUE RIPPLE @16100 RPM	32.79%	5.5%	114%
SPEED RIPPLE @16100 RPM	0.068%	0.036%	0.168%
EFFICIENCY			
@17000 RPM	85.2%	83.1%	45.5%
@16100 RPM	85%	84.3%	52.6%
@8000 RPM	61.94%	79.3%	40.6 %

Table 6.3: Summary of the performances of trapezoidal and FOC sensorless algorithms. The performances of this latter case are reported also in a post-phase failure conditions. The data have been collected with simulations parameter of "Case N^o 2".

Chapter 7
Summary and Conclusions

This thesis describes the design of a fault tolerant control for brushless DC motors and the results obtained with the simulation tests. Four different designes have been proposed as possible solutions, and tested in Matlab/Simulink®: trapezoidal and FOC controllers, both with and without encoders for the determination of speed and position of the rotor.

From the results obtained, as well as those proposed in the literature, the FOC algorithm has undoubted higher performances, but requires a more difficult implementation, due to its complex operating principle. The trapezoidal control, instead, is easy and low cost but it has lower efficiency, a narrower speed range and wider transient periods. Unfortunately, the advantages and drawbacks are not so clear when extending these algorithms to the sensorless versions. In fact, even if the FOC algorithm is still the one who has higher performances, the sensorless trapezoidal control seems to lose its simplicity, obtaining a design that may be less reliable than the one which uses hall sensors. This is because the zero-crossing BEMF technique, used to determine the instants of commutation in absence of hall sensors, requires a specific design of low pass filters to compensate the high noise that affects the voltage measurement. The filtering process limits not only the minimum speed that the system can tolerate, it introduces also a delay in the determination of speed and position, that is an effect much more degrading in the six-step control. As a matter of fact, the error introduced by the speed estimation is so high, above all with high deceleration rates, that needs a further filter to smooth big jumps in the speed demand. All this drawbacks have very bad influence in the reliability of the system out of the on-design conditions. All this problems, together with worse performances above mentioned, make wonder if the sensorless version is preferable to a six-step algorithm which uses hall sensors. This is why in the second part of the project, the results obtained with the simulation tests refer only to the fault-tolerant FOC model. It seems that this latter algorithm will be the most successful solution to control EMAs, equipped with just voltage and current sensors.

The fault-tolerant driver is able to keep on running the rotor but accepting a degraded efficiency. The performances, in terms of transient periods, is just slightly affected by the failure. While the torque and current ripples deeply increase in amplitude. Think about possible applications of this routine on a real plant, no strict requirements can be done for the maximum torque and speed fluctuations in post-failure conditions, but the motor can avoid unsafe blocks of the actuator. This could be crucial for the operability of the aircraft and goes in favour of a more reliable MEA concept.

All the tests have been runned in Matlab/Simulink®, thus not usable for statistical purposes. Consequently, as regard future works, the algorithm must be tested and improved employing data from a test bench, in order to validate or confute the results.

