Technische Universiteit Delft

Hardware in the Loop Simulation of an Electric Vehicle Powertrain

MSc. Thesis Report

J. Landívar López

TU Delft

Hardware in the Loop Simulation of an Electric Vehicle Powertrain

MSc. Thesis Report

by

J. Landívar López

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 18, 2019 at 10:00 AM.

Student number:	4724801	
Project duration:	December, 2018 – October, 2019	
Thesis committee:	Dr. J. Dong,	TU Delft, Supervisor
	Prof. dr. ir. P. Bauer,	TU Delft
	Dr. T. Batista Soeiro,	TU Delft
	Dr. M. Cvetkovic,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgements

I want to express my sincere gratitude to my supervisor, Dr Jianning Dong, for the valuable advice, guidance and the knowledge he shared with me during this thesis project. I appreciate his help when I faced difficulties and his valuable feedback. As well, I would like to thank the electrical machines laboratory's staff, Bart Roodenburg, Joris Koeners and Harrie Olsthoorn for their assistance and support during the hardware implementation of the project.

I would like to mention my classmates and friends Celia, Armando, Arturo and Jorge. For the afternoons and study nights that eventually turned into family reunions. My deepest gratitude to Katherine for her care, love and happiness while we grew up together for the past two years. To Katja who became part of my new family and was always there to listen to me. I thank my uncles, who made me feel I was not alone in a foreign land. To my parents, whom I never thank enough. And to my future self, who never gave up.

Finally, I would like to give my gratitude to the Government of the Republic of Ecuador (institutionally represented by SENESCYT), for having granted me with a full scholarship to finance my studies at the Delft University of Technology.

J. Landívar López
Delft, October 2019

Contents

Abstract

A hardware in the loop (HIL) simulation is a cost and time effective method to develop a simulation environment of an electric vehicle's powertrain. Control algorithms can be programmed into a digital signal processor and tested with real-time simulations of electrical machines and power electronic converters, eliminating the need for developing real prototypes.

Consequently, this project aims to develop a simulation environment for electric vehicle research by implementing a HIL simulation of an EV powertrain computer model and an external hardware controller.

The proposed HIL simulation runs in real time the transmission, inverter, and PMSM models of an EV powertrain inside an OP5700 simulator. On the other hand, the controller algorithm is executed from a TI LaunchXL-F28379D board. The controller samples current and rotor position signals from the motor model through the analog outputs of the simulator, and generates a PWM signal as a command to the inverter computer model. In addition, a software in the loop (SIL) simulation is implemented, running the controller algorithm in the real-time simulator's CPU to test the control strategy before moving on to the hardware implementation. The powertrain specifications are based on the released information of a battery-electric vehicle on the market. The HIL controller's FOC algorithm is based on MTPA and flux-weakening strategies from the reviewed literature.

The simulation results show the expected behaviour from the inverter's phase voltages, stator currents and the output torque of the simulated electric motor model. Finally, the response of the developed SIL and HIL simulations to a reference drive cycle is analysed, and the results validate the implemented controller algorithm and powertrain model.

1 Introduction

Throughout the world, the replacement of fossil fuels in exchange for renewable energy sources is needed to reduce the environmental impact brought by atmospheric pollutants related to the transportation sector. In this sense, the systematic replacement of vehicles powered by an internal combustion engine by fully electric cars becomes an important initiative to improve the environmental conditions of modern-day cities. An electric vehicle (EV) is defined as a vehicle where an electric motor provides the propulsion power. Among the types of EV, battery electric vehicles (BEV)[13] are fully electrical, powered by batteries, and do not release greenhouse gases into the atmosphere.

Because of the inherent environmental benefits of zero-emission vehicles, propulsion and energy storage research is a field open for development and technological innovation that is on the rise in automotive companies and universities. Nevertheless, with research comes in hand the necessity to test the performance of EV powertrain components and the respective controller strategies. Ultimately, experimentation can represent an issue considering the high costs and the time constraints of developing real scale models of a vehicle.

However, a cost-effective solution for testing and technology development is the implementation of real-time hardware in the loop (HIL) simulations of the electric vehicle. Through this method, the functional components of an EV can be simulated in real-time and interact with external controller hardware. Favourably, a real-time simulation platform from Opal-Rt is available at the electrical machines laboratory of the DCE&S group. The OP5700 real-time simulator is equipped with digital and analog interface modules necessary for HIL simulation.

Therefore, the purpose of this thesis work is to develop a laboratory environment for electric vehicle research in the university. And as a result, accelerate testing and controller development stages, making use of the Opal-RT OP5700 real-time simulator in the DCE&S laboratory.

1.1. Objective

- Model the transmission, power converter and electric motor of an EV powertrain and implement a hardware in the loop simulation using the platform in the laboratory and an external controller board.

To fulfil the project objective, some initial constraints and limitations are set:

- The variation of inductances due to saturation effects or current fluctuation are neglected when modelling the electric motor.
- The space vector (SV) modulation used in the power converter operation is not an object of investigation in this work. It is implemented with pre-existing models in the Simulink and RT-Events libraries.

1.2. Research Questions

Given the general purpose and objectives arranged for the present work, the following questions arise:

- What type of control strategy is necessary so that the electric motor of the EV can achieve a wide speed operation range?
- In which way can the implemented EV powertrain model and the controller algorithm be validated?
- According to the simulation needs, what is necessary to model and simulate the powertrain components in real-time?

1.3. Thesis Outline

The thesis report is structured in 6 main chapters. The first chapter is the introduction of the project and also where the objectives of the work are defined. Chapter 2 defines the main characteristics of an electric vehicle and its main functional blocks. As well, an introduction to the Nissan Leaf 2011 is given as the reference vehicle for this thesis work, and its powertrain parameters are listed. In this chapter the modelling stage of the resistance forces over the vehicle and the transmission is shown. The third chapter gives an introduction to the main characteristics of permanent magnet synchronous motors and shows the theory behind the flux oriented control (FOC) strategies to be implemented later in the controller. Chapter 4 presents the execution of the powertrain model in a real-time simulation using the Opal-RT platform and the implementation of the HIL controller using a LaunchXL-F28379D board. Finally, chapter 5 presents the real-time simulations results and chapter 6 the conclusions of the project.

2

Introduction and Dynamic Modelling of an Electric Vehicle Powertrain

In this chapter, the concept of an electric vehicle's powertrain is explained and the considerations taken to model the inverter and electrical motor in further chapters. Special consideration is given in this chapter, to the external forces that intervene in a vehicle's motion and how they are connected to the source of movement, the electrical motor. Initially, a brief introduction to the main specifications of the model vehicle chosen for this project, the Nissan Leaf 2011. Next, a general overview of vehicle movement is given, including the main motion opposing forces and the equations for a gear transmission.

2.1. Electric Vehicle

An Electric Vehicle (EV) is an automobile propelled by an electric motor and powered by a battery pack [7]. The powertrain of an electric vehicle is the system composed of all the main functional blocks that efficiently transform the energy stored in the battery into mechanical energy. It is comprised by an Energy Storage System (ESS), an electric motor, a power electronic conversion stage in most cases, a transmission, differentials, the final drive, and the drive shafts [10].

The term Purely Electric Vehicle (PEV) is used to define vehicles which are only powered by battery packs. These, are charged from the electric grid system at homes, workplaces or drive-in charging stations [34]. The main advantage of EVs over the Internal Combustion Engine (ICE) powered vehicles or even Hybrid Electric Vehicles (HEV) is the high efficiency the system can reach. Another advantage of EVs is their low lifetime maintenance needs.

Even though a significant drawback for electric vehicles is their reduced range or autonomy in comparison to ICE-powered vehicles or HEV, nevertheless, thanks to the continuous research and development of new materials and high-density energy storage this gap is being shortened.

2.1.1. The Nissan Leaf 2011 Powertrain

The purpose of this project, to some extent, is the simulation of the behaviour of an electric vehicle's powertrain. One of the first steps is to find information about a market vehicle and its main functional components. The Nissan Leaf from 2011, in figure 2.3, is the vehicle of interest for the present study. It was released in the year 2011 and is the first fully electric vehicle produced by a major car manufacturer. This precedent is one of the reasons why a considerable amount of information about its main functional blocks has been released. Generally, such an amount of detailed information is not available.

First, it is necessary to obtain a description of the components in the Nissan Leaf. K. Namdoo et al [21], provide a basic powertrain scheme shown in figure 2.2. As the figure shows, the Nissan Leaf powertrain consists of a battery bank, a DC-DC converter that feeds electronic accessories, and an electrical machine attached to a mechanical transmission system. However, this diagram is broad and does not clarify what type of electrical machine is used, or the power electronics stage connecting said machine to the battery bank.

More detailed information could be found in the literature released by the manufacturer [31][32] and the Argonne National Laboratory [14]. This information is condensed in table 2.1.

Figure 2.1: Nissan Leaf 2011 [31].

A 375 Volts high-density battery pack that serves as the power source for the entire system. The voltage from the battery is converted by a three-phase IGBT voltage source inverter, shown in figure 4.7. The inverter transforms the battery's DC voltage into an AC voltage that feeds the Interior Permanent Magnet Synchronous Motor (IPMSM) to generate the driving torque of the electric vehicle. Figure 2.3 helps to visualize the disposition of the motor, inverter, and battery inside the car.

Furthermore, the driving torque from the motor is transmitted to the driven wheels through the drive train, consisting of a torque coupling stage, a differential, and a single-speed gear transmission. The manufacturer states that the motor is combined with a 'three-parallel-shaft' gear reducer with a reduction ratio of 7.938 [32]. Once the power is transmitted to the driving wheels, it can counter-act the external forces that oppose the motion and that are exerted over the vehicle chassis and wheels.

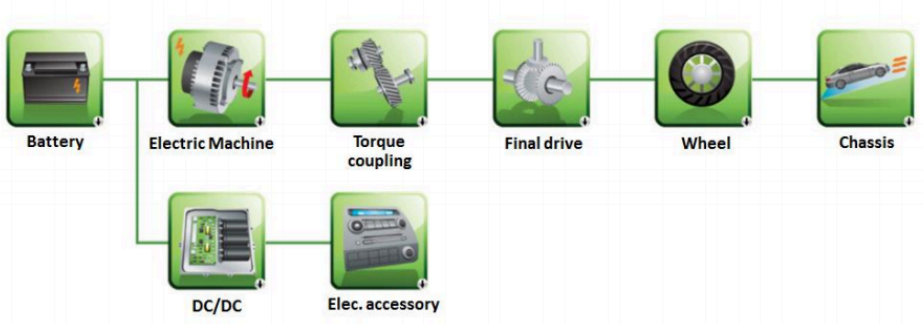


Figure 2.2: Nissan Leaf 2011 Powertrain [21].

Additionally, basic specifications of the Leaf's motor, are displayed in Table 2.2 below. Something to remark is that the Permanent Magnet Synchronous Motor (PMSM) in the powertrain is of the IPM kind. It has the magnets embedded in the interior of the rotor's structure, as displayed in Figure 2.5. Likewise, motor parameters have not been released by the manufacturer. Inductance and permanent magnet information shown in table 2.2 are taken from the experimental data provided by R.Yang [35]. Additionally, it must be stated that the inductance values showed, are averaged for the whole AC-current range of the IPM motor.

An electric vehicle's powertrain is the combination of different physical systems, where each draws energy from their primary source, the battery pack, and performs specific work. However, as in all real-world systems, energy is lost and each functional block within the powertrain like the inverter, motor or transmission, has an individual efficiency rate.

In this case, the efficiency of the electric powertrain is shown in Figure 2.6. It comprises the inverter, the electric motor and the cables connecting them. Nissan claims an efficiency of 95% at the highest efficiency point and a weighted average of 90%. Also, the manufacturer indicates the maximum motor speed as 10000 RPMs and the rated (base speed) of the machine is around the 2100 RPMs. This matches the torque vs. speed profile shown in the efficiency map.

Electric Vehicle Parameters		
Parameter	Nissan Leaf 2011	Unit
Curb Weight	1521	kg
Drag coefficient	0.22	
Gear ratio	7.938	
Battery voltage	375	V
Battery capacity	24	kWh
Battery type	Li Ion	
Max speed	144	kmh
0-60 mph	11.9	s
Wheel radius	0.316	m
Wheel type	205/55R16	
Wheel inertia	0.849	kg m^2
Gear efficiency	97	%
Inverter efficiency	98	%
Motor efficiency	95	%
Rolling res. coefficient	0.0098	
Switching frequency	5000	Hz
Inverter type	3 phase, IGBT	
Frontal area	2.27	m^2

Table 2.1: Nissan Leaf 2011 powertrain specifications [14][31][16].

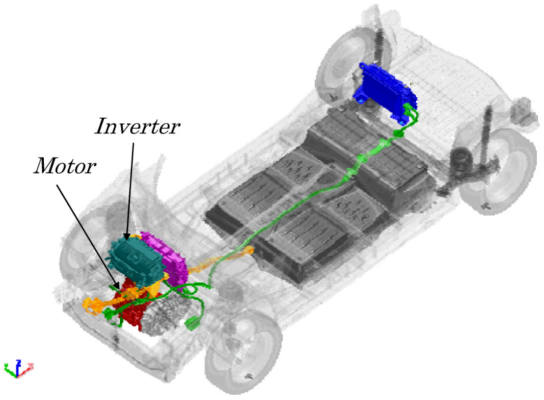


Figure 2.3: View of inverter, motor, charging module and battery for Nissan Leaf 2011 [31].

The inverter contains 3 IGBTs and 3 diodes per switch, 18 IGBTs in total. The literature claims a carrier frequency of 5 kHz and an operating input DC voltage range of 240 V to 403 V, although the rated voltage is stated as 375 volts.

2.1.2. Powertrain Modelling

In the previous section, a review of the main characteristics of the Nissan Leaf's powertrain was made. As a result, a set of powertrain parameters is obtained and listed in tables 2.1 and 2.2. Based on these parameters, a new block diagram of the powertrain is made.

Based on figure 2.7, the modeling of the electric vehicle powertrain is done in different stages. First, modelling the mechanical dynamics of the vehicle the resistance forces exerted over the chassis and the wheels. Second, model how these external loads are transmitted to the shaft of the vehicle's motor, therefore a transmission equation model is as well necessary. Likewise, the model of a PMSM and the three-phase inverter completes the modelling stages.

In the present work, some simplifications and constraints were taken before modelling the previously mentioned stages. In the first place, a battery model will not be included in the powertrain simulation.

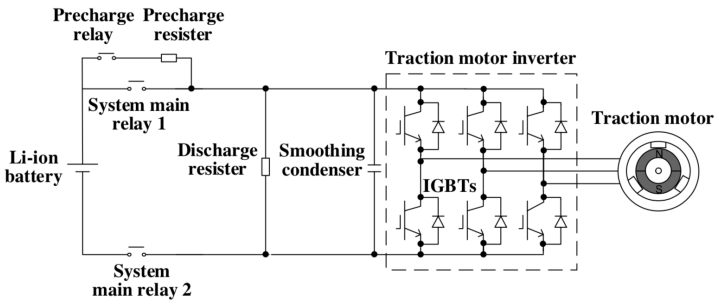


Figure 2.4: Nissan Leaf IGBT inverter diagram [35].

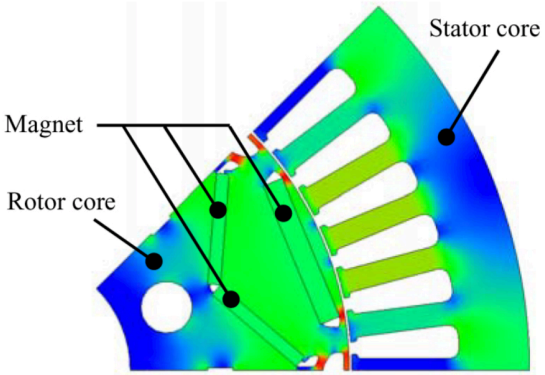


Figure 2.5: View of rotor and delta arrangement of embedded permanent magnets. Nissan Leaf 2011 [31].

