POLITECNICO DI TORINO

Master's degree in Mechatronics engineering

Modelling and design of control strategies for Permanent Magnet Synchronous Motors (PMSMs)

Supervisors:

Boglietti Aldo
Iqbal Hussain

Candidate:

Bertelli Alessandro

ACADEMIC YEAR 2019/2020

Index

1 Chapter I – Introduction 1

1.1 PMSM features and applications 1

1.2 PMSM structure 3

2 Chapter II – Theoretical basic concepts 5

2.1 Rotating field generation 5

2.2 Space vector representation 7

2.3 Clark and Park transformations 9

2.4 PMSM dynamical model 11

2.5 PWM: pulse with modulation 13

2.6 SVPWM: Space Vector PWM 21

2.7 A brief comparison between FOC and DTC 27

2.8 FOC control scheme and description 28

2.9 Feedforward compensation 29

2.10 Flux weakening control 31

2.11 Anti-windup PID controller 34

3 Chapter III: Simulink model and PI design 36

3.1 PMSM Simulink model 36

3.2 SVPWM and Inverter Design 40

3.3 Current and Speed control loops 44

3.4 PI parameters design 47

3.5 H infinity controller design 55

3.6 Flux weakening control implementation 74
4 Conclusions 83
5 Appendix 84
5.1 SVPWM: “Switch_Signal_Generator” MATLAB function 84
5.2 Hinfinity controller MATLAB code 87
6 Bibliography 91

List of Figures

FIGURE 1- PMSM 3-D STRUCTURE 3
FIGURE 2 – PMSM MAGNET CONFIGURATIONS: (A) SURFACE-MOUNTED, (B) SURFACE-INSET, (C) INTERIOR-RADIAL, AND (D) INTERIOR-CIRCUMFERENTIAL 4
FIGURE 3 – SPACE VECTOR REPRESENTATION 7
FIGURE 4 - SPACE VECTOR REPRESENTATION AT TIME T=0 (A) AND AT TIME T=$\pi 2\omega e$ 8
FIGURE 5 - ABC AND AB PLANE 9
FIGURE 6 - FIGURE 7 - AB AND DQ PLANE 10
FIGURE 8 – INVERTER CONNECTED TO A BALANCED LOAD 14
FIGURE 9 - PWM EXAMPLE 14
FIGURE 10 - MODULATION REGIONS 3-PHASE SINUSOIDAL PWM 16
FIGURE 11- FOURIER SPECTRUM OF THE PHASE VOLTAGES WITH VOLTAGES $V* = 200V$ $fo = 50$ Hz $fs = 2kHz$ 16
FIGURE 12 - CURRENT AND VOLTAGE REPRESENTATION FOR PHASE A 17
FIGURE 13 - SIX STEP OPERATION FROM TOP TO BOTTOM: LINE TO MIDPOINT VOLTAGE, PHASE VOLTAGE AND FUNDAMENTAL COMPONENT, NEUTRAL POINT VOLTAGE 18
FIGURE 14 - $vAO(t)$ AND $va(t)$ FOURIER SPECTRUM IN THE OVERMODULATION REGION 18
FIGURE 15 – COMMON MODE COMPONENT AND $va * t$, $vb * t$, $vc * t$ NORMALIZED VALUES OBTAINED MULTIPLYING FOR $VtrVdc$ 19
FIGURE 16 – REFERENCE SIGNALS AND OBJECTIVE VOLTAGES WITH $V* = Vdc2 = 200\,V$ 20
FIGURE 17 – MODULATION REGIONS WITH ZERO-SEQUENCE INJECTION 20
FIGURE 18 - REFERENCE SIGNALS AND OBJECTIVE VOLTAGES WITH $V* = Vdc3 = 230\,V$ 21
FIGURE 19 - INVERTER SCHEME SVPWM 22
FIGURE 20 – POSSIBLE SWITCHING CONFIGURATIONS (3-PHASE INVERTER) 23

FIGURE 21 - SPACE VECTOR REPRESENTATION OF THE SWITCH STATES 24
FIGURE 22 – TIME SLICES SEQUENCE 25
FIGURE 23 - FOC SCHEME 29
FIGURE 24 - FOC WITH FEEDFORWARD COMPENSATION 30
FIGURE 25 – CHARACTERISTIC CURVE OF A PMSM 31
FIGURE 26 - LIMITING CIRCLES OF BOTH VOLTAGE AND CURRENT 33
FIGURE 27 - ANTI-WINDUP BACK CALCULATION SCHEME 35
FIGURE 28 - PERFORMANCE COMPARISON OF PID CONTROLLER WITH AND WITHOUT ANTI-WINDUP FEATURE 35
FIGURE 29 – PMSM SUBSYSTEM 37
FIGURE 30 – "ABC-DQ TRANSFORMATION" SUBSYSTEM 38
FIGURE 31 – "ELECTRICAL EQUATIONS" SUBSYSTEM 38
FIGURE 32 – "MECHANICAL EQUATIONS" SUBSYSTEM 39
FIGURE 33 – "DQ-ABC TRANSFORMATION" SUBSYSTEM 39
FIGURE 34 - SVPWM IMPLEMENTATION 40
FIGURE 35 - INVERTER SETTINGS, ON THE LEFT THE MOST ACCURATE ONE, ON THE RIGHT THE ONE USED IN THE MODEL DESIGNED 41
FIGURE 36 – FILTER MATLAB CODE 42
FIGURE 37 – SIMULINK MODEL OF SVPWM, INVERTER AND PMSM 43
FIGURE 38 - PI CASCADE CONTROLLER SIMULINK IMPLEMENTATION 45
FIGURE 39 –CURRENT LOOP + FEEDFORWARD CONTROL SUBSYSTEM CONTENT 46
FIGURE 40 - NORM AND TETA COMPUTATION SUBSYSTEM 46
FIGURE 41 – SIMPLIFIED MODEL FOR PI PARAMETER COMPUTATION 47
FIGURE 42 - SPEED STEP RESPONSE OF BOTH SIMPLIFIED AND COMPLETE MODEL WITH WREF=3000RPM AND TLOAD=0 NM (1ST ATTEMPT) 51
FIGURE 43 - CURRENT SREP RESPONSE BEFORE AND AFTER CHANGING PARAMETERS 52
FIGURE 44 – SPEED STEP RESPONSE OF THE COMPLETE MODEL WITH 2ND ATTEMPT PARAMETERS 52
FIGURE 45- TORQUE OUTPUT TE OF THE COMPLETE MODEL WITH 3RD ATTEMPT PARAMETERS 54
FIGURE 46 – SPEED OUTPUT OF THE COMPLETE MODEL WITH 3RD ATTEMPT PARAMETERS 54
FIGURE 47 - SPEED RIPPLE 55
FIGURE 48 - H_INFINITY CONTROL SCHEME 56
FIGURE 49 - PMSM CONTROL SCHEME FOR H_INFINITY CONTROL DESIGN 58
FIGURE 50 - REQUIREMENTS ON THE RESONANCE PEACK OF T AND S 60
FIGURE 51 – W_S_INV BODE DIAGRAM 61
FIGURE 52 – W_T_INV BODE DIAGRAM 62
FIGURE 53 - SIMULINK MODEL USED FOR HINFINITY DESIGN 63

FIGURE 54 - S AND WS_INV BODE DIAGRAM COMPARISON 64
FIGURE 55 – IDEAL STEP RESPONSE (3000RPM) WITH NOMINAL STEP TORQUE (1.27N) INTRODUCED AT 0.05 S 65
FIGURE 56 - T AND WT_INV BODE DIAGRAM COMPARISON 65
FIGURE 57 - SPEED RESPONSE (3000 RPM STEP REFERENCE) COMPARISON FOR EACH DESCRITIZED CONTROLLER OBTAINED 67
FIGURE 58 - SPEED RESPONSE (300 RPM STEP REFERENCE) COMPARISON FOR EACH DESCRITIZED CONTROLLER OBTAINED ..... **ERRORE. IL SEGNALIBRO NON È DEFINITO.**
FIGURE 59 - RMS VALUE COMPARISON (300 RPM) 68
FIGURE 60 - RMS VALUE COMPARISON (300 RPM) 69
FIGURE 61 – SPEED RESPONSE COMPARISON BETWEEN HINFINITY BASED MODEL AND PI BASED MODEL WITH 3000 RPM STEP REFERENCE 70
FIGURE 62 – SPEED RESPONSE COMPARISON BETWEEN HINFINITY BASED MODEL AND PI BASED MODEL WITH 300 RPM STEP REFERENCE 70
FIGURE 63 - TORQUE RESPONSE (TE) COMPARISON BETWEEN HINFINITY BASED MODEL AND PI BASED MODEL WITH 3000 RPM STEP REFERENCE 71
FIGURE 64 - TORQUE RESPONSE (TE) COMPARISON BETWEEN HINFINITY BASED MODEL AND PI BASED MODEL WITH 300 RPM STEP REFERENCE 71
FIGURE 65 - RMS VALUE COMPARISON 73
FIGURE 66 - CURRENT VECTOR REPRESENTATION 74
FIGURE 67 - BC PARAMETER COMPUTATION SIMULINK SCHEME 75
FIGURE 68 - PI CASCADE CONTROLLER UPDATED TO OPERATE IN THE FLUX-WEAKENING AREA 76
FIGURE 69 - CURRENT LOOP + FEEDFORWARD CONTROL SUBSYSTEM 76
FIGURE 70 - CURRENT VECTOR COMPUTATION SUBSYSTEM 77
FIGURE 71 – CURRENT PI CONTROLLER WITH ANTI-WINDUP 78
FIGURE 72 - OUTPUT SPEED W_M WITH FLUX WEAKENING CONTROL AND MULTIPLE SPEED STEP REFERENCES. 78
FIGURE 73 - OUTPUT SPEED W_M WITH FLUX WEAKENING CONTROL AND MULTIPLE SPEED STEP REFERENCES. 79
FIGURE 74 - OUTPUT TORQUE TE WITH FLUX WEAKENING CONTROL AND MULTIPLE SPEED RAMP REFERENCES (LOAD TORQUE APPLIED AT T=0.12 80
FIGURE 75 - OUTPUT SPEED W_M WITH FLUX WEAKENING CONTROL AND MULTIPLE SPEED RAMP REFERENCES (LOAD TORQUE APPLIED AT T=0.12 80
FIGURE 77 - OUTPUT SPEED WITH FLUX WEAKENING CONTROL AND 300 RPM SPEED STEP REFERENCES (T_L=1.27 NM) 81
FIGURE 76 - OUTPUT SPEED WITH FLUX WEAKENING CONTROL AND 3500 RPM SPEED STEP REFERENCES (T_L=0.635 NM) 81

List of Tables

TABLE 1 - PHASE VOLTAGES AND SPACE VECTORS ON THE BASE OF THE SWITCHING CONFIGURATION.....23
TABLE 2 – EQUATIONS TO COMPUTE THE TIME SLICES RELATED TO EACH SECTOR.....26
TABLE 3 – EMJ-04APB22 PARAMETERS 47

Aknowledgements

I would like to thank the entire staff of the FREEDM System Center that has welcomed me during my first days in Raleigh, in particular Hulgize Kassa for the help he gave me, that goes further his duties, and Rebecca McLennan, guiding me through the bureaucratic aspects related to the Erasmus+ international experience. Unfortunately, the COVID-19 global pandemic has prevented me to exploit the great resources offered by the lab, which I could attend only for a week thanks to the effort of the staff. They have tried to make my internship the best possible, regardless the hard times we are all going through. In this regard, I appreciate the time spent by both my NCSU Supervisor, Iqbal Hussain, and by the Research Assistant Sodiq Agoro, to answer the questions and doubts emerged during the development of this thesis. Furthermore, I would like to thank my POLITO Supervisor, Aldo Boglietti, that have given me suggestions and connected me to his colleague,s in order to deeper understand one of the topic related to this work.

Finally, my sincerest thanks goes to my friends, that have made my academic path a period of my life that I will remember forever, and to my family, for their constant support and for believing in my capabilities.

There are no words to describe how I feel lucky to have such great people in my life.

1 Chapter I – Introduction

The aim of this thesis is to provide a wide view of the Permanent Magnet Synchronous Motor (PMSM), from its basic principles to its applications, pointing the interest on the *Field Oriented Control (FOC)* scheme and proposing different digital control techniques. A Simulink model that takes into account the computing limitation of a real control device and of its hardware components, such as the 3-phase inverter, is designed and used in order to evaluate the dynamic response of the controllers developed. In order to have an example in terms of hardware parameters and limitations, it has been chosen to design the models taking as reference the *TMDSHVMTRPFCKIT* developer kit and the *EMJ-04APB22 ESTUN* Permanent Magnet Synchronous Motor (PMSM) . In the first chapter the PMSM structure and its application are introduced. In the second one, it has been given a description of the theoretical concepts necessary to understand the mathematical equations behind the PMSM. As well, all the actors involved in the motor control are introduced. Furthermore, different control techniques are presented. In the third chapter, it is shown the Simulink implementation of the models developed and the design procedure followed to get each controller used. Finally in the conclusions chapter, the results obtained are briefly discussed, summarizing the advantages and disadvantages of the different techniques already explained in the previous chapters.

1.1 PMSM features and applications

Electrical motors are widely used nowadays in lots of different fields, from household appliance to industrial applications. Since different types of electrical motors exist, the right choice depends on the application usage end on the budget available. It is possible to divide the electrical motors in two big families: the DC motors and the AC motors.

With respect to DC motors, the AC motors don’t need any use of commutating parts (brushes and slip rings) eliminating the related drawbacks such as maintenance, power losses, and arcing, generating heat and electromagnetic interference (EMI). DC motor were at the beginning more suitable for those application where motor speed needed

to be finely controlled, but the progress made in the control algorithm of AC motors united with the advantage of a lower inertia, greater robustness and power efficiency, is now making the use of AC motors much more popular.

AC motors can be further divided in asynchronous motors (also called *Induction Motors*, IM) and synchronous motors. IM are characterized by a stator alternated current that induces another current in the rotor winding, generating an interaction between the 2 magnetic field produced. The two magnetic fields run at slightly different frequency generating torque. Instead, in synchronous motors, the magnetic field of the rotor is either due to a permanent magnet or to an excitation coil and has constant direction in a reference frame fixed to the rotor. The rotor at steady state condition moves with a frequency synchronized with the one of the stator magnetic field [1].

The PMSM is a synchronous motor where the rotor flux is produced by a permanent magnet. This have several advantages with respect to the IM that, together with the decrease of the cost related to permanent magnets, have made this kind of motors the most popular nowadays.

PMSM are characterized by higher torque/inertia ratio and better dynamic response due to the presence of the permanent magnet instead of a winding at the rotor side that increases the flux density at the air gap. They are also more efficient and easier to cool down since the losses related to the rotor current are eliminated. Furthermore, their power/weight ratio is higher if compared to induction motors and the control strategy is easier to implement since there is no necessity of slip speed calculation. However, disadvantages of these motors are the variation of their property during time and at different temperature and the need of high resolution position sensors, power inverter and a microprocessor [2]. Another limitation is the excessive cost for high power applications, that is why their usage is usually limited to a few kilowatts.

All the previous feature highlighted makes these motors suitable for lots of applications, from high performance ones, such as robotics and aerospace actuators, to electric vehicles, spindle motion operations, wind generation and motion control

application related to lots of different fields such as papermaking, packaging, textiles, ceramic, glass and woodworking [3].

1.2 PMSM structure

The PMSM is made by two main components, the rotor and the stator. A 3-phase winding is sinusoidally distributed around the stator in order to get an equivalent number of coils that changes on the base of the angle ϑ.

In a real machine it is impossible to achieve a perfect sinusoidal distribution due to the real dimension of coils and slots of the stator thus, only the first harmonic is considered to make a mathematical model:

$$N_{eq_a} = N_{eq} \sin(\vartheta) \tag{1.1}$$

N_{eq} can be computed considering the actual number of coils and a coefficient K_d that depends on the number of slots and of poles P of the stator. This means that:

$$N_{eq} = K_d \frac{2}{\pi} N_1 \tag{1.2}$$

where N_1 is the actual number of coils of each phase winding.

Due to the almost sinusoidal distribution, the stator is able to generate a rotating field when its windings are fed with 3-phase current.

For what concerns the rotor, on the bases of the placement of the permanent magnets, different configuration exists. The *surface-mounted topology* is characterized by magnets epoxy-glued or wedge-fixed to the rotor. Manufacture is really simple but the drawback is

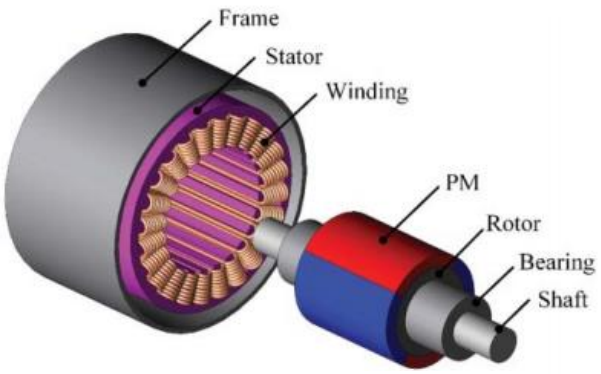

Figure 1- PMSM 3-D structure

that the mechanical strength of the rotor is limited by the one of the epoxy-glue, which can compromise high speed operations. The advantage is that, since the permeability of the permanent magnet is approximatively equal to the one of air, L_q and L_d inductances can be considered almost equal. For the *surface-inset configuration* the permanent magnets are placed in rotor slots making them more secured and producing a difference between L_q and L_d that generates what is called reluctance torque. The third type is *the interior magnet topology (salient-pole PMSMs)* that can be divided in *interior-radial* and *interior-circumferential*. In both, linear permanent magnets are buried inside the rotor improving the mechanical strength since the magnets can't fly away during rotation. The first one is characterized by permanent magnets radially magnetized while in the second they are circumferentially magnetized. The advantage of the second option is that, due to the magnet distribution, the air-gap flux density is greater than the remanent magnetization of the permanent magnet itself (flux-focusing or flux-concentration). The drawback is that, to reduce the flux leakage at the inner end of the magnets, a nonmagnetic shaft or collar must be used, compromising the torsional stiffness of the shaft. The different options are shown in Figure 2 while Figure 1shows the 3-D structure of the whole machine. [4]

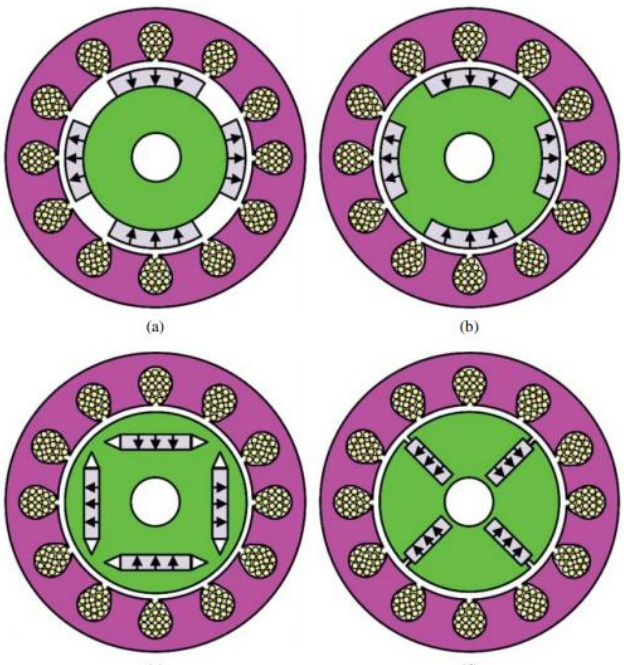

Figure 2 – PMSM magnet configurations: (a) surface-mounted, (b) surface-inset, (c) interior-radial, and (d) interior-circumferential

2 Chapter II – Theoretical basic concepts

Before introducing the design of the model and the possible control solutions it is necessary to understand which are the basic concepts behind the PMSM control.

In the following chapter the essential background is described in detail in order to give the reader all the instruments to understand the model design and control implementation.

2.1 Rotating field generation

Considering the sinusoidal winding distribution described in (2.53) for 3 windings shifted by 120°:

$$N_{eq_a} = N_{eq} \sin(\vartheta) \tag{2.1}$$

$$N_{eq_b} = N_{eq} \sin\left(\vartheta - \frac{2\pi}{3}\right) \tag{2.2}$$

$$N_{eq_c} = N_{eq} \sin\left(\vartheta + \frac{2\pi}{3}\right) \tag{2.3}$$

If the currents flowing in each winding are:

$$I_a = I \sin(\omega_e t) \tag{2.4}$$

$$I_b = I \sin(\omega_e t - \frac{2\pi}{3}) \tag{2.5}$$

$$I_c = I \sin(\omega_e t + \frac{2\pi}{3}) \tag{2.6}$$

Remembering that:

$$\sin(x)\sin(y) = \frac{1}{2}(\cos(x-y) - \cos(x+y)) \tag{2.7}$$

And considering the MMF produced by a single winding:

$$\begin{aligned} MMF_a = N_{eq_a} I_a &= N_{eq} I \sin(\vartheta)\sin(\omega_e) \\ &= \frac{1}{2} N_{eq} I[\cos(\vartheta - \omega_e t) - \cos(\vartheta + \omega_e t)] \end{aligned} \tag{2.8}$$

It is possible to compute the total MMF:

$$MMF_{tot} = \frac{1}{2} N_{eq}\, I[\cos(\vartheta - \omega_e t) - \cos(\vartheta + \omega_e t)]$$

$$+ \frac{1}{2} N_{eq}\, I\left[\cos\left(\vartheta - \frac{2\pi}{3} - \omega_e t + \frac{2\pi}{3}\right) - \cos(\vartheta - \frac{2\pi}{3} + \omega_e t - \frac{2\pi}{3})\right]$$

$$+ \frac{1}{2} N_{eq}\, I\left[\cos\left(\vartheta + \frac{2\pi}{3} - \omega_e t - \frac{2\pi}{3}\right) - \cos\left(\vartheta + \frac{2\pi}{3} + \omega_e t + \frac{2\pi}{3}\right)\right]$$

$$MMF_{tot} = \frac{3}{2} N_{eq} I \cos(\vartheta - \omega_e t) \tag{2.9}$$

Considering l_a the air gap length it is possible to compute the magnetic field:

$$H_{tot} = \frac{MMF_{tot}}{l_a} = \frac{3}{2}\frac{N_{eq} I}{l_a}\cos(\vartheta - \omega_e t) \tag{2.10}$$

From equation (2.10)(2.53) it is possible to see that H is constant when $\vartheta - \omega_e t$ is constant, making the derivative over time = 0:

$$\omega_m = \omega_e \tag{2.11}$$

which means that the angular frequency of the H space vector is equal to the electrical angular frequency ω_e of the 3-phase current flowing in the windings.

Relation (2.11) is true when the number of poles of the stator is 2, otherwise it is possible to demonstrate that:

$$\omega_m = \frac{\omega_e}{\frac{p}{2}} \tag{2.12}$$

where p is the number of poles.

In a PMSM the number of poles of stator and rotor must be the same, the mechanical angular velocity of the rotor in steady state conditions is equal to the angular velocity of the magnetic field produced by the stator currents.

2.2 Space vector representation

Since the distribution of the stator windings is sinusoidal, it is possible to represent the phase quantities in space in the form of space vectors. Taking into account the phase shift between the windings sinusoidal distributions we can represent each phase quantity on 3 vectors shifted by 120° as in Figure 3 [5].

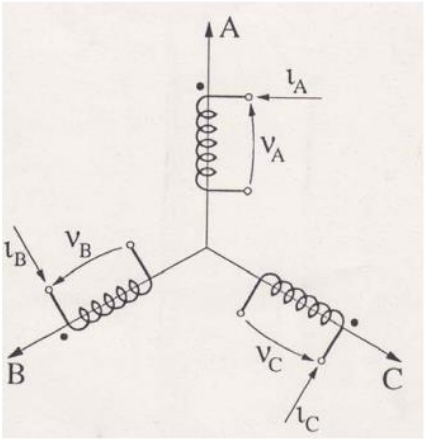

Figure 3 – Space vector representation

It is possible to consider each phase vector's module changing over time with electrical frequency ω_e. Let's consider for example (2.9), and compute the MMF at t = 0 and $\vartheta = 0$:

$$MMF_{tot,\vartheta,t} = \frac{3}{2} N_{eq} I \cos(0) = \frac{3}{2} N_{eq} I \tag{2.13}$$

Now the same result can be obtained considering (2.4), substituting t=0:

$$I_a = I \sin(0) = 0\ A \tag{2.14}$$

$$I_b = I \sin(-\frac{2\pi}{3}) = -\frac{\sqrt{3}}{2} I \tag{2.15}$$

$$I_c = I \sin\frac{2\pi}{3} = \frac{\sqrt{3}}{2} I \tag{2.16}$$

Considering the vectors $\bar{I_a} = I_a$, $\bar{I_b} = I_b\, e^{\frac{2}{3}j}$ and $\bar{I_c} = I_c\, e^{-\frac{2}{3}j}$ summing their projection along an axis shifted from A of ϑ (so coincident with A since $\vartheta = 0$):

$$I_\vartheta = I_a - \frac{\sqrt{3}}{2} I_b + \frac{\sqrt{3}}{2} I_c = 0 + \frac{3}{4} I + \frac{3}{4} I = \frac{3}{2} I \tag{2.17}$$

Since the phase quantities have been introduce, now we can consider the number of turns constant and equal to N_{eq} for each phase, thus:

$$MMF_{tot,\vartheta,\mathrm{t}} = N_{eq} I_{\vartheta} = \frac{3}{2} N_{eq} I \tag{2.18}$$

Which is the same result obtained in (2.13). In the next image they are shown the space vector resultant, abc components and the sinusoidal distribution to vary the angle ϑ.

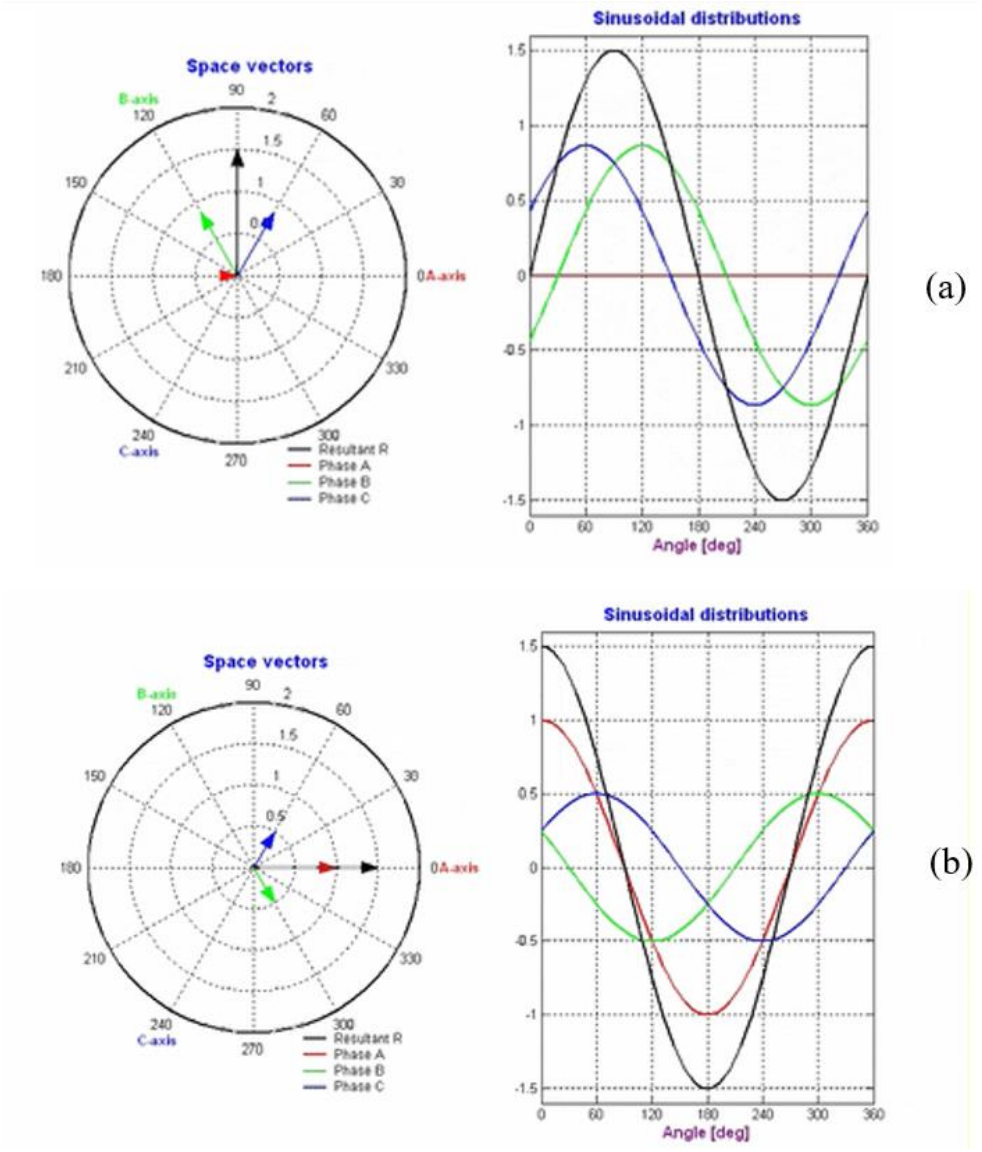

Figure 4 - Space vector representation at time t=0 (a) and at time t=$\frac{\frac{\pi}{2}}{\omega_e}$

2.3 Clark and Park transformations

In order to derive a mathematical model of the PMSM where the magnetic flux doesn't depend on time-variant inductances, it is useful to apply to the 3-phase quantities two consecutive transformations. The first one is called Clark transform and is used to change from the reference system *ABC* to the reference system *αβ0*, fixed to the stator with 3 orthogonal axes. The homopolar axis *0* is defined in order to derive a square matrix (invertible) but has not physical meaning and usually it is not represented (Figure 5).

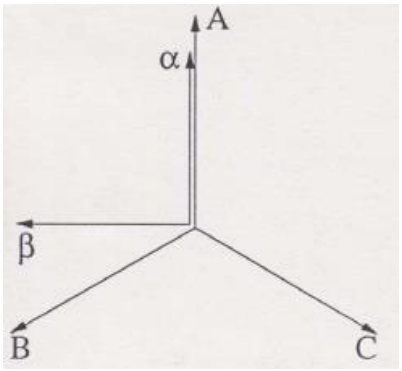

Figure 5 - abc and αβ plane

From simple trigonometric considerations it is possible to derive the following transformation matrix [6]:

$$T_{clark} = 2/3 \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.19}$$

Where, considering a general 3-phase vector and its counterpart in the αβ0 plane:

$$\mathrm{f_{abc}} = \begin{bmatrix} \mathrm{f_a} \\ \mathrm{f_b} \\ \mathrm{f_c} \end{bmatrix} \quad \mathrm{f_{\alpha\beta 0}} = \begin{bmatrix} \mathrm{f_\alpha} \\ \mathrm{f_\beta} \\ \mathrm{f_0} \end{bmatrix} \tag{2.20}$$

the αβ0 vector can be computed in the following way:

$$f_{\alpha\beta 0} = T_{clark}\, f_{abc} \tag{2.21}$$

The 2/3 gain in (2.19) is added to the matrix in order to keep the individual vector components magnitude the same after transformation when the abc input corresponds to a 3-phase sinusoidal signal.

The second transformation is done to change from the αβ0 reference frame fixed to the stator to the dq0 reference frame rotating at ω_e speed. Defining ϑ_e the angle between the dq0 and the αβ0 reference frame ($\vartheta_e = \omega_e t$), it is possible to derive the following relation [6]:

$$f_{dq0} = T_{park} \quad f_{\alpha\beta 0} \tag{2.22}$$

Where:

$$\mathrm{f}_{dq0} = \begin{bmatrix} \mathrm{f}_d \\ \mathrm{f}_q \\ \mathrm{f}_0 \end{bmatrix} \tag{2.23}$$

And:

$$T_{park} = \begin{bmatrix} cos\,\vartheta_e & sin\,\vartheta_e & 0 \\ -sin\,\vartheta_e & cos\,\vartheta_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

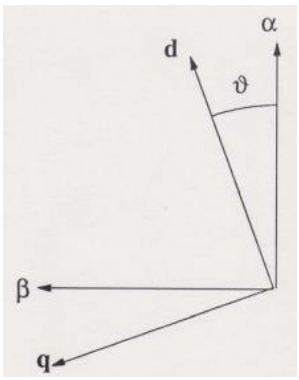

Figure 6 - Figure 7 - αβ and dq plane

When the machine has 2 poles ($\omega_e = \omega_m$) the dq0 reference frame is fixed to the rotor.