Acronyms List

BEMF Back Electromotive Force

BLDC Brushless DC

EMA Electro-Mechanical Actuator

FOC Field Oriented Control

MEA More Electric Aircraft

mmf Magnetomotive Force

PM Permanent Magnet

PMSM Permanent Magnet Synchronous Machine

PWM Pulse Width Modulation

RPM Rounds Per Minute

SVM Space Vector Modulation

Y Star Pattern

Z-N Ziegler-Nichols

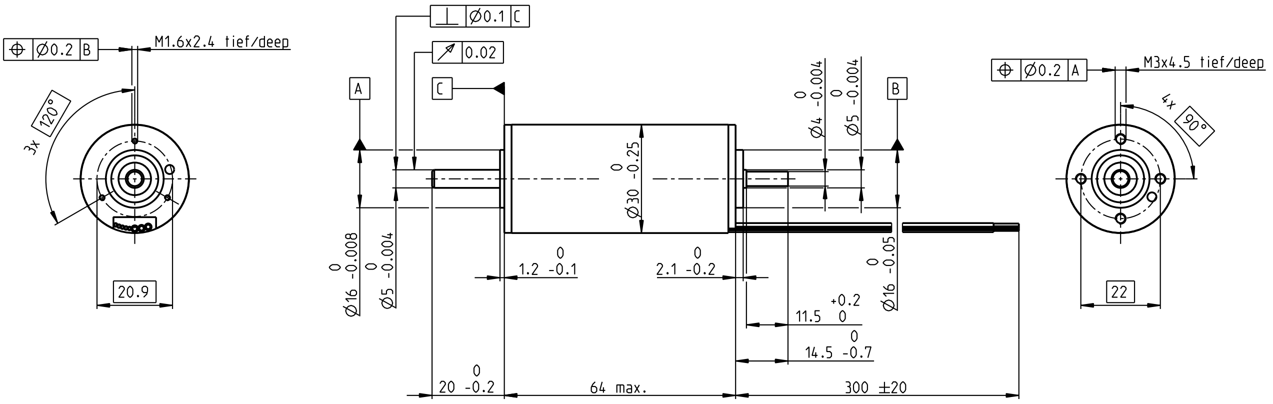
Appendix

Appendix A

EC-4pole 30 Maxon Motor Datasheet

EC-4pole 30 Ø30 mm, brushless, 200 Watt

High Power



M 1:2

🟥 Stock program
□ Standard program
▨ Special program (on request)

Part Numbers

	Motor Data		305013	305014	305015
	Values at nominal voltage				
1	Nominal voltage	V	24	36	48
2	No load speed	rpm	16700	16700	16500
3	No load current	mA	728	485	356
4	Nominal speed	rpm	16100	16200	16000
5	Nominal torque (max. continuous torque)	mNm	94.6	94.2	92.9
6	Nominal current (max. continuous current)	A	7.58	5.03	3.68
7	Stall torque	mNm	3220	3510	3430
8	Stall current	A	236	171	124
9	Max. efficiency	%	89	90	90
	Characteristics				
10	Terminal resistance phase to phase	Ω	0.102	0.21	0.386
11	Terminal inductance phase to phase	mH	0.016	0.037	0.065
12	Torque constant	mNm/A	13.6	20.5	27.6
13	Speed constant	rpm/V	700	466	346
14	Speed/torque gradient	rpm/mNm	5.21	4.78	4.83
15	Mechanical time constant	ms	1.82	1.67	1.69
16	Rotor inertia	gcm^2	33.3	33.3	33.3

Specifications

	Thermal data	
17	Thermal resistance housing-ambient	7.4 K/W
18	Thermal resistance winding-housing	0.21 K/W
19	Thermal time constant winding	2.11 s
20	Thermal time constant motor	1180 s
21	Ambient temperature	-20…+100°C
22	Max. winding temperature	+155°C
	Mechanical data (preloaded ball bearings)	
23	Max. speed	25000 rpm
24	Axial play at axial load < 8.0 N	0 mm
	> 8.0 N	0.14 mm
25	Radial play	preloaded
26	Max. axial load (dynamic)	5.5 N
27	Max. force for press fits (static)	73 N
	(static, shaft supported)	1300 N
28	Max. radial load, 5 mm from flange	25 N
	Other specifications	
29	Number of pole pairs	2
30	Number of phases	3
31	Weight of motor	300 g

Values listed in the table are nominal.

Connection motor (Cable AWG 18)

black	Motor winding 2
white	Motor winding 3
red	Motor winding 1

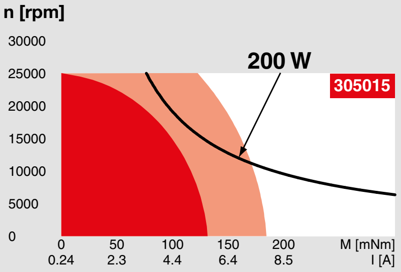
Connection sensors (Cable AWG 26)

black/grey	Hall sensor 2
blue	GND
green	V_{Hall} 3…24 VDC
red/grey	Hall sensor 1
white/grey	Hall sensor 3

Wiring diagram for Hall sensors see p. 41

Operating Range

Comments



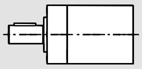
🟥 Continuous operation
▨ Continuous operation with reduced thermal resistance R_{th2} 50%
□ Intermittent operation

— Assigned power rating

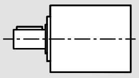
maxon Modular System

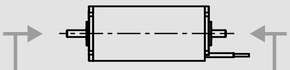
Overview on page 28–36

Planetary Gearhead
Ø32 mm
4 - 8 Nm
Page 341



Planetary Gearhead
Ø42 mm
3 - 15 Nm
Page 346





Recommended Electronics:	
Notes	**Page 32**
ESCON Mod. 50/5	427
ESCON Mod. 50/4 EC-S	427
ESCON 50/5	428
ESCON 70/10	428
DEC Module 50/5	430
EPOS2 50/5	435
EPOS2 70/10	435
EPOS4 Module/CB 50/5	442
EPOS4 Module 50/8	443
EPOS4 Comp. 50/8 CAN	443
MAXPOS 50/5	447