For the sake of simplicity, the battery is simulated as a constant DC voltage at the rated value. Secondly, since detailed information about the inverter components is not available, the model of the inverter is made of 6 IGBTs and six diodes. An exact simulation of the powertrain is not the objective of this project. Finally, inductances and magnet flux linkage values are average estimations, and they do not consider saturation or thermal behaviour of the motor. As well, for simplicity purposes, losses are not modelled either.

On the next section, an overview of the forces that affect a vehicle while moving is given. Also, the first modeling stage, the transmission model is done.

Motor Parameters		
Parameter		Unit
Magnet Flux Linkage	0.067523	Wb
Number of poles	8	
Peak phase current	600 (4 seconds)	A
Rotor inertia	7.24e-2	kg m^2
Phase resistance	5.67e-3	Ω
Ld	120	μH
Lq	375	μH
Max torque	280	Nm
Output power	80	kW
Motor type	IPMSM, 3 phase	
Base speed	2100	RPM
Max speed	10000	RPM

Table 2.2: Nissan Leaf 2011 IPM Motor Specifications [14][31][16].

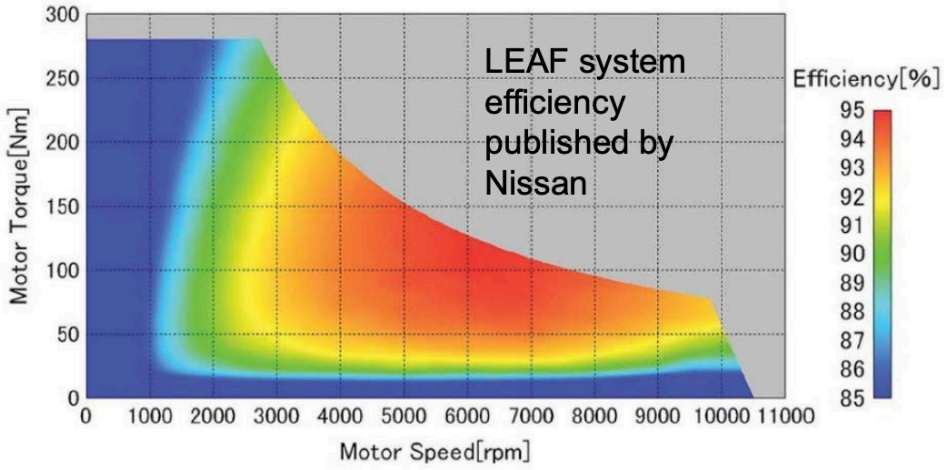


Figure 2.6: Nissan Leaf 2011 electric powertrain efficiency map [4].

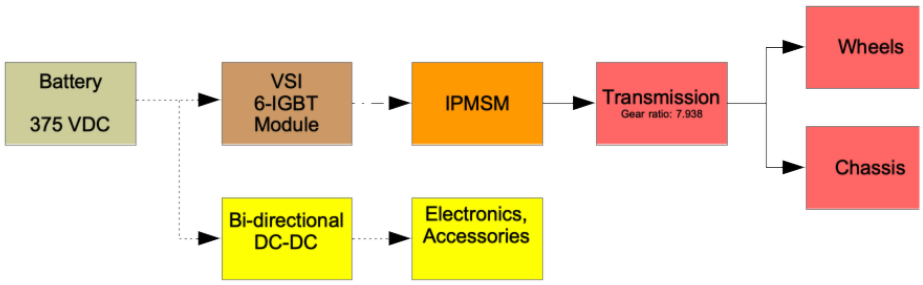


Figure 2.7: Powertrain block diagram.

2.2. Overview of Vehicle Movement

Figure 2.8 shows the forces involved in the longitudinal movement of a primary automobile. The forward-moving force of the vehicle F_w and the tractive force F_t. The source of the tractive force is an internal combustion engine or an electric motor. This force is transmitted to the wheels generally through a set of torque increasing gears defined as a transmission.

The necessary conditions so that the tractive effort of the power plant can accelerate the vehicle are given when this force is greater in magnitude than the movement-opposing forces or resistance forces. These resistance forces basically consist of the aerodynamic drag, uphill resistance, and tire rolling resistance.

From Newton's second law of motion, the linear acceleration of a vehicle is proportional to the difference between the tractive force and the resistance forces over the car, divided by the mass of the vehicle as equation 2.1 shows.

$$\frac{dV}{dt} = \frac{\sum F_t - \sum F_r}{M_v} \tag{2.1}$$

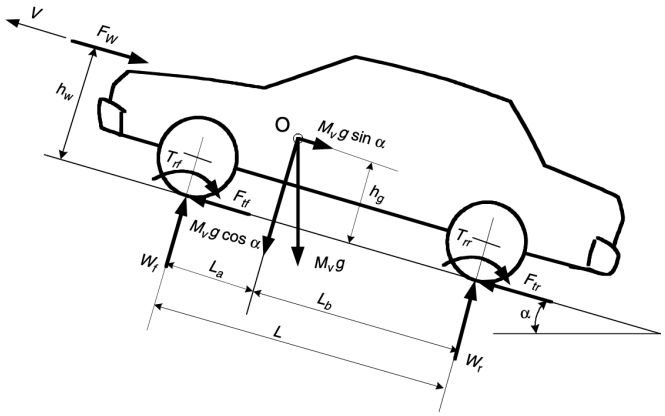


Figure 2.8: Forces acting over a vehicle [12].

In equation 2.1, V is the speed of the vehicle, $\sum F_t$ represents the total tractive force, $\sum F_r$ is the sum of all the movement opposing forces over the vehicle and M_v is the mass of the moving vehicle.

2.3. Vehicle Resistance Forces

This section is meant to provide insight into the most relevant external forces that take part in the dynamics of a vehicle. These external forces oppose the movement and must be overcome by the torque of the motor sufficiently so that the system develops acceleration.

2.3.1. Rolling resistance force

The rolling resistance force is a result of the deformation of the wheel in the contact area against the rolling surface. The load force P in figure 2.9 is distributed symmetrically over the contact area. Consequently, a reaction force P_z is generated, aligned with the load force but in the opposite direction.

The wheel deformation z results in a hysteretic behaviour of the reaction force P_z and the deformation of the wheel. The reaction force is asymmetrically distributed and more significant in the frontal part of the contact area. As a result, a reaction force shifted in direction takes place. Consequently, it has a force component that will oppose the movement of the vehicle [12].

Figure 2.9.(a) shows load force P and its reaction force P_z under the frontal part of the contact area. Figure 2.9.(b) shows how the change in deformation z results in the generation of a reaction force P_x parallel to the rolling surface and opposite to the moving direction.

$$F_r = Pf_r cos\alpha = Mgf_r cos\alpha \tag{2.2}$$

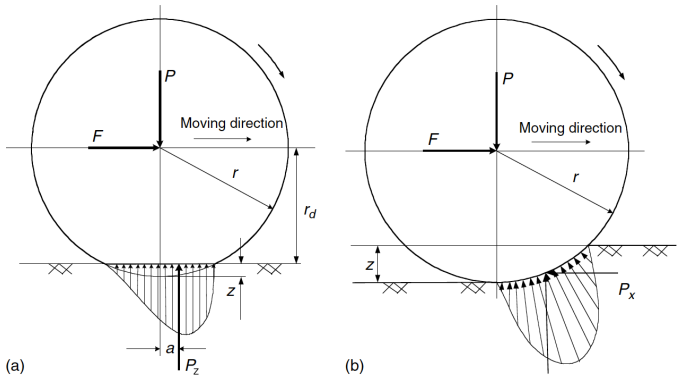


Figure 2.9: Deflection of load force P and rolling resistance force generation [12].

As a result, the magnitude of the rolling resistance force is given in equation 2.2. The term $f_r =$ is called the rolling resistance coefficient. Force P is the normal load in the wheel, therefore, from a simplistic approach it can be said is equal to the product between the mass of the vehicle and gravity's acceleration (Mg). If the vehicle is rolling over a surface with slope, the new normal load force is the perpendicular component to the road as given by equation 2.2, where the angle α is the slope angle in radians.

The rolling resistance coefficient depends on the tire material and structure, among others. A typical value for the rolling resistance coefficient for tires against asphalt is $f_r = 0.012$ [12].

2.3.2. Air Drag Force

Air exerts a resistance force over any moving body. When a vehicle is moving, it is pushing the air in front of it, accumulating and building up air pressure. A high-pressure wall appears at the side of the vehicle that is facing the direction of motion. In contrast, the opposite occurs at the rear side of the vehicle where the air is not able to fill the void left by the forward-moving body quickly enough. As a result, a low air pressure area is created behind the moving vehicle, as shown in figure 2.10.

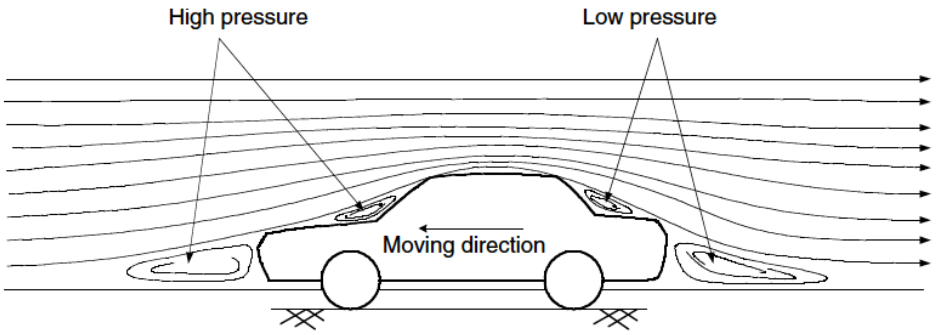


Figure 2.10: Aerodynamic drag over vehicle [12].

The pressure difference results in a force that opposes the movement of the vehicle through the surrounding air. This particular phenomenon is attributed to the shape of the moving body that causes the high pressure due to the accumulation of the air that gets trapped at the front of the vehicle and the low-pressure zone at the rear where the fluid (air) is missing in volume. The high-pressure front pushes the body against the movement direction, and the low-pressure side pulls the vehicle backward. This is denominated as shape drag [12].

The second phenomenon that also contributes to the aerodynamic drag force is the skin effect. It is based on the friction between the different moving layers of air close to moving vehicle. The air particles nearest to the surface of the vehicle, move almost at the same speed of the vehicle. The outer air layer remains static. Altogether, an additional resistance force is generated.

Based on the mentioned behaviours, the aerodynamic drag force depends on the geometry or shape

of the vehicle. Its magnitude is given by equation 2.3.

$$F_w = \frac{1}{2}\rho A_f C_D (V + V_m)^2 \tag{2.3}$$

Where:
ρ= air density
A_f= vehicle frontal area
C_D= aerodynamic drag coefficient
V= vehicle speed
V_w= wind speed

The aerodynamic drag or lift coefficient (C_D) varies according to the geometry of the car [1]. The V_w term is the speed of the wind and is represented with a positive sign when the wind is moving contrary to the vehicle.

2.3.3. Gradient Resistance

Any time a vehicle is moving up or down a road with gradient, its weight generates a resistance force that is parallel to the road surface. As observed in figure 2.8, this component can either oppose the movement of the vehicle when moving upwards or it can contribute to the forward motion when descending. The gradient resistance force, as seen in the image above, can be defined as:

$$F_g = M_v g \sin(\alpha) \tag{2.4}$$

M_v is the mass of the vehicle, g gravity's acceleration and α represents the road's gradient angle.

2.4. Vehicle's Dynamic Equation

The total resistance force is the total sum of the motion opposing forces which are summarised in the equation below in 2.5.

$$F_{res} = M_v g(f_r \cos\alpha + \sin\alpha) + \frac{1}{2}\rho A_f C_D V^2 \tag{2.5}$$

Overall, the most representative forces over the moving vehicle are the rolling resistance force of the front and rear axles (F_{rf}, F_{rr}). Next, the aerodynamic drag F_w and gradient resistance F_g forces. Finally, the forces that initiate movement these being the front and rear tractive forces F_{tf}, F_{tr}. F_{tf} is equal to zero in the case of a vehicle with traction only in the rear wheels and F_{tr} is zero when the car is front-wheel-driven.

As a result, the dynamic equation of the vehicle is expressed as follows:

$$M_v \frac{dV}{dt} = (F_{tf} + F_{tr}) - \underbrace{(F_{rf} + F_{rr} + F_w + F_g)}_{F_{res}} \tag{2.6}$$

2.4.1. Powertrain Tractive Force

Re-arranging equation 2.6 into expression 2.7, an equation for the needed tractive force applied to the wheels is obtained.

$$F_t = M_v \frac{dV}{dt} + (F_{rf} + F_{rr} + F_w + F_g) \tag{2.7}$$

The tractive force, is the force the vehicle's motor needs to transmit to the the wheels[12].

Generally the powertrain of a vehicle, consists of a set of components which include the motor, a clutch (if the transmission is manual) or a torque converter (in automatic transmissions), a gearbox, differential gears and a drive shaft connecting the driven wheels of the vehicle to the rest of the

transmission system. Depending if the vehicle is hybrid, electric or powered by an internal combustion engine, some of the elements above are not present. The functional blocks of a general automotive powertrain are shown in figure 2.11 [12].

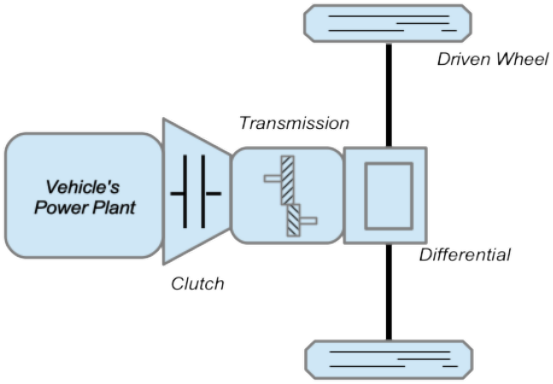


Figure 2.11: Automobile's general powertrain [12].

In essence, the torque and the speed generated at the shaft of the motor are transmitted to the wheels of the vehicle through the clutch or torque converter, gearbox, differential and final drive shaft. The clutch or torque converter's function, is to decouple the motor from the gear box. The gear box presents a gear ratio that generates an angular speed reduction and torque distribution to the wheels through the differential gears[12].

2.5. Gear Transmission

Gear transmissions are used widely, mainly in industrial applications where rotational motion needs to be transmitted from a motor to a load and these two can not be directly connected. Another reason can also be a mismatch (due to the system conditions) between the rotating speed of the motor and the speed needed by the load. For this reason, gears or pulleys are used. However, gears are able to transmit force without slip because of the coupling of their teeth.

A diagram of the vehicle transmission used in the present work, is shown in figure 2.12. The relationship between the torques exerted in each gear can be inferred from Newton's second and third laws. The image represents the generation of motion from the electric motor that generates torque T_e. The engine is connected by a shaft with inertia J_m to gear 1 with radius r_1 and angular speed ω_m. Next, the motor is connected to the wheel with rotational inertia J_w, load torque T_t. The angular speed of the wheel is ω_w and it keeps a linear proportion to the angular speed of the electric motor ω_m.

In figure 2.12, no slip and gears without inertia are assumed. Both gears have the same tangential speed in the point of contact, and no power is lost on them. Therefore, the angular speed and torque relations for the transmission of the vehicle are shown in equation 2.8 [29].

$$\frac{r_1}{r_2} = \frac{\omega_w}{\omega_m} = \frac{T_1}{T_2} \tag{2.8}$$

In general, the angular speed of the motor is larger than the speed imparted to the wheel, which has been reduced by the gear ratio of the transmission given in equation 2.9. The term G_r is the gear ratio for the Nissan Leaf shown in table 2.1.

$$G_r = \frac{r_2}{r_1} \tag{2.9}$$

The traction torque T_t is equal to the traction force F_t given in equation 2.7 multiplied by the radius of the wheel of the vehicle, r_w.