For the sake of completeness it is also shown a different version of the abc- αβ matrix (2.25) that is used for power invariant considerations.

$$T = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{2.25}$$

The coefficient at the third row of this matrix and the gain are chosen in order to make it orthonormal ($T^T = T^{-1}$). Thanks to this property the electrical power can be computed after transformation leading to the same result obtained before transformation as shown in (2.26) [7].

$$P = \begin{bmatrix} v_\alpha \\ v_\beta \\ v_0 \end{bmatrix}^T \begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = \left[T \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \right]^T T \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}^T T^T T \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}^T \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \tag{2.26}$$

From (2.19) and (2.24) it is also possible to derive the inverse Clark and Park matrixes:

$${T_{clark}}^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \tag{2.27}$$

$${T_{park}}^{-1} = \begin{bmatrix} cos\,\vartheta_e & -sin\,\vartheta_e & 0 \\ sin\,\vartheta_e & cos\,\vartheta_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

Note that T_{park} is an orthonormal matrix.

2.4 PMSM dynamical model

Through the use of the transformations seen in the previous section it is possible to derive the following equation in the d-q axes:

$$v_q = r_s i_q + \frac{d}{dt}\lambda_q + \omega_e\,\lambda_d \tag{2.29}$$

$$v_d = r_s i_d + \frac{d}{dt}\lambda_d - \omega_e\,\lambda_q \tag{2.30}$$

Where the linked fluxes in the d and q axes are:

$$\lambda_q = L_q i_q \tag{2.31}$$

$$\lambda_d = L_d i_d + \lambda_f \tag{2.32}$$

With:

$\lambda_f = constant\ linked\ rotor\ flux$

$r_s = stator\ resistence\ considered\ equal\ for\ each\ phase$

$L_d, L_q = self\ inductances\ in\ the\ d\ and\ q\ axes$

Moreover, the motor is able to produce the following electrical torque:

$$T_e = \frac{3}{2} 0.5p[\lambda_f i_q + (L_d - L_q) i_d i_q] \tag{2.33}$$

Where p is the number of couple of poles of the motor.

Note that the torque produced is the sum of 2 contributions, one that depends only on a control variable, the q-current i_q, and the second that depends on both i_d and i_q currents and that is non-null only when the two self-inductances along the d and q axes are different (which is the case of interior-magnet motors). This second contribution is usually called reluctance torque.

$\omega_e \lambda_d$ and $\omega_e \lambda_q$ in (2.29) (2.30) are the back e.m.f contribution proportional to the speed of the motor. Note that increasing the speed of the motor ω_m also ω_e increases (see (2.12)) and thus the back e.m.f. increases. This means that to generate the same torque at higher speed also the applied voltage v_q must increase in order to keep i_q constant.

For what concerns the mechanical domain, the following equation is valid:

$$T_e - T_{load} - B\omega_m = J \frac{d\omega_m}{dt} \tag{2.34}$$

Where:

$T_{load} = mechanical\ torque\ generated\ by\ the\ load$

$B = viscous\ friction\ of\ the\ motor$

$J = Inertia\ of\ the\ motor$

Equation (2.29), (2.30) and (2.34) can be rewritten in the state-space representation form:

$$\frac{di_d}{dt} = -\frac{r_s}{L_d} i_d + \frac{L_q}{L_d}\frac{P}{2}\omega_m i_q + \frac{v_d}{L_d} \tag{2.35}$$

$$\frac{di_q}{dt} = -\frac{r_s}{L_q} i_q - \frac{L_d}{L_q}\frac{P}{2}\omega_m i_d - \frac{1}{L_q}\frac{P}{2}\omega_m \lambda_f + \frac{v_q}{L_q} \tag{2.36}$$

$$\frac{d\omega_m}{dt} = \frac{1}{J}(T_e - T_{load} - B\omega_m) \tag{2.37}$$

Which in the s-domain become:

$$i_d = \frac{1}{s}\left(-\frac{r_s}{L_d} i_d + \frac{L_q}{L_d}\frac{P}{2}\omega_m i_q + \frac{v_d}{L_d}\right) \tag{2.38}$$

$$i_q = \frac{1}{s}\left(-\frac{r_s}{L_q} i_q - \frac{L_d}{L_q}\frac{P}{2}\omega_m i_d - \frac{1}{L_q}\frac{P}{2}\omega_m \lambda_f + \frac{v_q}{L_q}\right) \tag{2.39}$$

$$\omega_m = \frac{1}{sJ}(T_e - T_{load} - B\omega_m) \tag{2.40}$$

This equation will be used in Chapter 3 to derive the Simulink model of the PMSM.

2.5 PWM: pulse with modulation

In order to spin the motor a 3-phase inverter is used to convert the voltage coming from a DC generator into the 3 sinusoidal voltages V_a, V_b and V_c.

The 3-phase inverter is made by three legs, one for each output voltage. For each leg there is an upper and a lower switch that connects the output voltage respectively to the upper and to the lower potential side of the DC generator (Figure 8). The switches can be implemented in different ways by a back to back connection between a diode and a controllable device. Si-IGBT are used when the objective switching frequency f <20 kHz while Si MOSFET are used for lower power application but higher frequency. SiC MOSFET can work at even higher frequency and reduce losses.

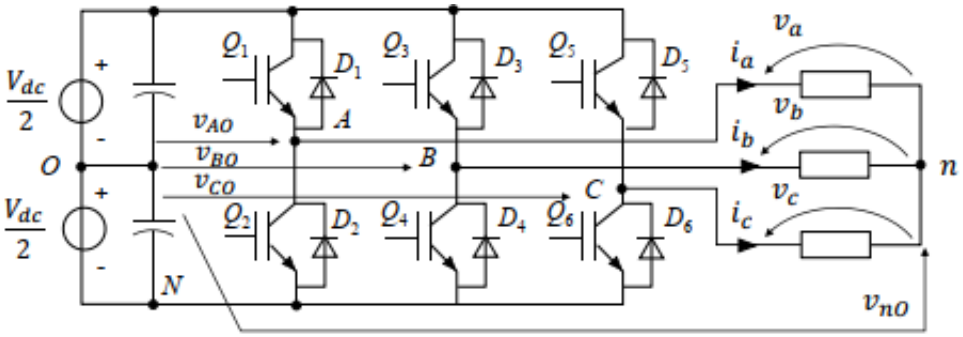

Figure 8 – Inverter connected to a balanced load

One of the possible techniques in order to control the pulses sent to the inverter gates is the PWM generation. Pulses are generated comparing the reference voltage for each leg with a triangular waveform V_{tr} called carrier. Calling q(t) the pulse signal the control law is the following: each time the reference signal $v_c(t) > V_{tr}$ q(t)=1 otherwise q(t)=0 (Figure 9).

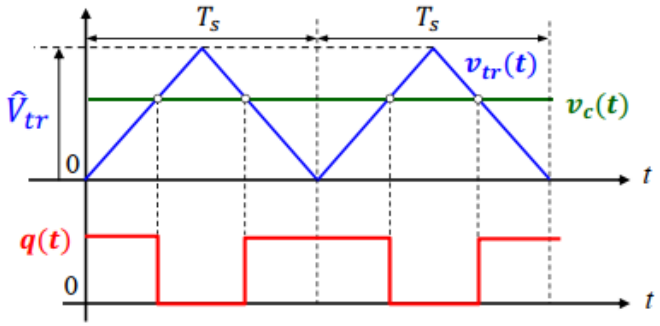

Figure 9 - PWM example

Choosing the reference signal:

$$v_c(t) = \frac{\hat{V}_{tr}}{V_{dc}} v_o^*(t) + 0.5 \qquad (2.41)$$

where $v_o^*(t)$ is the objective voltage, is possible to demonstrate [8] that, if the carrier frequency is much higher than the reference waveform frequency ($\frac{f_s}{f_o} > 20$), the mobile mean value of the voltage generated by the inverter ($\overline{v_0}$(t)) will be really close to the objective voltage.

$$\overline{v_o}(t) = \frac{1}{T_s}\int_0^{T_s} v_0(t)dt \cong v_o^*(t) \tag{2.42}$$

Where T_s is the carrier period: $T_s = \frac{1}{f_s}$

An important parameter when dealing with PWM is the amplitude modulation index:

$$m_a = \frac{\hat{V}^*}{0.5V_{DC}} \tag{2.43}$$

where $\hat{V}^*$ is the peak value of the objective voltage for each leg.

$$\overline{v_{AO}}(t) \cong v_{o,A}^*(t) = \hat{V}^* \sin(\omega t) \tag{2.44}$$

$$\overline{v_{BO}}(t) \cong v_{o,B}^*(t) = \hat{V}^* \sin(\omega t - \frac{2\pi}{3}) \tag{2.45}$$

$$\overline{v_{CO}}(t) \cong v_{o,C}^*(t) = \hat{V}^* \sin(\omega t + \frac{2\pi}{3}) \tag{2.46}$$

with $\omega = 2\pi f_o$.

The modulation is said to be linear when the peak value of the fundamental output voltages of the inverter is equal to the peak voltage of the mobile mean value and thus, also equal to the peak voltage of the objective voltage.

$$\hat{v}_{AO,1} = \hat{V}^* \qquad \hat{v}_{BO,1} = \hat{V}^* \qquad \hat{v}_{CO,1} = \hat{V}^* \tag{2.47}$$

Where the subscript "1" stands for: "fundamental component".

For sinusoidal PWM the modulation is linear when $m_a < 1$ which means $\hat{V}^* < \frac{1}{2} V_{DC}$.

The following graph can be considered where $\hat{V}_{o,1}$ is the peak voltage of the first harmonic (fundamental) of any of the 3 legs output voltages.

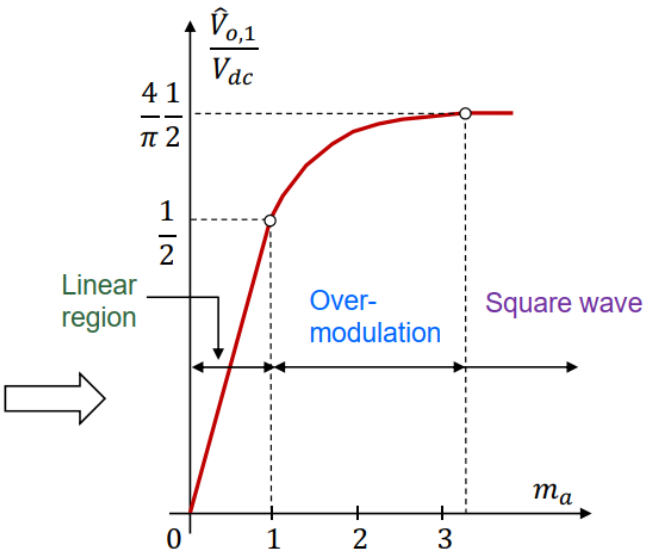

Figure 10 - Modulation regions 3-phase sinusoidal PWM

In the linear region the dominant harmonic distortion of the output voltage has the same frequency of the carrier, therefore, it can be easily eliminated by the use of a filter without affecting the amplitude of the first harmonic since $f_s >> f_o$.

The Fourier spectrum at the limit of the linear region of the phase voltage $v_a(t)$ considering $V_{DC} = 400\,V$,setting the peak of the objective voltage $\hat{V}^* = 200V$ $f_o = 50$ Hz and $f_s = 2kHz$ is the following:

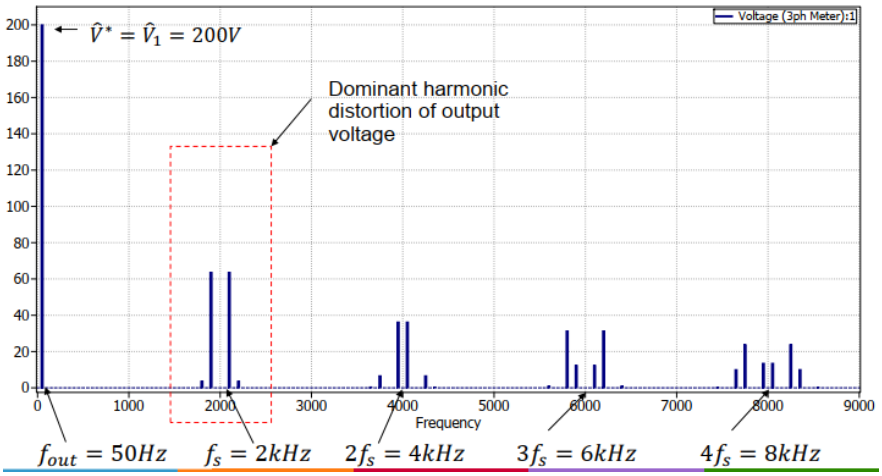

Figure 11- Fourier spectrum of the phase voltages with voltages $\hat{V}^* = 200V$ $f_o = 50\ Hz$ $f_s = 2kHz$

The phase current contains a fundamental component due to the phase voltage fundamental component and ripple due to the voltage harmonic distortion.

A representation of the phase current when not filtered is given in Figure 12 .

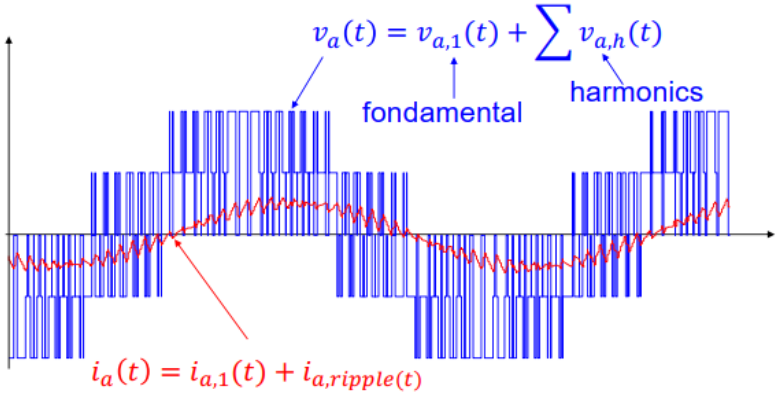

Figure 12 - Current and voltage representation for phase a

In the over-modulation region, the phase voltages will contain lower harmonics not multiple of 3:

$$f_k = (6k \pm 1)\, f_o, \quad k = 1,2\ldots \quad \left(\frac{f_k}{f_o} = 5,7,11,13\ldots.\right) \tag{2.48}$$

This happens because the harmonics multiple of 3 that appear in the line to virtual midpoint O voltages are canceled in the phase voltages thanks to the star connection of the load, since the considered load is balanced.

In the overmodulation region the phase voltage has a trapezoidal shape that becomes a square waveform when the modulation index $m_a \gg 1$.

In this case the fundamental component of the phase voltage reaches the peak limit ($\hat{V}_{phase,1} = \frac{4}{\pi}\frac{V_{DC}}{2}$) and we speak of six-step operation.

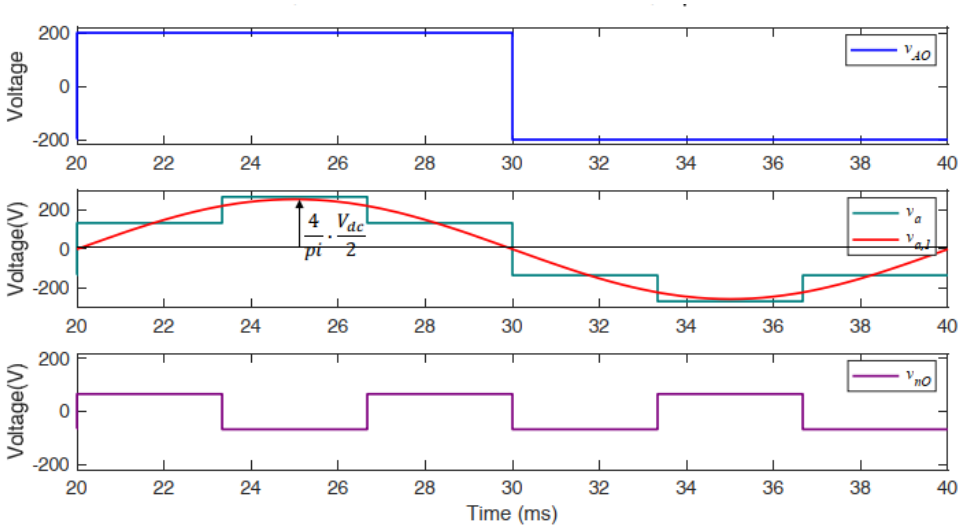

Figure 13 - Six step operation from top to bottom: line to midpoint voltage, phase voltage and fundamental component, neutral point voltage

The Fourier spectrum in Figure 14 shows the amplitude of the lower harmonics that affects $v_{AO}(t)$ and $v_a(t)$ when the inverter is working in the overmodulation region.

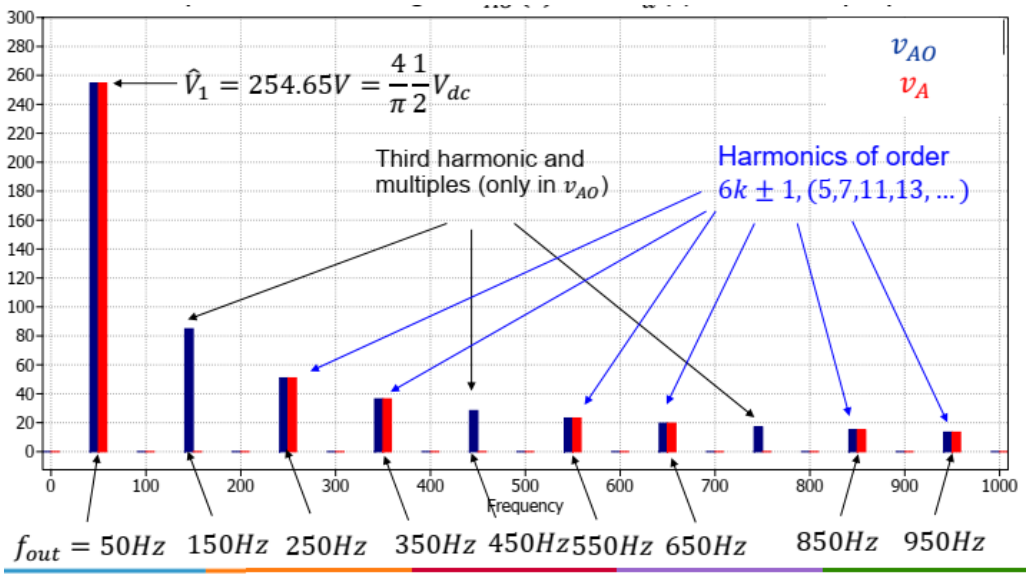

Figure 14 - $v_{AO}(t)$ and $v_a(t)$ Fourier spectrum in the overmodulation region

It is possible to extend the inverter linearity with the zero-sequence injection technique [8], the injection of a third harmonic in the reference voltages, which translates in an additional third harmonic in the mobile mean value of the common mode voltage $v_{nO}(t)$. The injection of the third harmonic has no influence on the phase voltages but reduces the peak of the reference voltages, increasing the amplitude of the fundamental

phase voltages at the limit of the linear region. The reference signal for each inverter leg will become:

$$v_{cA}(t) = \frac{\hat{V}_{tr}}{V_{dc}}(v_a^*(t) + v_{CM}) + 0.5 \tag{2.49}$$

$$v_{cB}(t) = \frac{\hat{V}_{tr}}{V_{dc}}(v_b^*(t) + v_{CM}) + 0.5 \tag{2.50}$$

$$v_{cC}(t) = \frac{\hat{V}_{tr}}{V_{dc}}(v_c^*(t) + v_{CM}) + 0.5 \tag{2.51}$$

Where v_{CM} is the third harmonic injected and can be computed for example in the following way:

$$v_{CM}(t) = -\frac{1}{2}[\max(v_a^*, v_b^*, v_c^*) + \min(v_a^*, v_b^*, v_c^*)] \tag{2.52}$$

In Figure 15 the computation of the common mode component when $m_a = 1$ is shown.

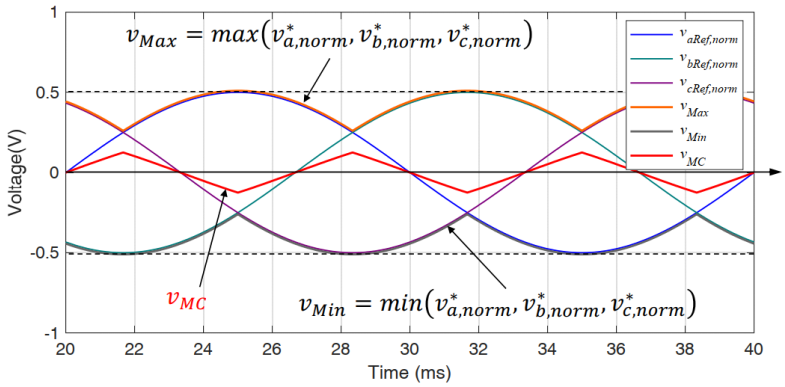

Figure 15 – Common mode component and $v_a^*(t)$, $v_b^*(t)$, $v_c^*(t)$ normalized values obtained multiplying for $\frac{\hat{V}_{tr}}{V_{dc}}$

Looking at Figure 16 it is clear that the resultant $v_{cA}(t), v_{cB}(t)$ and $v_{cC}(t)$ haven't reach the duty cycle saturation limit (their peak is lower than the one of the carrier) thus the objective voltage $v_a^*(t)$, $v_b^*(t)$, $v_c^*(t)$ peak, set to $\frac{V_{dc}}{2}$, can still be increased.

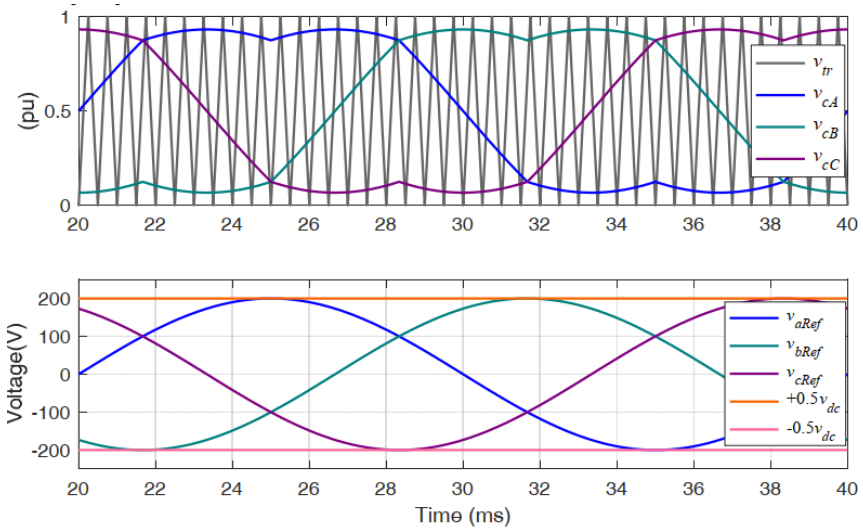

Figure 16 – Reference signals and objective voltages with $\hat{V}^* = \frac{V_{dc}}{2} = 200\ V$

The linear modulation are is extended as shown in Figure 17.

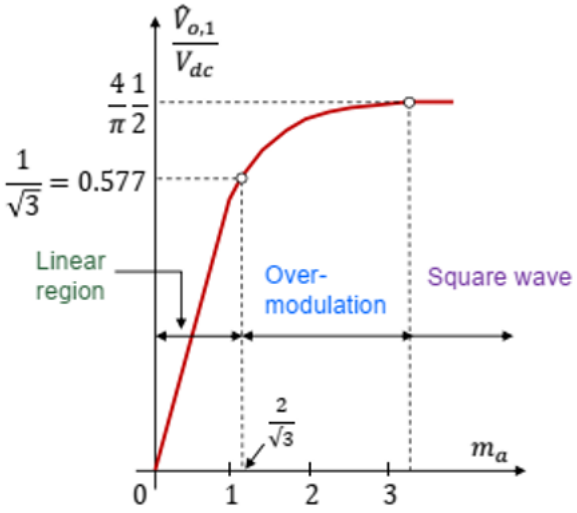

Figure 17 – Modulation regions with zero-sequence injection

The phase voltages fundamental harmonic can now reach an higher peak value at the limit of the linear modulation region without be affected by low order harmonics:

$$\hat{V}_{phase,1} = \frac{1}{\sqrt{3}} V_{dc} \tag{2.53}$$

The resultant reference waveform and objective voltages when it is chosen $\hat{V}^* = 230\,V$ with $V_{dc} = 400\,V$ are shown in Figure 18.

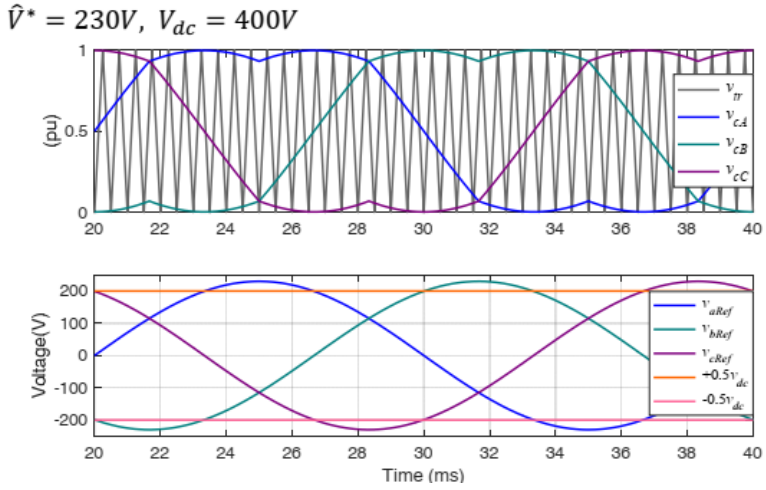

Figure 18 - Reference signals and objective voltages with $\hat{V}^* = \frac{V_{dc}}{\sqrt{3}} = 230\,V$

2.6 SVPWM: Space Vector PWM

The space vector modulation technique makes possible to obtain the same results of the Zero-sequence injection. The fundamental component of the phase voltage generated with either SVPWM or Zero-sequence injection is characterized by a maximum peak value equal to $\frac{Vdc}{\sqrt{3}}$ in the linear modulation area. Since for sinusoidal PWM the maximum peak value in the linear region is fixed to $\frac{Vdc}{2}$ it means that SVPWM and Zero-sequence injection increase the DC bus utilization by 15,47%.

The SVPWM algorithm can be easily translated in code. The objective of SVPWM is to approximate the reference vector:

$$\overrightarrow{V^*} = {v^*}_\alpha + j{v^*}_\beta = \sqrt{{v_\alpha}^2 + {v_\beta}^2}\, e^{j\,tan^{-1}\frac{v_\beta}{v_\alpha}} = \overline{|v_r|} e^{j\psi} \qquad (2.54)$$

Where ${v^*}_\alpha$ and ${v^*}_\beta$ are the objective voltages in the fixed reference frame.

Let's consider now the inverter scheme in Figure 19.

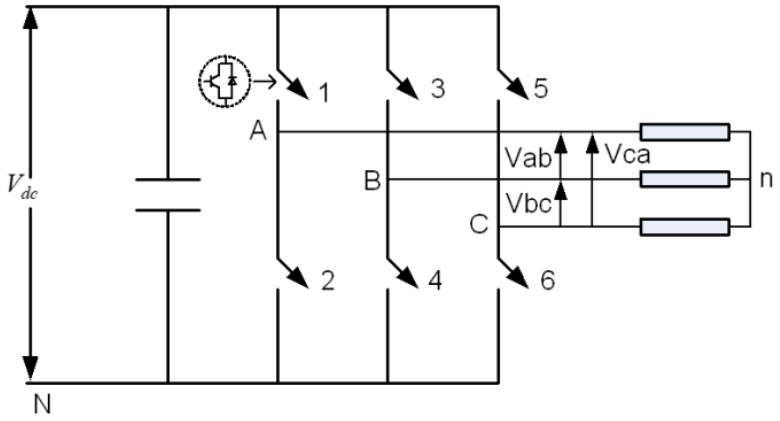

Figure 19 - Inverter scheme SVPWM

The space vector related to the 3-phase voltage can be computed with the following formula:

$$\overrightarrow{v_s} = \frac{2}{3}\left(v_a + \vec{a}v_a + \overrightarrow{a^2}v_c\right) \quad where \quad \vec{a} = e^{\frac{2}{3}j} \; and \; \overrightarrow{a^2} = e^{\frac{4}{3}j} \tag{2.55}$$

With reference to Figure 19 it is possible to derive the following equations:

$$v_A = v_{AN} = v_a + v_{nN} \tag{2.56}$$

$$v_B = v_{BN} = v_b + v_{nN} \tag{2.57}$$

$$v_C = v_{CN} = v_c + v_{nN} \tag{2.58}$$

Summing equations (2.56)-(2.58) and considering a balanced load, where the sum of the 3-phase voltages is null:

$$v_{nN} = \frac{1}{3}(v_A + v_B + v_C) \tag{2.59}$$

Then it is possible to derive for each phase voltage:

$$v_a = \frac{2}{3}v_A - \frac{1}{3}(v_B + v_C) \tag{2.60}$$

$$v_b = \frac{2}{3}v_B - \frac{1}{3}(v_A + v_C) \tag{2.61}$$

$$v_c = \frac{2}{3}v_C - \frac{1}{3}(v_A + v_B) \tag{2.62}$$

Remembering that for each inverter leg only one switch per time can be closed there exist only 8 possible switching configuration which are shown in Figure 20.

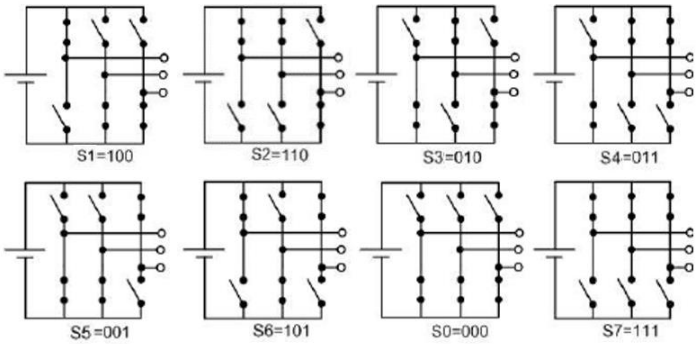

Figure 20 – Possible switching configurations (3-phase inverter)

Switches are considered ON when closed. It is possible to assign to each configuration a binary number made by 3 digits, one for each inverter leg switch configuration from left to right. 1 is assigned when the upper switch is closed (lower one opened) and 0 when the upper switch is open (lower one closed). With reference to Figure 19, Figure 20 and equations (2.55)(2.62) the following table can be derived:

Table 1 - Phase voltages and space vectors on the base of the switching configuration

State	***Switch ON***	***Binary sequence***	v_a	v_b	v_c	$\overrightarrow{v_s}$
0	2,4,6	000	0	0	0	0
1	1,4,6	100	$\frac{2}{3}V_{DC}$	$-\frac{1}{3}V_{DC}$	$-\frac{1}{3}V_{DC}$	$\frac{2}{3}V_{DC}$
2	1,3,6	110	$\frac{1}{3}V_{DC}$	$\frac{1}{3}V_{DC}$	$-\frac{2}{3}V_{DC}$	$\frac{2}{3}V_{DC}e^{j\frac{\pi}{3}}$
3	2,3,6	010	$-\frac{1}{3}V_{DC}$	$\frac{2}{3}V_{DC}$	$-\frac{1}{3}V_{DC}$	$\frac{2}{3}V_{DC}e^{j\frac{2\pi}{3}}$
4	2,3,5	011	$-\frac{2}{3}V_{DC}$	$\frac{1}{3}V_{DC}$	$\frac{1}{3}V_{DC}$	$\frac{2}{3}V_{DC}e^{j\pi}$
5	2,4,5	001	$-\frac{1}{3}V_{DC}$	$-\frac{1}{3}V_{DC}$	$\frac{2}{3}V_{DC}$	$\frac{2}{3}V_{DC}e^{j\frac{4\pi}{3}}$
6	1,4,5	101	$\frac{1}{3}V_{DC}$	$-\frac{2}{3}V_{DC}$	$\frac{1}{3}V_{DC}$	$\frac{2}{3}V_{DC}e^{j\frac{5\pi}{3}}$
7	1,3,5	111	0	0	0	0

The space vectors associated to each switch state can be represented on an hexagon as shown in Figure 21.