Encoder 16 EASY
128 - 1024 CPT, 3 channels
Page 395

Encoder 16 EASY Abs.
4096 steps, Single Turn
Page 396

Encoder 16 RIO
512 - 65536 CPT, 3 channels
Page 408

Encoder 2RMHF
3000 - 5000 CPT, 3 channels
Page 410

Encoder AEDL 5810
1024 - 5000 CPT, 3 channels
Page 412

Encoder HEDL 5540
500 CPT, 3 channels
Page 417

Brake AB 20
24 VDC, 0.1 Nm
Page 456

List of Figures

List of Tables

Bibliography

[1] M. M. P. Association et al. *MMPA Standard No. 0100-00, Standard Specifications for Permanent Magnet Materials.* Available www. intl-magnetics. org.

[2] C. Controls. *What is Field Oriented Control and what good is it?* Copley Controls Corp.

[3] M. Inc. *Park, Inverse Park and Clarke, Inverse Clarke Transformations MSS Software Implementation User Guide.* Microsemi.

[4] B. Kimco. *What is a brushless DC motor.* URL http://www.beikimco.com/resources-downloads/about-bldc-motors/what-is-a-brushless-DC-motor.

[5] E. Magnetics. *NdFeB Magnets / Neodymium Iron Boron Magnets Datasheet.* Available www. eclipsemagnetics .com.

[6] TMotor. *TMotor Drone UAV Brushless Motor U5 KV400.*

[7] T. Wilson and P. Trickey. *DC machine with solid-state commutation.* Electrical Engineering, *volume 81*(11):879–884 [1962].

[8] K. Iizuka, H. Uzuhashi, M. Kano, T. Endo and K. Mohri. *Microcomputer control for sensorless brushless motor.* IEEE Transactions on Industry Applications, (3):595–601 [1985].

[9] T. Miller. *Permanent Magnet and Reluctance Motor Drives*, (Oxford, UK: Oxford Science Publications,1989).

[10] IEEE. *Four-quadrant sensorless brushless ECM drive.*

[11] J. C. Moreira. *Indirect sensing for rotor flux position of permanent magnet AC motors operating over a wide speed range.* IEEE Transactions on Industry Applications, *volume 32*(6):1394–1401 [1996].

[12] K. Rajashekara, A. Kawamura and K. Matsuse. *Sensorless control of AC motor drives: speed and position sensorless operation*, (IEEE press New York,1996).

[13] IEEE. *Adaptive fault tolerance and graceful degradation under dynamic hard real-time scheduling.*

[14] J. Haylock, B. Mecrow, A. Jack and D. Atkinson. *Operation of a fault tolerant PM drive for an aerospace fuel pump application.* IEE Proceedings-Electric Power Applications, *volume 145*(5):441–448 [1998].

[15] M. E. S. Graham C. Goodwin, Stefan F. Graebe. *Control System Design*, pages 157–173, figure 6.1 and 6.6, (Prentice Hall,2000).

[16] M. Barr. *Introduction to Pulse Width Modulation (PWM)* [2001]. URL https://barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation.

[17] F. Wang. *Sine-triangle versus space-vector modulation for three-level PWM voltage-source inverters.* IEEE transactions on industry applications, *volume 38*(2):500–506 [2002].

[18] B. C. Mecrow, A. G. Jack, D. J. Atkinson, S. R. Green, G. J. Atkinson, A. King and B. Green. *Design and testing of a four-phase fault-tolerant permanent-magnet machine for an engine fuel pump.* IEEE Transactions on Energy Conversion, *volume 19*(4):671–678 [2004].

[19] L. Parsa and H. A. Toliyat. *Fault-tolerant five-phase permanent magnet motor drives.* In *Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE*, volume 2, pages 1048–1054, (IEEE2004).

[20] IEEE. *State of the art and future trends in position sensorless brushless DC motor/generator drives.*

[21] L. Parsa and H. A. Toliyat. *Five-phase permanent-magnet motor drives.* IEEE Transactions on Industry applications, *volume 41*(1):30–37 [2005].