The relation between the torques T_1 and T_2 on each side of the transmission, the electric torque T_e and the tractive torque on the load T_t shown in figure 2.12, is given by equation 2.10 [29].

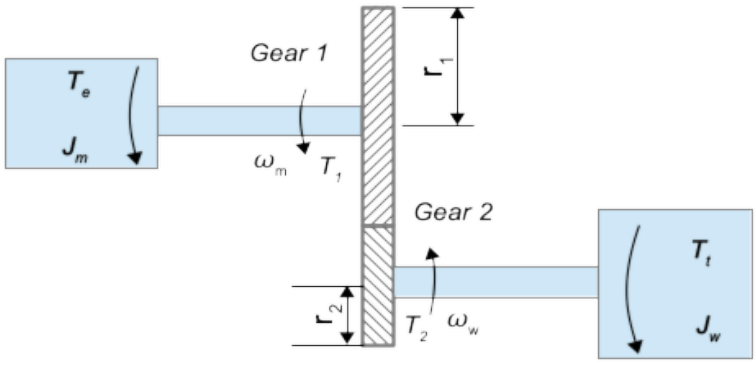


Figure 2.12: General gear transmission system.

$$\underbrace{\left(T_e - J_m \frac{d\omega_m}{dt}\right)}_{T_1} \frac{r_2}{r_1} = \underbrace{\left(T_t + J_w \frac{d\omega_w}{dt}\right)}_{T_2} \tag{2.10}$$

Organizing the previous equation to obtain an expression for the torque generated by the electric motor, equation 2.11 is obtained. Speeds and torques are referred to the motor side of the transmission.

$$T_e = J_m \frac{d\omega_m}{dt} + \left(\frac{r_1}{r_2}\right)^2 J_w \frac{d\omega_m}{dt} + \left(\frac{r_1}{r_2}\right) r_w F_t \tag{2.11}$$

As in equation 2.11, the electric torque is divided in two components. The first represents the torque that accelerates inertia J_m of the motor's shaft. The second and third components are due to the influence of the load forces and the acceleration of the rotating parts. In other words, they are the torque that the electric motor must transfer to the load in order to provide motion [9] [29].

Re-arranging and replacing the expression in 2.9 into equation 2.11, equation 2.12 below is obtained.

$$T_e = \left(J_m + J_w \frac{1}{G_r^2}\right) \frac{d\omega_m}{dt} + \frac{r_w}{G_r} F_t \tag{2.12}$$

If the traction force shown above is decomposed using equation 2.7, the final expression for modelling the vehicle's transmission is obtained below.

$$T_e - \frac{r_w}{G_r} F_{res} = \underbrace{\left[J_m + J_w \frac{1}{G_r^2} + M_v \frac{1}{G_r^2}\right]}_{J_{tot}} \frac{d\omega_m}{dt} \tag{2.13}$$

The equation above shows that the torque produced by the motor must be enough to overcome the resistance to rotation of the whole system and the resistance torque from external forces. Here, the inertia J_{tot} term is introduced, and it consists of the inertia of the motor's shaft and the inertia of the rotating parts of the load system as seen from the electric motor's side of the gear transmission.

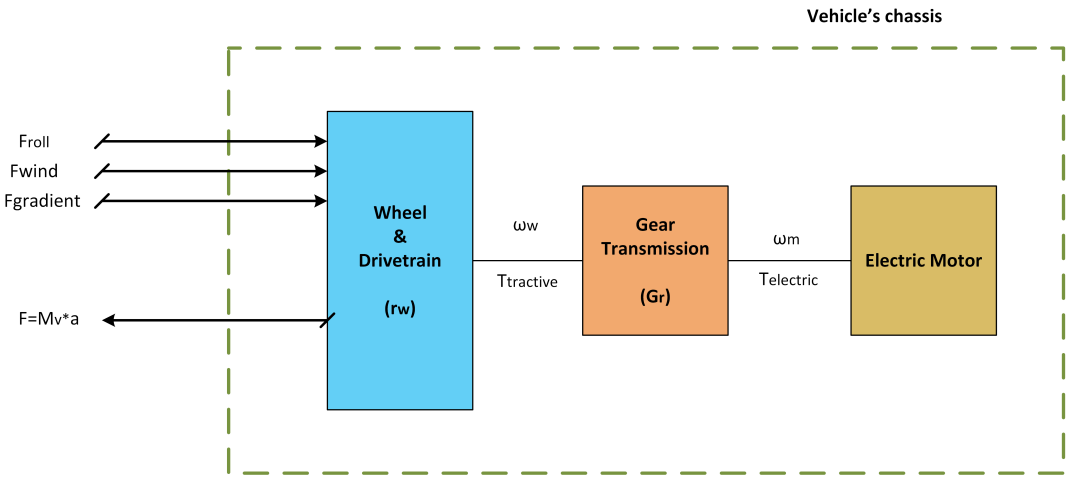


Figure 2.13: Vehicle's mechanical system.

Finally, figure 2.13 shows the final scheme of the vehicle's mechanical system, where the tractive torque is transferred to the wheel from the electric motor, through the gear transmission. After overcoming the vehicle's resistance forces, the transferred torque rotates the wheels, and the vehicle accelerates.

3

Modelling and Control of Permanent Magnet Synchronous Machines (PMSM)

The present chapter is an introduction the permanent magnet synchronous machine (PMSM). Special attention is given to a particular case, the interior permanent magnet synchronous motor which is the type of electric motor used by the Nissan Leaf. Accordingly, an overview of field oriented control is given based on the studied PMSM.

3.1. Permanent Magnet Synchronous Machines

The PMSM has become one the most preferred options when it comes for a powertrain's traction motor in electric vehicle (EV) applications. This is because of their high power density, high torque and specially its high efficiency.

Although induction machines have a mature technology supported by its continuous use throughout history in industrial applications, they have lower efficiency in comparison to synchronous machines. Additionally, the choice of permanent magnets of rare-earth materials, as a source of excitation, improves the magnetic flux distribution in the air-gap and the power and torque capabilities of the machine.

A special case of PMSM are interior permanent magnet motors (IPMSM), which offers a wider range of capabilities. In the meanwhile it requires sophisticated control algorithms in comparison to their counterpart, surface-mounted permanent magnet machines (SPMSM).

In this chapter, an introduction to the most important characteristics of permanent magnet synchronous machines will be given with respect to their application as a traction motor inside an electric vehicle's powertrain.

3.1.1. Classification of PMSMs

The source of excitation in a synchronous machine can come from a field winding in the rotor carrying an excitation current, or from permanent magnets attached to the machine rotor part. Either way, the distribution of the magnetic flux in the air-gap generated by the excitation, is assumed to have a sinusoidal form. On the basis on where the magnets are located in the structure of the rotor, PMSM are can be classified as surface-mounted (SPMSM) or interior permanent magnet synchronous machines (IPMSM).

In a SPMSM the magnets are attached to the rotor surface. This refrains the motor to be operating at relatively high speeds because is prone to the detachment of the magnets. However, it has the advantage of being easier to manufacture in comparison to their counterpart, the IPMSM. As it can be observed from figure 3.1, IPM machines rotor's configuration is more robust since the magnets are buried inside the rotor's laminated body.

The interior magnet embedded in the rotor structure, brings saliency to this type of machine that a SPMSM does not have. As shown in figure 3.1, the magnetic flux of the magnets, defines the direct-axis as the axis that is always aligned with the permanent magnet. In contrast, the quadrature-axis is located 90 electrical degrees in advance of the direct-axis [3]. It can be seen in figure 3.2.a and 3.2.b that for SPMSM the flux path is the same for both direct and quadrature-axis assuming that the

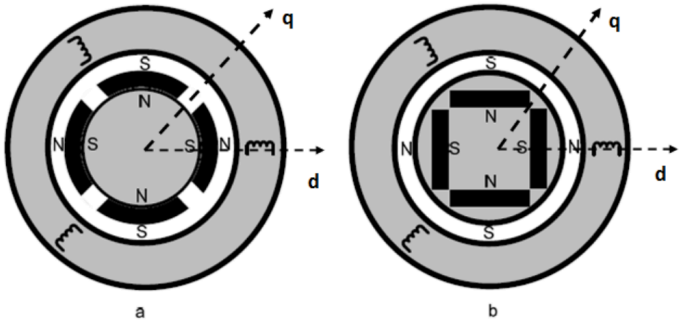


Figure 3.1: Examples of PMSM rotors types.(a) Surface-mounted magnet type, (b) Interior magnet type [30].

permeability of the permanent magnet is almost the same as the air's. Therefore, the effective air-gap is the same for both magnetic paths. As a result, for SPMSM $L_d = L_d$.

In contrast, observing at figures 3.2.c and 3.2.d, it is noticeable that for IPMSM the effective air-gap has changed according to the rotor position and the reluctance is higher for the direct-axis flux path. Hence, for IPMSM $L_q > L_d$. As a result, the saliency ratio of the machine is defined as shown in equation 3.1.

$$\xi = \frac{L_q}{L_d} \tag{3.1}$$

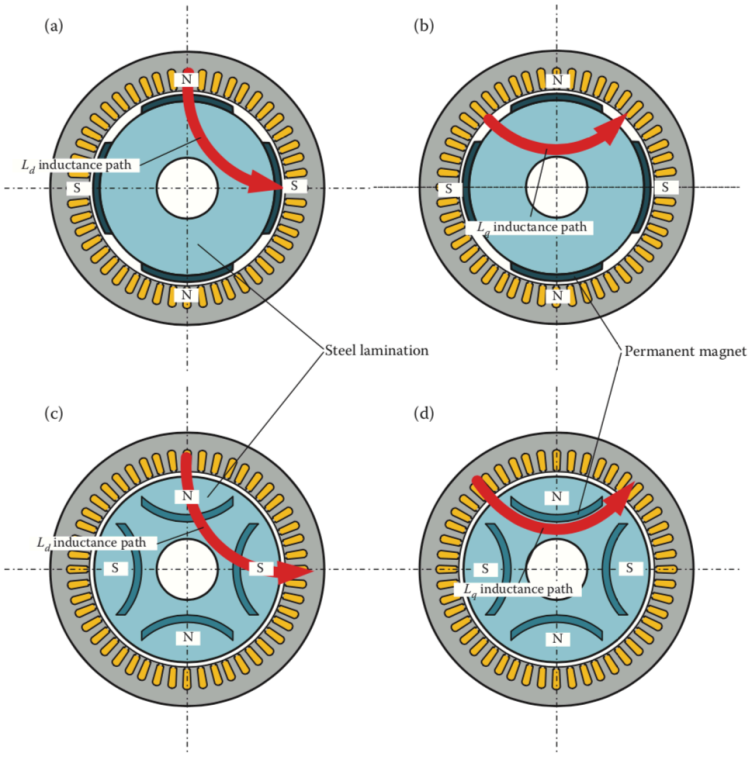


Figure 3.2: Examples of magnetic flux path for different PMSM rotor types (a) and (b) surface-mounted magnet type, (c) and (d) interior magnet type [3].

3.1.2. Rotor Oriented model of a PMSM

The following mathematical model of a synchronous machine is based on the IPMSM and is developed under the assumptions that the resistance of the winding and the permanent magnet's flux do not

suffer from thermal effects and therefore they are considered constant values. The same occurs for the inductances since no saturation effects will be considered and it is assumed that the three-phase voltages that feed the PMSM are a balanced system.

For the purposes of control, it is desirable to have a system where variables are DC quantities although the real variables are from a three-phase sinusoidal system. This is achieved by having a machine model that is based in a reference frame rotating at the same angular frequency as the sinusoidal variables. When the reference frame is moving at an angular speed equal to that of the sinusoidal three-phase supply, the relative speed between them becomes zero. As a result, the sinusoidal variables (e.g. stator voltages and currents) will now be perceived as a DC signals.

Starting from the three-phase stator phase-voltage in the winding of the PMSM, equation set 3.2 is provided.

$$\begin{aligned} u_a &= R_s i_a + \frac{d\psi_a}{dt} \\ u_b &= R_s i_b + \frac{d\psi_b}{dt} \\ u_c &= R_s i_c + \frac{d\psi_c}{dt} \end{aligned} \tag{3.2}$$

Where u_a, u_b and u_c are the phase voltages, R_s is the stator's winding resistance equal for all phases and ψ_a, ψ_b and ψ_c the flux linkages for each phase.

In order to change from a three-phase sinusoidal system to a rotating D-Q rotor oriented frame of reference, the Park transformation is applied to the set of equations in 3.2. The new D-Q system is oriented with the machine's rotor and is rotating at the same synchronous frequency of the stator's voltage. The transformation matrix is shown in equation 3.3.

$$\begin{bmatrix} u_{sd} \\ u_{sq} \\ u_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \tag{3.3}$$

The terms u_{sd} and u_{sq} of expression 3.3 represent the direct and quadrature components of the stator phase voltages. u_0 is referred as the zero sequence voltage and for the case of a balanced system, this component is equal to zero [24]. θ is defined as the rotor electrical angle, or as well as shown in figure 3.3 the angle between the stator's phase a-axis and the rotating direct-axis.

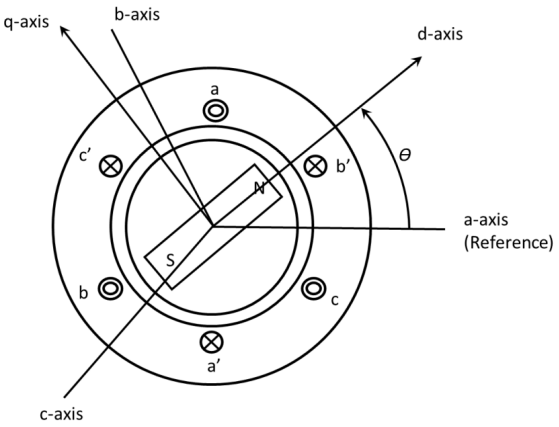


Figure 3.3: Vector representation of the Park transformation from the fixed three-phase to the D-Q rotating frame of reference. [25].

The angular frequency at which the D-Q plane is rotating in figure 3.3 is defined as the synchronous electrical speed ω_e, where $\omega_e = \frac{d\theta}{dt}$. As a result, the new stator voltage equations expressed in the D-Q rotating frame of reference are shown in Eq.3.4 and Eq.3.5 [25] [30].

$$u_{sd} = R_s i_{sd} + \frac{d\psi_d}{dt} - \omega_e \psi_q \tag{3.4}$$

$$u_{sq} = R_s i_{sq} + \frac{d\psi_q}{dt} + \omega_e \psi_d \tag{3.5}$$

The terms i_{sd} and i_{sq} are the direct-axis and quadrature-axis stator currents of the machine. Likewise, ψ_d and ψ_q are the direct and quadrature-axis flux linkages terms and they are equal to:

$$\psi_d = L_d i_{sd} + \psi_f \tag{3.6}$$

$$\psi_q = L_q i_{sq} \tag{3.7}$$

Where ψ_f represents the rotor's permanent magnet flux linkage and as mentioned in the previous section, L_d and L_q correspond to the new direct and quadrature inductances of the PMSM. Replacing equations 3.6 and 3.7 into 3.4 and 3.5, the final expressions for the stator voltage of the PMSM in the rotor-oriented frame of reference are given in Eq.3.8 and Eq.3.9.

$$u_{sd} = R_s i_{sd} - \omega_e L_q i_{sq} + L_d \frac{di_{sd}}{dt} \tag{3.8}$$

$$u_{sq} = R_s i_{sq} + \omega_e L_d i_{sd} + L_q \frac{di_{sq}}{dt} + \omega_e \psi_f \tag{3.9}$$

However, the previous equations represent a linear model of the PMSM. This standard motor model is made non-linear when making the PMSM parameters current and temperature dependant. Thus, the current dependant terms $L_d(i_{sd}, i_{sq})$, $L_q(i_{sd}, i_{sq})$, $\psi_f(i_{sd}, i_{sq})$ and temperature T_{cu} dependant resistance $R_s(T_{cu})$ can be replaced into equations 3.8 and 3.9.