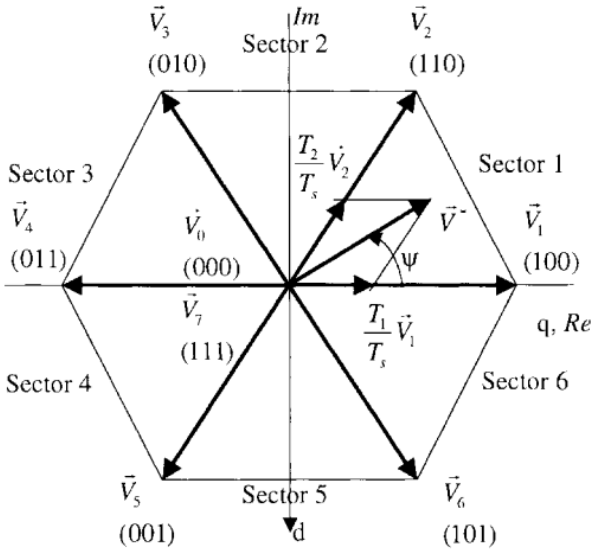

Figure 21 - Space vector representation of the switch states

The reference space vector computed with (2.54) formula rotates with radial frequency ω_e. If once every "Ts" the reference vector position and amplitude are sampled it is possible to approximate it with the 2 space vectors that delimitate the sector, tuning the time they are active, which means tuning the duty cycle of the pulses sent to the inverter legs. In order to achieve good results, the reference vector is considered fixed during Ts interval and the following formula is applied:

$$\overrightarrow{V^*}T_s = \overrightarrow{V_1}T_1 + \overrightarrow{V_2}T_2 \tag{2.63}$$

Where $\overline{V_1}$ and $\overline{V_2}$ are the vectors that delimitate the sector where $\overline{V^*}$ lays in that time instant [9].

Since the angle between the space vectors that delimitates a sector is $\frac{\pi}{3}$ and since

$|\overline{V_1}| = |\overline{V_2}| = \mathrm{V} = \frac{2}{3}\mathrm{Vdc}$ doing the projection of equation (2.63) along the imaginary axis:

$$V^*T_s \cos(90 - \psi)) = V\,T_2 \cos\left(90 - \frac{\pi}{3}\right)$$

$$V^* T_s \sin(\psi) = V\, T_2 \sin\frac{\pi}{3}$$

$$T_2 = \frac{V^*}{V} \; \frac{sin\,\psi}{sin\frac{\pi}{3}} T_s \tag{2.64}$$

Doing the projection on the real axis:

$$V^* T_s \cos\psi = V\, T_1 + V\, T_2 \cos\frac{\pi}{3}$$

$$V^* T_s \cos\psi = V\, T_1 + V \frac{V^*}{V} \; \frac{\sin\psi \cos\frac{\pi}{3}}{\sin\frac{\pi}{3}} T_s$$

$$T_1 = \frac{V^*}{V}(\cos\psi - \frac{\sin\psi \cos\frac{\pi}{3}}{\sin\frac{\pi}{3}}) T_s$$

$$T_1 = \frac{V^*}{V} \; \frac{\cos\psi \sin\frac{\pi}{3} - \sin\psi \cos\frac{\pi}{3}}{\sin\frac{\pi}{3}} \; T_s$$

$$T_1 = \frac{V^*}{V} \; \frac{sin(\frac{\pi}{3} - \psi)}{sin\frac{\pi}{3}} \; T_s \tag{2.65}$$

The null vectors $\overrightarrow{V_0}\ and\ \overrightarrow{V_7}$ are applied to modulate the amplitude of the approximated space vector computing the time for which they are appllied in the following way [9]:

$$T_{null} = T_0 + T_7 = \; T_s - (T_1 + T_2) \tag{2.66}$$

In order to have a switching frequency equal to $f_s = 1/T_s$ for the switches associated to each leg, the time slices previously computed must be sequenced accordingly. Only one switch between 1 and 0 should happen each T_s. A possible way to sequence the time slices is given in Figure 22.

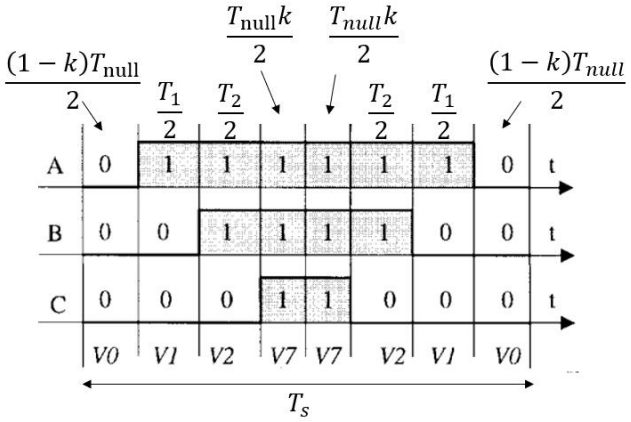

Figure 22 – Time slices sequence

K is a design parameter, it's choice influence the shape of the resulting waveforms related to the line to ground voltage mean value.

The authors in [10] have demonstrated that, choosing $k = \frac{1}{2}$, which means choosing $T_0 = T_7 = \frac{T_{null}}{2}$, leads to the same results obtained by the Zero-sequence injection of the triangular waveform shown in Figure 15. Thus, all previous considetation regarding the results of the zero-sequence injection are still valid.

Defining:

$$\mathrm{m} = \frac{V^*}{\frac{Vdc}{2}} \tag{2.67}$$

Table 2, derived applying the same method shown for the first sector, can be used to determine the time slices related to the different space vectors that delimitates each sector.

Table 2 – Equations to compute the time slices related to each sector

Sector I $(0 \leq \omega t \leq \frac{\pi}{3})$	Sector II $(\frac{\pi}{3} \leq \omega t \leq \frac{2\pi}{3})$	Sector III $(\frac{2\pi}{3} \leq \omega t \leq \pi)$
$T_1 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{\pi}{6})$ $T_2 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{3\pi}{2})$ $T_0 + T_7 = T_s - T_1 - T_2$	$T_2 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{11\pi}{6})$ $T_3 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{7\pi}{6})$ $T_0 + T_7 = T_s - T_2 - T_3$	$T_3 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{3\pi}{2})$ $T_4 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{5\pi}{6})$ $T_0 + T_7 = T_s - T_3 - T_4$
Sector IV $(\pi \leq \omega t \leq \frac{4\pi}{3})$	**Sector V** $(\frac{4\pi}{3} \leq \omega t \leq \frac{5\pi}{3})$	**Sector VI** $(\frac{5\pi}{3} \leq \omega t \leq 2\pi)$
$T_4 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{7\pi}{6})$ $T_5 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{\pi}{2})$ $T_0 + T_7 = T_s - T_4 - T_5$	$T_5 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{5\pi}{6})$ $T_6 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{\pi}{6})$ $T_0 + T_7 = T_s - T_5 - T_6$	$T_6 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{\pi}{2})$ $T_1 = \frac{\sqrt{3}}{2} m T_s \cos(\omega t + \frac{11\pi}{6})$ $T_0 + T_7 = T_s - T_1 - T_6$

The maximum amplitude of the reference phase vector that can be reached without overmodulation can be found imposing $T_0 + T_7 = 0$.

Considering equation (2.66), in the specific case where $T_1 = T_2 = \frac{T_s}{2}$ and thus $\psi = \frac{\pi}{6}$ and applying (2.64):

$$\frac{T_s}{2} = \frac{V^*}{\frac{2}{3}Vdc} \frac{\frac{1}{2}}{\frac{\sqrt{3}}{2}} T_s$$

$$1 = \frac{V^*}{Vdc}\sqrt{3}$$

$$V^* = \frac{Vdc}{\sqrt{3}} \tag{2.68}$$

This result is true for all the possible angle of the reference space vector which means that, in order to work in the linear region, the reference vector must rotate inside the circle inscribed in the hexagon of Figure 21.

2.7 A brief comparison between FOC and DTC

In order to control a PMSM there are several techniques that can be used, the most popular are the FOC (Field Oriented Control) and the DTC (Direct Torque Control). Both methods are able to decouple the torque control from the flux control during transient and steady-state, but the results obtained in terms of response in time are different. It is not easy to compare these two techniques for different reasons, first of all, the switching frequency of the inverter is constant when applying FOC while it's variable with DTC and depends on the operating point of the motor. This means that the tests must be chosen in order to meet an operating point that causes similar inverter switching frequency for both methods. Another comparison issue regards the sampling time chosen, since one of the main advantages of DTC is to require lower computational power this method should be applied setting a higher sampling time.

The authors in [2] have conduct different simulation tests according to the previous consideration which led them to precise comparison conclusions. DTC provides a quicker response for torque step transient if compared to FOC of at least an order of magnitude while, the speed step variation is track almost in the same way by the two control strategies. However, the drawback of the DTC is to be characterized by higher flux, current and torque ripple which translates in high audible noise at low speed. This makes the FOC considerable superior to DTC in terms of steady-state behaviour. In this thesis it has been chosen to focus the attention on the FOC technique also because it makes possible to improve performance acting both on current and speed control loops.

2.8 FOC control scheme and description

The FOC technique exploits the equation related to the d and q axes (2.38)-(2.39) and the one related to the torque (2.33) in order to apply a control similar to the one of a DC machine. Since the torque produced by the machine is directly proportional to the i_q current, and the rotor, due to the permanent magnet, generates a constant flux itself, there is no need to produce flux through the i_d current, which can be kept null. This reduce the stator losses and increase efficiency. For what concerns the d-axis the reference i_d current will be set to:

$$i^*_d = 0\ A \tag{2.69}$$

Since the magnetic flux produced by the stator will be related only to the i_q current the objective angle between stator and rotor flux will be 90° which corresponds to maximum efficiency. In order to track the reference currents (i^*_d and i^*_q) two digital PI controllers (PI_c) are implemented inside a close loop whose purpose is to generate the v^*_d and v^*_q input signals. These are translated in the v^*_α and v^*_β voltages through the inverse park transformation (2.28) and used to compute the reference vector (2.54), input of the SVPWM algorithm. The SVPWM algorithm generates the impulses needed to control the inverter switches. In this way the 3-phase voltages are generated. Through the use of 2 transducers, the 3-phase currents are sampled (from i_a and i_b it is possible to derive i_c) and the same is done for the rotor position ϑ_m using a resolver or an encoder. The rotor angle in multiplied for the number of couple of poles of the motor in order to obtain the electrical angle ϑ_e. This value is used to transform the 3-phase currents applying first the Clark transformation and then the Park one. Finally, the i_q and i_d currents, output of the feedback path, are compared with the reference currents in order to generate the error, input of the PI_c controllers. The reference current i^*_q is the output of another digital PI controller implemented inside a speed loop (PI_v). The feedback signal of the loop is the rotor angular speed ω_m obtained doing the derivative of the rotor angular position. The scheme in Figure 23 represent the control technique actuated [11].

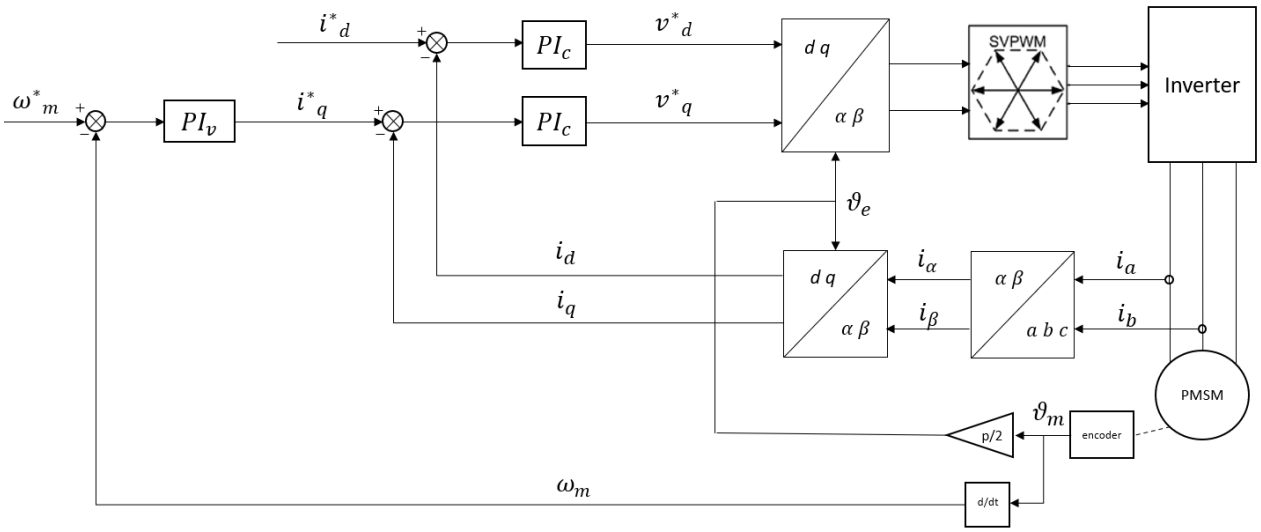

Figure 23 - FOC scheme

2.9 Feedforward compensation

The concept of feedforward compensation has been exploited in order to simplify the transfer functions between the output of the PI_c controllers related to the current loops and the output currents i_d and i_q.

Let's consider equations (2.38)and (2.39), they can be rewritten in the following form:

$$i_d = \frac{v_d}{r_s + L_d s} + \frac{L_q \frac{P}{2} \omega_m i_q}{r_s + L_d s} \tag{2.70}$$

$$i_q = \frac{v_q}{r_s + L_q s} - \frac{\frac{P}{2} \omega_m \left(L_d\, i_d + \lambda_f\right)}{r_s + L_q s} \tag{2.71}$$

The back e.m.f contribution in each of the 2 equations can be considered as a disturbance. It is possible to compensate the effect of the disturbance applying:

$$v_d = v_d{}' - L_q \frac{P}{2} \omega_m i_q \tag{2.72}$$

$$v_q = v_q{}' + \frac{P}{2} \omega_m \left(L_d\, i_d + \lambda_f\right) \tag{2.73}$$

Replacing v_d and v_q in (2.70) and (2.71):

$$\frac{i_d}{v_d{}'} = \frac{1}{r_s + L_d s} \tag{2.74}$$

$$\frac{i_q}{v_q{}'} = \frac{1}{r_s + L_q s} \tag{2.75}$$

Since L_q and L_d are known, i_q and i_d can be computed from the 3-phase currents and the motor speed can be derived from the angular position of the motor ϑ_m, it is possible to modify the scheme in Figure 23 in the following way:

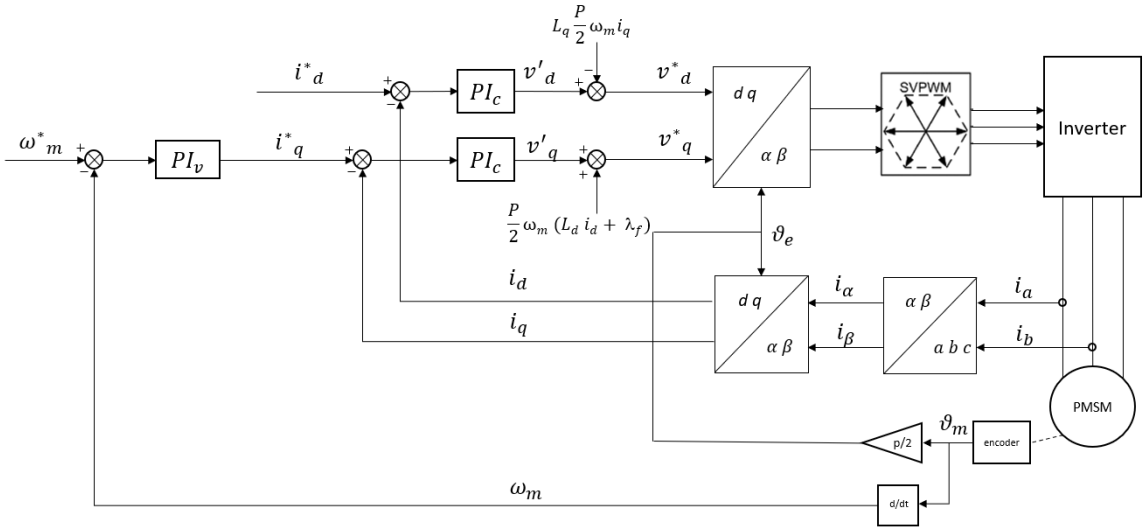

Figure 24 - FOC with feedforward compensation

2.10 Flux weakening control

In order to control the motor above its rated speed (base speed) a different technique must be used. The characteristic curve of a PMSM is shown in Figure 25

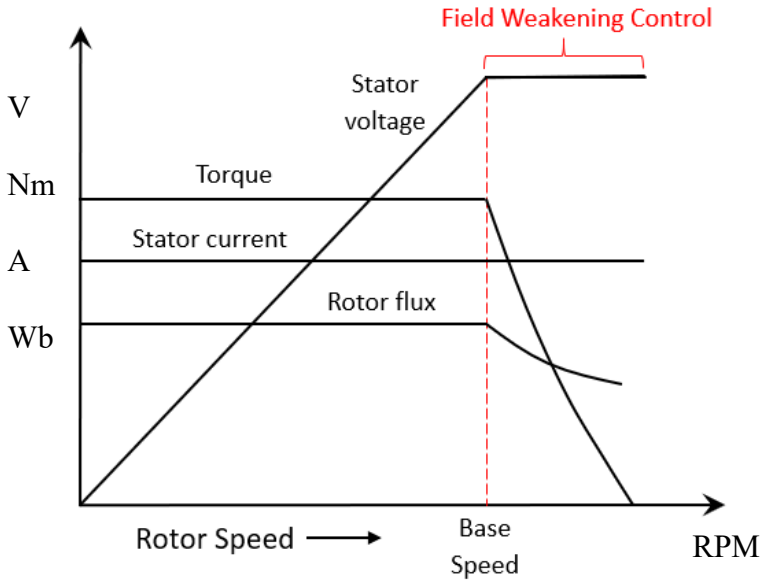

Figure 25 – Characteristic curve of a PMSM

In the constant torque region, the motor can increase its speed until the back E.M.F produced reaches the inverter output voltage limit. Ones this limit has been reached, the motor is working at maximum power and the torque capability of the motor starts decreasing when speed is increased. Knowing the rated speed, torque and current it is possible to find the voltage at the base speed point and thus the minimum DC bus voltage needed to drive the motor. Equation (2.29)(2.30) are written again below for the sake of clarity:

$$v_q = r_s i_q + \frac{d}{dt}\lambda_q + \frac{\mathrm{P}}{2}\omega_m \left(L_d i_d + \lambda_f\right)$$

$$v_d = r_s i_d + \frac{d}{dt}\lambda_d - \frac{\mathrm{P}}{2}\omega_m L_q i_q$$

From the motor datasheet taken as reference *(EMJ-04APB22)* the following parameters are found:

$$I_{rated} = 2.7\ A$$

$$T_{rated} = 1.27\ Nm$$

$$w_{rated} = 3000\ RPM = 314.16\ rad/s$$

While λ_f can be computed from (2.33) ($T_e = \frac{3}{2} 0.5p[\lambda_f i_q + (L_d - L_q) i_d i_q]$):

$$\lambda_f = 0.07846\ Wb$$

Considering the FOC strategy applied at rated speed, if the rated torque is applied to the motor:

$$i_q = I_{rated} \tag{2.76}$$

$$i_d = 0\ A \tag{2.77}$$

And these values will be constant, thus $\frac{d}{dt}\lambda_q$ and $\frac{d}{dt}\lambda_d$ will be null.

Replacing these values in (2.29)(2.30) the following voltage values are obtained:

$$v_q = 104.94\ \text{V}$$

$$v_d = -22.05\ \text{V}$$

Thus it is possible to compute the voltage reference value which, according to (2.68) ($V^* = \frac{Vdc}{\sqrt{3}}$) can be used to obtain the DC bus voltage needed to feed the inverter.

$$V^*_{max} = \sqrt{v_q^2 + v_d^2} = 106.94\ V \tag{2.78}$$

$$V_{dc} = V^* \sqrt{3} = 185.22V \tag{2.79}$$

Equations (2.29)(2.30) can be further simplified neglecting the voltage contribution due to the stator resistance (which can be not considered, since its value is really low if compared to the back E.M.F. value).

$$|v_q| = \frac{\mathrm{P}}{2} \omega_m\ (L_d i_d + \lambda_f) \tag{2.80}$$

$$|v_d| = \frac{\mathrm{P}}{2} \omega_m L_q i_q \tag{2.81}$$

Combining equations (2.80), (2.81) and (2.78) the constant voltage set of locus can be found:

$$V_{\max}^{*2} = \frac{\mathrm{P}^2}{4}\omega_m^2\left(L_q^2 i_q^2 + \left(L_d i_d + \lambda_f\right)^2\right)$$

$$\frac{V_{max}^{*2}}{\frac{\mathrm{P}^2}{4}\omega_m^2 L_q^2} = i_q^2 + \frac{L_d^2}{L_q^2}\left(i_d + \frac{\lambda_f}{L_d}\right)^2 \tag{2.82}$$

Equation (2.82) describes a set of ellipses in a $i_d - i_q$ chart centered in $\left(-\frac{\lambda_f}{L_d},\ 0\right)$ which becomes a set of circles with radius $R = \sqrt{\frac{V_{max}^{*2}}{\frac{\mathrm{P}^2}{4}\omega_m^2 L_q^2}}$ when $L_d = L_q$.

Considering the rated current:

$$I_{rated} = \sqrt{I_q^2 + I_d^2} \tag{2.83}$$

which describes a circle centered in (0, 0) in the a $i_d - i_q$ chart it is possible to represent both the constant voltage locus and the maximum current locus in the same graph (Figure 26) [12].

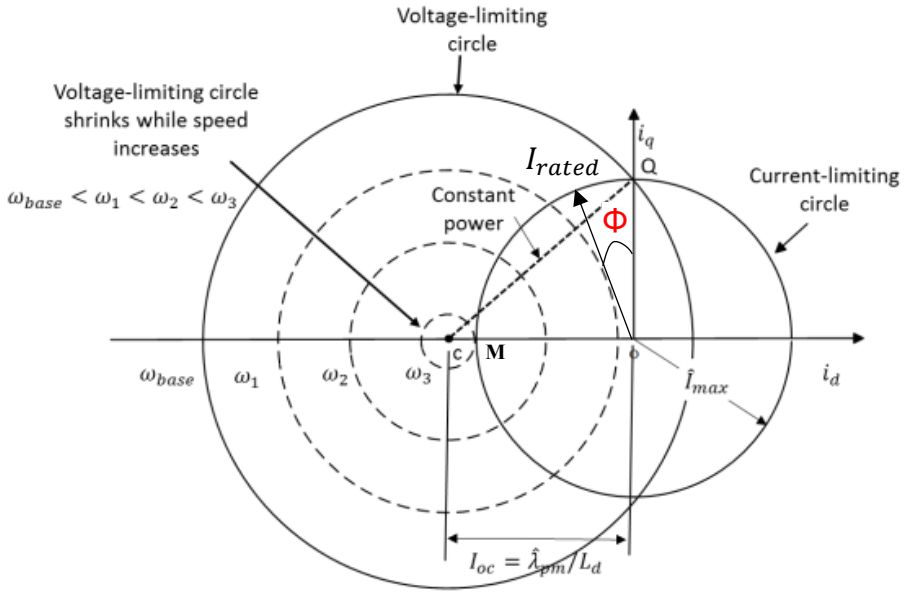

Figure 26 - Limiting circles of both voltage and current

When the base speed is reached, I_q is maximum while $I_d = 0\,A$. This situation is represented by the working point Q in Figure 26. In order to increase speed, the angle Φ of the current vector must be increased, which means generating a negative I_d current whose action is decreasing the linked flux along the d-axis ($\lambda_d = L_d i_d + \lambda_f$). However, in order to increase the absolute value of the i_d current the i_q current must decrease $I_q = \sqrt{I_{rated}^2 - I_d^2}$, which means decreasing the maximum electrical torque that the motor is able to produce.

Furthermore, if $I_{rated} < \frac{\lambda_f}{L_d}$, which is true for most of the cases, the maximum theoretical speed that the motor can reach is limited and corresponds to the operating point M in Figure 26 when $I_q = 0\,A$ and $I_d = -I_{rated}$ [13].

2.11 Anti-windup PID controller

Due to their wide use and their simplicity it has been decided to study the PID controller with the anti-windup feature and to apply them for what concerns the flux weakening control. The windup it's a phenomena that happens when the controller output saturates due to real application limitations. In the PMSM case, the current reference cannot be greater than the maximum current indicated in the motor datasheet, thus, the output of the controller related to the speed loop must be saturated. Due to this saturation, the rise time and settling time become higher and the integrated error causes an higher overshoot than the one expected in the ideal case. Different techniques can be used to improve the controller behavior known as anti-windup techniques [14]. The most common approach is the back-calculation one, which purpose is to prevent the integral term to accumulate a large value. The difference between the ideal output and the real one, multiplied by a tunable gain, is added to the input of the integral term. A scheme example is proposed in Figure 27 [15]. This technique can't reduce the rise time but improves the behavior in terms of overshoot and settling time.

Figure 28 compares the performance over time of the same plant with a PID controller without output saturation (blue line), with output saturation (red line) and with both output saturation and back-calculation (green line).

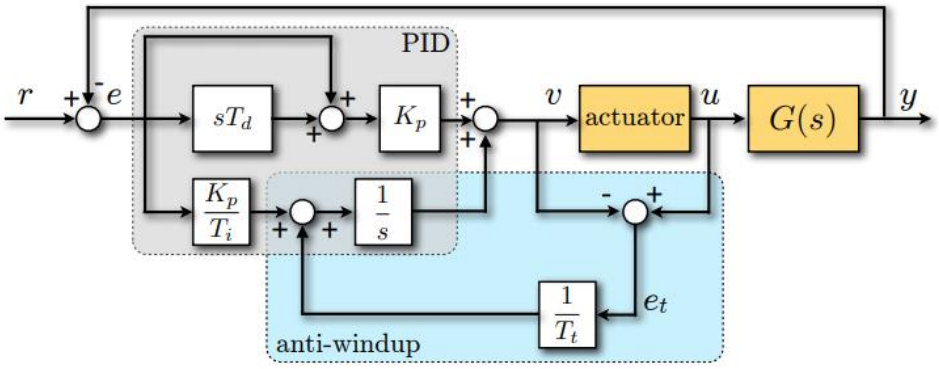

Figure 27 - Anti-windup back calculation scheme

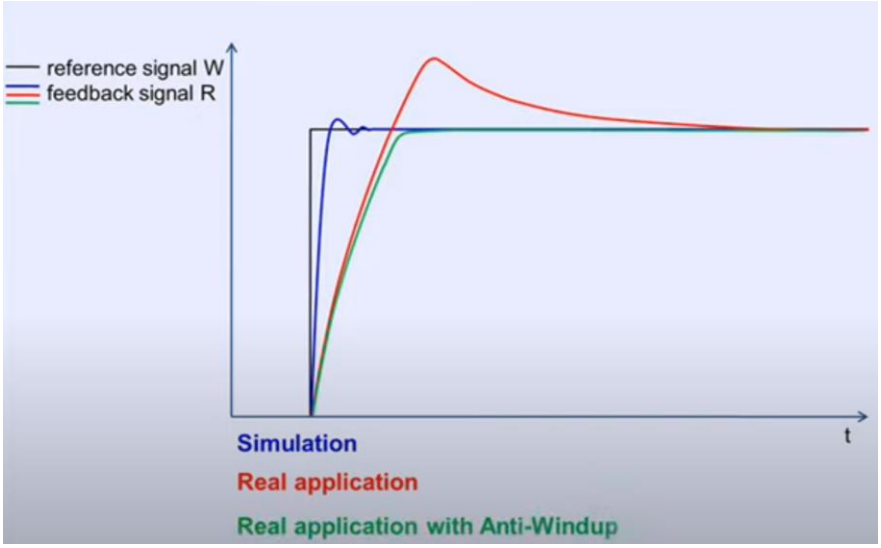

Figure 28 - Performance comparison of PID controller with and without anti-windup feature

Since the saturation constraint is known a-priori no sensors are needed to compute the difference between the saturated value and the controller output. The gain applied to this difference $\left(\frac{1}{T_t}\right)$ define how fast the input of the integral term will go to 0.

When the output of the controller is not saturated this difference is 0 and the PID works as usual.

3 Chapter III: Simulink model and PI design

In this chapter it will be shown how the previous theoretical considerations have been exploited in order to build the Simulink [13]model of the PMSM and of the control scheme shown in Figure 24. The Simulink model will be used at first to tune the controllers parameters, taking as reference the developers kit built by Texas Instrument (*TMDSHVMTRPFCKIT)*. The kit includes all the hardware components necessary for FOC. The digital control card "*Piccolo F28035*" elaborates the control inputs that are sent to the inverter. The inverter is attached to the Estun PMSM (*EMJ-04APB22*). The PI controllers and the SVPWM algorithm in the Simulink model will be placed inside triggered layers (discrete domain) since they are implemented inside the *Piccolo* control card. Instead, the PMSM model will run in the continuous domain.

3.1 PMSM Simulink model

The PMSM Simulink model is designed considering the equation (2.39)-(2.40) and built inside the "PMSM" subsystem. The three phase voltages V_{fa}, V_{fb} and V_{fc} and the load torque T_load are the inputs of the subsystem while the three phase currents vector I_{abc}, the rotor angular speed w_m and the electric torque T_e are the outputs. The content of the subsystem is shown in Figure 29. It is possible to notice that 3 other subsystems are designed inside the previous one, the first (light blue background on the left) implement both the Clark and Park transformations, the second (yellow background) is derived from equations (2.39), the third (grey background) is derived from equation (2.40) and the last one (light blue background on the right) implement the inverse Clarke and Park transformations. Their content is shown from Figure 30 to Figure 33, constants and gains have been highlighted in yellow.