[22] P. P. Acarnley and J. F. Watson. *Review of position-sensorless operation of brushless permanent-magnet machines.* IEEE Transactions on Industrial Electronics, *volume 53*(2):352–362 [2006].

[23] IEEE. *Fault diagnosis and condition monitoring of electrical machines-A Review.*

[24] IEEE. *Speed and position estimation for pm synchronous motor using self-compensated back-emf observers.*

[25] R. Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*, (Springer Science & Business Media2006).

[26] E. Levi, M. Jones, S. N. Vukosavic, A. Iqbal and H. A. Toliyat. *Modeling, control, and experimental investigation of a five-phase series-connected two-motor drive with single inverter supply.* IEEE Transactions on industrial electronics, *volume 54*(3):1504–1516 [2007].

[27] W. Rippel. *Induction Versus DC Brushless Motors* [2007]. URL https://www.tesla.com/blog/induction-versus-dc-brushless-motors.

[28] J. Zambada. *Sensorless field oriented control of PMSM motors.* Microchip AN1078, pages 1–30 [2007].

[29] N. Bianchi, S. Bolognani and M. D. P. Pre. *Impact of stator winding of a five-phase permanent-magnet motor on postfault operations.* IEEE Transactions on Industrial Electronics, *volume 55*(5):1978–1987 [2008].

[30] P. K.Gopakumar. *Lecture - 29 Space Vector PWM Signal Generation* [2008]. URL https://www.youtube.com/watch?v=fCQe5rSpa3o, CentreforElectronicsDesignandTechnology,IIScBangalore.

[31] N. P. Quang, Dittrich et al. *Vector control of three-phase AC machines*, page 24 table 2.3, (Springer,2008).

[32] B. Singh and S. Singh. *State of the art on permanent magnet brushless DC motor drives.* journal of power electronics, *volume 9*(1):1–17 [2009].

[33] IEEE. *Sensorless speed control of BLDC motor using six step square wave and rotor position detection.*

[34] TDK. *Micronas showcases system solution for brushless DC motor control applications* [2010]. URL https://www.micronas.com/en/news-events/trade-news/micronas-showcases-system-solution-brushless-dc-motor-control-applications-0.

[35] R. Isermann. *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*, (Springer Science & Business Media2011).

[36] F. V. Mario. *Applicazione della trasformata Wavelet Continua all'Individuazione di Guasti e alla Diagnosi dei Motori Brushless DC.* Ph.D. thesis [2012].

[37] A. Hughes and W. Drury. *Electric motors and drives: fundamentals, types and applications*, pages 87, 95, (Newnes,2013).

[38] J. Specht. *Terminology Proposal: Redundancy for Fault Tolerance.* University Lecture [2013].

[39] Š. Kozák. *State-of-the-art in control engineering.* Journal of Electrical Systems and Information Technology, *volume 1*(1):1–9 [2014].

[40] D. P. P. Yadu Kiran. *A Review of Brushless Motor Control Techniques.* Electrical Engineering, *volume 3*(8):10 963–10 971 [2014].

[41] B. Pera. *4-Pole brushless DC Motor* [2015]. URL https://www.researchgate.net/post/How_can_I_find_structure_and_parameter_of_Brushless_DC_Motor2.

[42] C. Puttaswamy. *Analysis, design and control of permanent magnet brushless motors.* Ph.D. thesis [2015].

[43] IEEE. *Trapezoidal control of Brushless DC motor based on DSP F28335.*

[44] ST. *STM32 32-bit ARM Cortex MCUs* [2017]. URL http://www.st.com/en/microcontrollers/stm32-32-bit-arm-cortex-mcus.html.

[45] M. Wallace. *How to Design an RC Snubber By Mike Wallace* [2017]. URL https://sciencing.com/design-rc-snubber-12030206.html.

[46] Mathworks. *PID Tuner* [Matlab 2017]. URL https://it.mathworks.com/help/slcontrol/gs/automated-tuning-of-simulink-pid-controller-block.html#d119e1367.

[47] Mathworks. *slTuner* [Matlab 2017]. URL https://it.mathworks.com/help/slcontrol/ug/sltuner.html?searchHighlight=slTuner&s_tid=doc_srchtitle.

[48] Mathworks. *Tuning Multiloop Control Systems* [Matlab 2017]. URL https://it.mathworks.com/help/control/examples/tuning-multi-loop-control-systems.html#d119e14345.