3.1.3. Electromagnetic Torque

By decoupling the stator currents, variables as the flux and the torque of the machine can be manipulated independently. An expression for the electromagnetic torque of the IPMSM can be derived from the power balance equation.

Starting from the three-phase voltages and currents feeding the machine, we obtain the expression in Eq. 3.10 for the input power of the electric motor. Computing the Park transformation as in Eq.3.11, to the set of three-phase voltages and currents in the power equation, we obtain that the instantaneous input power of the PMSM in the rotor-oriented frame of reference is given by equation 3.12.

$$P = v_a i_a + v_b i_b + v_c i_c = [v_a v_b v_c]^T [i_a i_b i_c] = {V_{abc}}^T I_{abc} \tag{3.10}$$

$$P = {V_{abc}}^T I_{abc} = \left[T^{-1}(\theta) V_{dq0}\right]^T \left[T^{-1}(\theta) I_{dq0}\right] \tag{3.11}$$

$$P = \frac{3}{2}(V_{dqn})^T I_{dqn} = \frac{3}{2}(v_{sd} i_{sd} + v_{sq} i_{sq} + v_0 i_0) = \frac{3}{2}(v_{sd} i_{sd} + v_{sq} i_{sq}) \tag{3.12}$$

Next, substituting equations 3.8 and 3.9 into 3.12, we obtain the power balance equation of the permanent magnet synchronous machine [22][30].

$$P = \frac{3}{2}\left[R_s({i_{sd}}^2 + {i_{sq}}^2) + \left(i_{sd}\frac{d\psi_d}{dt} + i_{sq}\frac{d\psi_q}{dt}\right) + \omega_e\left(\psi_f i_{sq} + (L_d - L_q) i_{sd} i_{sq}\right)\right] \tag{3.13}$$

Equation 3.13 is clearly divided in three representative components, each of them representing where the input power of the machine is allocated, or consumed. Starting from left to right, the first power term of the equation represents the stator copper losses. The second term is attributed to the variation of the magnetic energy. The last term, represents the mechanical output power of the motor.

$$P_{out} = T_e \omega_r \tag{3.14}$$

Considering equation 3.14 and defining ω_r as the mechanical rotational speed of the rotor where, ω_e=$p\omega_r$ the output torque of the machine is found dividing the mechanical power term in 3.13 by ω_r [22]. Therefore, the torque equation is expressed as:

$$T_e = \frac{3}{2} p \left[\psi_f i_{sq} + (L_d - L_q) i_{sd} i_{sq} \right] \tag{3.15}$$

From equation 3.15 it can be noticed it has two components. The first is the magnetic torque produced by the permanent magnet flux linkage and the quadrature-axis current. The second component is named as the reluctance torque and is a result of the PMSM's saliency. As a result, it can be said that IPMSM posses an extra torque component, whereas in non-salient SPMSM, the reluctance torque is eliminated from the expression since L_d=L_q.

The direct and quadrature-axis inductances are prone to variation due to magnetic saturation or as a response to the change of current. As a result, the torque is affected by the inductance change. Initially the relation between the current and the resultant magnetic flux is linear. However, once the core of the machine starts to saturate, the magnetic flux increases slowly when the current rises, meaning the inductance starts to decrease.
Similarly, as figure 3.2 shows, the direct-axis flux has a higher reluctance path. As a result, L_d is less sensitive to changes in i_{sd}. Conversely, in the quadrature-axis the path of the flux has lower a reluctance since it is mostly iron. In consequence, if i_{sq} grows large, is easier for the magnetic circuit to saturate and as a result L_q decreases drastically, as shown in figure 3.4.

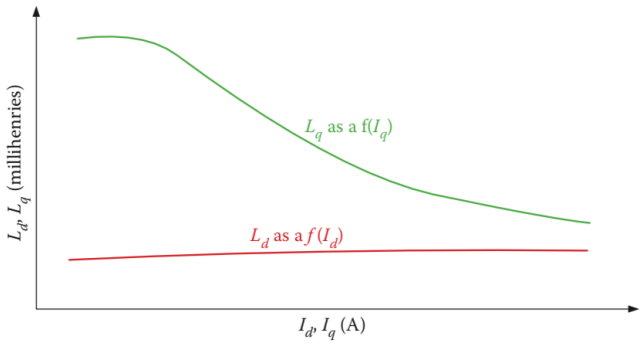


Figure 3.4: L_d and L_q as a function of currents i_{sd} and i_{sq} repectively. [3].

3.2. Control of a PMSM

Normally there are two types of control methods used for PMSM: Scalar control and vector control. And within these two methods, the three main ones are:

- Volt/hertz control in open loop.
- Direct torque control (DTC), closed loop.
- Field oriented control (FOC), closed loop.

Volt/hertz control is a scalar control method that operates in open loop, so feedback signals are not required. This is why is deemed as low cost, simple to implement and popular for low performance applications like pumps and fans. Likewise, its main drawback is the lack of control of variables such as current and torque [6] . Therefore for is not fit for traction applications as in electric vehicles.

Next, within the vector control strategies, is DTC. As it name says, it is aimed to control the torque and the flux of the machine directly without using a speed feedback signal, but estimating the controlled variables from the machine's currents and voltages. One of its advantages is the faster torque response when compared to FOC. However, it is more used for applications with induction motors, requires torque and flux estimators and presents torque and flux ripple [36].

Consequently, the chosen option for this application is FOC. FOC transforms a three-phase stator current system into two DC current components from which the torque and the flux of the machine are indirectly controlled. FOC is widely used in high performance applications and usually has two control loops, where the outer loop is the speed control loop. Therefore this control method does need a speed feedback signal.

3.2.1. Field Oriented Control (FOC)

As said before, FOC in principle decouples the indirect control of the flux and the torque by transforming a regular three-phase set of stator currents, into direct-axis and quadrature-axis current components. The direct-axis is aligned with the flux linkage vector of the permanent in the machine's rotor as seen in figure 3.5.

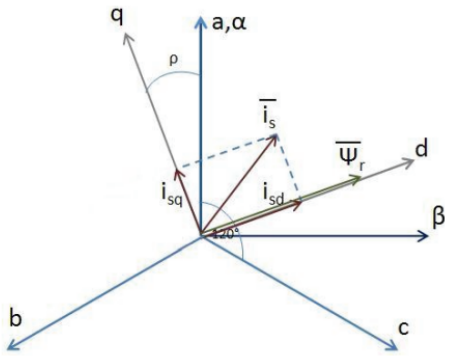


Figure 3.5: Vector diagram of D-Q reference frame [23].

As previously mentioned, in the implemented FOC method, the control of the torque and flux is done indirectly by controlling i_{sd} and i_{sq} currents. This is the first control loop. The outer control loop, as shown in figure 3.6 is the speed control loop. As a result, the desired speed of the motor is the input of the controller. The output of the speed regulator is taken as the desired or reference stator current vector amplitude. With the help of MTPA and FW control strategies, the i_{sd} and i_{sq} reference current pair is generated. The new reference currents are the input of the current control loop. The output of the current regulator are two D-Q voltage commands that will drive the inverter output voltage through a space vector modulation.

The maximum torque per ampere (MTPA) and flux-weakening (FW) control strategies are explained with more detail in the subsequent section. The MTPA control strategy is applied during the constant torque region when the motor is running at low speeds. The FW control takes place once the machine reaches its base speed and further.

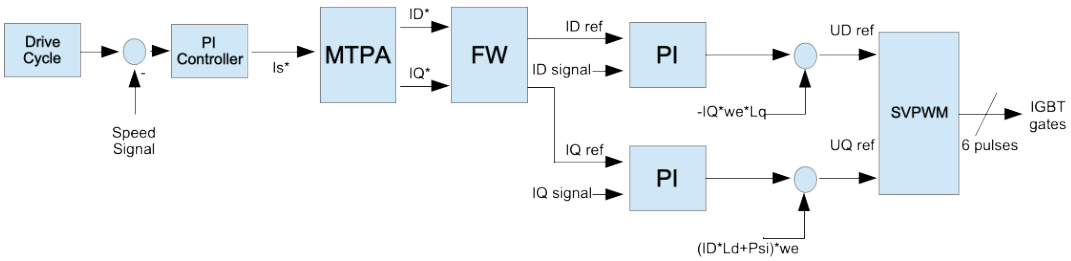


Figure 3.6: Field oriented controller block diagram

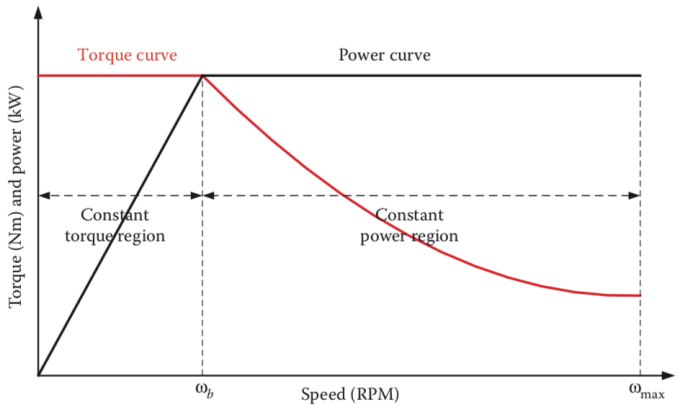


Figure 3.7: Torque vs. Speed curve for PMSM traction motors [3].

3.2.2. Maximum Torque per Ampere (MTPA)

When it is desired to produce the maximum torque possible at an specific current value, the torque equation for PMSM (see equation 3.15) has to be reviewed. As explained previously, for non-salient SPMSM the reluctance component of the torque is equal to zero. Therefore, the torque expression for SPMSM is reduced to equation 3.16.

$$T_e = \frac{3}{2} p \psi_f i_{sq} \tag{3.16}$$

As a result, is logical that the best option to control the torque of an inverter fed SPMSM is to keep $i_{sq} = I_s$ and $i_{sd} = 0$. Furthermore, given that IPM synchronous machines are salient machines and have a reluctance torque component available, the calculation of the maximum torque possible is influenced by how the reluctance and magnet excitation torque components are combined, and consequently, which direct and quadrature current pair is used. Observing figure 3.8, we obtain expressions 3.17 and 3.18 for i_{sd} and i_{sq} respectively, that are function of angle γ, often named as the torque angle [25]. I_s is the amplitude of the stator's phase current vector and γ in this case, is the angle between the current vector and the positive direct-axis.

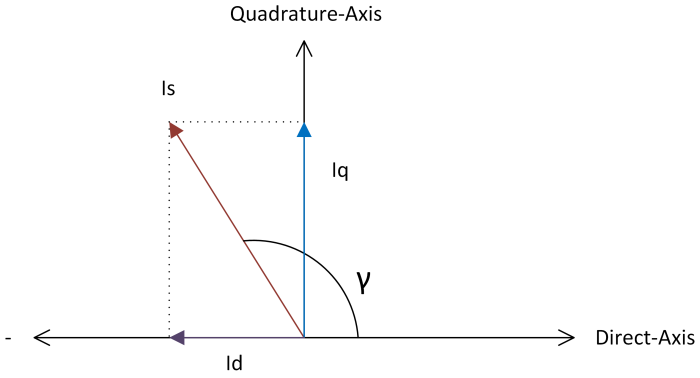


Figure 3.8: Torque angle γ .

$$i_{sd} = \hat{I}_s \cos\gamma \tag{3.17}$$

$$i_{sq} = \hat{I}_s \sin\gamma \tag{3.18}$$

$$\hat{I}_s = \sqrt{i_{sd}^2 + i_{sq}^2} \tag{3.19}$$

By replacing both equations 3.17 and 3.18 into the electric torque equation 3.15, equation 3.20 is obtained. It shows how the excitation and reluctance torque components are a function of the torque

angle, and if the angle value γ changes from zero to π also the two torque components do, as seen in figure 3.9 that is computed with the machine parameters from table 2.2.

$$T_e = \frac{3}{2}p\left(\psi_f \hat{I}_s \sin\gamma + (L_d - L_q)\hat{I}_s^2 \frac{\sin 2\gamma}{2}\right) \tag{3.20}$$

Looking at figure 3.9, it can be noticed that there are regions where the reluctance torque component is higher than the magnetic excitation torque. For non-salient SPMSM, the resulting electric torque waveform is the same as the magnetic excitation torque. This means that for salient machines the maximum possible torque can be higher in comparison, thanks to the additional reluctance torque. In order to obtain an expression for the maximum torque, the derivative of equation 3.20 with respect to angle γ is set to be equal to zero. The result is given by equations 3.21 and 3.22.

$$\frac{dT_e}{d\gamma} = \frac{3}{2}p\left(\psi_f \hat{I}_s \cos\gamma + (L_d - L_q)\hat{I}_s^2 \cos 2\gamma\right) = 0 \tag{3.21}$$

$$\psi_f(\hat{I}_s \cos\gamma) + (L_d - L_q)\left((\hat{I}_s \cos\gamma)^2 - (\hat{I}_s \sin\gamma)^2\right) = 0 \tag{3.22}$$

For further simplification, equations 3.17, 3.18 and 3.19 can be replaced in 3.22 and the result equation is 3.23.

$$2(L_d - L_q)i_{sd}^2 + \psi_f i_{sd} - (L_d - L_q)\hat{I}_s = 0 \tag{3.23}$$

Equation 3.23 is a second degree equation and has two solutions when solved for i_{sd}. Keeping in mind that L_q is larger than L_d for IPMSM and that the torque angle γ must be greater than $90°$ so that the reluctance torque always adds up to the overall developed torque [25], the chosen solution for the maximum torque is shown in equations 3.24 and 3.25.

$$i_{sd\_mtpa} = \frac{\psi_f - \sqrt{\psi_f^2 + 8(L_q - L_d)^2 \hat{I}_s^2}}{4(L_q - L_d)} \tag{3.24}$$

$$i_{sq\_mtpa} = \sqrt{\hat{I}_s^2 - i_{sd\_mtpa}^2} \tag{3.25}$$

The maximum torque per ampere control strategy does not consider the voltage operation limits but, the torque is constrained by limiting the magnitude of $\hat{I}_s$ as the maximum allowed operation current. For each value of $\hat{I}_s$, equations 3.24 and 3.25 show that there is a different i_{sd} and i_{sq} current pair that develops the maximum torque possible at a desired current reference.

Finally, using the machine data in table 2.2 and equations 3.24 and 3.25, the MTPA curve is calculated and shown in figure 3.10 for the whole range of allowed stator current I_s values. The point where the MTPA curve and the maximum I_s amplitude circle intersect, is referred as the point where the maximum torque of the machine is developed at speeds below base speed, while still operating inside the ”Constant torque region” as showed in figure 3.8. For speeds above base speed, the maximum torque of the machine decreases as the speed rises, entering in the flux-weakening region.

3.2.3. Flux-Weakening Control of a PMSM

When PMSM are used for traction applications as in electric vehicles, it is desirable that they can operate within a wide speed range. However, the MTPA control strategy cannot be used to control the PMSM for speeds above the base because of the back electromotive force (BEMF). The BEMF is proportional to the electrical speed of the machine ω_s and the permanent magnet flux ψ_f, as observed in equation 3.9. When the BEMF becomes larger than the stator voltage fed to the machine, the motor is no longer capable of drawing current, it loses power and therefore, it cannot generate torque.