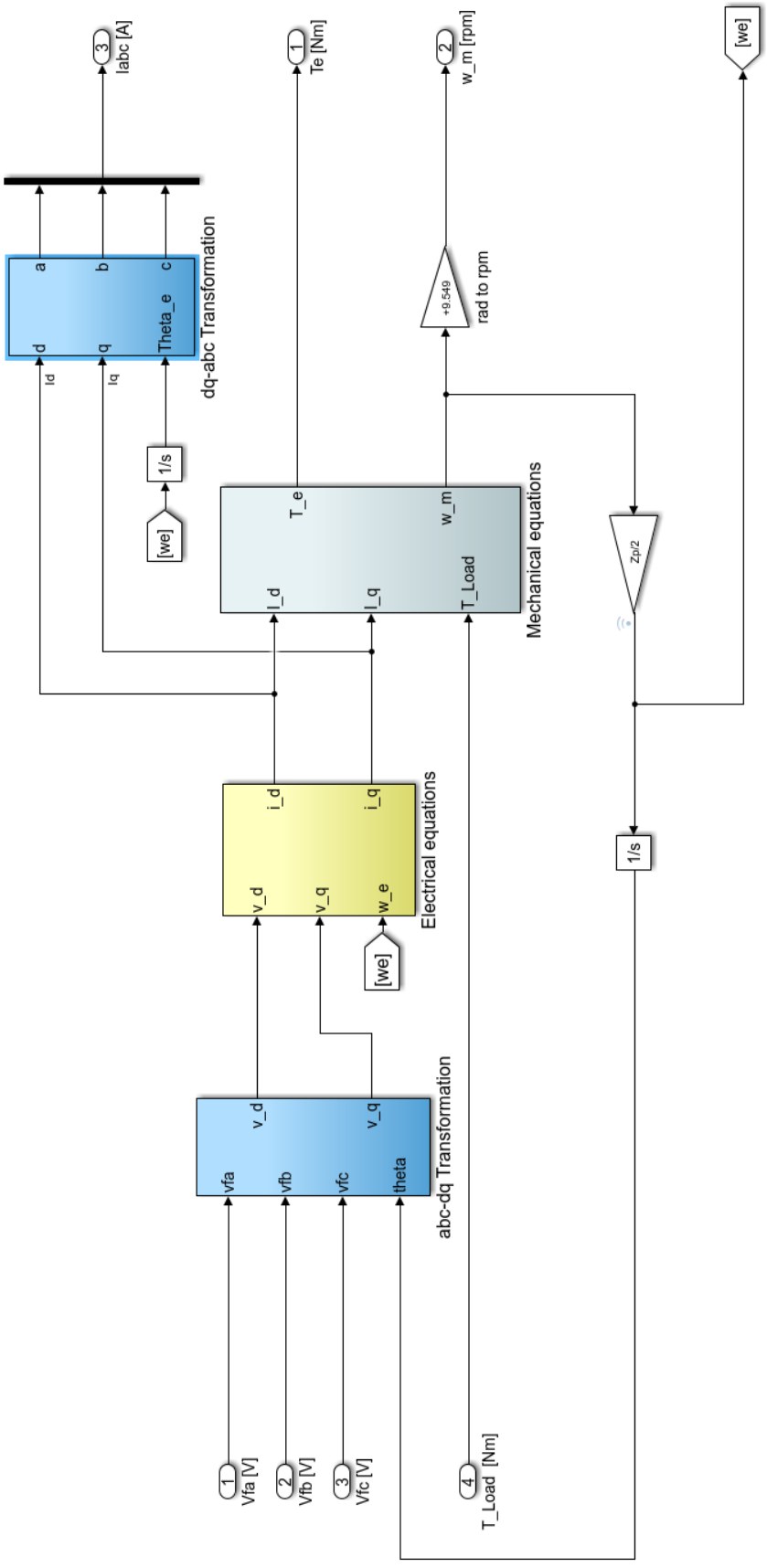

Figure 29 – PMSM subsystem

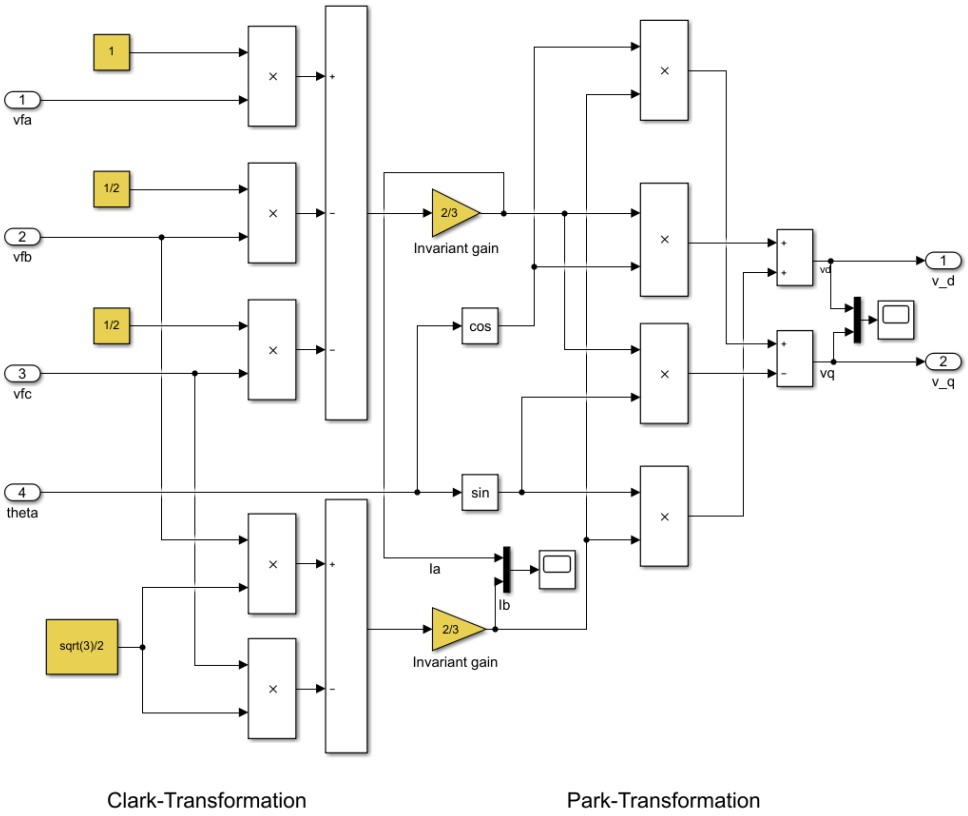

Figure 30 – "abc-dq Transformation" subsystem

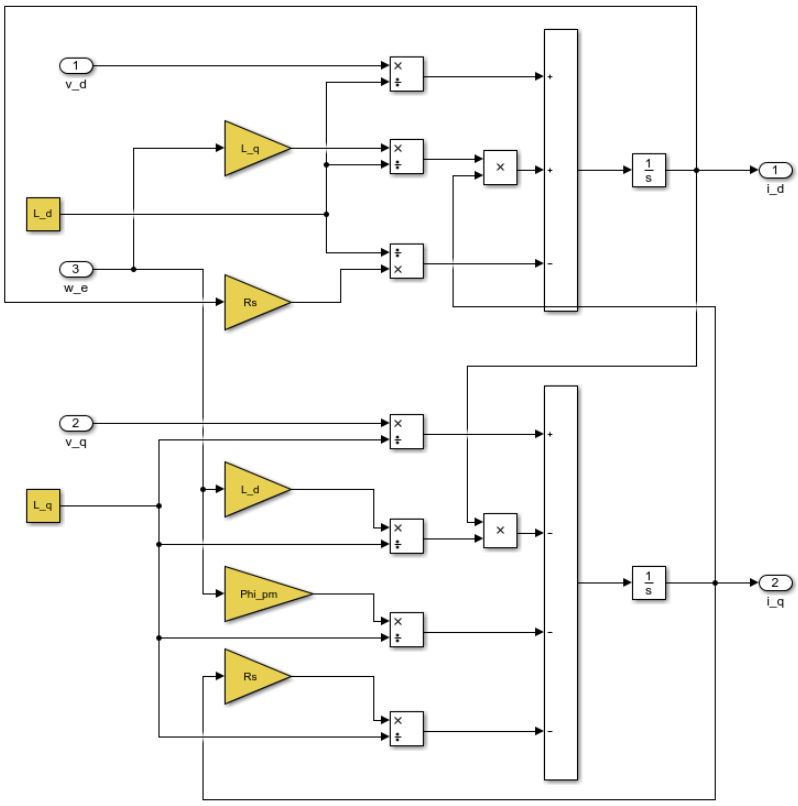

Figure 31 – "Electrical equations" subsystem

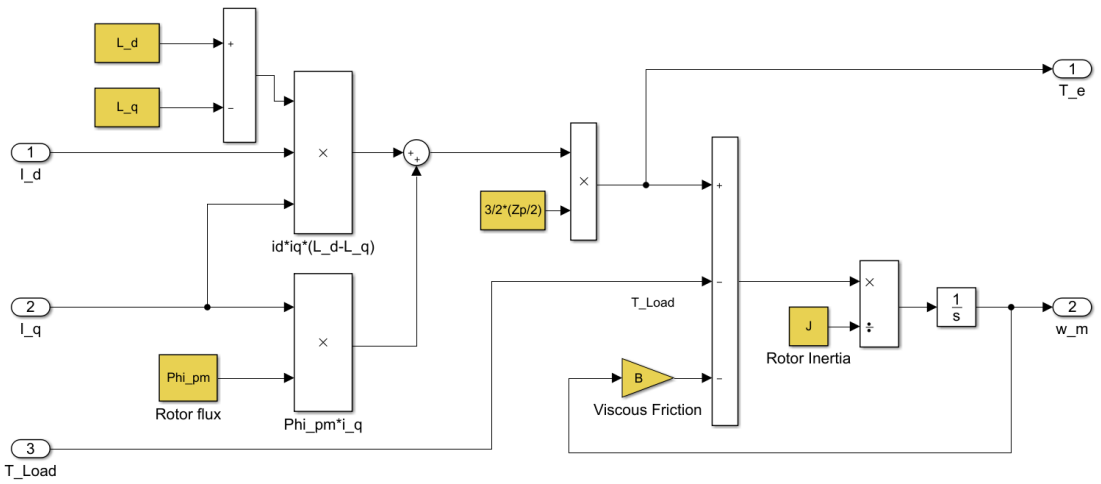

Figure 32 – "Mechanical equations" subsystem

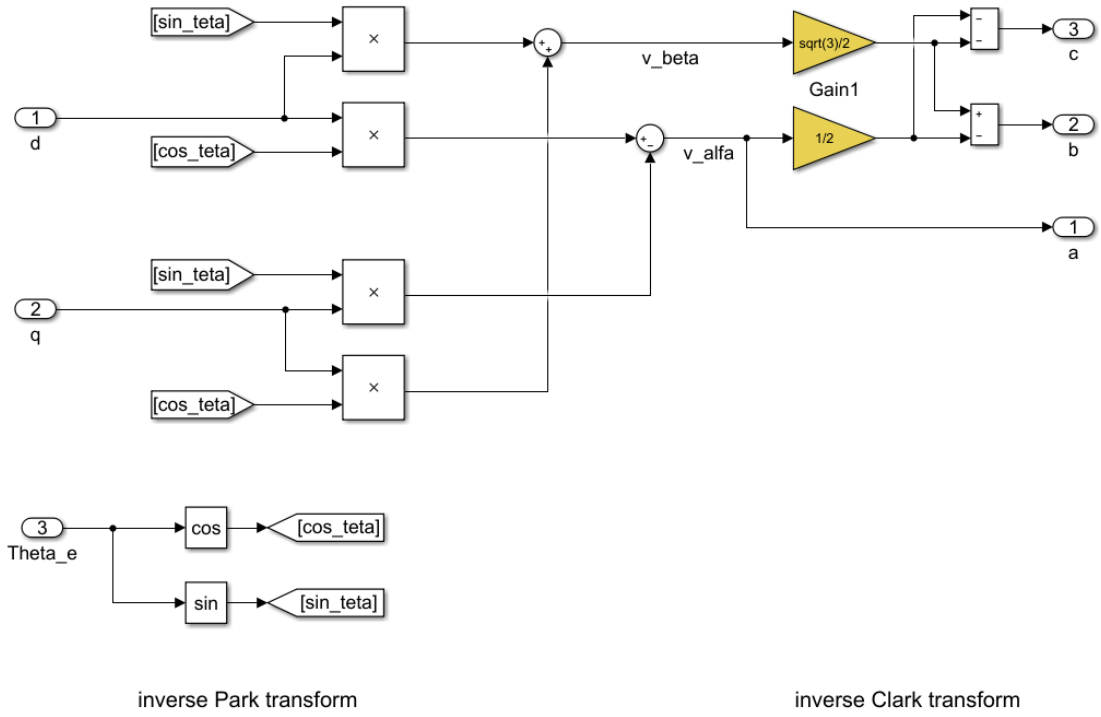

Figure 33 – "dq-abc Transformation" subsystem

3.2 SVPWM and Inverter Design

The theory explained in chapter 2.6 has been exploited in order to implement the SVPWM algorithm in Simulink through the Matlab function block shown in Figure 34 (see appendix A for code details).

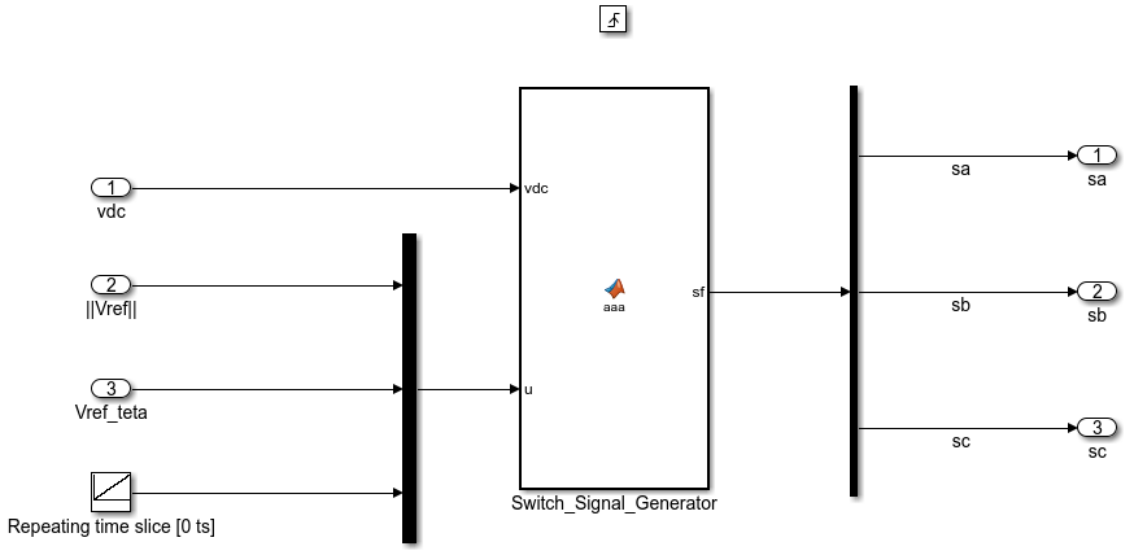

Figure 34 - SVPWM implementation

The reference vector is computed each time "ts" thanks to the current control loop, which will be described in the next chapter. Then it is used by the SVPWM to compute the inverter gate signals. The time "ts" corresponds to the T_s of equation (2.63) and thus the inverter switching frequency is $f_s = \frac{1}{ts}$. The Matlab function block is placed inside a subsystem which is triggered each ts/N. N is a design parameter that depends on the computing power of the processor that will be used to implement the control. The higher N is, the faster the algorithm will be run, which translates in sending the pulse signals with major time accuracy.

The outputs of the function are sa, sb and sc which are the pulses sent to the gates of the upper switches of the inverter legs. The outputs are only 3 because the other 3 gate pulses (related to the inverter lower switches) are computed as opposite of the upper ones inside the inverter block.

The inputs of the Matlab function block are:

- The continuous voltage vdc attached to the inverter.

- The reference voltage magnitude ||Vref|| and its angle with respect to the fixed reference frame α- β.

- A discretized ramp signal between 0 and ts, implemented through the "repeating sequence stair" Simulink block, which will be used to compare the actual time with the time slices computed through the algorithm. The "repeating sequence stair" block request as input a vector which as been computed as linspace(0, ts*(1-1/N), N). For N → ∞ this signal tends to a continuous ramp signal which means a more accurate comparison between the actual time and the time slices computed in the algorithm. Since this comparison leads to a change in the output of the function, increasing N value means having the gates pulse sent at the right time.

For what concerns the inverter, it has been implemented through the Inverter(Three phase) block of the Simscape Power System library. This makes possible to easily increase model accuracy when the inverter physical parameters are known just changing the inverter settings as shown in Figure 35.

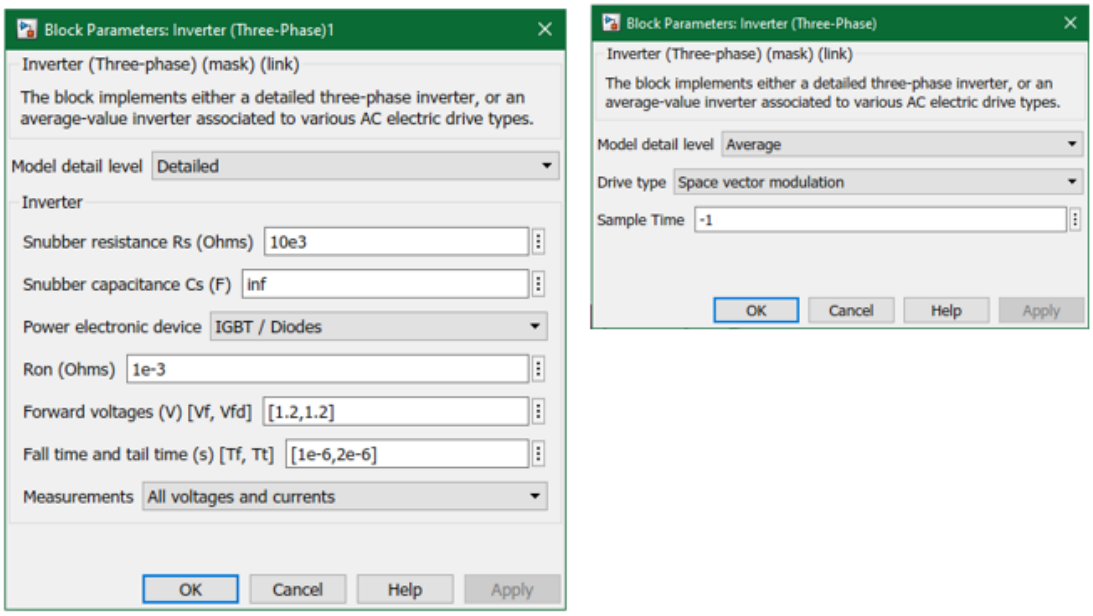

Figure 35 - Inverter settings, on the left the most accurate one, on the right the one used in the model designed

Figure 37 (pp. 43) shows how the SVPWM algorithm triggered block, the inverter block and the PMSM block are linked together.

The "variable reassignment" function is just used for Simulink computational purpose since the software asks to reassign the pulses variables before sending them to the inverter. The gains highlighted with a yellow background are used to convert the line to ground voltages into the phase voltages translating equations (2.60)(2.62) in suitable Simulink representation. A filter has been used in order to remove the high order harmonics generated by the inverter at frequency f_s equal to the switching frequency of the inverter (which in standard PWM corresponds to the carrier frequency). It has been designed a low pass filter with cutting frequency $f_c = \frac{f_s}{10}$ thus the filter transfer function obtained is the following:

$$filter(s) = \frac{1}{1 + \frac{s}{2\pi \frac{f_s}{10}}} \tag{3.1}$$

This filter has been applied to each of the voltages coming from the inverter. The "Filter" block shown in Figure 37 corresponds to the following Matlab code:

```
tau=1/(0.1*2*pi()*(1/ts));
filter =1/(1+s*tau);
Filter= [filter           0                0;
         0              filter             0;
         0                        0        filter];
```

Figure 36 – Filter Matlab code

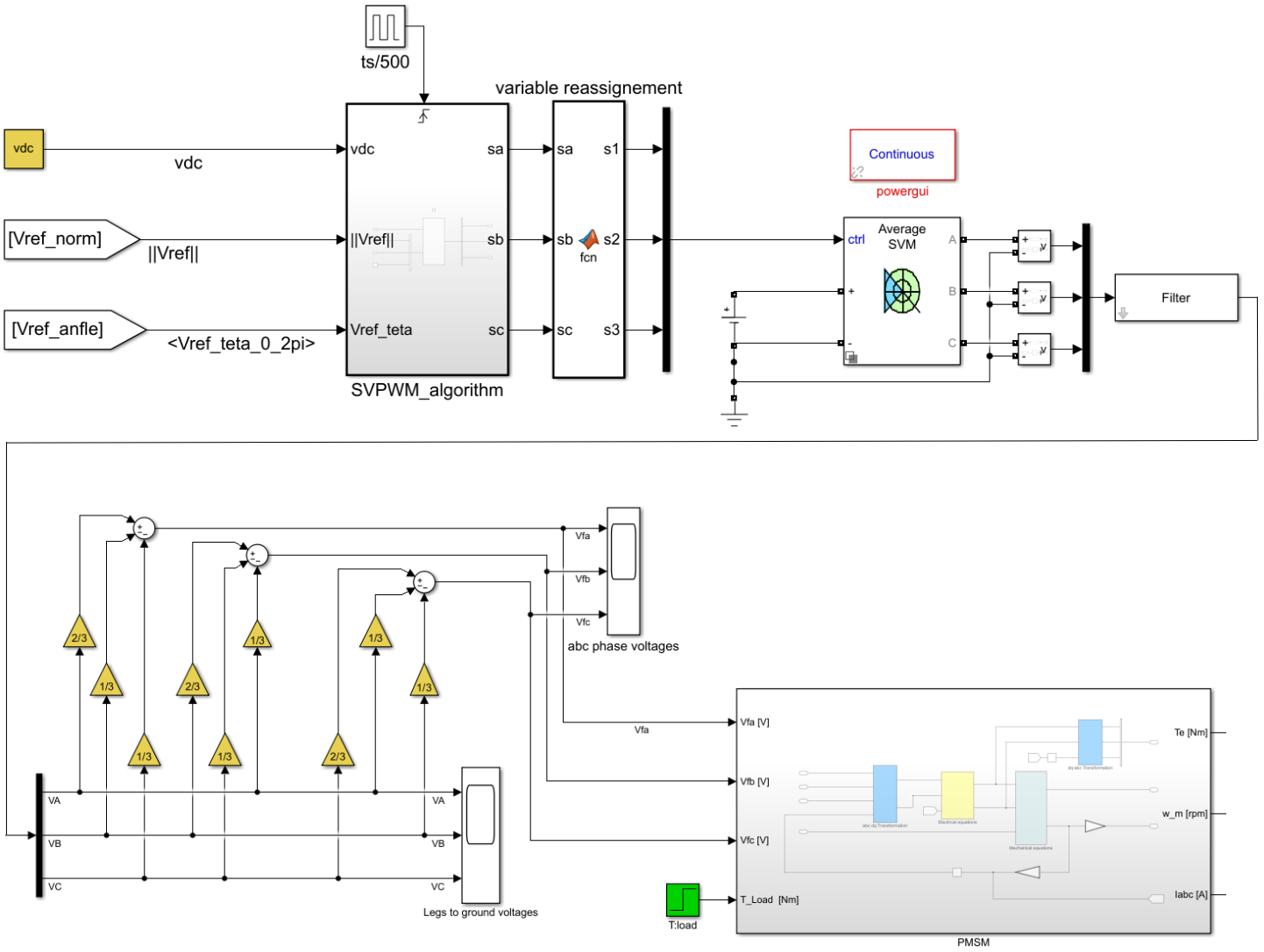

Figure 37 – Simulink model of SVPWM, Inverter and PMSM

3.3 Current and Speed control loops

The FOC scheme described in chapter 2.8 has been implemented as shown in Figure 38 and Figure 39. For what concerns the speed loop the reference signal is given as a step signal in RPM, the error between reference and actual speed is then multiplied for a constant gain to convert it to rad/s and sampled by a discrete PID controller block. The output, which is the i^*_q current reference, input of the "*Current loop + Feedforward control*" Subsystem is saturated to ±8.1 A which is the maximum limit for the starting current stated in the *EMJ-04APB22* datasheet. The two current loops have been placed in a separate subsystem in order to be able to set the sample time of the discrete current PIDs at the same value just changing the triggering signal input of the subsystem. Moreover, the PID related to the speed loop is left outside so that it possible to decide different sample times for the speed and the current loops. The three phase currents and the electrical angular speed, obtained from the angular speed of the motor, are the other inputs of the "*Current loop + Feedforward control*" Subsystem.

Inside the subsystem, the electrical angular position ϑ_e is obtained through the use of a discrete integrator and used inside the Clark-Park transformation subsystem to get, from the three phase currents, the i_d and i_q currents. These are multiplied for the motor speed and for the λ_q and λ_d linked fluxes to compute the voltage values that will be respectively added and subtracted to the output of the d and q PIDs following the basic principles of feedforward compensation described in chapter 2.9 (see equations (2.72)-(2.73)). The same i_d and i_q are used to compute the error signals, inputs of the discrete PIDs used for the current loops. Finally, the v_d and v_q output voltages are converted in the correspondent v_α and v_β and used to compute the norm and the angle of the reference space vector $\overline{v_r}$, which will be the inputs of the "*SVPWM_algorithm*" described in the previous chapter. In Figure 39 it is possible to see that the norm of the reference space vector $\overline{|v_r|}$ is saturated to $\frac{1}{\sqrt{3}}V_{DC}$, which is the limit value in order to work in the linear modulation area. The reference angle is converted from the [-π π] interval to [0 2 π] interval just for computation purposes.

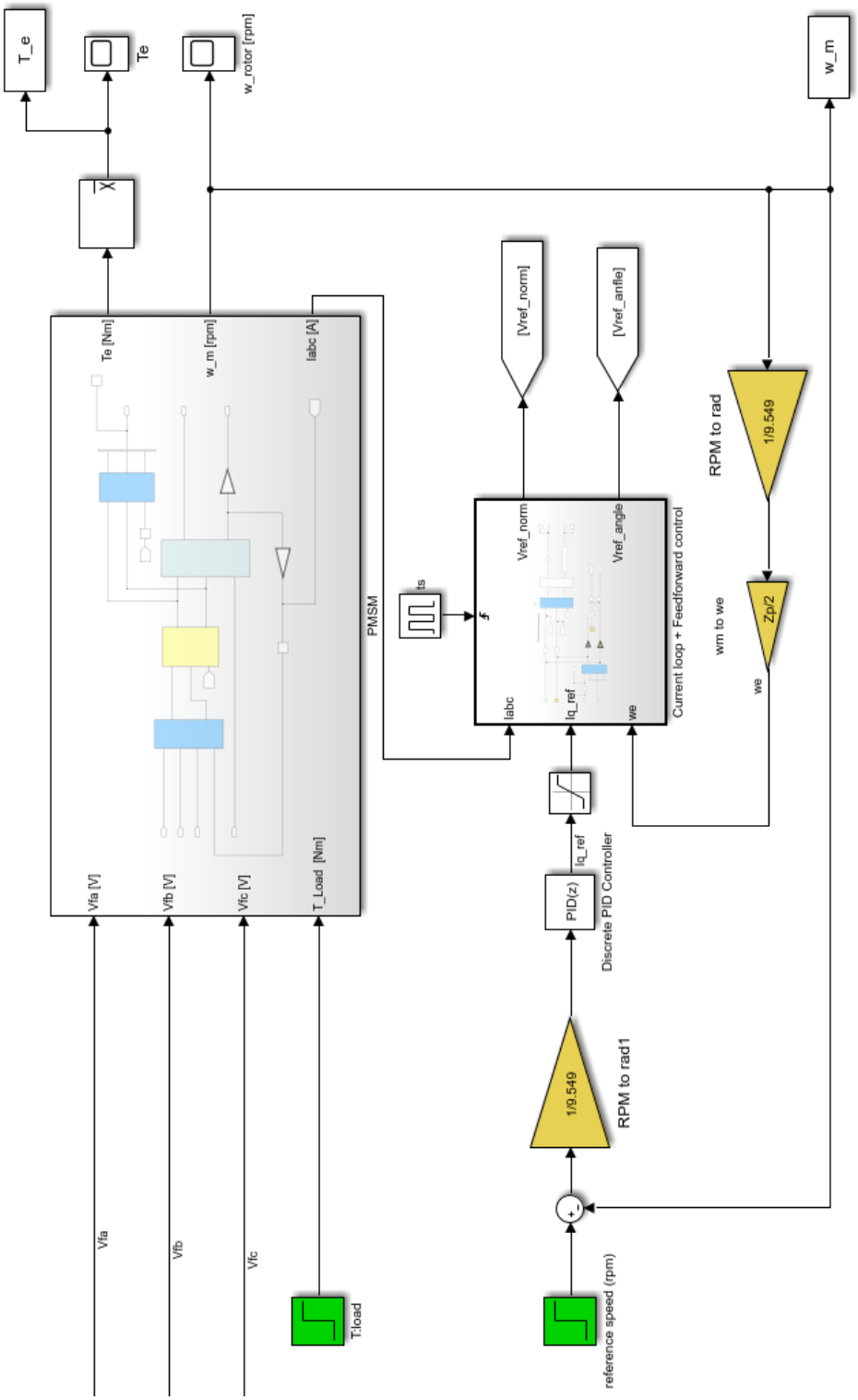

Figure 38 - PI cascade controller Simulink implementation

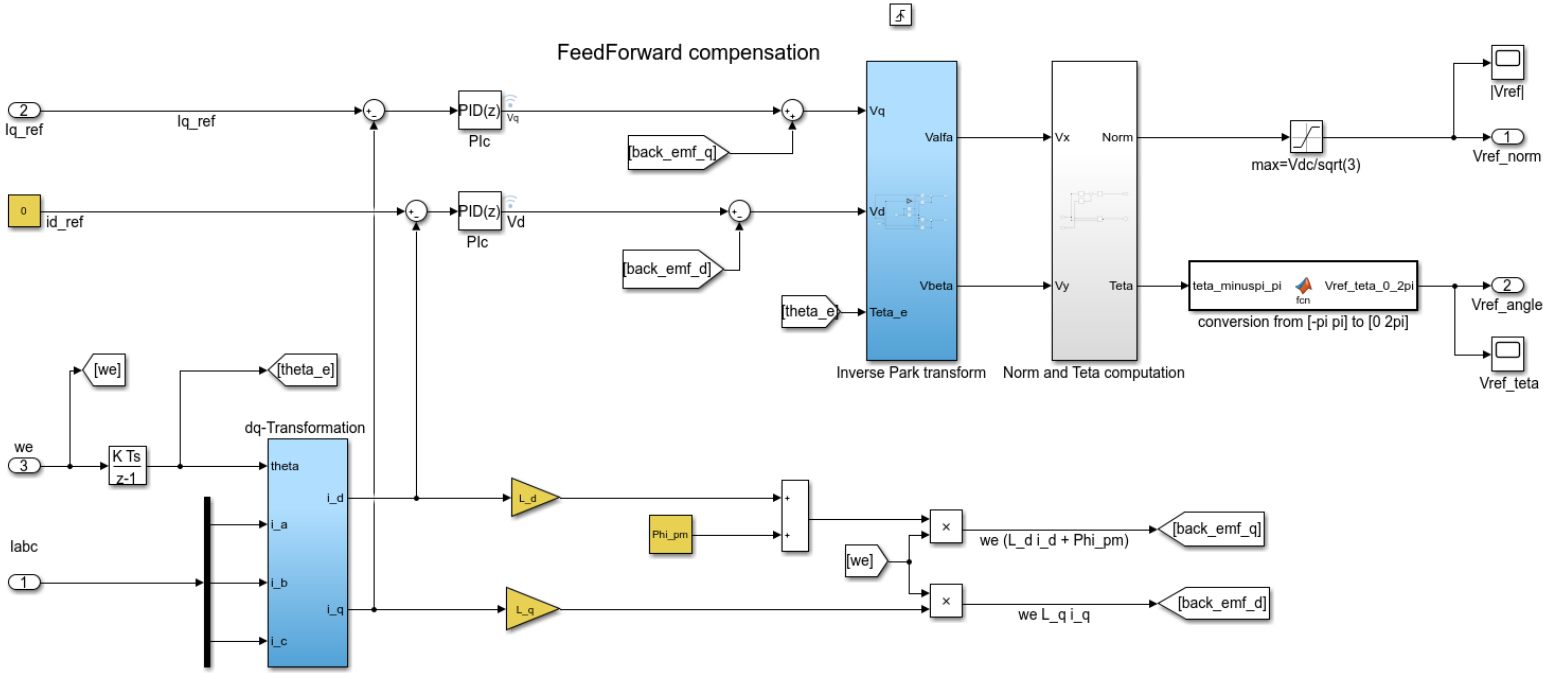

Figure 39 –Current loop + Feedforward control Subsystem content

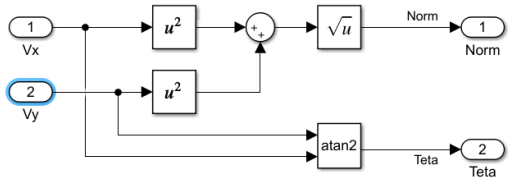

Figure 40 - Norm and Teta computation Subsystem

3.4 PI parameters design

In order to tune the PI parameters at first the simplified model in Figure 41 has been considered.

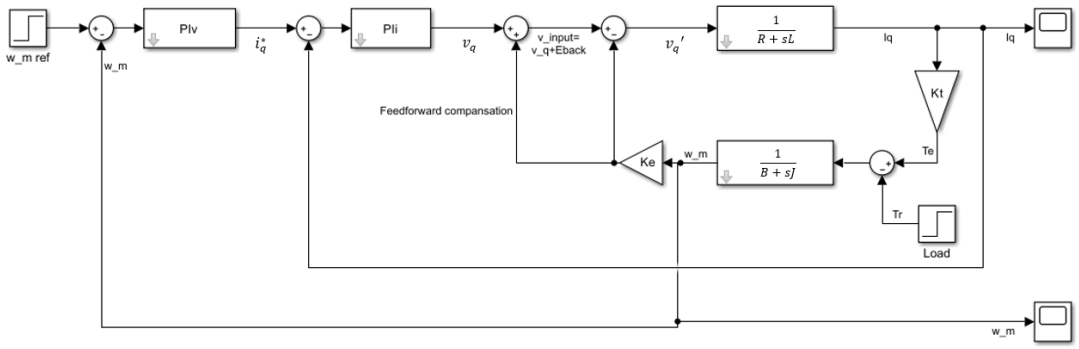

Figure 41 – Simplified model for PI parameter computation

It is possible to compute the closed loop transfer function following the procedure below.

First the parameter of the *EMJ-04APB22* motor have been identified:

Table 3 – EMJ-04APB22 parameters

Parameter	Value	Unit of measurement
r_s	2.35	Ω
L_q	6,5 10^{-3}	mH
L_d	6,5 10^{-3}	mH
λ_f	7.846 10^{-2}	Wb
J	3,169 10^{-5}	$kg\ m^2$
B	52.79 10^{-6}	Ns/m
P	8	number of poles

From the above parameter it is possible to compute the torque constant and the back emf constant, considering:

$$T_e = K_t\, i_q \quad and \quad EMF_{back} = K_e\, \omega_m \tag{3.2}$$

Then, from equations (2.29), (2.33):

$$K_t = \frac{3}{2}\frac{P}{2}\,\lambda_f \qquad K_e = \frac{P}{2}\lambda_f \tag{3.3}$$

Considering the plant transfer function:

$$G_c(s) = \frac{i_q}{v_q{}'} = \frac{1}{R + sL} \tag{3.4}$$

where $L = L_q = L_d$ and $R = r_s$

Given the transfer function related to the PI current controller:

$$PI_c(s) = \frac{K_p \cdot s + K_i}{s} \tag{3.5}$$

It is possible to derive the current open loop transfer function:

$$L_c(s) = PI_c(s) \cdot G_c(s) = \frac{K_p\left(s + \frac{K_i}{K_p}\right)}{s} \cdot \frac{1}{L\left(\frac{R}{L} + s\right)} \tag{3.6}$$

Imposing:

$$\frac{K_i}{K_p} = \frac{R}{L} \tag{3.7}$$

The current close loop transfer function is computed:

$$W_c(s) = \frac{L_c(s)}{1 + L_c(s)} = \frac{\frac{K_p}{L} \cdot \frac{1}{s}}{1 + \frac{K_p}{L} \cdot \frac{1}{s}} = \frac{K_p}{L} \cdot \frac{1}{s} \cdot \frac{1}{\frac{L \cdot s + K_p}{L \cdot s}} = \frac{K_p}{K_p + L \cdot s} \tag{3.8}$$

The procedure is repeated for what concerns the speed loop obtaining the following transfer functions:

$$G_v(s) = W_c(s) \cdot K_T \cdot \frac{1}{B + sj} \tag{3.9}$$

$$PI_v(s) = \frac{K_{pc} \cdot s + K_{iv}}{s} \tag{3.10}$$

$$L_v(s) = PI_v(s) \cdot G_v(s) = \frac{K_{pv}\left(s + \frac{K_{iv}}{K_{pv}}\right)}{s} \cdot \frac{K_p}{K_p + L \cdot s} \cdot K_T \cdot \frac{1}{B + sJ} =$$

$$= K_{pv} \frac{s + \frac{K_{iv}}{K_{pv}}}{s} \cdot \frac{K_p}{K_p + L \cdot s} \cdot K_T \cdot \frac{1}{J\left(\frac{B}{J} + s\right)} \tag{3.11}$$

Imposing:

$$\frac{K_{iv}}{K_{pv}} = \frac{B}{J} \tag{3.12}$$

The speed open loop transfer function becomes:

$$L_v(s) = K_{pv} \cdot \frac{K_p}{K_p + L \cdot s} \cdot \frac{K_T}{J} \cdot \frac{1}{s} \tag{3.13}$$

Computing the speed closed loop transfer function the following result is obtained:

$$W_v(s) = \frac{L_v(s)}{1 + L_v(s)} = \frac{\frac{K_{pv} \cdot K_T \cdot K_p}{s \cdot J} \cdot \frac{K_p}{(K_p + Ls)}}{\frac{j(K_p + L \cdot s) \cdot s + K_{pv} \cdot K_T \cdot K_p}{j(K_p + Ls) \cdot s}} = \frac{K_{pv}K_TK_p}{K_{pv}K_TK_p + JK_p \cdot s + JL \cdot s^2}$$

$$W_v(s) = \frac{1}{s^2 \frac{JL}{K_{pv}K_TK_p} + s \cdot \frac{J}{K_{pv}K_T} + 1} \tag{3.14}$$

It is possible to see that a second order closed loop transfer function is obtained. Considering the prototype second order transfer function:

$$W_p(s) = \frac{1}{s^2 \frac{1}{{w_n}^2} + s \cdot \frac{2\zeta}{w_n} + 1} \tag{3.15}$$

Knowing the objective settling time and overshoot related to a step response it is possible to compute the dumping factor ζ and the natural frequency w_n of the second order system with the well known formulas:

$$\zeta = \frac{\left|\ln\left(\frac{\hat{s}}{100}\right)\right|}{\sqrt{\pi^2 + \ln^2\left(\frac{\hat{s}}{100}\right)}} \tag{3.16}$$

$$w_n = -\frac{\ln\left(\frac{\alpha}{100}\right)}{\zeta \cdot t_s} \tag{3.17}$$

Where:

$\hat{s} = overshoot\ \%$ and $t_{s,\alpha\%} = setting\ time$

Once this two parameters are defined the PI parameters can be computed combining (3.14) and (3.15).

From:

$$w_n{}^2 = \frac{K_{pv}K_T K_p}{JL} \quad \text{and} \quad \frac{2\zeta}{w_n} = \frac{J}{K_{pv}K_T}$$

$$K_{pv} = \frac{J}{K_T} \cdot \frac{w_n}{2\zeta} \tag{3.18}$$

$$K_p = \frac{w_n{}^2 \cdot J \cdot L}{K_{pv} \cdot K_T} \tag{3.19}$$

Recalling (3.7) and (3.12):

$$K_i = K_p \frac{R}{L} \tag{3.20}$$

$$K_{iv} = K_{pv} \frac{B}{J} \tag{3.21}$$

Imposing:

$\hat{s} = 5\%$ and $t_{s,5\%} = 0.02\ s$

The following PI parameters are obtained:

$K_p = 2.99$ $K_i = 1081$

$K_{pv} = 0.0163$ $K_{iv} = 0.0271$

The parameters are imposed both in the simplified model (Figure 41) and in the complete model in Figure 38 (setting the sample time of the discrete PI blocks of the complete model to sample_time=ts).

Then the simulation results are compared. Additional simulation parameter must be fixed before starting. The Vdc voltage, which depends on the inverter capabilities and on the voltage supplier, has been fixed in this case to 282.84 V since on the motor it is indicated the line-line rms voltage rating $V_{ll,RMS} = \frac{V_{dc}}{\sqrt{2}}$. The step time if fixed to step_time=ts/N/2 where ts and N are the parameter introduced in chapter 3.2.

In the first simulation the load torque is considered T_L=0 Nm. The speed and the current PI are triggered at the same frequency f=1/ts.

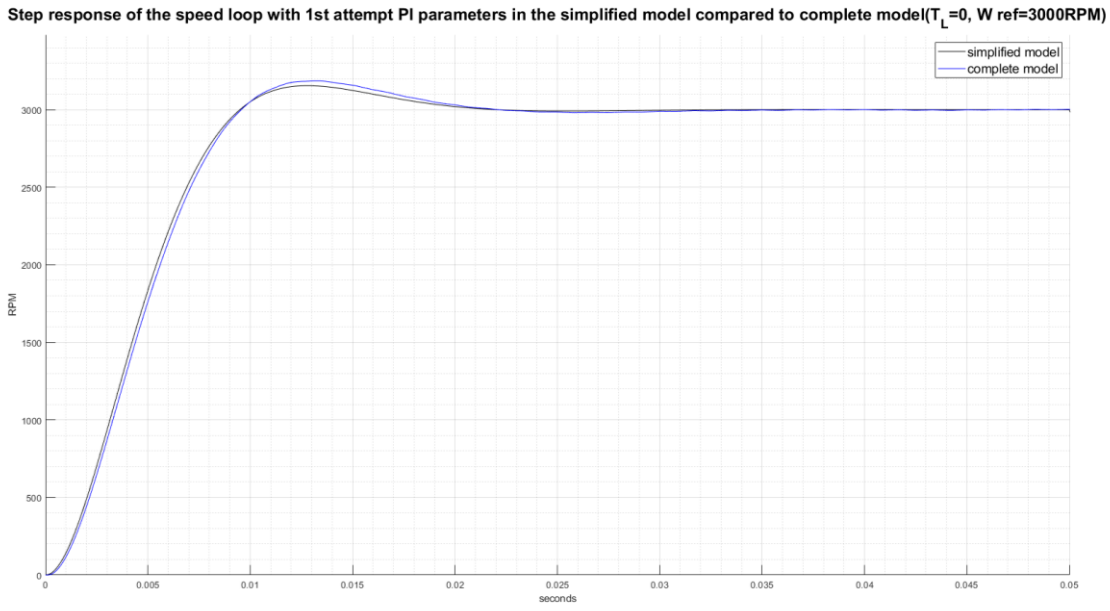

Figure 42 - Speed step response of both simplified and complete model with Wref=3000RPM and Tload=0 Nm (1st attempt)

It is possible to notice that in the complete model, due to the non-idealities introduced by the inverter and by the discrete PI blocks a slightly increased overshoot is obtained.

In order to reduce the overshoot in both models the settling time of the current loop is reduced of 10 times increasing the PIc parameters:

$K_{p2} = 6\,K_p \qquad K_{i2} = 6\,K_i$

The following graph compares the step responses of the current loop obtained before (blue line) and after (black line) the change of the parameters.

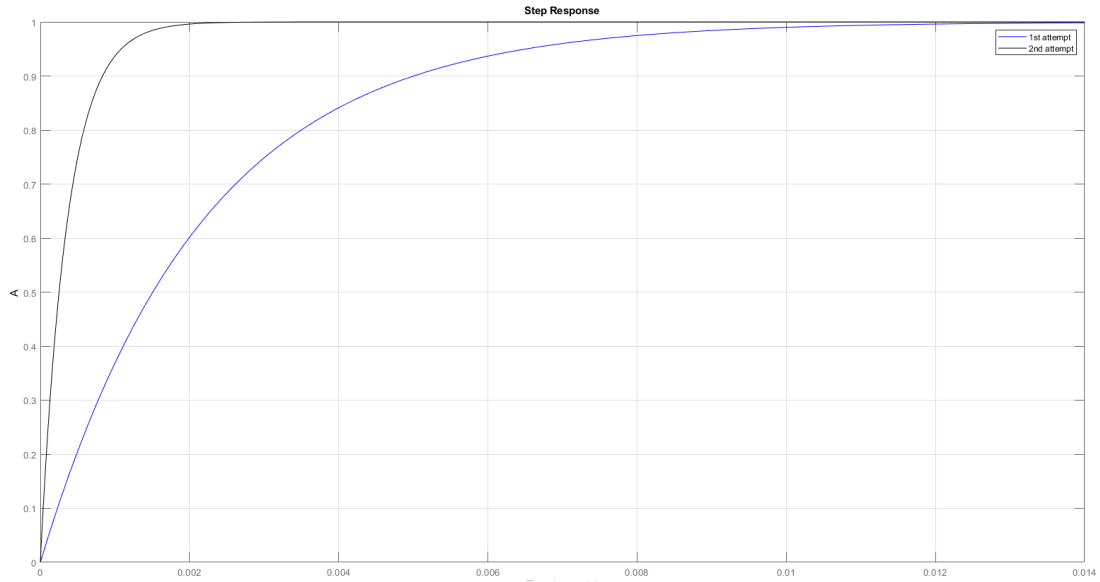

Figure 43 - Current srep response before and after changing parameters

Now the speed loop should behave like a first order system since the reference current i^*_q is tracked at a very high speed.

In order to verify the behavior of the new system under load condition a step load torque is introduced at time t=0.05 of amplitude $T_{load} = T_{rated} = 1.27N$The result of the simulation is the following:

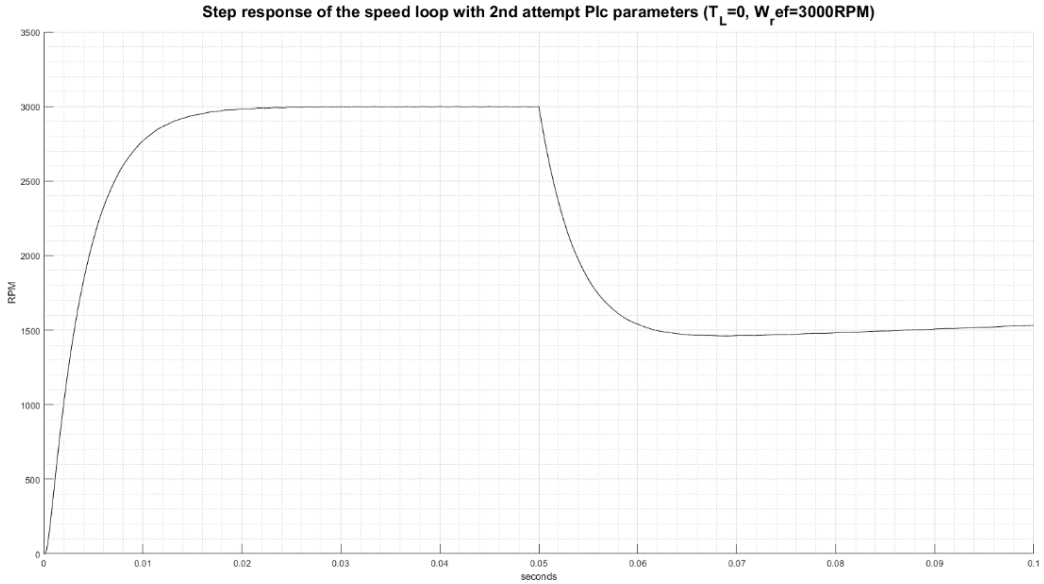

Figure 44 – Speed step response of the complete model with 2nd attempt parameters

As expected, the speed response at motor start is the one of a first order system, with no overshoot, however the load torque introduces a speed error too high, which is not acceptable, and that is rejected too slowly.

In order to improve the performance, the PIv parameters are modified by trial and error procedure noticing the following properties:

- with high K_{iv} , when K_{pv} is increased, the overshoot at starting and the undershoot due to the load torque decrease, while the time to reach again w_{ref} after the load torque is applied increases.

- increasing K_{iv} the disturbance introduced by the load torque is rejected faster but the overshoot at starting increases

In order to reach good performance in terms of time to reject the load torque disturbance and, at the same time, to reduce the undershoot caused by that, Ki and Kp are set equal to the values computed in the second attempt ($K_{p2}\ and\ K_{i2}$) while the following PIv parameters are found:

$K_{pv2} = 5\ K_{pv}$ and $K_{iv2} = 1000\ K_{iv}$

The simulation results regarding both the speed and the torque output of the model can be seen in the two charts in the next page (Figure 46-Figure 45).
For what concerns speed, the high overshoot in starting condition is compensated by a low settling time ($t_{s,5\%} = 9.65\ ms$) and by good performance in terms of rejection of the disturbance introduced by the load torque. After the load torque is applied the reference speed is reached again in less than 10 ms. Looking at the torque chart (obtained computing the mean value of Te setting "ts" as integration time) it is possible to see that the torque produced at steady state by the motor is slightly greater than the load torque because it has also to compensate the friction forces.
In Figure 47 it is possible to notice the speed oscillation at frequency $f_s = \frac{1}{ts} = 10\ kHz$ due to the high order harmonic contribution introduced in the current waveform by the inverter switching frequency.

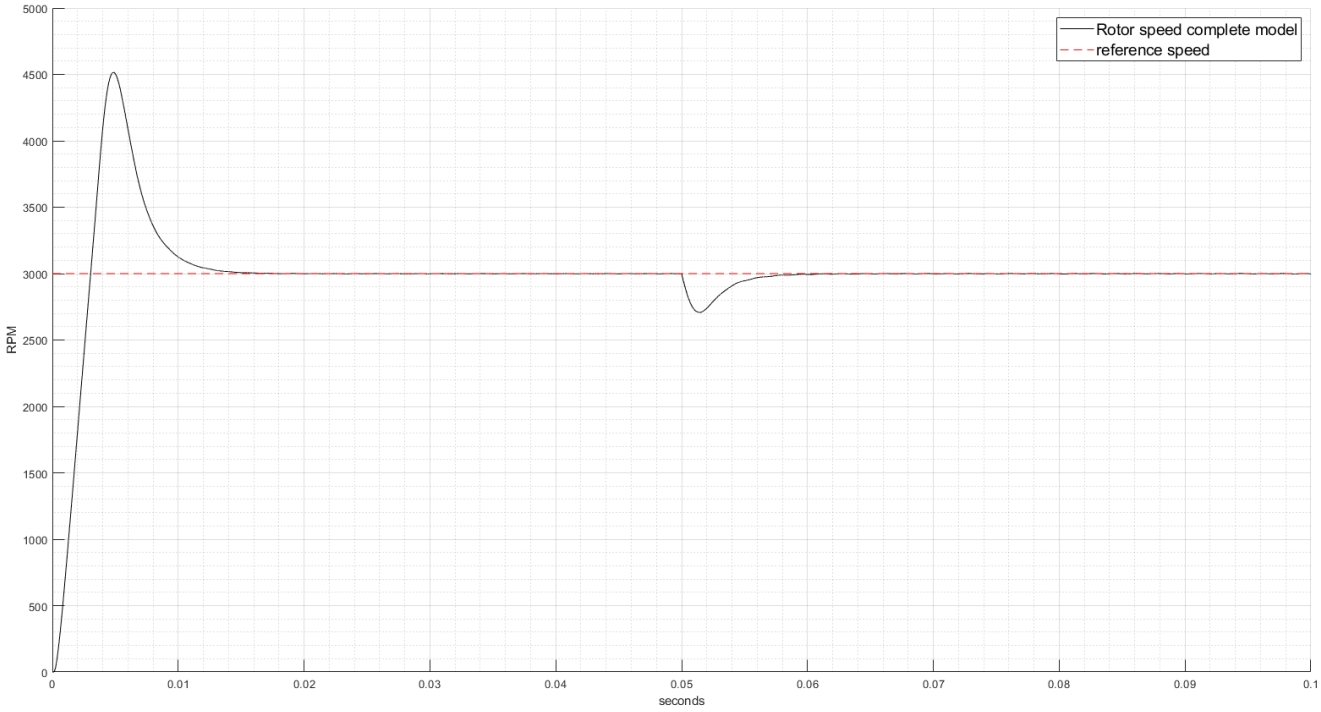

Figure 46 – Speed output of the complete model with 3rd attempt parameters

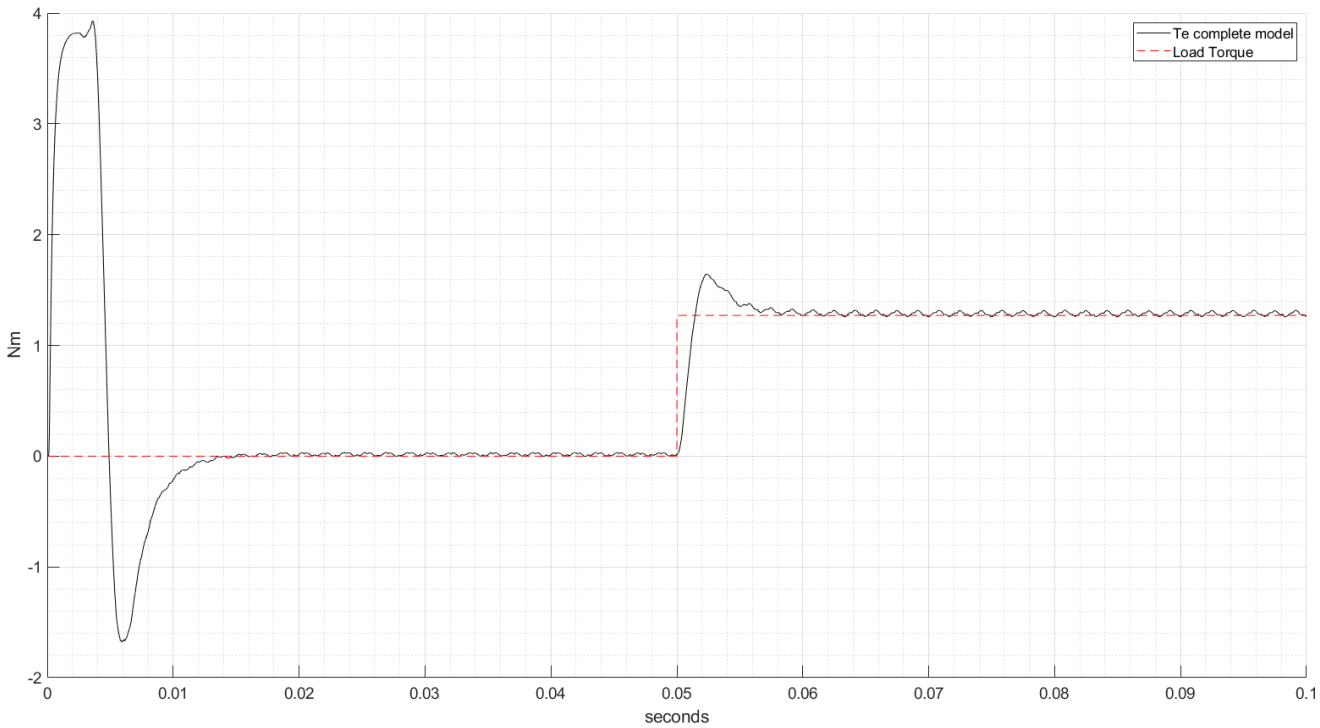

Figure 45- Torque output Te of the complete model with 3rd attempt parameters

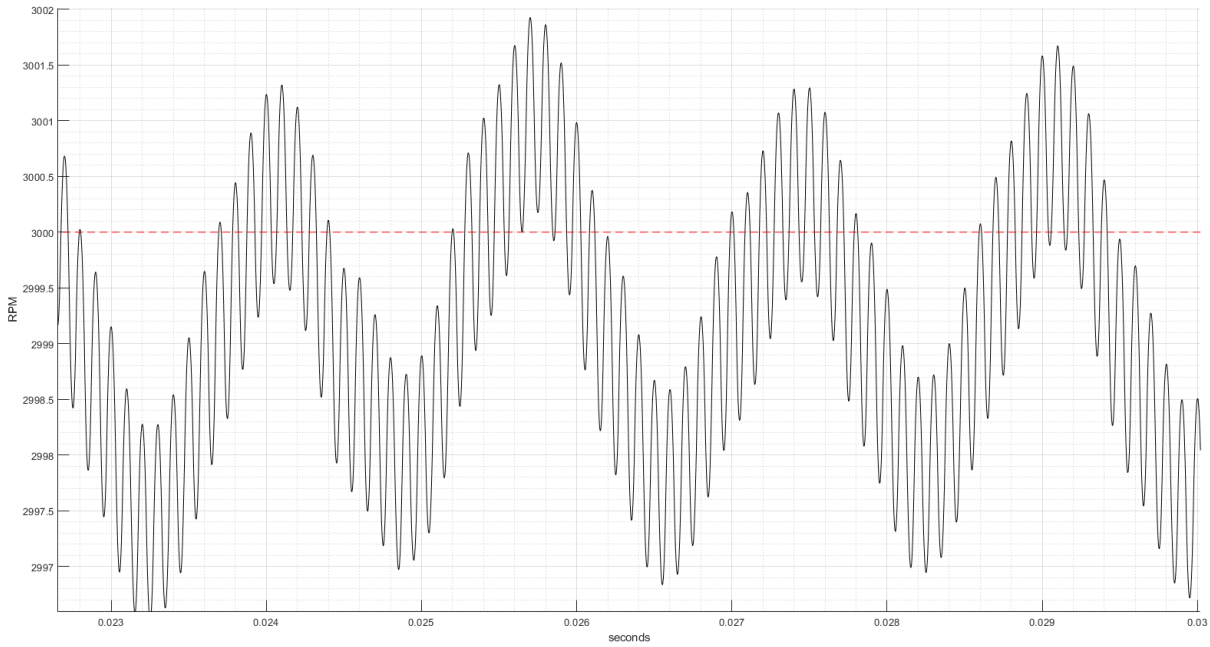

Figure 47 - Speed ripple

The inverter output was not filtered during simulation in order to highlight the controller influence on speed output and make a fair comparison between the different parameter choices.

3.5 H infinity controller design

In order to enhance performance, it has been decided to design a new discrete controller that will substitute the PI controller in the speed loop following the Hinfinity based approach (through the use of the related Matlab toolbox).

The Hinfinity design is based on an algorithm that takes as input the inverse of the objective sensitivity and complementary sensitivity functions, as weighting functions, and returns the controller suitable to generate with good approximation the functions given as input.

Let's consider the following scheme:

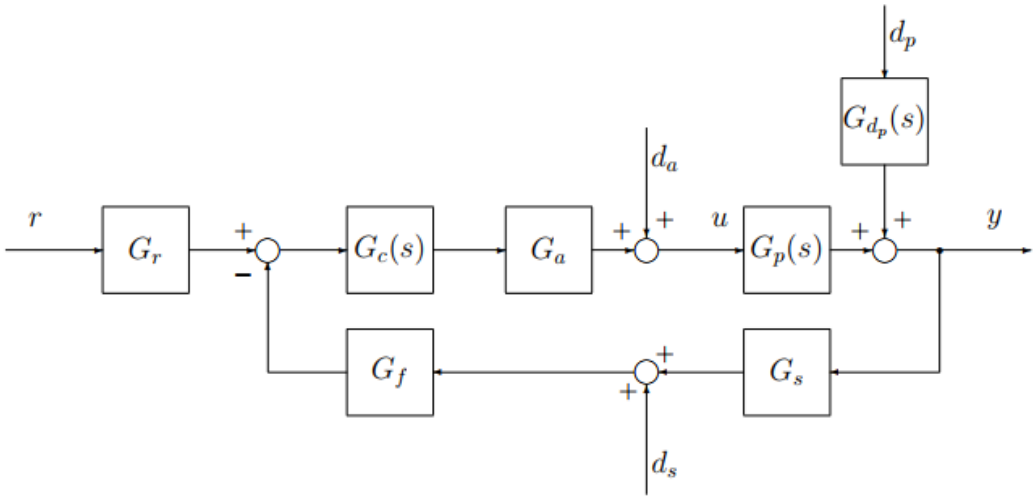

Figure 48 - H_infinity control scheme

The open loop transfer function is given by:

$$L(s) = G_c G_a G_p G_s G_f \tag{3.22}$$

The sensitivity function can be computed as:

$$S(s) = \frac{1}{1+L} \tag{3.23}$$

While the complementary sensitivity function is given by:

$$T(s) = \frac{L}{1+L} \tag{3.24}$$

The first thing to do is to define the requirements the closed loop system should fulfill.

In order to obtain a close loop behavior similar to the one obtained with the PI controller, and make a fair comparison, the main objective defined is to reject in the best way possible the disturbance introduced in the speed output by the load torque. A step signal that raises from 0 to Trated = 1.27 Nm at time t=0.05s has been considered.

In order to reduce the time needed to reach again the reference speed after the load torque is applied (which will be called from now on "t_load_rise"), it has been chosen for the design of the controller a low rise time :

$$\text{t\_rise} = 2.5\ 10^{-3}\ \text{s} \tag{3.25}$$

Then a suitable overshoot and settling time are defined:

$$\hat{s} = 5\% \tag{3.26}$$

$$t_{settling} = 0.02\ s \tag{3.27}$$

From (3.25(3.27) it is possible to find the natural frequency and the damping factor of a prototype second order system that fulfill the given requirements. The following formulas are applied:

$$\zeta \geq \frac{\left|\ln\left(\frac{\hat{s}}{100}\right)\right|}{\sqrt{\pi^2 + \ln^2\left(\frac{\hat{s}}{100}\right)}} \tag{3.28}$$

$$w_{n\ settling\ 5\ \%} = \frac{4.6}{\zeta\ \ t_{settling,5\%}} \tag{3.29}$$

$$w_{n\ rise} = \frac{\pi - \cos^{-1}(\zeta)}{t_{rise}\sqrt{1-\zeta^2}} \tag{3.30}$$

$$w_n \geq \max\left(w_{n\ settling\ 5\ \%}, w_{n\ rise}\right) \tag{3.31}$$

The following results are obtained:

$w_{n\ settling\ 5\ \%} = 333.2818$

$w_n = w_{n\ rise} =$ 1.2892e+03

$\zeta =$ 0.6901

In order to define the steady state requirements, the PMSM control model must have the same shape of the one in Figure 48. A possible scheme is proposed in the next page.

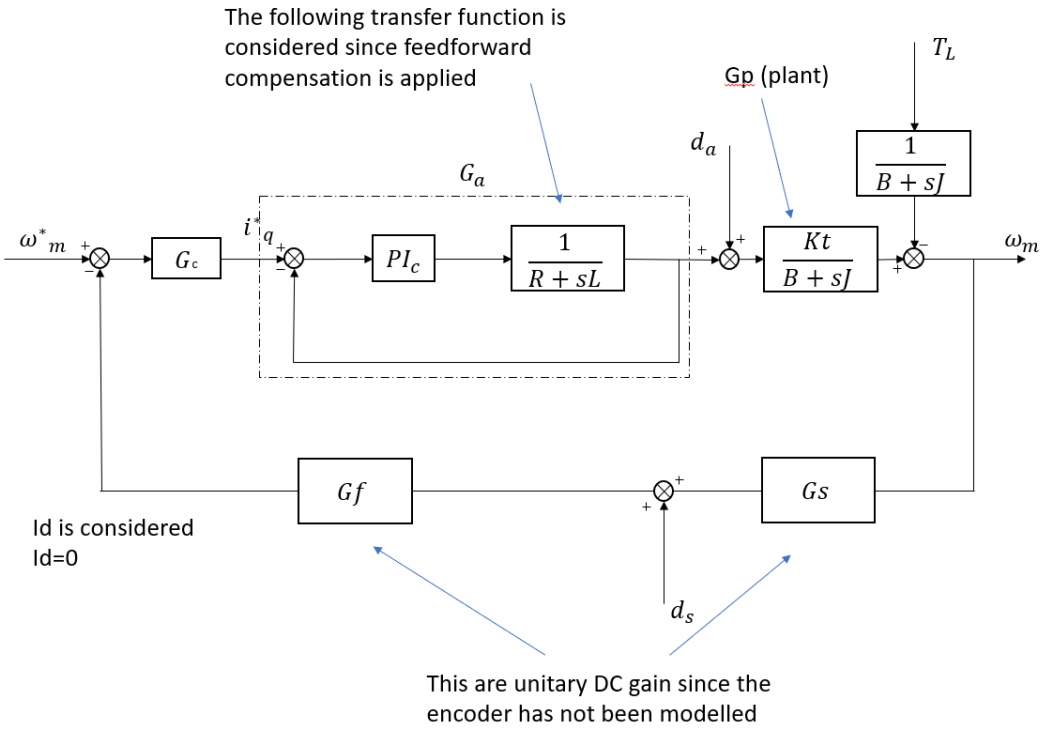

Figure 49 - PMSM control scheme for H_infinity control design

The current closed loop is considered as the actuator transfer function, the plant is given by the transfer function related to the mechanical equation (2.36). The mechanical load is considered as a filtered disturbance added to the output. The Gf and Gs transfer function can be modelled according to the encoder actually used, in this case, since an ideal encoder is considered, they are just unitary gains.

It is possible to derive a list of constraints from the steady state requirements.