Consequently, the motor speed cannot be increased unless the effect of the rising BEMF is reduced. One method to achieve this, is to counteract the magnetic flux of the magnets in the rotor of the PMSM. However, the magnet flux cannot be directly controlled. Instead, the air-gap flux can be weakened through the current in the direct-axis [28]. This procedure is named ‘Flux-Weakening’, and with its implementation an extension of the speed range of the motor can be achieved.

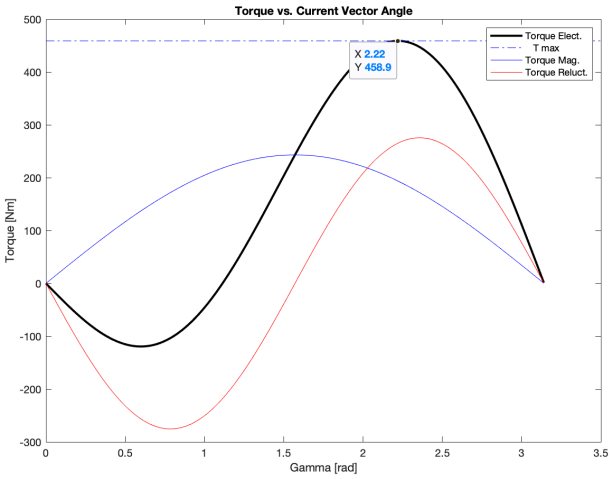


Figure 3.9: Electric torque decomposed into the magnetic and reluctance torques vs. Current vector angle

Taking a look back at equations 3.8 and 3.9 and rearranging them into their steady state expressions, we obtain equations 3.26 and 3.27.

$$u_{sd} = R_s i_{sd} - \omega_e L_{sq} i_{sq} \tag{3.26}$$

$$u_{sq} = R_s i_{sq} + \omega_e L_{sd} i_{sd} + \omega_e \psi_f \tag{3.27}$$

Another set of constraints for the operation of the PMSM at all times, are the voltage and current limits, expressed in equations 3.28 and 3.29 below [28]. Both constraints are meant for the operation of a PMSM, and refer to the maximum phase-voltage amplitude of the inverter and the maximum operation current of the system. Normally, what sets the voltage limits of a PMSM operation are the maximum phase-voltage from the inverter and the DC link voltage. The maximum current limit is as well, set by the inverter operation limits.

$$U_{s\_max}^2 \geq u_{sd}^2 + u_{sq}^2 \tag{3.28}$$

$$I_{s\_max}^2 \geq i_{sd}^2 + i_{sq}^2 \tag{3.29}$$

If the resistance of the stator's winding is neglected, and substituting equations 3.26 and 3.27 into equation 3.28, the expression in equation 3.30 is obtained.

$$\left(\frac{U_{s\max}}{\omega_e}\right)^2 \geq L_d^2 \left(\frac{\psi_f}{L_d} + i_{sd}\right)^2 + (L_q i_{sq})^2 \tag{3.30}$$

Equation 3.30 represents an ellipse centred at $(-\frac{\psi_f}{L_d},0)$ in the D-Q plane. This ellipse equation is referred as the voltage limit ellipse, and as the speed rises the ellipse shrinks towards its center point. The center of the ellipse, represents the value of the stator current at which the motor reaches its theoretical infinite speed. The voltage ellipse equation and the maximum current circle (equation 3.29) can be represented in the D-Q plane and represent both operation constraints.

Observing figure 3.11 it can be mentioned that if the voltage ellipse center is outside of the maximum current circle, the maximum theoretical speed cannot be reached without breaking the operation boundaries. Using the parameters from table 2.2 the center of the voltage ellipse is calculated to be in point $(-562.69, 0)$ of the D-Q plane. Since $I_{smax} = 600A$, it can be said that the center of the voltage ellipses of the machine is located within the current constraint and theoretically, the machine is capable to reach infinite speed.

Likewise, using the same machine parameters, the voltage ellipses for different speed values are plotted with the MTPA curve and the current limit circle in figure 3.12.

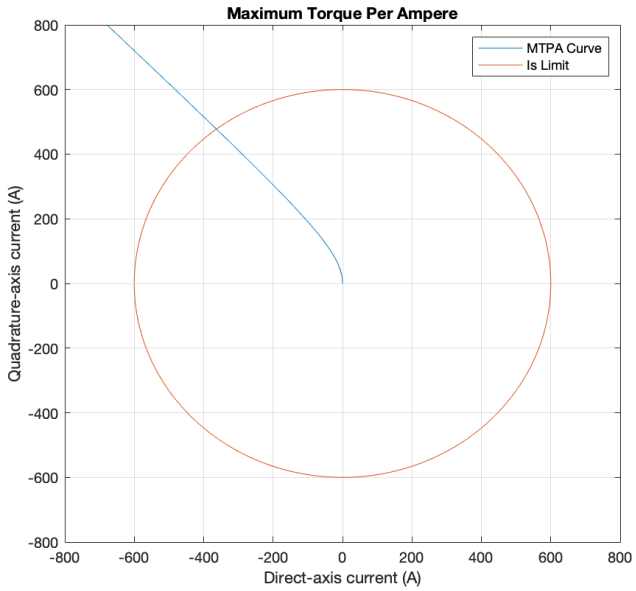


Figure 3.10: MTPA curve and maximum stator current circle.

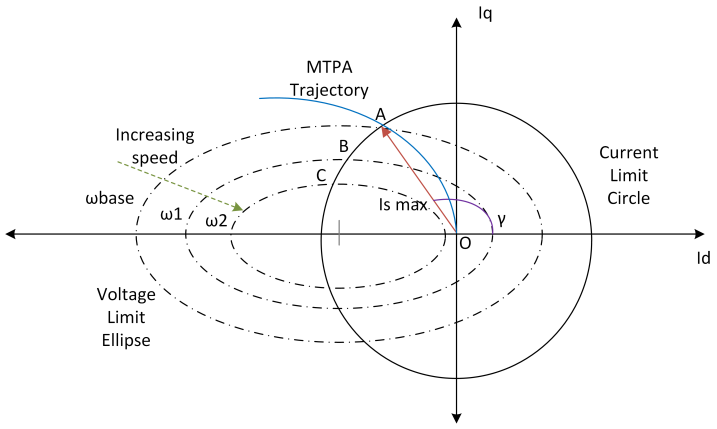


Figure 3.11: Voltage and current limits in the direct and quadrature currents plane.

The importance of figures 3.11 and 3.12 is that they give a clear view of how the current and voltage constraints of the PMSM intersect with each other and give an idea on which points of the i_{sd} and i_{sq} plane the machine can operate. As stated before in section 3.2.2, the intersection of the MTPA curve and the current limit circle, is the i_{sd} and i_{sq} current pair that corresponds to the maximum torque of the machine at the maximum current magnitude, this point is equal to point A in figure 3.11. The PMSM can operate from point O to point A over the MTPA trajectory from speed zero until reaching base speed ω_b. Once above base speed, the machine cannot generate the maximum torque as before and the MTPA strategy is no longer in use. In order to rise the speed from ω_b to ω_1 (see figure 3.11), flux-weakening is needed.

Flux weakening is achieved when the i_{sd} component of the I_s current vector grows towards the negative direct-axis. When i_{sd} becomes more negative, the $\omega_e \psi_f$ term which is the BEMF in equation 3.27 is counteracted for positive speeds. In other words, the machine is capable of extending its speed range when the negative i_{sd} current grows by increasing the angle γ or shifting the I_{smax} vector towards the negative direct-axis as for example point B for speed ω_1 and point C for speed ω_2 of figure 3.11.

Overall, the FOC control strategy is summarised as follows. First, the motor can operate following the MTPA trajectory for speeds within the zero to base speed range. Second, for higher speeds, the points the flux-weakening operation are the intersections of the respective voltage ellipses with

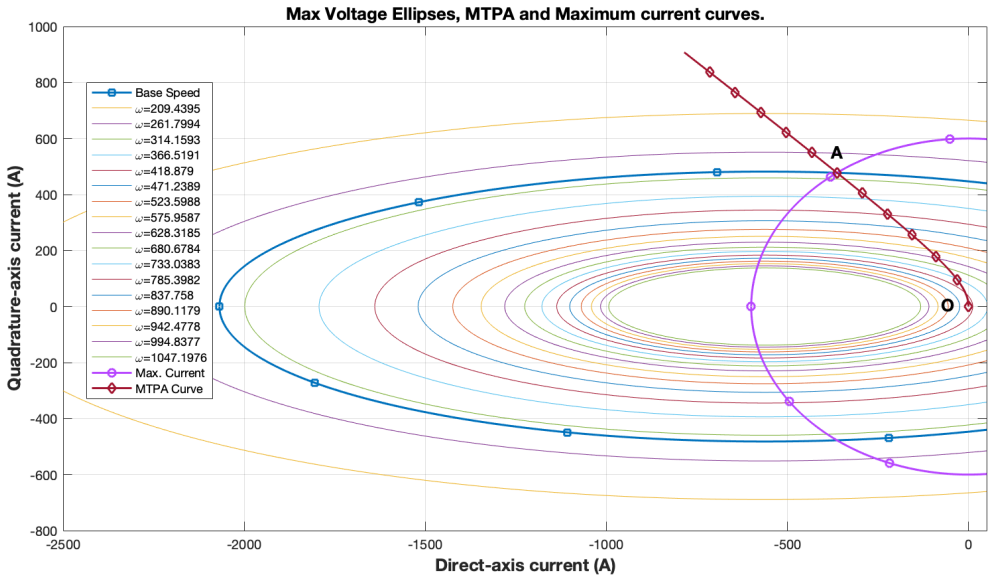


Figure 3.12: Maximum voltage ellipses according the PMSM's angular speed in rad/s.

the maximum current circle as shown in figure 3.12. Each one of the intersection points provides an operation point where the maximum voltage and maximum currents constraints are followed and additionally the maximum torque possible is generated [26].

3.2.4. Adopted Flux-Weakening Control Strategy

The flux-weakening control strategy implemented for this project was proposed by J.Wai and T.Jahns [33]. It is based on the control of the complimentary angle of the stator current vector. When the magnitude of the stator voltage of the PMSM (see Eq.3.28) grows almost to be equal to the maximum inverter phase-voltage, the complimentary angle of the vector I_s is reduced, shifting the current vector towards the negative direct-axis. How the control method is implemented will be further explained in section 4.5.2.

This flux oriented control strategy is intended for PMSMs operating above their base speed. The algorithm, as shown in figure 3.13, constrains the phase voltage of a PMSM which is being current-controlled.

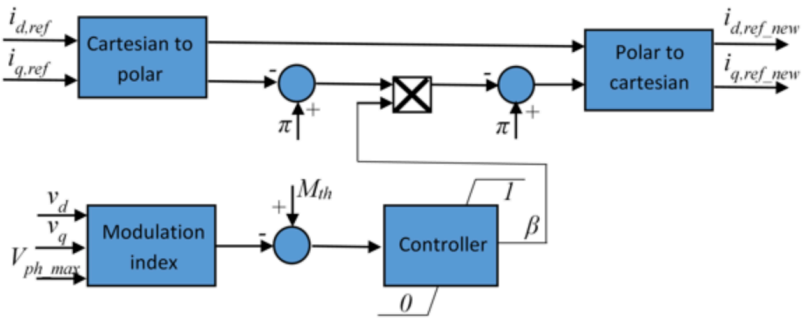


Figure 3.13: Flux weakening strategy scheme [33].

This algorithm decreases the phase voltage at the stator of the electric motor by changing the angle of the of the reference current vector. The angle adjustment, is done when the desired magnitude of the stator's voltage vector, is about to exceed its maximum limit.
When the motor's speed is reaching its base speed, it means that the BEMF voltage is about to exceed the voltage constraint that is stablished by the inverter and the battery voltage.

In the case of a PMSM driven by a three-phase VSI inverter, that is operating under a Space Vector PWM, the maximum phase voltage possible is:

$$U_{\max}^{ph} = \frac{U_{DC}}{\sqrt{3}} \tag{3.31}$$

The FW algorithm, contains an integral regulator which is fed by the difference between a user-set modulation threshold and a computed modulation index according to equation 3.32.

$$M = \frac{\sqrt{V_d^2 + V_q^2}}{U_{\max}^{ph}} \tag{3.32}$$

This difference quantifies how close is the stator voltage, to reach its operation limit. The following integral controller and anti-windup stage in figure 3.13, generate an a β factor between 0 and 1. When M is about to exceed threeshold M_{th}, β tends to zero, reducing angle 180-γ and making more negative the I_D component of the reference current vector.

The beta factor is used to add a fraction of the I_s vector's supplementary angle to angle γ. Taking a look at equation 3.9, it can be noticed that a more negative I_{sd} current, counteracts the effects of the speed dependent BEMF. On the other hand, when the stator voltage is within the limit, β tends to be one, and the current vector's angle remains unchanged as the initial reference.

4

Implementation of EV Powertrain HIL Simulation

After describing the key aspects considered in this projects on how to create a PMSM model and a vehicle model, in this chapter, the process of setting up a real-time simulator with a hardware in the loop controller is unfold.

In the previous chapters, a description of a three-phase inverter model circuit has not been given. The reason is the simplicity of its implementation for the current the powertrain model. In this chapter an explanation on how the inverter and the IPMSM model were simulated using OPAL-RT's real time simulator is given.

The physical dynamics of the powertrain, are represented by the gear transmission model equation previously described in chapter two.

The concepts of a software in the loop or hardware in the loop controller are briefly described. Furthermore, once a powertrain model is built, the different system signals are mapped, scaled and routed to the different outputs of the real-time simulator. Next, they are processed by an external controller or looped back into the real-time model, to a simulated controller.

The controller, reads signals from the simulated powertrain model, and enforces a field oriented controller (FOC). Two FOC strategies are implemented as explained in chapter three, a maximum torque per ampere and a field weakening strategy.

4.1. OPAL-RT and Real-time Simulation

The following section, introduces the OPAL-RT ecosystem. The combination of a real-time simulator and model based design simulation software, provides the vital means for a real-time and consequently hardware in the loop simulation. In addition, a brief description of CPU-based and FPGA-based simulations is given, and why the present work implementation is based on the latter.

4.1.1. Real-time simulations

Simulation software of widespread use as Simulink, is used for validation of either the accuracy or the performance of mathematical, physical and designed systems. Nonetheless, these type of simulators do not run in real-time, but in an "offline" mode with computing times longer than the reaction of the real system itself. Normally, this prevents simulated models to be interfaced with external hardware, limiting the range of application of the simulator. If it is desired to interface a simulated model of a system and external hardware, as for example a controller board, the simulation software needs to receive data from the external hardware and compute the model's response within the same time-step, meaning that the required calculations must be done in less than the simulation time-step. If a model is desired to be simulated in real-time, the simulation software will have to produce the internal variables and their outputs in the same time as the physical counterpart.

4.1.2. Types of Real-time Simulations

Real-time simulators are mainly divided in three modes: Rapid control prototyping (RCP), hardware in the loop (HIL) and software in the loop (SIL).

Rapid Control Prototyping:
RCP is the denomination given to the application where a physical plant or system is connected to a controller that is running inside a real-time simulator as detailed in figure 4.1. An advantage of RCP is that simulated controller offers flexibility, less required implementation time and debugging is less complex since every internal state or variable is available and easier to access.

Hardware in the Loop:
In HIL applications, an external controller device is interfaced to a virtual plant running in a real-time simulator, as shown in Fig.4.2. In addition, a HIL controller allows for early testing, as well, developing a virtual plant is obviously cheaper and constant since a simulated system offers repeatability of results and the opportunity to test scenarios that are not possible in real hardware, as extreme events or failure.

Software in the Loop:
SIL is said to be a combination of RCP and HIL modes, where both controller and plant are executed in a real-time simulator. In this type of application, the times of execution less important since a slower or faster simulation in comparison of the physical counterpart has no impact in the validity of the simulation [5].

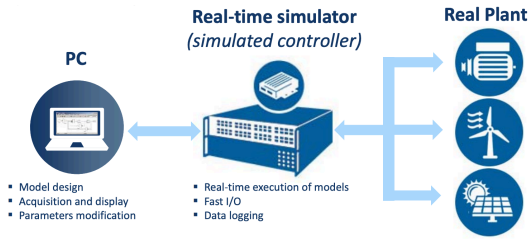


Figure 4.1: RCP architecture [5].

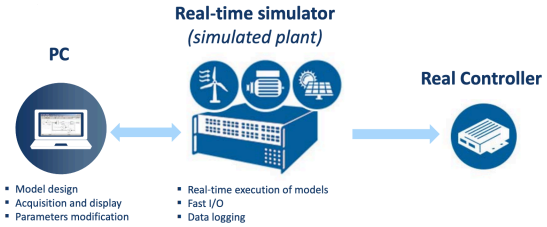


Figure 4.2: HIL architecture [5].

4.1.3. OPAL-RT Real-Time Simulator

The OPAL-RT system consists of two main components, the host computer and the real-time simulator. The host computer contains OPAL-RT's software tool, RT-Lab. The host computer is a PC where compatible versions of RT-LAB and Matlab Simulink run together. RT-Lab allows to import or edit Simulink models and then transform them to a real-time application through code generation done by RT-Lab itself. On the other hand, the real-time simulator is the hardware component of the OPAL-RT system and is the part responsible of the real-time execution of the previously made Simulink model. The real-time simulator is often called the "target" computer and the communication between the target and the host computer, is done through TCP/IP protocols.

4.1.4. Simulink to real-time simulation

Once the plant model has been built in Simulink and validated offline, is necessary to import said model into an RT-LAB project. However, it does not end there. Simulink models must be rearranged

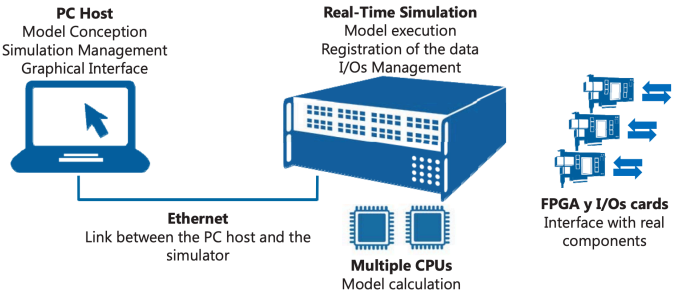


Figure 4.3: Master and console subsystems [2]

.

into subsystems. There are three types of subsystems. Each subsystem is computed on a different computation node in the OPAL-RT system, as shown in figure 4.3.

- Master subsystem.
- Console subsystem.
- Slave subsystem.

In all RT-LAB projects there is only one console and master subsystems and the addition of a slave sub-system is optional.

In more detail, the master subsystem is conformed of the mathematical operations, computational blocks, varying signals and I/O blocks of the real-time model. The slave sub-system is similar to the master but is used when simulating large plant models and the user desires to distribute the model instead of keeping it in one sub-system entirely. Both the master and slave subsystem are executed on a CPU core of the real-time simulator.

In comparison, the console sub-system of the RT-LAB project is the interface between the user and the simulated model and it contains the set of parameters of commands the user desires to change or send to the plant model. As well, this subsystem is executed on the host computer and is where scopes are placed to monitor the real-time model's signals if desired [2].

A real-time simulation can only be executed in a fixed-step mode. This means that the model is solved by the simulator at regular time intervals The fixed-step size necessary for the simulation depends of the hardware capabilities of the simulator and the dynamics of the model.

Consequently, the question on how to choose a determined time-step arises. The rule of thumb is to achieve 10 or 20 samples per period of an AC or fast changing signal inside the model. And this obviously depends on the plant model that is being simulated. For example, mechanical application run in the order of 1 ms time-step, whereas an electrical circuit model on average runs at around 50 μs.

Therefore, for the present application described in this document, the general simulation time-step is calculated considering a powertrain model with a three-phase IGBT inverter, with a switching frequency of 5 kHz, as shown in equation 4.1.

$$T_{solver} = \frac{1}{20 \cdot f_s} = 10\mu s \tag{4.1}$$

For more information on how to set-up a real-time simulation using RT-LAB please refer to RT-LAB's user manual found at OPAL-RT's website.

4.1.5. The OP5700

The OP5700 real-time simulator at the EEMCS faculty of the Delft University of Technology, as seen in figure 4.4, is the device used for the implementation of the HIL simulation of the EV powertrain model. The relevant specifications of the OP5700 used are shown in table 4.1.

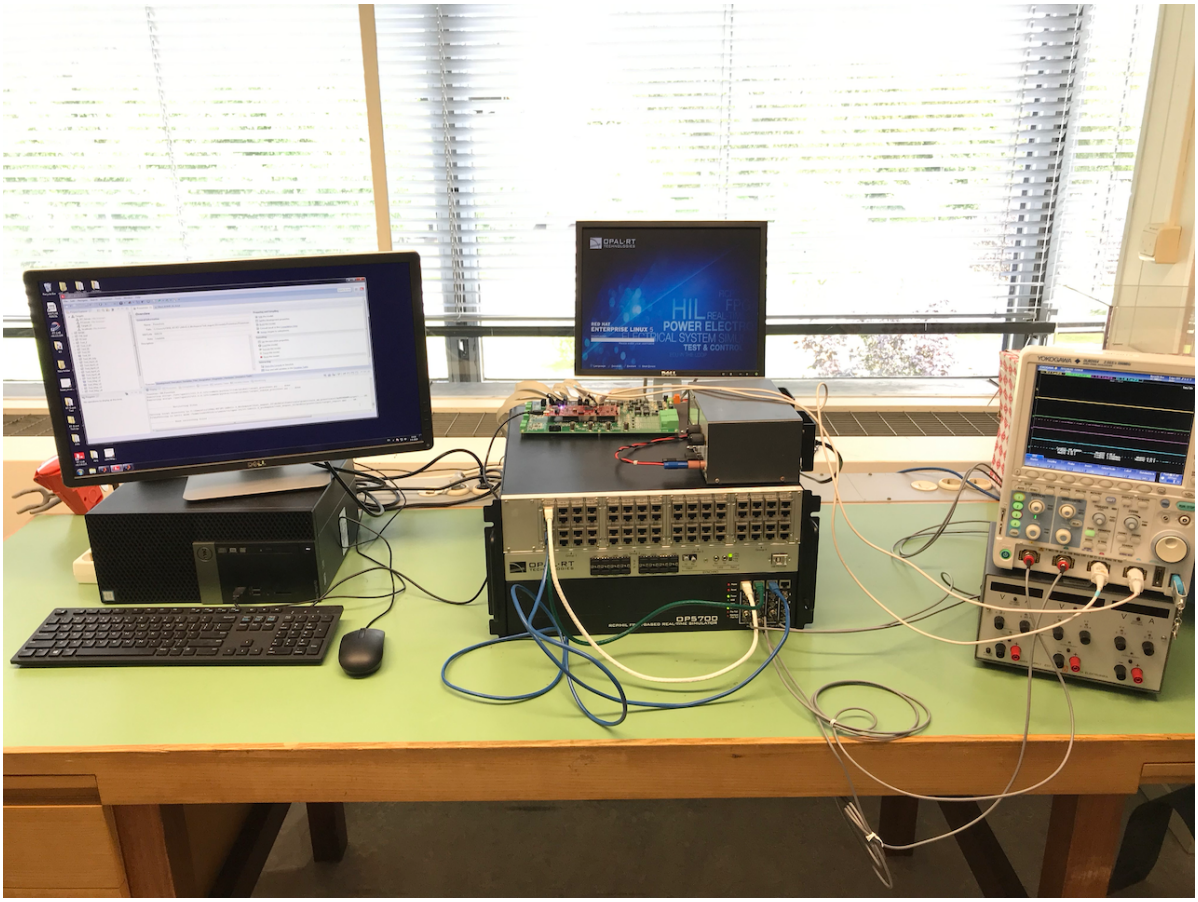


Figure 4.4: OP5700 real-time simulator and host machine station in the laboratory.

4.1.6. CPU vs FPGA based HIL Simulations

Real-time simulations can be performed in the CPU core or in the FPGA card of an OPAL-RT simulator device. Even though a complex plant model might be running in real-time, the nature of CPU-base simulations limits the minimum allowed time-step to 5-10 μs and in some HIL applications the time-step is constrained to no less than 25 μs. This represents a serious drawback for the simulation platform when is desired to simulate fast switching circuits as power converters. In the same way, the simulation of high frequency converters in HIL applications goes in hand with two basic requirements:

- Low latency between the external controller and the virtual plant.
- High PWM sampling resolution.

Latency is defined as the time delay between the command from the HIL controller and a change being detected at the output of the simulated plant. To precisely sample a PWM signal, a high sampling resolution is critical to avoid undesired effects as jitter [11]. Consequently, a HIL application based in CPU-level simulations, will fail to meet the low latency and high sampling frequency requirement.

Conversely, FPGA-based HIL simulations present reduced latencies and high sampling rates because of time-steps that can go as low as 10 ns [2].

Therefore, an FPGA-based HIL simulation is the chosen solution for the present work. Since an external controller will be sending PWM signals to a real-time simulation of a three-phase inverter, it is desirable that the model is running at a small time-step so that the PWM signal is properly sampled to the IGBT's gates. Likewise, a PMSM model that is driven by said inverter, also needs to be simulated at a low time-step to secure low latency and present a fast reaction under the controller commands.

Items	Quantity	Description
Chassis Type	1	OP5700
CPU	1	Intel Xeon E5, 3.2 GHz
Total cores	8	
Memory	2	8 GB
OP5330 Module	1	16 Analog Outputs
OP5340 Module	1	16 Analog Inputs
OP5353 Module	1	32 Digital Inputs
OP5360-2 Module	1	32 Digital Outputs

Table 4.1: OP5700 simulator specifications

4.2. Inverter and PMSM Real-Time Simulation in RT-LAB

There is more than one option to simulate an electrical circuit using the OP5700 + RT-LAB platform. It all depends on the necessities of the simulation, the complexity of the circuit and how small the time-step of the simulation is desired to be. As mentioned in the previous section, for the purpose of a HIL simulation, the external hardware will exchange information with the simulated EV powertrain model. Therefore, the simulation time-step needs to be as small as possible. Consequently an FPGA-based HIL simulation will be implemented in the present description.

Using RT-LAB, the most used software options available are the RT-Events and the eFPGASIM/eHS libraries. RT-Events provides the option to simulate a DC-AC inverter circuit with a fixed time-step as low as 10 μs. Nevertheless, the RT-Events library offers only CPU-based simulations. In comparison, the eHS software tool is capable of simulating electric circuits at FPGA level with a smaller time-step of 250 ns. Additionally, according to C. Dufour et al [11], the RT-Events blockset is presented as a solution for fully numerical real-time applications (e.g. SIL), whereas, in HIL applications is recommended that fast switching signals from external hardware, are sampled by an Opal-RT FPGA card. Consequently, the chosen software solution is the eFPGASIM/eHS library. The eHS solver, allows to create power electronic converter models in Simulink and program them into the OP5700's FPGA. Likewise, the electrical machine model is programmed into the FPGA card with a PMSM block from the eFPGASIM library.

To design and implement a real-time inverter and machine model, there are four main steps to follow:

1. Build and set-up the RT-LAB project.
2. Add and set the parameters of the eHS and eFPGASim electric motor blocks.
3. Build the an eHS inverter circuit.
4. Establish a software link between the simulated electric motor and inverter circuit.

The first two items in the list are omitted in this document since they are general procedure steps on how to initialise an RT-LAB project.
Generally speaking, the goal of this part of the implementation process, is to run a simulation of the model topology described in Figure 4.5. The inverter and electrical machine models are executed in the FPGA of the OP5700 exchanging voltage and current signals between them.

4.2.1. Three-phase Inverter Model

The 'eHS Gen3 CommBlk' block located in the eHS library, is the tool used to model the three-phase IGBT inverter in the FPGA. However, the inverter circuit is built in a separate Simulink file and the block is the link between the CPU and model running in the FPGA of the OP5700. The eHS solver computes the minimum time-step allowed for the FPGA to execute the inverter electrical circuit. Normally, the time-step depends on the amount of sources and passive elements in the circuit. The minimum time-step allowed for the current implementation is $Ts = 250n$s, as shown in Figure 4.6.

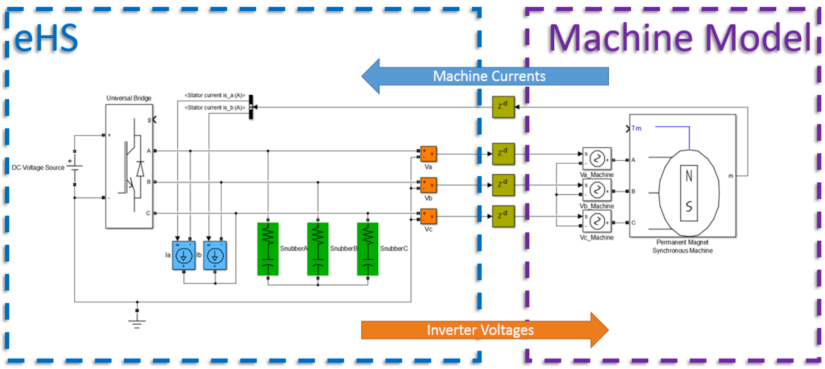


Figure 4.5: Circuit partitionning between the eHS and the machine model [18].

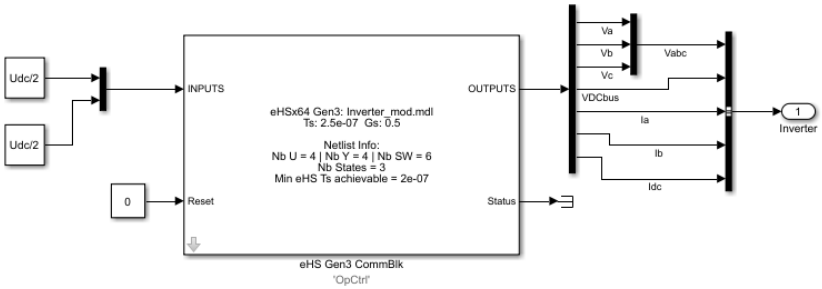


Figure 4.6: eHS Gen3 with loaded inverter circuit file.

The inverter circuit in figure 4.7 is coupled to the PMSM model by exchanging voltage and current signals. These are listed in table 4.2. The information exchange from and to the inverter model in the FPGA is described in figure 4.8. Signals $U01$ and $U04$ are constants in the CPU's powertrain model, and they emulate the EV battery voltage. On the other hand, signals $U02$ and $U03$ represent two controlled current sources (see Fig.4.7) inside the inverter. They insert phase currents A and B from the PMSM model, establishing the link between the eHS inverter and motor model (seen in figure 4.5).

On the same note, The voltage measurement blocks seen in figure 4.7, obtain the inverter's three-phase line voltages and map them through signals $Y01$, $Y02$ and $Y03$. These are linked externally to the PMSM model, and used to feed the three-phase winding of the PMSM that drives the electric vehicle.

There are six IGBTs arranged in the three-phase inverter in the Nissan Leaf model. As seen in figure 4.8, the gate driving signals are inputs of the inverter model and have been configured to be sampled by the FPGA card directly from the first 6 digital inputs of the OP5700 real-time simulator. The source is the controller commanding the real-time model. The configuration of the blocks used from the eHS library to implement the inverter simulation, is given in detail in the appendix section A.1.