1 – Steady state gain Kd=1 with step reference

$$\lim_{s\to 0} s\,\frac{1}{s}\,\frac{G(s)}{1+H\,G(s)} = 1 \tag{3.32}$$

Where $G(s) = G_c G_a G_p$ and $H = G_s G_f$

Since we expect G(s) to have poles in 0 for $s \to 0$ $G(s) \to \infty$, equation (3.32) becomes:

$$\frac{1}{H(s)} = \frac{1}{G_s G_f} = 1 \quad \rightarrow \quad G_s = 1 \tag{3.33}$$

2 – Steady state error = 0 in presence of both step and ramp reference: $\frac{R_0}{s^2}$

$$e_r(s) = \lim_{s\to 0} s \ \frac{R_0}{s^2} \ S^*(s)\, s^{v+p} = 0 \tag{3.34}$$

Where:

$S(s) = S^*(s)\, s^{v+p}$,

$v = poles\ in\ 0\ of\ the\ controller\ G_c$

$p\ = poles\ in\ 0\ of\ the\ plant\ G_p = 0$

From equation (3.34) $v = 2$ is found.

3 – Steady state error = 0 in presence of a ramp or step load torque $T_l(s) = \frac{T_L}{s^2}$

Considering all the other inputs of the system = 0:

$$w_m(s) = \lim_{s\to 0} s \ \frac{T_L}{s^2} \ S^*(s)\, s^v \frac{1}{sJ + B} = 0 \tag{3.35}$$

This requirement is already met since $v = 2$ is imposed from the previous requirement.

For what concerns the current loop the parameters of the PI controller are computed in the following way:

$K_a = L \ \ current\ bandwidth = L\frac{R}{L} = R = 2.35$

$K_b = \frac{R}{L} = \frac{2.35}{0.0065} = 361.54$

$K_p = 5\, K_a = 11.75$

$K_I = 5\, K_a K_b = 4248$

Once the requirements are defined it is possible to start defining the weighting function that will be used to derive a proper controller.

The first weighting function designed is Ws(s).

The inverse of this function must approximate the prototype second order transfer function S2(s) for middle frequencies.

$$S2(s) = s\,\frac{s + 2\,\zeta\,w_n}{w_n^2 + 2\,\zeta\,w_n\,s + s^2} \tag{3.36}$$

Substituting:

$$S2(s) = \frac{s^2 + 1779\,s}{s^2 + 1779\,s + 1.662e06} \tag{3.37}$$

For low frequency it should approximate the following function:

$$S_{low} = s^2\,S0 \tag{3.38}$$

Where S0 is a free parameter in this case.

Furthermore, the requirement on ζ translates in the following requirement:

$$|W_s^{-1}(s)| \leq S_p = 1.28 \quad \forall s \tag{3.39}$$

Where S_p is found from the following chart:

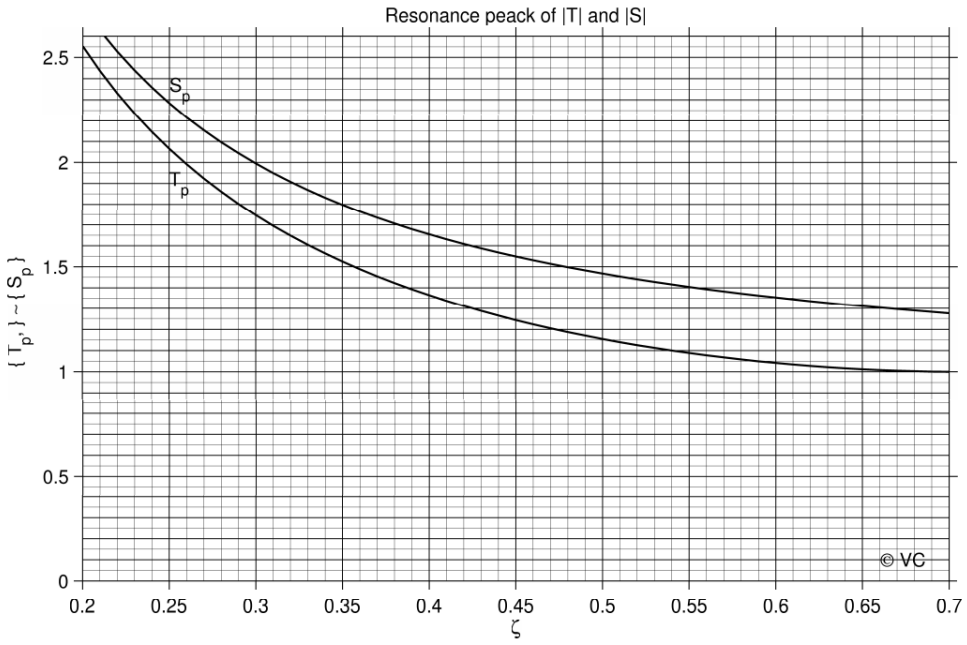

Figure 50 - Requirements on the resonance peak of T and S

The following weighting function is designed:

$$W_s^{-1}(s) = \frac{s^2}{s+1}\, S0\, \frac{1+\frac{s}{w_1}}{1+\frac{1.414}{w_2}s+\frac{s^2}{w_2^2}} \tag{3.40}$$

Where w_1 is a design parameter and:

$$w_2 = \sqrt{w_1 \frac{S_p}{S0}} \tag{3.41}$$

Choosing $w_1 = \frac{w_n}{250}$ and $S0$ =0.00001:

$$W_s^{-1}(s) = 1.28\, \frac{s^2(s+5.157)}{(s+1)(s^2+\ 1149s\ +\ 6.6e05)} \tag{3.42}$$

S_{low}(s), S2(s) (red lines) and $W_s^{-1}(s)$ (black line) bode diagrams are shown in the following figure:

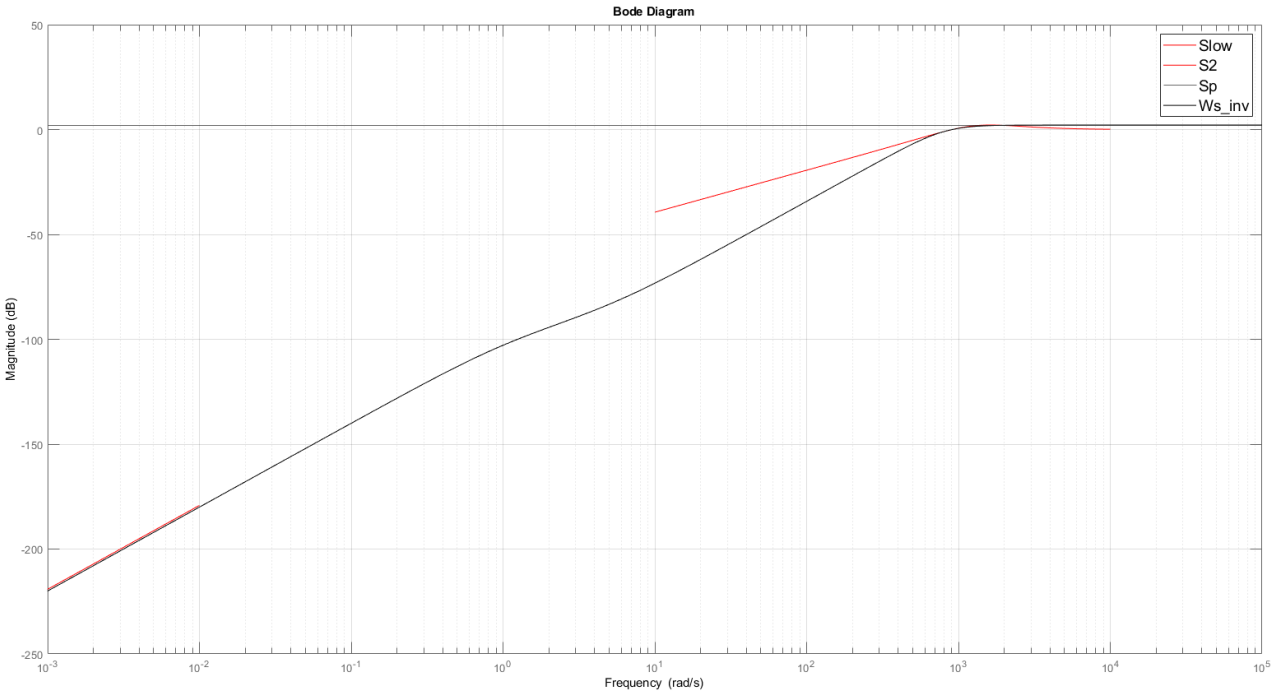

Figure 51 – W_s_inv bode diagram

The second weighting function to design is $W_t^{-1}(s)$. The only requirement is:

$$|W_t^{-1}(s)| \leq T_p = 1 \quad \forall s \tag{3.43}$$

Which comes from the chart in Figure 50.

It should have the shape of a second order transfer function, low pass filter with cutoff frequency comparable to the one of $W_s^{-1}(s)$.

The following transfer function is designed:

$$W_t^{-1}(s) = \frac{Tp}{1 + \left(\frac{s}{w_{nt}}\right)^2} = \frac{5.9829e07}{(s + 7735)^2} \tag{3.44}$$

Where $w_{nt} = 6\, w_n$ and $Tp = 1$.

The following figure shows the bode diagram obtained:

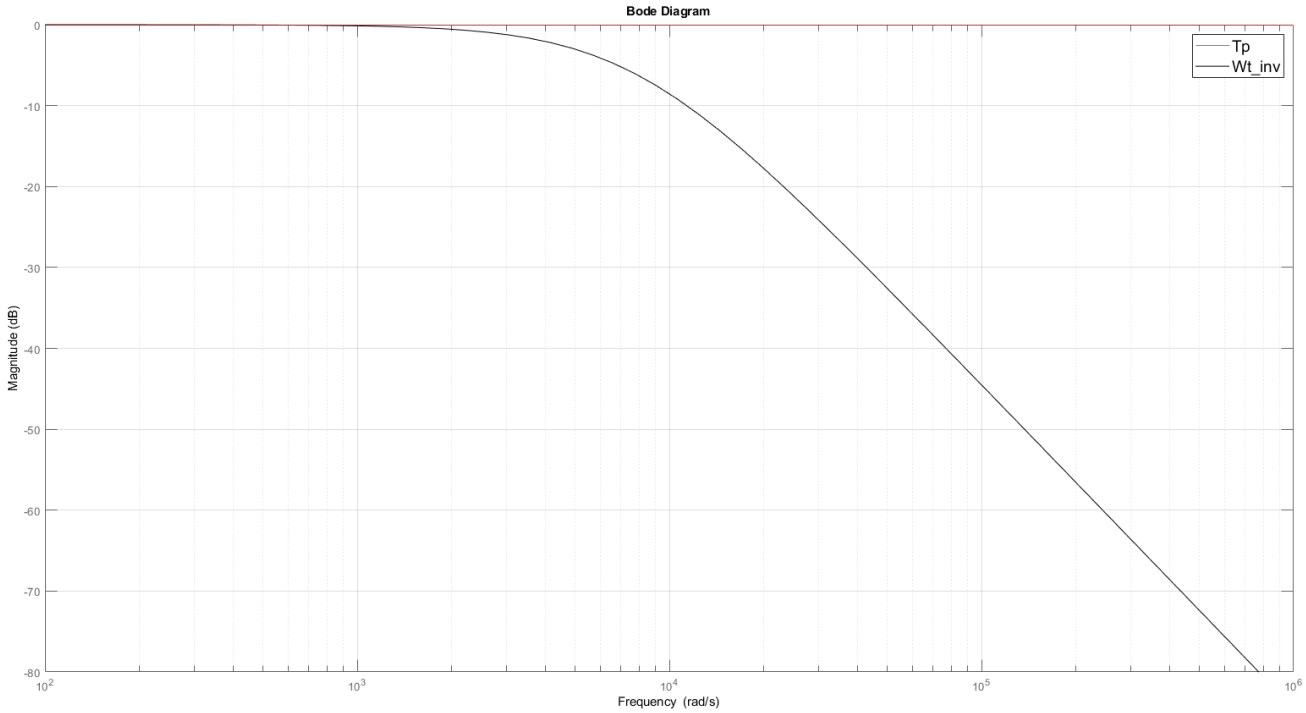

Figure 52 – W_t_inv bode diagram

From equation (3.44), (3.42) W_t *and* W_s are obtained, from these 2 weighting functions, W1 and W2 are derived and then tuned through an iterative procedure until the nominal requirements are met, which happens when:

$$S(s) \leq W_t^{-1}(s) \;\; \forall s \quad \text{and} \quad T(s) \leq W_s^{-1}(s) \;\; \forall s \tag{3.45}$$

Two modified versions of W1 and W2 are used in the Simulink model in Figure 53. This model will be used to derive the structures needed by the MATLAB function that is able to generate a suitable controller.

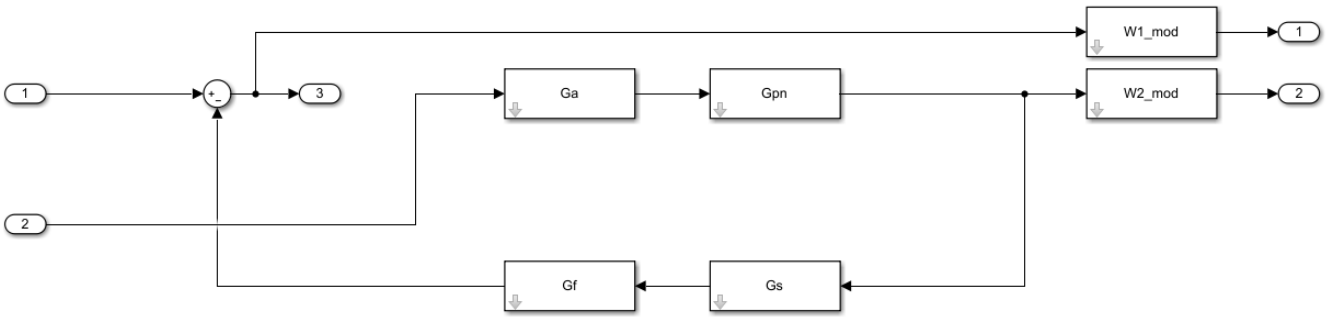

Figure 53 - Simulink model used for Hinfinity design

The W1 and W2 used in this case are obtained from:

$$W1^{-1}(s) = \frac{s^2(s+5.157)}{(s+1)(s^2+1015s+5.157e05)} \tag{3.46}$$

Which is obtained from (3.40)(3.41) changing the Sp parameter to Sp=1.

$$and \qquad W2^{-1}(s) = W_t^{-1}(s) \tag{3.47}$$

Once that W1 and W2 are found they must be modified.

For what concerns W1, the poles in zeros are substituted with low frequency poles in order to avoid computational errors related to the Hinfinity algorithm.

$$W1_{mod} = W1\frac{s^2}{\left(s+\frac{w_n}{10000}\right)^2} \tag{3.48}$$

For what concerns W2 it must be modified in order to obtain a proper transfer function to put in the Simulink model. This means to simplify both the zeros and then add those zeros simplified to the model through the following command lines:

```
[Am,Bm,Cm,Dm] = linmod('H_inf')
M = ltisys(Am,Bm,Cm,Dm);
M = sderiv(M,2,[1/wnt2 1]);
```

Where 'H_inf' is the name of the Simulink model derived.

Thus,

$$W2_{mod} = \frac{W2}{\left(1 + \frac{s}{w_{nt}}\right)^2} = 1 \tag{3.49}$$

The "hinflmi" function takes as argument the structure "M" derived from the Simulink model and generates the controller Gc_mod. This controller must be modified adding a number of poles in 0 equal to the ones removed from W1 (substituting the same amount of low frequency poles). It is also possible to remove or substitute some of the high frequency poles obtained provided that the controller is kept a proper transfer function.

In this case the output of the "hinflmi" function is:

$$Gcmod = \\ = 1.0673e07 \frac{(s + 1216)(s + 400.5)(s + 348.1)(s + 1.673)(s + 1)}{(s + 4.053e06)(s + 1.937e04)(s + 361.5)(s + 5.047)(s^2 - 0.1969s + 0.01386)} \tag{3.50}$$

Which is modified in order to obtain the controller final form Gc:

$$G_c = 1.0876 \frac{(s + 1216)(s + 400.5)(s + 348.1)(s + 1.673)(s + 1)}{s^2(s + 361.5)(s + 8000)(s + 5.047)} \tag{3.51}$$

Then nominal performance are checked comparing the sensitivity and complementary sensitivity functions, computed with the formulas (3.23)-(3.24) that include the controller G_c, with the weighting functions $W_t^{-1}(s)$ and $W_s^{-1}(s)$ as already stated in (3.45).

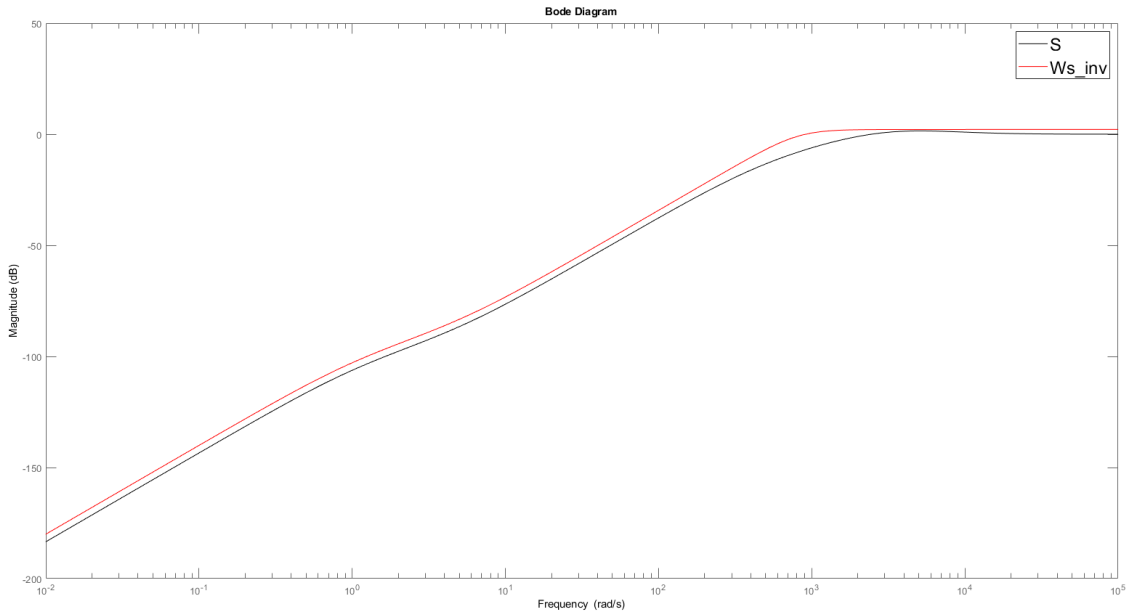

Figure 54 - S and Ws_inv bode diagram comparison

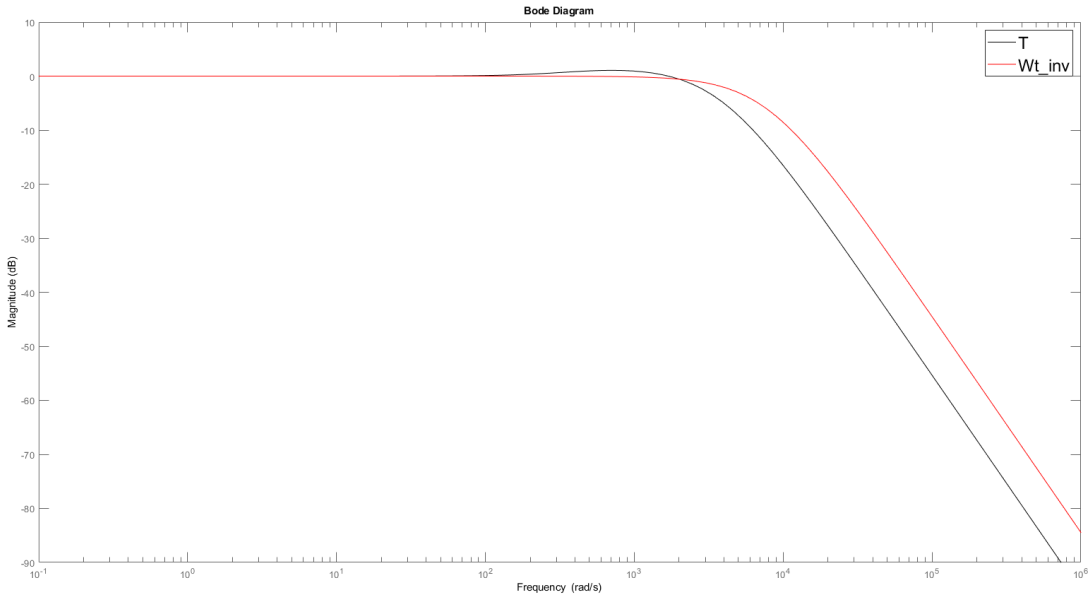

Figure 55 – T and Wt_inv bode diagram comparison

It is possible to notice that the constraint on T is not met, even after tuning several times the weighting functions. This could be due to the requirement defined at the beginning regarding the overshoot. This requirement, that translates first in the definition of the damping factor ζ and thus in the maximum peak of T (T_p), could be too stringent. The performance of the controller obtained and directly tested in the Simulink model shown in Figure 55 and verified through the use of a scope in a simplified model (Figure 49).

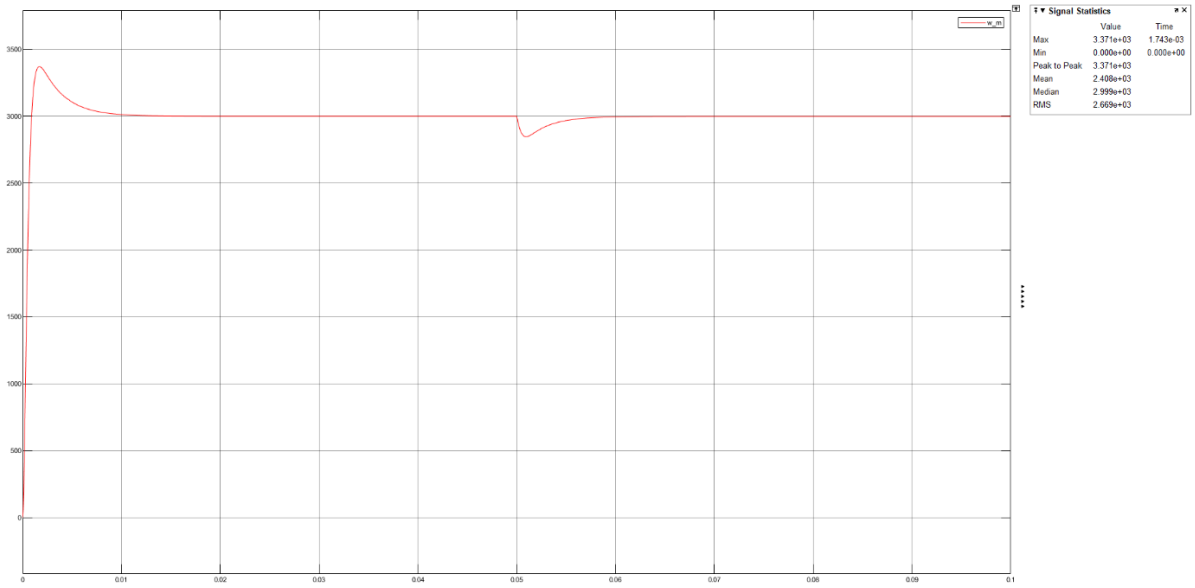

Figure 56 - Ideal step response (3000RPM) with nominal step torque (1.27N) introduced at 0.05 s

The simulation has been carried out with an input step reference of 3000 RPM and a load torque step signal T_L introduced at t=0.05 s.
As expected the overshoot is greater than the one stated in the requirements but still acceptable:

$$\hat{s} = \frac{3371 - 3000}{3000} = 12.37\ \% \tag{3.52}$$

The requirements on rise and settling time are both satisfied:

$$t_{settling,5\%} = 4.16\ 10^{-3} < 2\ 10^{-2} \tag{3.53}$$

$$t_{rise} = 9\ 10^{-4} < 2.5\ 10^{-3} \tag{3.54}$$

The load torque disturbance introduced at t=0.05 s is completely cancelled in less than 10 ms.

Now that a suitable controller has been obtained it must be tested inside the complete model. Its behavior will be compared with the one of the PI controller to understand the advantages of the Hinfinity approach. The first thing to do is to discretize the controller obtained in the continuous time domain. This can be done through the use of the "c2d" Matlab function. Since the function offers difference methods to discretize the continuous transfer function all of them have been tried and the behavior of each controller obtained in this way has been compared.

The discretization method exploited, imposing sampling time equal to *ts* (inverter switching period*),* are:

- ZOH (Zero Order Hold)

-FOH (First Order Hold)

- Tustin

-Matched

-Least-squares

All simulations have been carried both at 300 RPM and 3000 RPM considering a load torque $T_L = 1.27N$ at t = 0.05 s. The following graphs compare the speed responses:

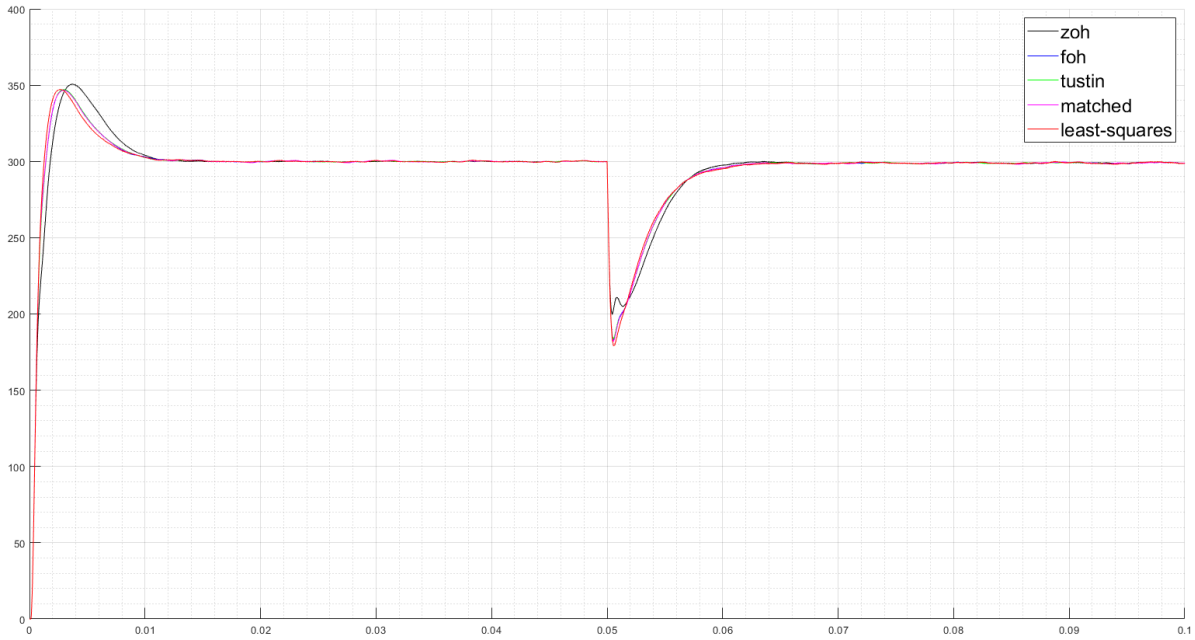

Figure 57 - Speed response (300 RPM step reference) comparison for each descritized controller obtained

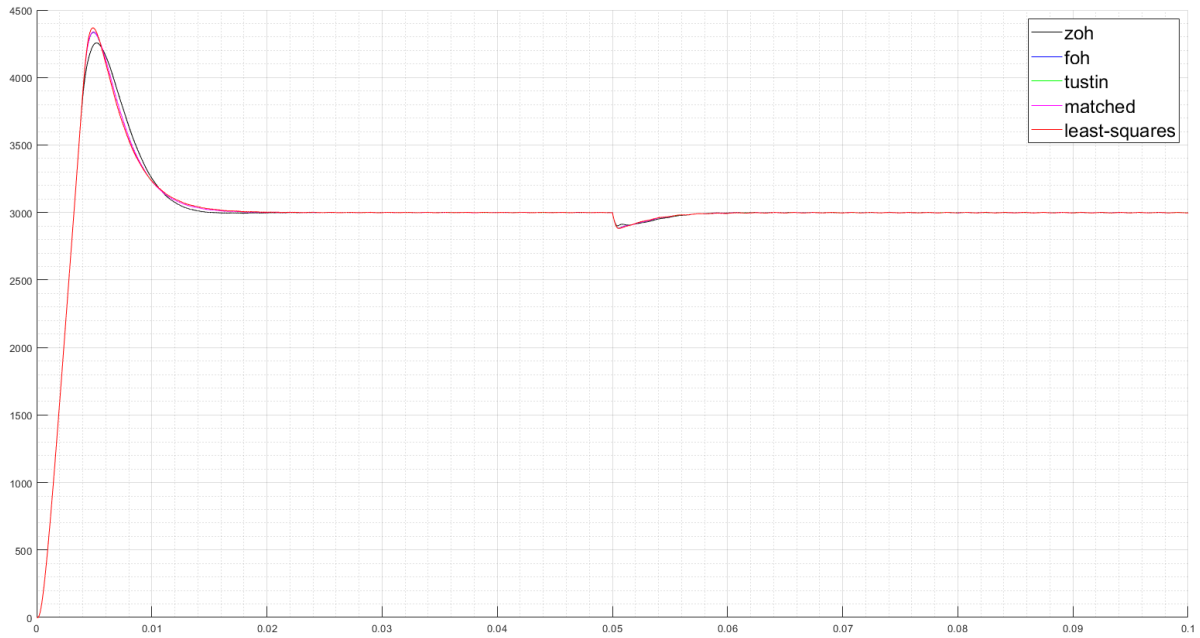

Figure 58 - Speed response (3000 RPM step reference) comparison for each descritized controller obtained

It can be seen that the controller obtained with the ZOH discretization technique shows a better behavior in terms of overshoot for what concerns the speed response with 3000 RPM step reference while at 300 RPM the performance are slightly worst then with the other discretization methods. Anyway, in both cases, the undershoot due to the load torque is lower with the ZOH controller.