4.2.2. PMSM Model

The permanent magnet motor of the EV powertrain model is simulated using blocks from eFPGASIM library. Shown in figure 4.9, the 'PMSM Motors VDQ' block is the means to set the machine parameters in table 2.2 into the PMSM model. Additionally, the 'Machine Interface' block routes the three line voltages from the inverter circuit, into the winding of the PMSM model.

As mentioned in the previous section. The inverter model is linked to the PMSM model through an exchange of line voltages and phase currents. These signals are simulated in real-time in the FPGA. However, the PMSM motor block, also receives signals from the powertrain model running in the CPU. For example, the angular speed of the rotor is generated in the vehicle's transmission model, simulated in the CPU. It must be mentioned that time constants of mechanical systems are larger than electrical

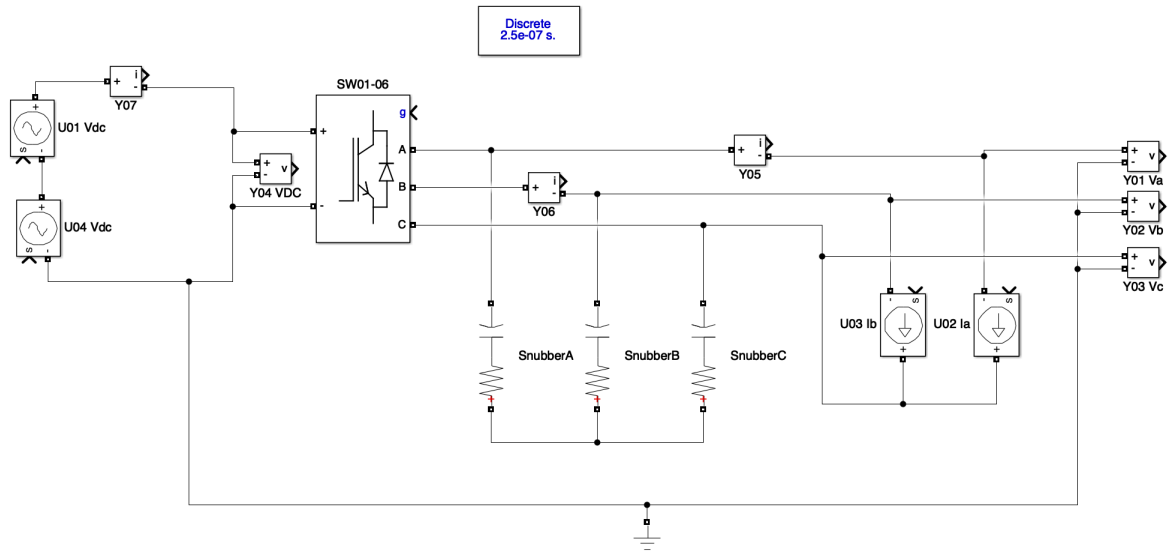


Figure 4.7: eHS three-phase inverter circuit for the powertrain model.

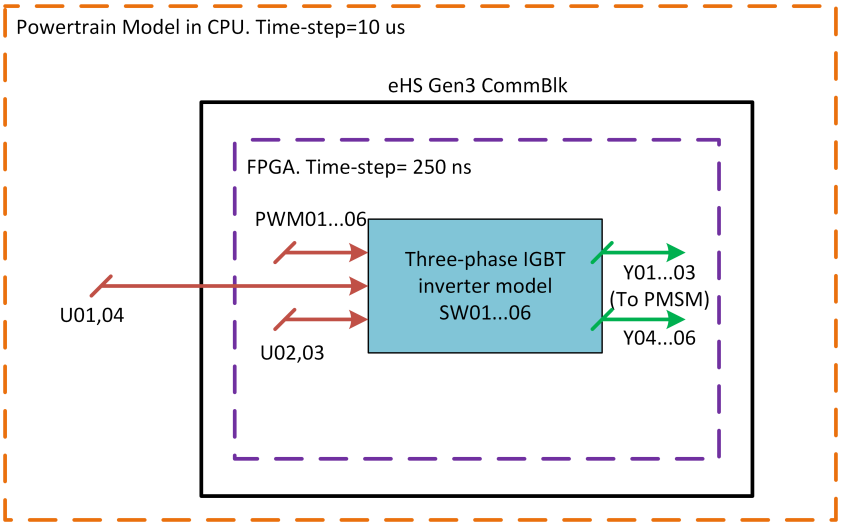


Figure 4.8: eHS three-phase inverter model signal diagram.

ones. In other words, changes in the variables of the physical model of the powertrain do not occur as fast as the variables from the electric motor or inverter circuit might. Therefore, it is appropriate that the transmission and the resistance forces model are simulated at a larger time-step in the CPU.

Figure 4.10 illustrates the origin of the signals going into the PMSM model. The rotor speed signal is mapped from the transmission model. Conversely, the $Y01$, $Y02$ and $Y03$ voltage signals are generated in the inverter. From the FPGA, signals as the electrical torque and rotor position are generated and transmitted to the CPU. Nevertheless, the motor's currents I_a and I_b are directly transmitted to the FPGA inverter model.

Signal Name	Type	Function
U01	Input	Battery input voltage
U02	Input	PMSM phase A line current
U03	Input	PMSM phase B line current
U04	Input	Battery input voltage
Y01	Output	PMSM phase A line voltage
Y02	Output	PMSM phase B line voltage
Y03	Output	PMSM phase C line voltage
Y04	Output	DC link measured voltage
Y05	Output	Inverter phase A line current measurement
Y06	Output	Inverter phase B line current measurement
Y07	Output	DC link measured current

Table 4.2: Input, output, and measurement signals in three-phase IGBT inverter model.

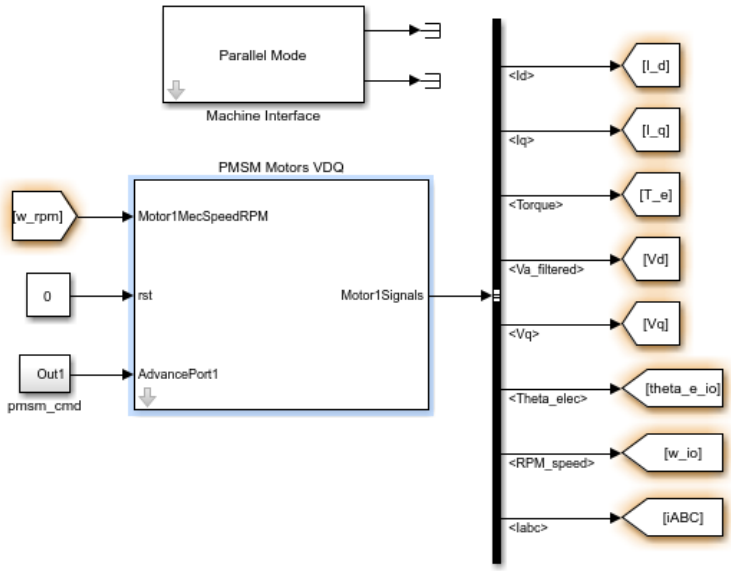


Figure 4.9: PMSM and Machine interface blocks from eFPGASIM library.

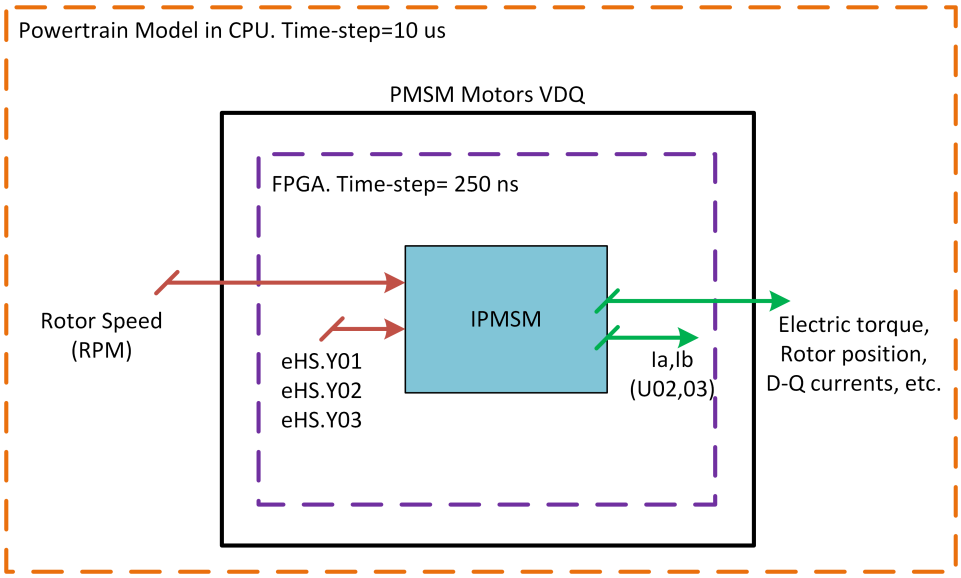


Figure 4.10: PMSM model signal diagram.

4.3. Gear Transmission Model

The EV's gear transmission model and the resistance forces of the vehicle, are simulated in the CPU of the OP5700. They are modelled as displayed in Figure 4.11.

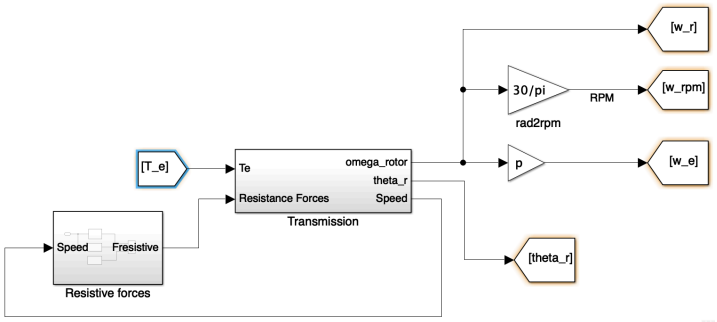


Figure 4.11: Mechanical System model of the EV powertrain.

The equation that models the dynamics and the movement of the vehicle was previously developed in Eq. 2.13. The final expression is shown again in equation 4.2, and it has been arranged in a way that all the variables are referred to the motor side of the transmission expressed in terms of the rotor's angular acceleration and the combined rotational inertia of the system, J_{tot}. Although equation 4.2 can be expanded to include more vehicle variables, as the vehicle's acceleration, it has been simplified as shown in figure 4.12 to avoid algebraic loops and consequently, errors in during the compilation of the model in RT-LAB.

$$T_e - \frac{r_w}{G_r} F_{res} = J_{tot} \frac{d\omega_m}{dt} \tag{4.2}$$

The term F_{res} accounts for the resistance forces exerted over the vehicle (see chapter 2), and T_e represents the electrical torque signal that is generated in the PMSM model. G_r and r_w are constants that stand for the Nissan Leaf's reduction gear ratio and the radius of the wheel, respectively. Lastly, the term ω_m represents the angular speed of the electric motor.

Using equation 4.2, is easier to obtain the PMSM's rotor speed by integration of the angular acceleration, as shown in figure 4.12. As a result, the motor speed in RPMs can be obtained and used as an input signal for the PMSM FPGA model.

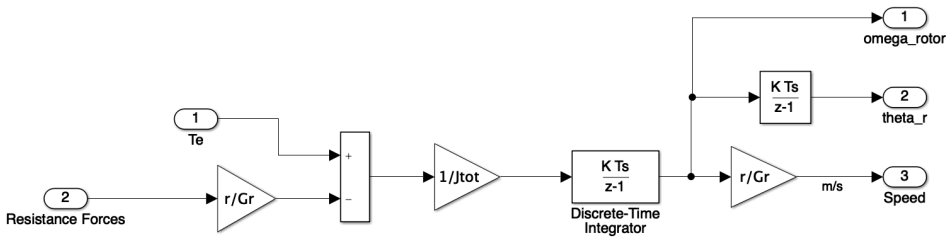


Figure 4.12: Powertrain's transmission model.

4.4. General Control Strategy

Whether the operation is using a hardware or simulated controller, said controller will have to receive signals from the simulated plant and send back a command. Figure 4.13 shows a scheme of the infor-

mation exchange between the powertrain model and the controller.

In the controller board, the speed and motor FOC control algorithms are executed. The speed controller receives a desired vehicle speed signal and generates a desired current reference. Next, from the current reference the MTPA and flux-weakening control strategies in chapter 3, are implemented to control the operation of the vehicle's motor. Finally, the controller computes a voltage vector command that will be given to the powertrain model in the form of a space vector pulse width modulated (SVPWM) signal.

In the case of a SIL simulation, the external controller is replaced by a simulated version and the control algorithms are executed inside the real-time simulator. The SVPWM signal is delivered to the digital inputs (DI) of the OP5700 device where the powertrain model is running in real-time.

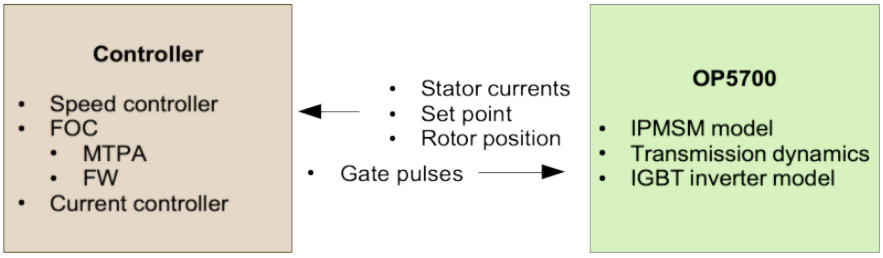


Figure 4.13: Overview of the information exchange in the powertrain simulation.

Other signals as motor currents and rotor position, are forwarded to the controller from the analog output (AO) ports of the OP5700. A description on how to to transmit simulated signals to the AO ports of the OP5700 and how to sample signals into the real-time model from the AI and DI ports, is given in the appendix section B.

4.5. Simulated Controller Implementation - SIL Controller

After designing the vehicle's transmission and resistance forces, the three-phase inverter and the PMSM model for real-time simulation, the next stage is to design a motor controller according to the strategies mentioned in chapter 3. Before doing a HIL simulation of the EV powertrain, a SIL powertrain simulation is implemented as a preliminary design phase. The purpose of the SIL powertrain simulation in this project is primarily to test the control algorithm and strategies together with the powertrain model, before implementing an external controller.

The proposed SIL simulation, will execute the powertrain and the controller in the OP5700. However, the signal exchange between these two is implemented through the simulator's IO ports. In Figure 4.14, the controller, powertrain model and the signal disposition is shown. The real-time controller simulation samples the powertrain signals that have been looped back into the model from the AI module of the OP5700. The rotor speed, electrical angle and the phase current signals, have been scaled as detailed in section B and routed to the AO module directly from the FPGA motor model.
The feedback signals that are looped back to the AI module are rescaled to their original magnitudes before being sampled by the simulated controller subsystem.