In order to get an idea also of the steady state performance, the RMS value have been evaluated both at 300 and 3000 RPM, for each discretized controller obtained, considering the speed values between 0.2 and 0.5 seconds. The following chart compares the value obtained:

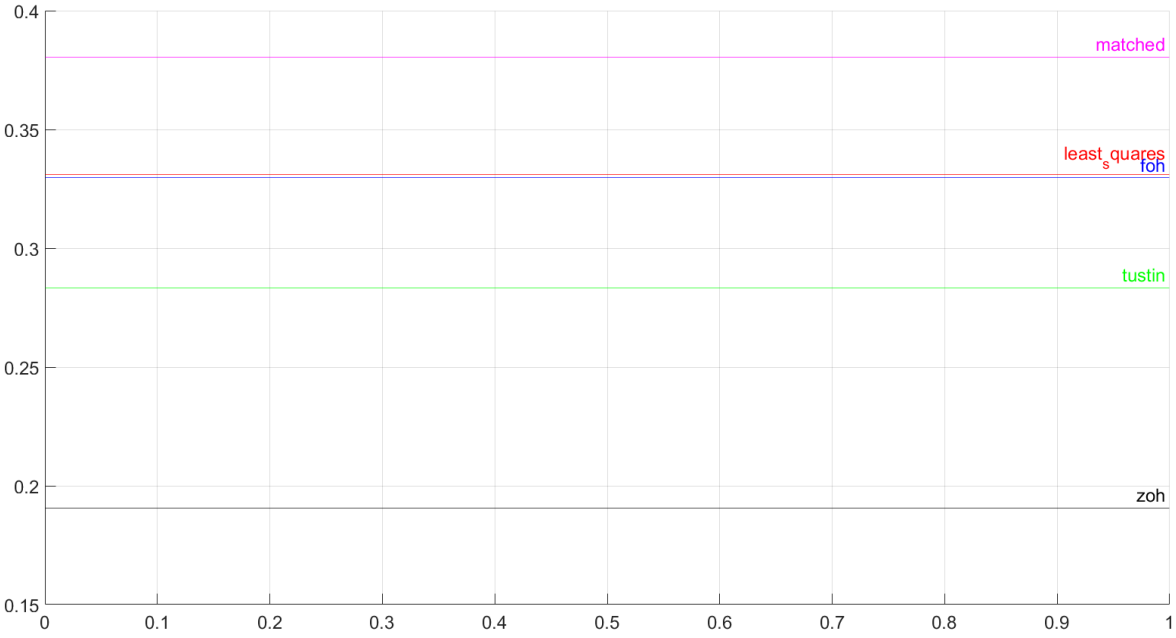

Figure 59 - RMS value comparison (300 RPM)

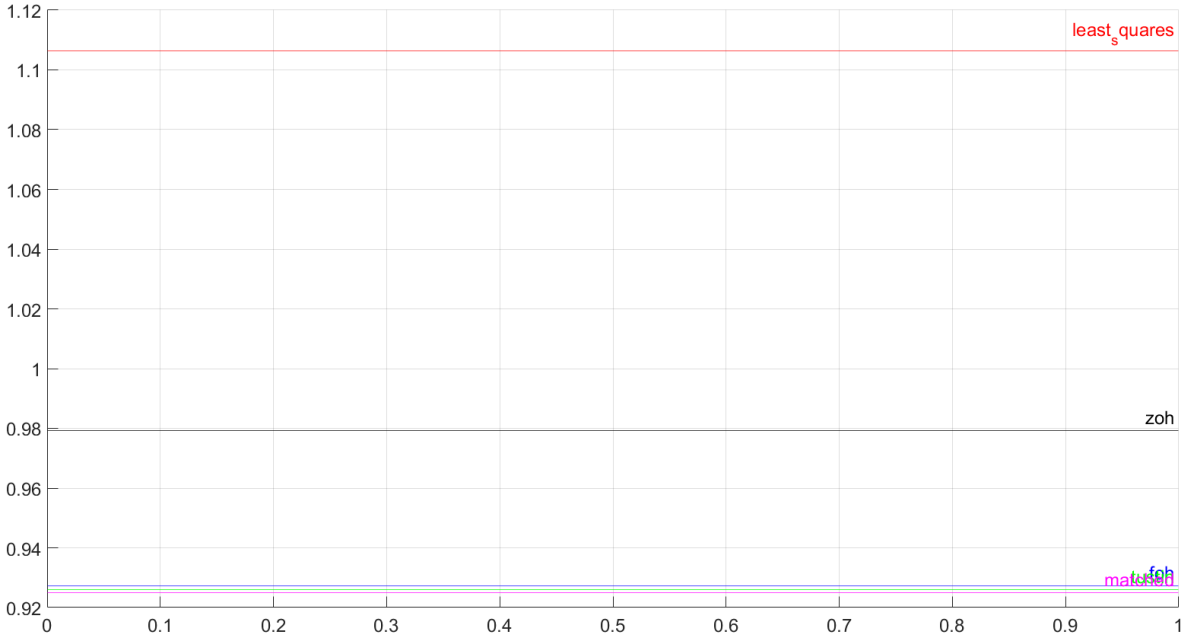

Figure 60 - RMS value comparison (3000 RPM)

From Figure 59 it can be seen that the ZOH controller assure the lowest RMS value at 300 RPM while it's still comparable with the values obtained with the other techniques at 3000 RPM. For the reasons listed above the controller obtained with the ZOH discretization method has been chosen to compare the speed response with the one obtained with the PI controller.

Two simulations have been carried out in order to evaluate the behavior of the system both at high (3000 RPM) and low speed (300 RPM) , the results are shown in the next pages.

As can be seen in Figure 61 and Figure 62 the model that exploit the controller obtained through the Hinfinity approach performs better in terms of overshoot in the speed response and undershoot caused by the load torque applied while the settling time, the rise time and the time to restore the steady state value once the load torque is applied, are all comparable for both the PI and the Hinfinity models.

For what concerns the torque Figure 63 and Figure 64 show how the load torque profile is better followed in the case of the Hinfinity controller based model.

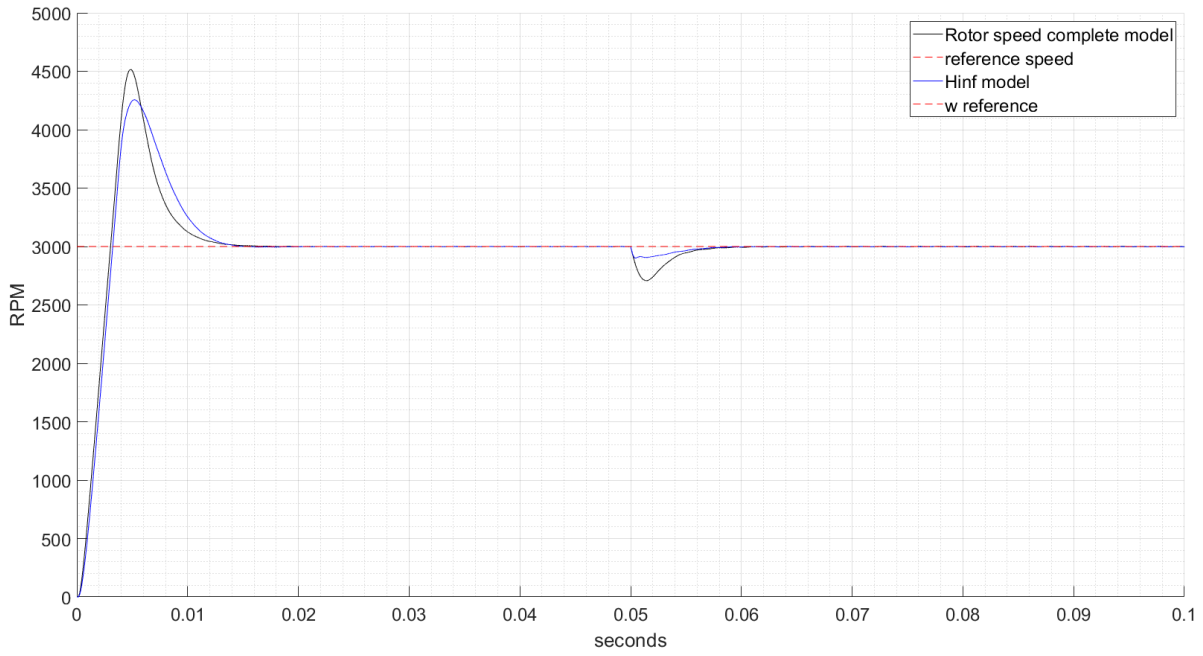

Figure 61 – Speed response comparison between Hinfinity based model and PI based model with 3000 RPM step reference

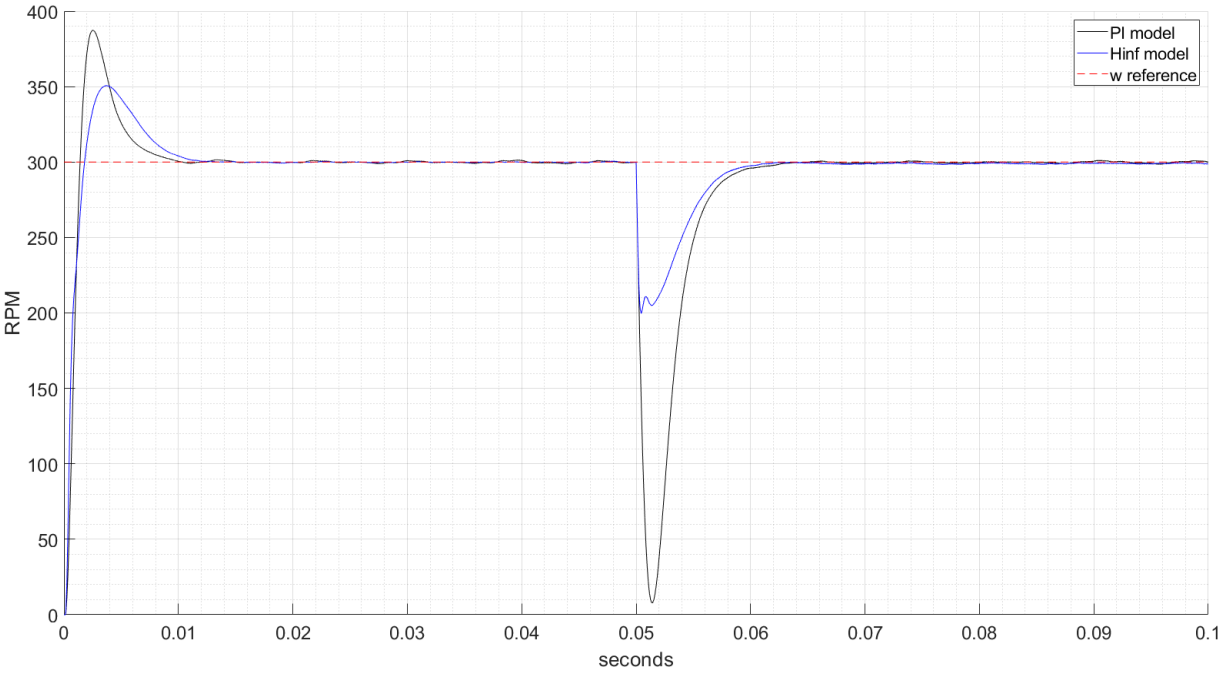

Figure 62 – Speed response comparison between Hinfinity based model and PI based model with 300 RPM step reference

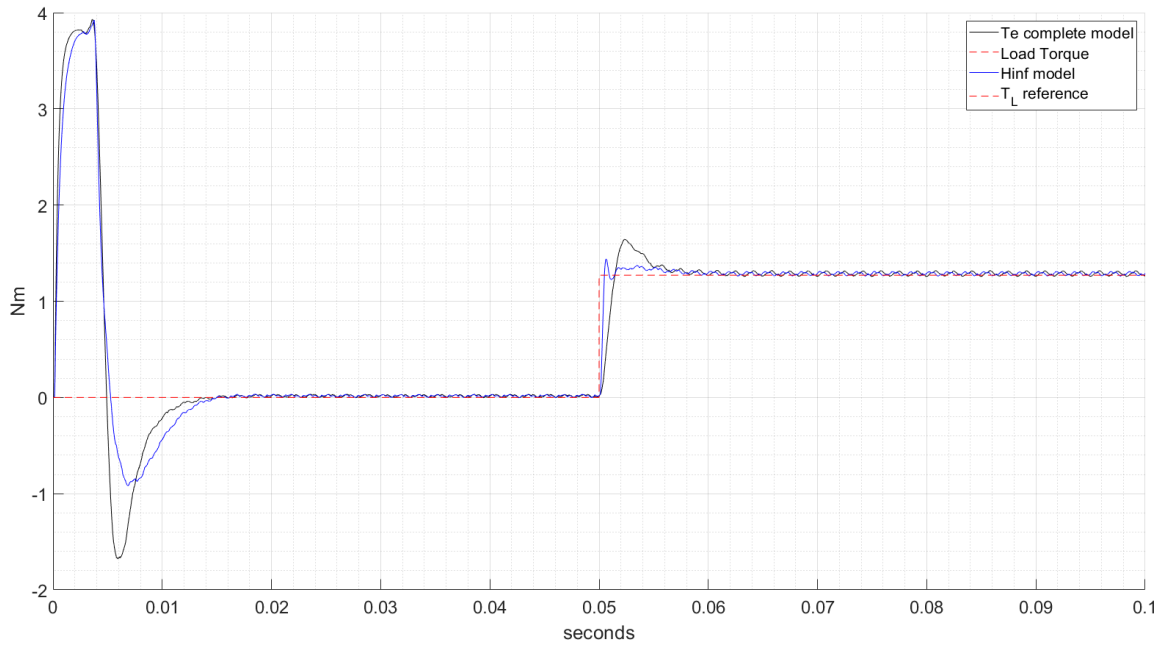

Figure 63 - Torque response (Te) comparison between Hinfinity based model and PI based model with 3000 RPM step reference

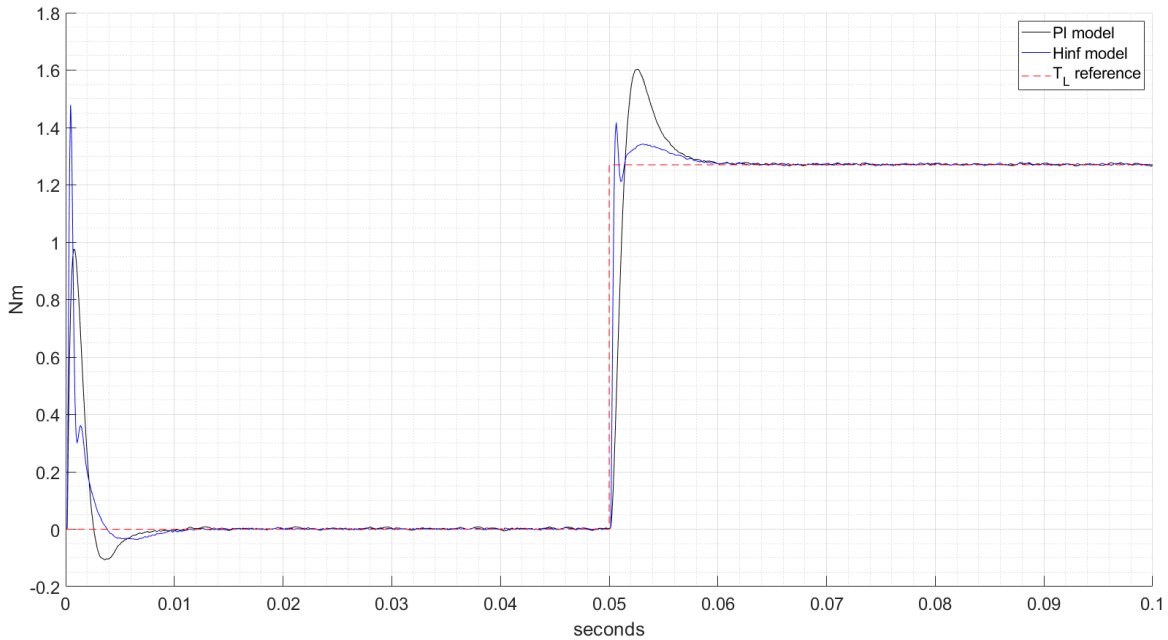

Figure 64 - Torque response (Te) comparison between Hinfinity based model and PI based model with 300 RPM step reference

The advantages of the Hinfinity controller are even greater in the low speed case since the overshoot percentage value is almost half than the one in the PI controller based model:

$$\widehat{s_{PI}} = \frac{387.4 - 300}{300} = 29.1\% \tag{3.55}$$

$$\widehat{s_{Hinfinity}} = \frac{350.7 - 300}{300} = 16.9\% \tag{3.56}$$

Furthermore, the undershoot due to the load torque is less than half than the one produced in the case of the PI control based model, which cause the motor almost to stop.

The overshoot obtained in the case of the Hinfinity based model with step reference speed equal to 300 RPM is slightly higher than the one computed in (3.52) (which was obtained carrying simulation in a simplified model). This could be due to the non-idealities introduced by the space vector modulation algorithm used to control the inverter, by the discretization of both the speed and current loop controllers, and by the saturation block introduced to limit the current reference to 8.1 A, which is the maximum current indicated in the motor datasheet. The influence of these non-idealities becomes even stronger in the case of the high speed reference where the overshoot reaches the value:

$$\widehat{s_{Hinfinity}} = \frac{4257 - 3000}{3000} = 41.9\% \tag{3.57}$$

For what concerns the rise time and settling time they are, in the case of the 3000 RPM reference:

$$t_{settling,5\%} = 1.09\ 10^{-2}\ s \tag{3.58}$$

$$t_{rise} = 3.24\ 10^{-3}\ s \tag{3.59}$$

Greater than the ideal case values but still acceptable.

In the case of the 300 RPM step reference:

$$t_{settling,5\%} = 7.65\ 10^{-3}\ s \tag{3.60}$$

$$t_{rise} = 1.8\ 10^{-3}\ s \tag{3.61}$$

These values are really close to the ones found in the ideal case (3.53 (3.54), which suggest that, the less is the speed, the more the model behavior is closed to the ideal case.

Finally, in order to have a comparison between the PI controllers-based model and the Hinfinity controller based model, also in terms of steady state value, the RMS speed values are compared, both at 300 and 3000 RPM.

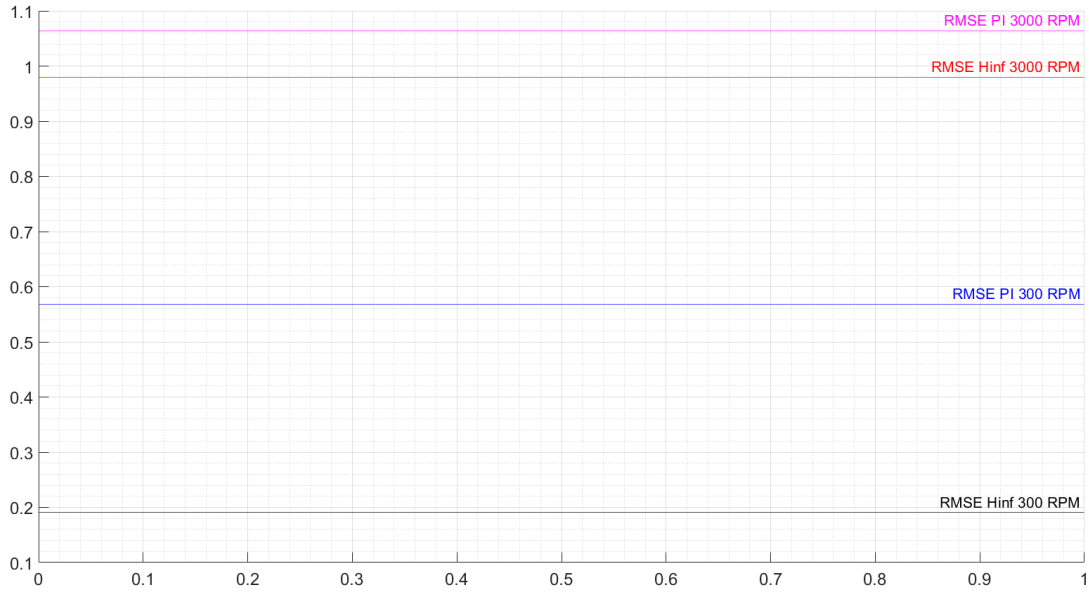

Figure 65 - RMS value comparison

It is well evident that the amplitude of the oscillations around the steady state value is reduced using the Hinfinity controller, and this effect is enhanced when a low speed reference is imposed.

3.6 Flux weakening control implementation

In order to implement the control algorithm needed to work in the flux weakening area it has been made reference to the theory explained in chapter 2.10. The main idea considered in this implementation is to increase the current vector angle Φ when the base speed is reached in order to further increase the motor speed. This is done multiplying the complementary angle $\pi - \Phi$ for a gain β_c. This gain should be 1 when the motor is working below its base speed and should tend to 0 when the motor is reaching its limit speed. The current vector and the complementary angle are represented in Figure 66.

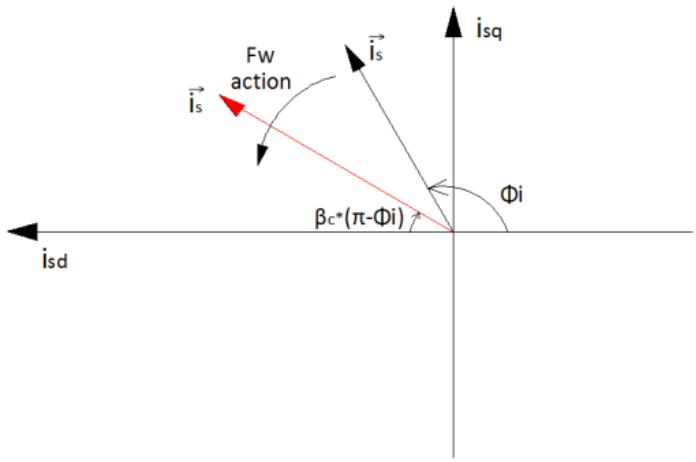

Figure 66 - Current vector representation

In order to generate the β_c reference, an integral action has been implemented that receives as input the difference between the modulation index M and a tunable parameter M* where:

$$M = \frac{V_{ref}}{\frac{Vdc}{\sqrt{3}}} = \frac{\sqrt{V_q^2 + V_d^2}}{\frac{Vdc}{\sqrt{3}}} \tag{3.62}$$

It is known that the reference voltage V_{ref} should reach its absolute value limit at the base speed, when a load torque equal to the rated one is applied. In the case of the *EMJ-04APB22* this value has been computed in chapter 2.10 and it is equal to:

$$V_{ref} = \sqrt{V_q^2 + V_d^2} = \frac{V_{dc}}{\sqrt{3}} = \frac{185.22}{\sqrt{3}} = 106.94\ V \tag{3.63}$$

Which means that the modulation index should be $M = 1$ in the condition just described. Since Φ should start increasing (and thus β_c should start decreasing) when the motor is close to the base speed, the tunable parameter M* is chosen to be:

$$M^* = 0.99 \tag{3.64}$$

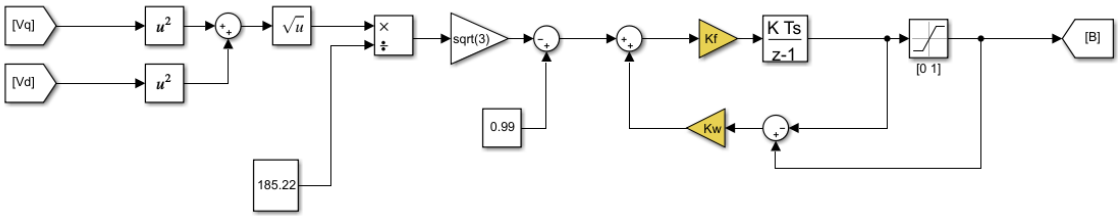

Figure 67 - Bc parameter computation Simulink scheme

Moreover, since the parameter β_c is saturated to 1 as upper limit and to 0 as lower one, an anti-windup back calculation scheme has been designed in Simulink (Figure 67). The Kf and Kw gains have been tuned by trial and error procedure, the final value chosen are listed below:

$$K_f = 2000 \tag{3.65}$$

$$K_w = 0.5 \tag{3.66}$$

The β_c value is sent to the Current loop + Feedforward control subsystem as well as the reference value for the current vector. This value is the output of a discrete PI block that has been tuned with the same values previously used in chapter 3.4 computed with (3.18) and (3.21) formulas. The PI block has been modified adding an anti-windup feature through back calculation technique in order to avoid big overshoots around the base speed that, due to the flux-weakening algorithm, would cause oscillation around the base speed for a while before reaching the steady state value. In this case back calculation gain Kb has been chosen:

$$K_b = 2000 \tag{3.67}$$

Figure 68 represent the speed control loop of Figure 38 updated with the Bc computation and the anti-windup feature.

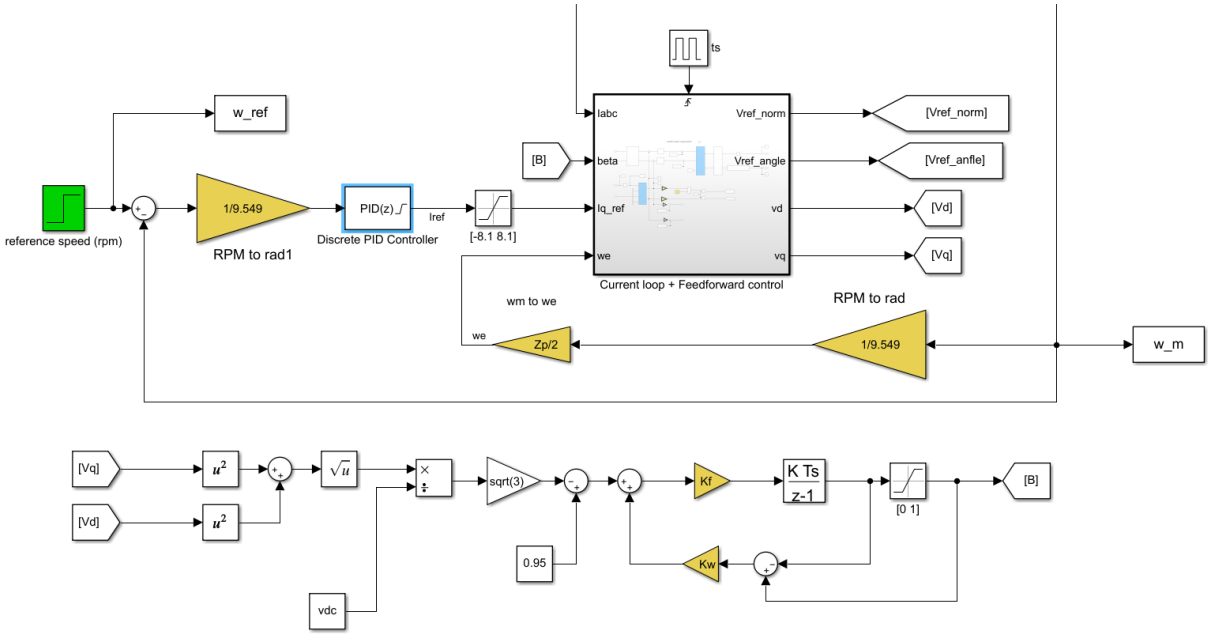

Figure 68 - PI cascade controller updated to operate in the flux-weakening area

Also the current loop has been modified in order to compute the Iq and Id reference current from the new inputs I_{ref} and β_c, and to return as output the V_d *and* V_q voltages, used to compute β_c. The new *"Current loop + Feedforward control"* subsystem is shown in Figure 69.

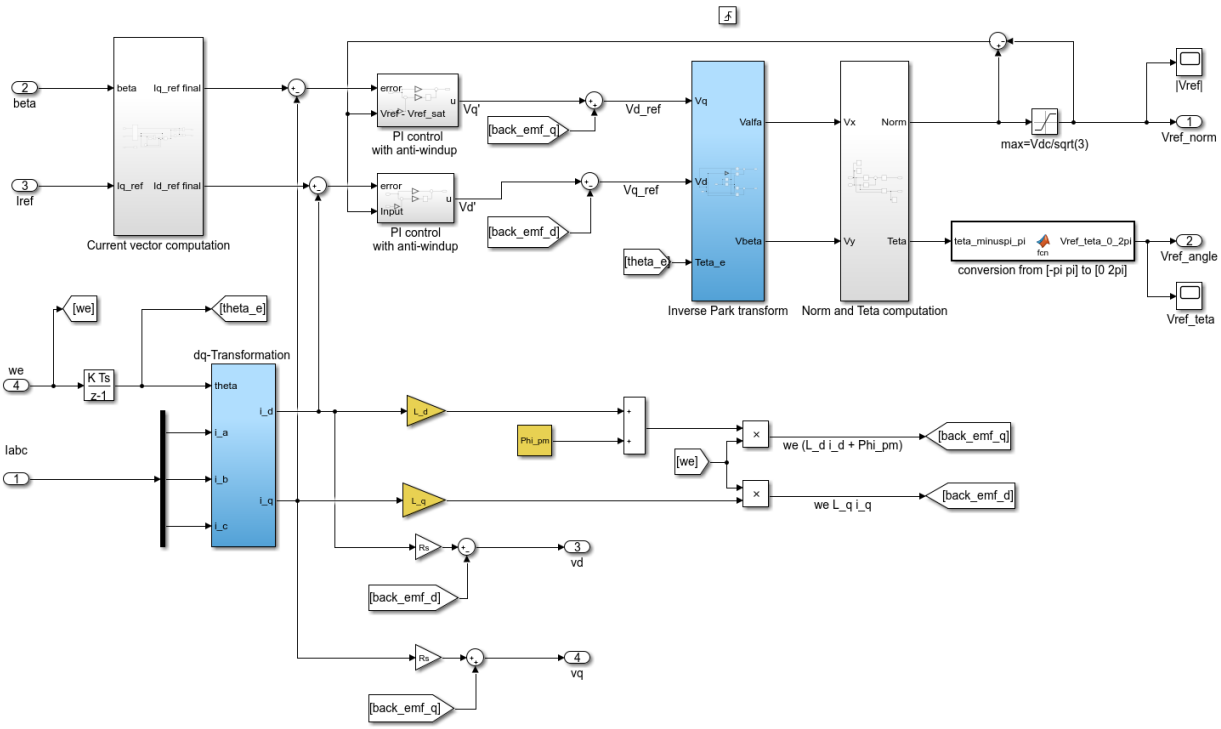

Figure 69 - Current loop + Feedforward control subsystem

I_{ref} and β_c are the inputs of the "*Current vector computation"* subsystem, which is shown in Figure 70.

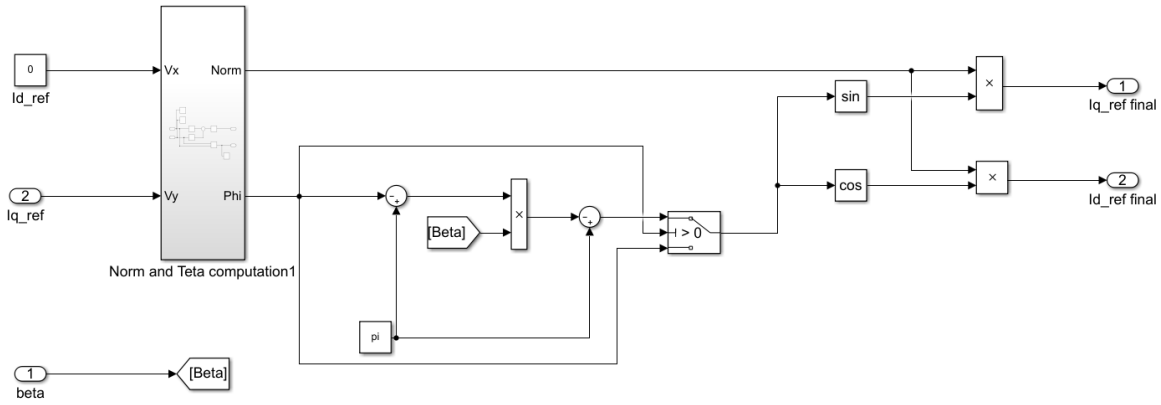

Figure 70 - Current vector computation subsystem

The I_{ref} is first used as I_{q0} reference while the I_{d0} reference is set to 0. The norm and the angle (with respect to the d-axis) of the current vector are computed. Then the angle is modified in two different ways on the base of the sign of I_{ref}.

If $I_{ref} > 0$, $I_{q0} > 0$ and thus $\Phi > 0$:

The complementary angle is multiplied for β_c and reduced ($\beta_c < 1$). Then the complementary angle is subtracted from π the final value is used to compute the final I_q and I_d references.

If $I_{ref} < 0$, $I_{q0} < 0$ and thus $\Phi < 0$:

It means that the motor speed is greater than the reference value. In order to reduce it, a negative I_q current is desired, thus, the angle is kept negative (-90°) without multiplying it for β_c.

For what concerns the PI controllers inside the current loop, a modified back-calculation technique has been applied. When the rated condition are met the reference voltage is saturated (see (3.63) formula), which means that the PI controller outputs, V_{qref} *and* V_{dref} can't be applied. Thus, the error between the saturated refence voltage V_{ref}^{*} and the objective value V_{ref} is computed, multiplied for a tunable gain (Kz) and fed back in order to reduce the error integrated by the current PI controller (Figure 69).

The final gain chosen through trial and error procedure is:

$$Kz = 0.5 \tag{3.68}$$

Figure 71 shows the current PI controller subsystem.

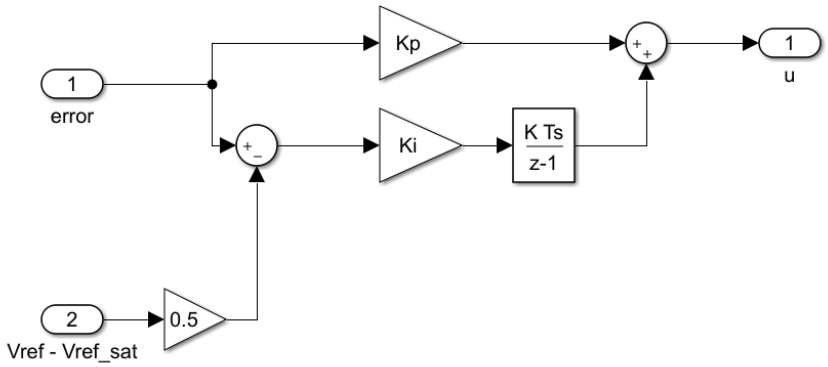

Figure 71 – Current PI controller with anti-windup

Multiple simulation with different reference speed profile have been carried out. The speed and torque outputs of the first one are shown in Figure 72 and Figure 73

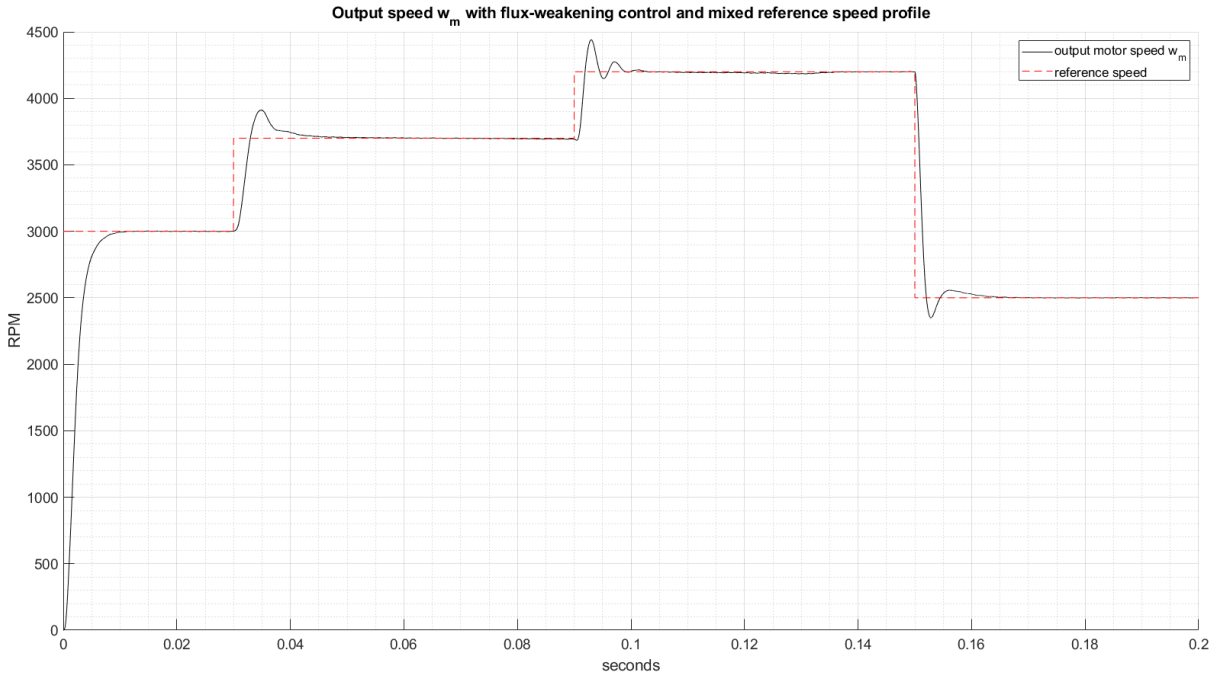

Figure 72 - Output speed w_m with flux weakening control and multiple speed step references

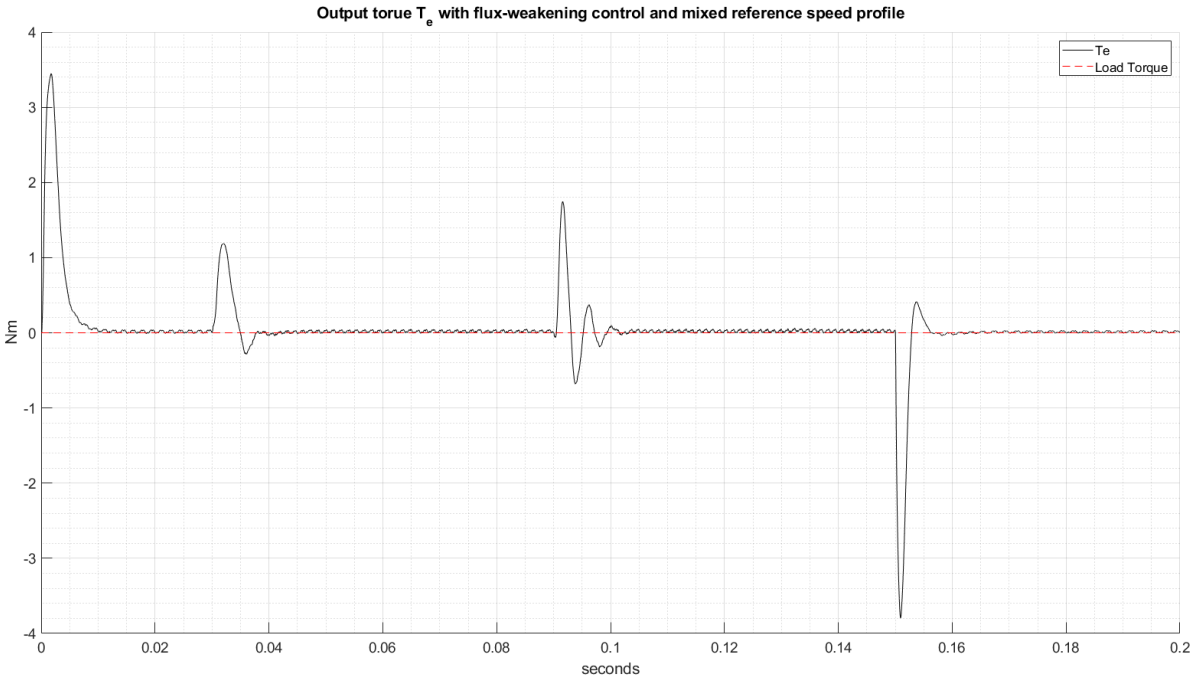

Figure 73 - Output speed w_m with flux weakening control and multiple speed step references

This first simulation highlights the advantages of the anti-windup technique applied to the speed PI controller. The overshoot when a 3000 RPM step reference is applied is 0%. Furthermore, the settling time is reduced to $t_{settling,5\%} = 5.4\,ms$. For what concerns all the other changes in the value of the reference speed, the settling time is always below 10 ms. For what concerns the Te graph it has been decided to plot the mean value of the torque generated by the motor computed over the time interval t_s in order to better identify the torque profile.

In the second simulation the reference profile has been built through the use of the Signal Editor tool, to test the motor behavior when ramps reference signals are applied. In addition, a load torque $T_l = \frac{1}{2}T_{rated} = 0.635$ has been applied at t=0.12.

The speed and torque outputs are shown Figure 74 and Figure 75.

In this case the overshoot due to the ramp reference signal from 0 to 3000 RPM is the highest one and it's equal to

$$\hat{s} = \frac{3224.7 - 3000}{3000} = 7.5\ \% \tag{3.69}$$

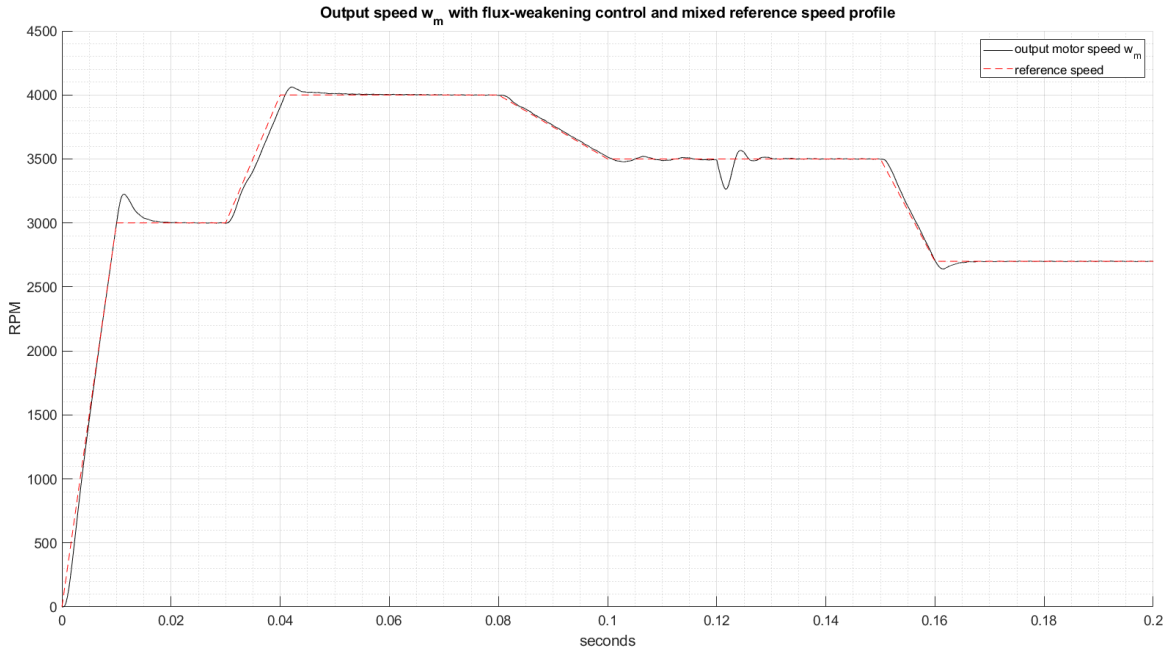

Figure 75 - Output speed w_m with flux weakening control and multiple speed ramp references (load torque applied at t=0.12

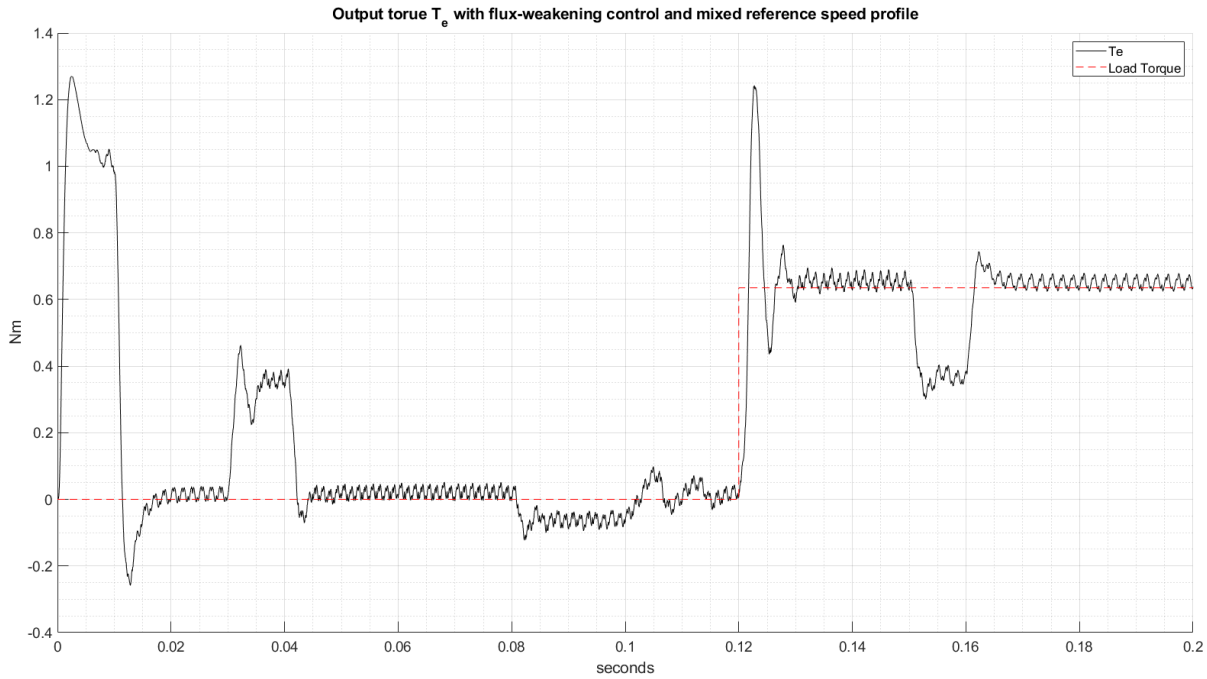

Figure 74 - Output torque Te with flux weakening control and multiple speed ramp references (load torque applied at t=0.12

It can be seen that the speed reference is followed with good accuracy and the load torque disturbance is cancelled in less than 10 ms.

Finally, 2 other simulations, one with step reference equal to 3500 RPM (in the flux-weakening area) and one at 300 RPM have been carried out. The load torque has been set respectively equal to $T_l = \frac{1}{2} T_{rated} = 0.635\ Nm$ and $T_l = T_{rated} = 1.27\ Nm$ in order to compare the steady state RMSE value with the previous simulation related to the Hinfinity controller and to the first implementation of the FOC. The results, for what concerns the speed output, are shown in Figure 77 and Figure 76.

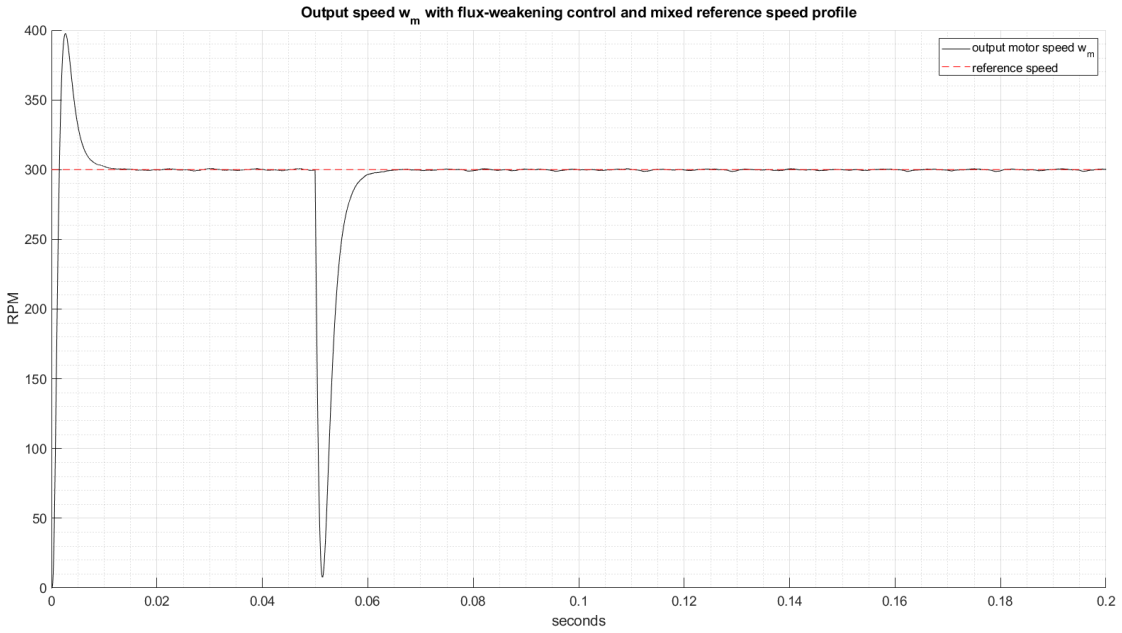

Figure 77 - Output speed with flux weakening control and 3500 RPM speed step references (T_L=0.635 Nm)

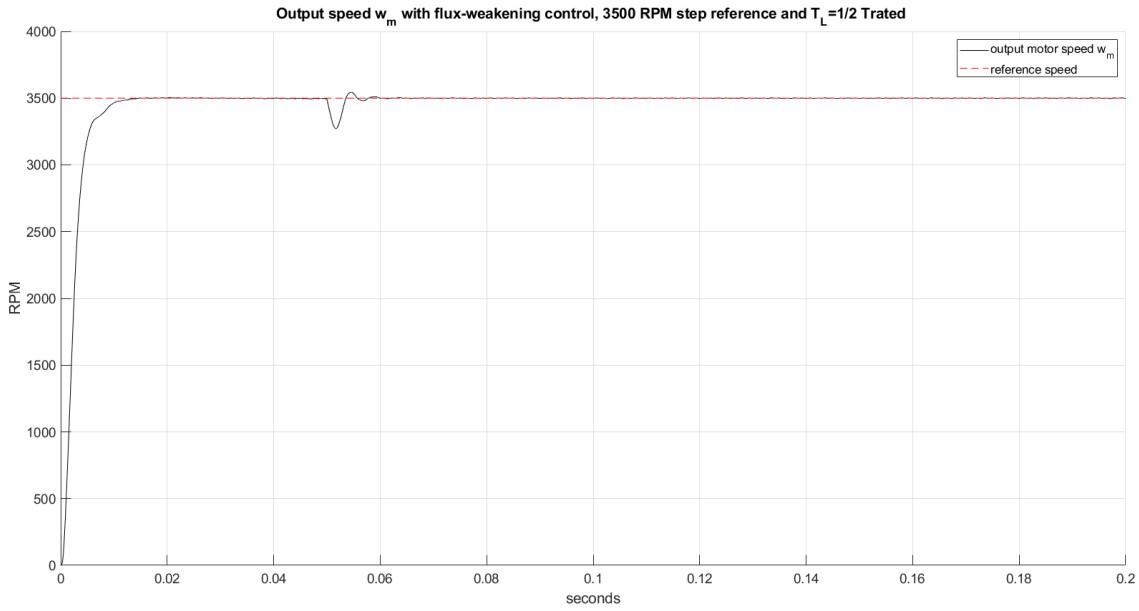

Figure 76 - Output speed with flux weakening control and 300 RPM speed step references (T_L=1.27 Nm)

For what concerns the simulation with reference speed equal to 3500 RPM the RMSE value, computed analyzing the data between t=0.12 and t=0.19 is:

$$RMSE_{3500} = 1.1485 \text{ RPM} \tag{3.70}$$

Which is comparable with the results obtained in Figure 65.

The simulation with 300 RPM step reference shows results comparable with the one carried out with the standard PI controllers. The flux weakening control at low speed is not active and the outputs of the PI controllers are not saturated, that is why the results are comparable with the one shown in Figure 62 (black line related to the first PI model). In this case, the following RMSE value has been found analyzing performance in steady state:

$$RMSE_{300} = 0.4405 \text{ RPM} \tag{3.71}$$

Which means that the oscillations around the steady state value have amplitude in between the one found with the standard PI model and the one related to the Hinfinity model (see Figure 65 for further details).

4 Conclusions

During the development of this thesis it has been decided to focus attention on the theory and the mathematical models behind the control of a PMSM. In “Chapter II – Theoretical basic concepts”, it is given the theoretical background necessary to develop the different control techniques proposed and an accurate model of the motor. In the next chapter it has been decided to show the design work developed in MATLAB and Simulink, step by step. Finally, the results obtained with the different controllers applied are presented and compared. The use of an Hinfinity controller have led to better performance in terms of overshoot and rejection of the load torque disturbance at low speed, which makes it particularly interesting in case of low speed application such as materials handling, warehousing and related industrial applications. On the contrary, the PI controller are easier to design and anti-windup methods can be applied, which makes them more suitable for high speed application with limited DC voltage supply (thus, working in the flux weakening area). All the models developed are thought for a future digital implementation of the controllers designed. For this reason, the different domains (continuous for the PMSM model and the inverter and digital for what concerns the control algorithms) have been separated, leaving the user the possibility to decide different sampling time for the speed and current loops on the base of the hardware computational power available.

5 Appendix

5.1 SVPWM: “Switch_Signal_Generator” MATLAB function

```
function sf = aaa(vdc, u)

%{
Matlab Code to generate Switching functions
Inputs are:  u1(:) magnitude of reference space vector
             u2(:) angle of reference space vector
             u3(:) ramp time signal for comparison from 0 to ts
%}

ts=0.0002;  %Inverter switching period
x=u(2);
y=u(3);
sa=0;
sb=0;
sc=0;

generic_space_vector_module=2/3*vdc;

%note that |Vref| (u(1)) can't exceed Vdc/sqrt(3)

mag=u(1)/generic_space_vector_module * ts/sin(pi/3);

%sector I
if (x>=0) & (x<pi/3)

    ta = mag * sin(pi/3-x);
    tb = mag * sin(x);
    t0 =(ts-ta-tb);

    t1=[t0/4 ta/2 tb/2 t0/2 tb/2 ta/2 t0/4];
    t1=cumsum(t1);

    v1=[0 1 1 1 1 1 0];
    v2=[0 0 1 1 1 0 0];
    v3=[0 0 0 1 0 0 0];

      for j=1:7
        if(y<t1(j))
            break
        end
      end

    sa=v1(j);
    sb=v2(j);
    sc=v3(j);
end

% sector II
if (x>=pi/3) & (x<2*pi/3)
```

```
    adv= x-pi/3;
    tb = mag * sin(pi/3-adv);
    ta = mag * sin(adv);
    t0 =(ts-ta-tb);
    t1=[t0/4 ta/2 tb/2 t0/2 tb/2 ta/2 t0/4];
    t1=cumsum(t1);

    v1=[0 0 1 1 1 0 0];
    v2=[0 1 1 1 1 1 0];
    v3=[0 0 0 1 0 0 0];

    for j=1:7
        if(y<t1(j))
            break
        end
    end

    sa=v1(j);
    sb=v2(j);
    sc=v3(j);
end

%sector III
if (x>=2*pi/3) & (x<pi)
    adv=x-2*pi/3;
    ta = mag * sin(pi/3-adv);
    tb = mag * sin(adv);
    t0 =(ts-ta-tb);
    t1=[t0/4 ta/2 tb/2 t0/2 tb/2 ta/2 t0/4];
    t1=cumsum(t1);
    v1=[0 0 0 1 0 0 0];
    v2=[0 1 1 1 1 1 0];
    v3=[0 0 1 1 1 0 0];

    for j=1:7
        if(y<t1(j))
            break
        end
    end

    sa=v1(j);
    sb=v2(j);
    sc=v3(j);
end

%sector IV
if (x>=pi) & (x<4*pi/3)
    adv = x - pi;
    tb= mag * sin(pi/3 - adv);
    ta = mag * sin(adv);
    t0 =(ts-ta-tb);
    t1=[t0/4 ta/2 tb/2 t0/2 tb/2 ta/2 t0/4];
    t1=cumsum(t1);
    v1=[0 0 0 1 0 0 0];
    v2=[0 0 1 1 1 0 0];
```

```
    v3=[0 1 1 1 1 1 0];
    for j=1:7
        if(y<t1(j))
                break
        end
    end
    sa=v1(j);
    sb=v2(j);
    sc=v3(j);
end

% sector V
if (x>=4*pi/3) & (x<5*pi/3)
    adv = x-4*pi/3;
    ta = mag * sin(pi/3-adv);
    tb = mag * sin(adv);
    t0 =(ts-ta-tb);
    t1=[t0/4 ta/2 tb/2 t0/2 tb/2 ta/2 t0/4];
    t1=cumsum(t1);
    v1=[0 0 1 1 1 0 0];
    v2=[0 0 0 1 0 0 0];
    v3=[0 1 1 1 1 1 0];
    for j=1:7
        if(y<t1(j))
            break
        end
    end

    sa=v1(j);
    sb=v2(j);
    sc=v3(j);
end

%Sector VI
if (x>=5*pi/3) & (x<2*pi)
    adv = x-5*pi/3;
    tb = mag * sin(pi/3-adv);
    ta = mag * sin(adv);
    t0 =(ts-ta-tb);
    t1=[t0/4 ta/2 tb/2 t0/2 tb/2 ta/2 t0/4];
    t1=cumsum(t1);
    v1=[0 1 1 1 1 1 0];
    v2=[0 0 0 1 0 0 0];
    v3=[0 0 1 1 1 0 0];

    for j=1:7
        if(y<t1(j))
            break
        end
    end
    sa=v1(j);
    sb=v2(j);
    sc=v3(j);
end

sf=[sa, sb, sc];
```

5.2 Hinfinity controller MATLAB code

```
close all
clear all
clc
% Defining motor current PI controller parameters
s=tf('s');

L=6.5*10^-3
R=2.35;
Phi_pm= 0.07846;
P=8;
J=0.00003169;
B=52.79*10^-6;
Kp=11.75;
Ki=4248.08;

%% Design of the Wu weighting function to verify robust stability

Jn=0.00003169;
% considering 55% uncertainty on J
J_low=0.55*Jn;
J_high=1.45*Jn

Bn=52.79*10^-6;
% 50% for B
B_low=0.5*Bn;
B_high=1.5*Bn;

Kt=3/2*0.5*P*Phi_pm;

Ga=(Kp*s+Ki)/(s^2*L+(Kp+R)*s+Ki);
Gpn=Kt/(s*Jn+Bn);

w=logspace(-3,3,200);
maxim=zeros(200,1);
N1=20;
for J=J_low:(J_high-J_low)/N1:J_high
    for B=B_low:(B_high-B_low)/N1:B_high
        Gp=Kt/(s*J+B)
        delta=Gp/Gpn-1;
        [d,f]=bode(delta,w);
         d=squeeze(d);
          figure(1),
          loglog(w,d,'b')
          hold on
          for i=1:200
              if(d(i)>maxim(i))
                  maxim(i)=d(i);
              end
          end
   end
end
```

```
J=Jn;
B=Bn
% % %
figure(10),
magg = vpck(maxim,w);
Wa = fitmag(magg);
[A,B1,C,D] = unpck(Wa)
Wu=zpk(ss(A,B1,C,D));

% figure(1),
% [wu,f]=bode(Wu,w);
% wu=squeeze(wu);
% loglog(w,wu,'r')

%% Defining constraints
s_hat=0.05;
tsettling=0.02;
zeta = abs(log(s_hat))/sqrt(pi()^2+(log(s_hat))^2);
trise=tsettling/8;
wn_rise=1/trise/sqrt(1-zeta^2)*(pi-acos(zeta))

wn_settling=4.6/tsettling/zeta;
wn=max(wn_rise,wn_settling);
Gf=1;
Gs=1;
Tp=1.0;
Sp=1.28;
%% Designing Ws
S2=s*(s+2*zeta*wn)/(wn^2+2*zeta*wn*s+s^2);
S0=0.00001 %free paramater

w1=wn/250
w2=sqrt(w1*Sp/S0);
Ws_inv=s^2*S0*(1+s/w1)/(1+1.414/w2*s+s^2/w2^2)/(s+1);
S_low=s^2*S0;

wlow=logspace(-3,-2,300);
whigh=logspace(2,3,300);
wmid=logspace(1,4,300);
w=logspace(-3,5,300);

%% Plotting Ws^-1, s*S0 and S2
figure(20),
bodemag(S_low*1.1,wlow,'r')
hold on,
bodemag(S2,wmid,'r')
yline(20*log10(Sp));
bodemag(Ws_inv,w,'k')
grid on

%% Designing and plotting Wt

wnt=wn*6;
wmthf=linspace(62000,100000);
Wt_inv=1/(1+s/wnt)^2;
```

```
figure(2),
hold on
grid on
yline(20*log10(Tp),'r');
bodemag(Wt_inv,'k')

%% Defining W1, W2, W1_mod and W2_mod
wnt2=wnt;
W2_inv=1/(1+s/wnt2)^2;
w11=w1;
S02=S0;
w22=sqrt(w11*1/S02);
W1_inv=s^2*S02*(1+s/w11)/(1+1.414/w22*s+s^2/w22^2)/(s+1);

W1=inv(W1_inv);
W2=inv(W2_inv);
%
W2_mod=minreal(W2/(1+s/wnt2)^2,1e-3);
W1_mod=minreal(W1*s^2/(s+wn/10000)^2,1e-3);

%% Use of the hinflmi function to compute the controller Gc_mod

[Am,Bm,Cm,Dm] = linmod('H_inf')
M = ltisys(Am,Bm,Cm,Dm);
M = sderiv(M,2,[1/wnt2 1]);
M = sderiv(M,2,[1/wnt2 1]);
[gopt,Gcmod] = hinflmi(M,[1 1],0,1e-2,[0 0 0]);
[Ac,Bc,Cc,Dc]=ltiss(Gcmod);
Gcmod = zpk(ss(Ac,Bc,Cc,Dc))
%% Getting Gc from Gcmod
Gc=minreal(Gcmod*(s^2 - 0.1969*s +
0.01386)*(1+s/4.053e06)*(1+s/1.937e04)/(1+s/8000)/s^2,1e-3);

%% Verifying nominal requirements
L=Gc*Gpn*Gs*Gf*Ga;

T=L/(1+L);
S=1/(1+L);
% Ws^-1 vs S
figure(5),
bodemag(S,'k')
hold on,
bodemag(Ws_inv,'r')
% Wt^-1 vs T
figure(6),
bodemag(T,'k')
hold on,
bodemag(Wt_inv,'r')

%% Nichols chart of the open loop t.f. compared with Tp and Sp locus
figure(7),
myngridst(Tp,Sp)
hold on
w=logspace(-2, 5);
nichols(L,w)
```

```
%% Checking controller stability in presence of parameter
uncertainty
figure(12),
bodemag(inv(Wu),{0.001,10^5},'r')
hold on
bodemag(T,'b')
```

6 Bibliography

[1] T. A. Zarma, A. A. Galadima and A. A. Maruf, "A Review of Motors for Electrical Vehicles," *Journal of Scientific Research and Reports,* October 2019.

[2] X. T. Garcia, B. Zigmund, A. Terlizzi, R. Pavlanin and L. Salvatore, "Comparison Between FOC and DTC strategies for Permanent Magnet Synchronous Motors," *Advances in Electrical and Electronic Engineering,* pp. 76-81.

[3] D. Dhivyaa and B. Karunamoorthy, "Design of PMSM and its application," pp. Vol. 9, Issue, 05, pp.51047-51050, May 2017.

[4] K. Chau, Electric vehicle machines and drives, Hong Kong: IEEE Wiley, 2015.

[5] D. G. Holmes and T. A. Lipo, Pulse Width Modulation For Power Converters, IEEE PRESS, 2003.

[6] S. Chattopadhyay, "Clarke and Park Transform," in *Electric Power Quality*, Springer Science+Business Media, 2011, pp. 89-96.

[7] G. Holmes and T. Lipo, Pulse Width Modulation For Power Converters, IEEE Press, 2003.

[8] R. Bojoi, "Course of Electrical Technologies for eMobility," Politecnico di Torino, Italy, 2020.

[9] I. Atif, "Space Vector PWM Techniques for a three-phase VSI".*Encyclopedia.*

[10] Z. Keliang and W. Danwei , "Relationship Between Space-Vector Modulation and Three-Phase Carrier-Based PWM: A Comprehensive Analysis," *IEEE*

TRANSACTIONS ON INDUSTRIAL ELECTRONICS, vol. 49, no. 1, February 2002.

[11] P. Ramesh and R. Prathyusha, "Field Oriented Control of Permanent Magnet Synchronous Motor," *International Journal of Computer Sciance and Mobile Computing,* pp. 269-275, March 2014.

[12] M. Li, "Flux-Weakening Control for Permanent-Magnet Synchronous Motors Based on Z-Source Inverters," 2014.

[13] H. MAHMOUDI and A. LAGRIOUI, "Flux-Weakening control of Permanent Magnet Synchronous Machines," *Journal of Theoretical and Applied Information Technology,* 31 12 2011.

[14] L. R. da Silva, R. C. Flesch and J. E. Normey-Rico, "Analysis of Anti-windup Techniques in PID Control of Processes with Measurement Noise," IFAC PapersOnLine, 2018.

[15] A. Bemporad, "Automatic Control 2 - Anti-windup techniques," University of Trento, 2010-2011.

[16] C. Junxi, "Implementation and Analysis of Direct Torque Control for Permanent Magnet Synchronous Motor Using Gallium Nitride based Inverter," *Electronic Theses and Dissertations. 7502,* 2018.

[17] C. Xing, H. Jibin , C. Kai and P. Zengxiong , Modeling of electromagnetic torque considering saturation and magnetic field harmonics in permanent magnet synchronous motor for HEV, vol. 66, Elsevier B.V, 2016, pp. 212-225.

[18] D. Y. Ohm, "Dynamic model of PMSM".*Drivetech, Inc.*

[19] . M. MUŠÁK and M. ŠTULRAJTER, "Novel methods for parameters investigation of PM Synchronous Motors," *Acta Technica Corviniensis - Bulletin of Engineering,* January-March 2013.

[20] V. Blasko, "Analysis of a Hybrid PWM Based on Modified Space-Vector and Triangle-Comparison Methods," *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS,* vol. 33, no. 3, May/June 1997.

[21] M. K. Mosi, S. Venugopal and G. Narayanan, "Space vector-based analysis of overmodulation in triangle-comparison based PWM for voltage source inverter," in *Sadhana*, vol. 38, Indian Academy of Sciences, 2013, pp. 331-358.

[22] N. Maleki, M. R. A. Pahlavani and I. Soltani, "A Detailed Comparison Between FOC and DTC Methods of a Permanent Magnet Synchronous Motor Drive," *Journal of Electrical and Electronic Engineering,* January 2015.