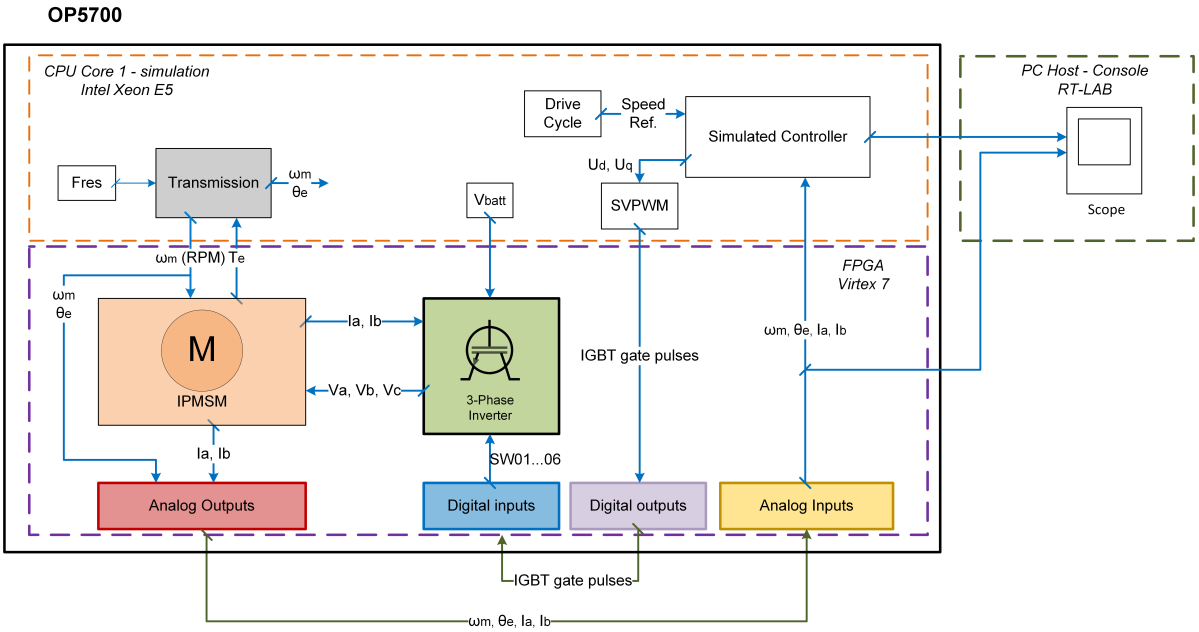


Figure 4.14: SIL Powertrain Simulation Diagram

Figure 4.15 shows the implemented simulated controller subsystem. The subsystem is triggered at a 5 kHz frequency. Therefore, the input signals are sampled, and the outputs generated every 200 μs. Unlike the rest of the CPU simulation with a 10 μs time-step or the inverter and PMSM FPGA models that run at 250 ns, the controller is synchronized with the 5 kHz switching signal that drives the inverter IGBTs. Some controller inputs must be scaled first. For instance, the speed reference and feedback signals must be in rad/s units. The electrical angle must be in radians. Also, the feedback stator currents must go through a Park transformation and expressed in the D-Q frame of reference before implementing the FOC control. Finally, from the console subsystem it is possible to monitor the simulated powertrain and controller variables.

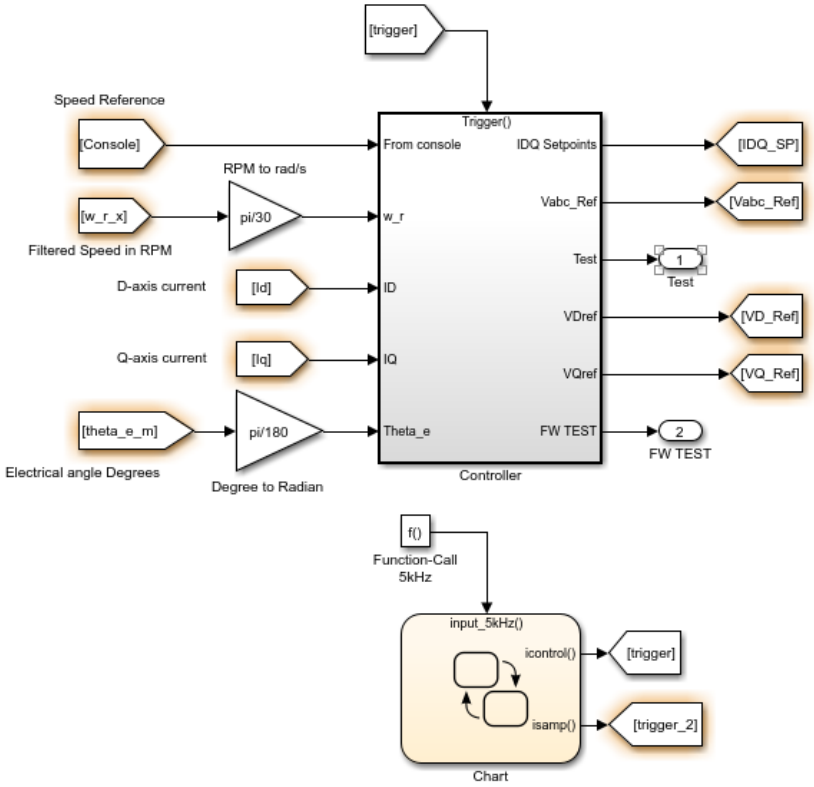


Figure 4.15: Simulated controller subsystem

The description of each of the internal functional blocks of the controller subsystem in figure 4.16, is presented in the next section.

4.5.1. Speed Controller

The implemented speed controller consists of a discrete proportional-integral regulator. For the case of the SIL controller, the speed control is inside of the general controller subsystem, simulated in the OP5700's CPU. The speed regulator samples the reference and feedback speed signals, as shown in figure A.10. The reference signal source can be either a 'Drive Cycle Source' block or a step input command. Internally, the reference signal is converted to rad/s units using the transmission equation on Eq.2.13. On the other hand, the feedback signal is obtained from the AI module and then filtered and scaled to rad/s units.

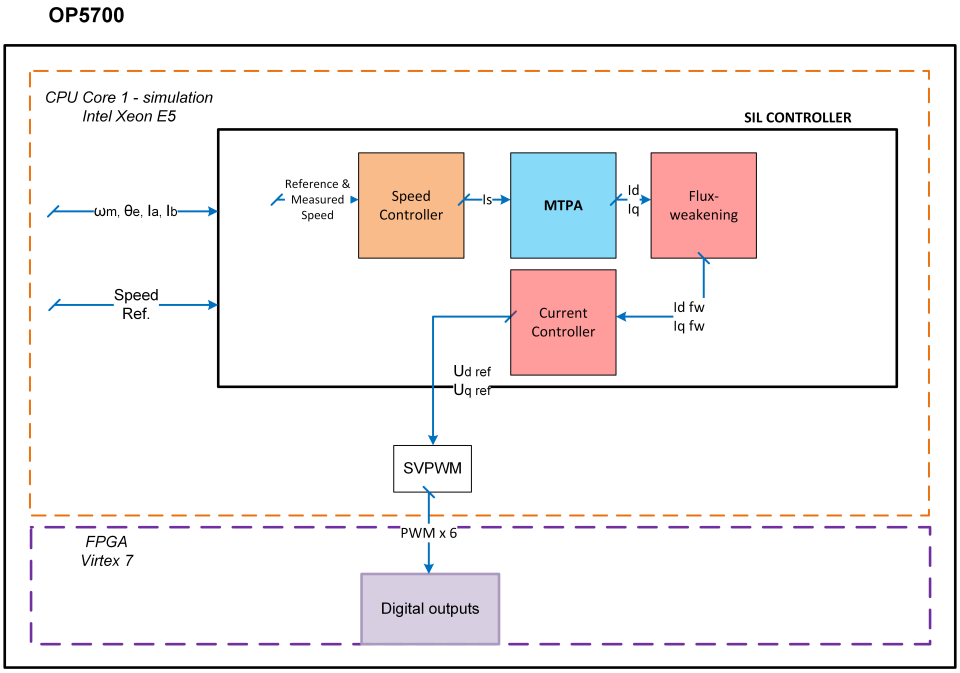


Figure 4.16: Simulated controller block diagram

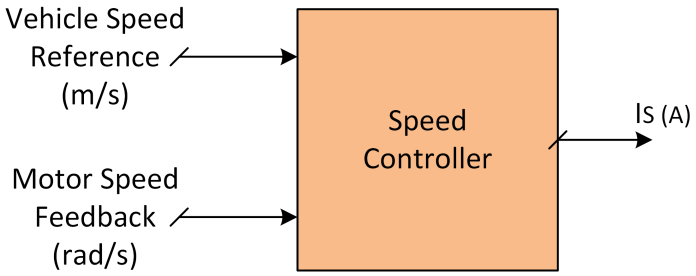


Figure 4.17: Speed controller diagram.

The output of the discrete PI block is equivalent to a reference current vector magnitude I_s. This current set-point is the input of the Maximum Torque per Ampere strategy subsystem. The implemented speed controller model is shown in figure A.10 in the appendix section.

4.5.2. Maximum Torque per Ampere Control Strategy

Following from chapter 3 and MTPA equations 3.24 and 3.25. The MTPA control strategy samples the desired stator current magnitude I_s from the speed regulator's output. I_s is decomposed into direct-axis and quadrature-axis reference currents. As a result, a reference I_{dx} and I_{qx} current pair obtained. From them, the desired angle γ^* is computed. Together, the I_{dx} and I_{qx} components maintain the same initial magnitude I_s, but additionally, they can develop the maximum torque possible for a particular current magnitude. Therefore, if operating below base speed, the PMSM can reach the reference speed at the maximum torque. For higher motor speeds, the flux-weakening algorithm will manipulate the angle γ^* to counter-act the increasing BEMF. The implemented MTPA model, is shown in figure A.11 in the appendix section.

4.5.3. Flux Weakening Control Strategy

The Flux-Weakening (FW) control algorithm of the simulated controller is implemented according strategies proposed by Courtay and Kelly [8] and Wai et al[33], previously described in chapter 3. This flux oriented control strategy is intended for operation of the electric motor above its base speed. Basically,

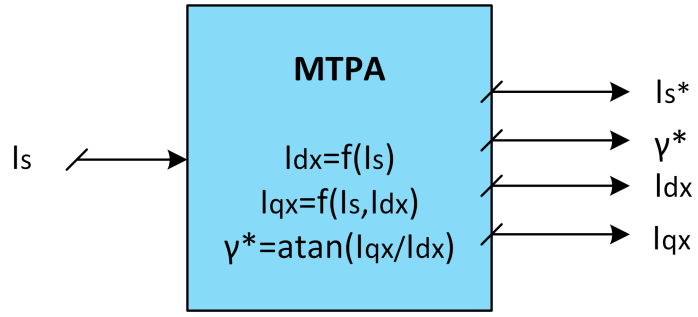


Figure 4.18: MTPA subsystem diagram.

the proposed algorithm, as shown in figure 3.13, constrains the phase voltage of a PMSM which is being current-controlled.
The diagram in figure 4.19 represents the implemented FW controller. The input I_{dx} and I_{qx} current components are the result of the executed MTPA strategy, described in the previous section. From them, the algorithm obtains the current vector's magnitude I_s and the angle γ. Next, from $U_{dcontrol}$ and $U_{qcontrol}$ voltage reference components, the modulation index is obtained, as in equation 3.32.

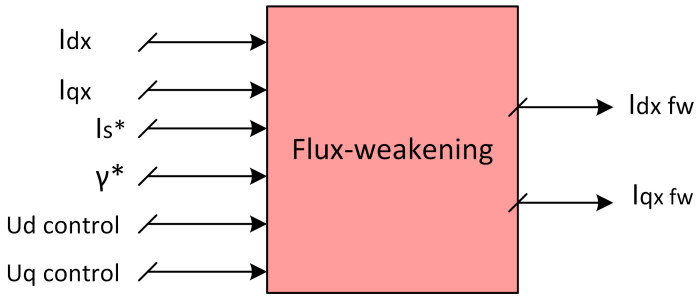


Figure 4.19: Flux-weakening subsystem diagram.

The implemented FW algorithm is shown in Figure 4.20. As can be seen, the stator voltage's modulation index is compared against a 'Modulation Index Threshold.' The error goes through an integral controller with anti-windup that generates a number between zero and one, named as $Beta$. If M, the calculated modulation index, becomes close to the threshold 0.97 (as in Fig.4.20), then $Beta$ tends to zero, and as a result, the complimentary angle of γ is reduced.

When the complementary angle $(\pi - \gamma)$ decreases, γ increases. Therefore, the current vector I_s shifts towards the negative direct-axis, making I_{dx} more negative and I_{qx} smaller. As a result, the $\omega_e \psi_f$ term in equation 3.9 is subtracted, and the stator voltage of the PMSM is kept within the limit. Finally, the PMSM is able to increase its speed range above the base speed.
Consequently, the new I_{dxfw} and I_{qxfw} current components are obtained and limited to be kept within the machine's operation boundaries. Both new I_{dxfw} and I_{qxfw} set-points are the inputs to the following current controller stage. However, if the PMSM operates below base speed, the FW algorithm does not intervene, $Beta$ is equal to one, and the initial I_{dx} and I_{qx} components remain unchanged.

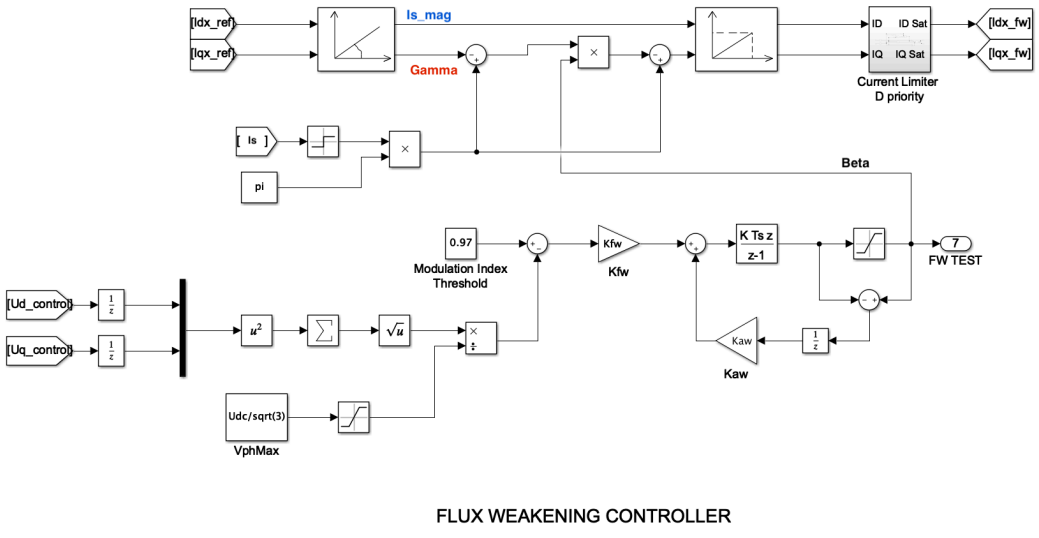


Figure 4.20: Flux weakening simulated controller.

4.5.4. Current Controller and PWM Generation

The new I_{dxfw} and I_{qxfw} current set-points from the flux weakening stage, feed their respective direct and quadrature-axis current controller subsystems as shown in figure 4.21.

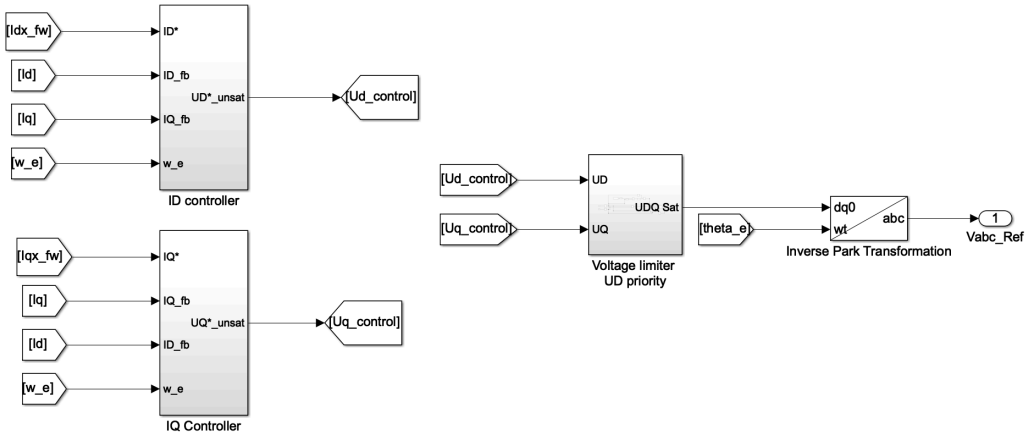


Figure 4.21: I_D and I_Q simulated controllers.

Each direct-axis and quadrature-axis current controller consist of a discrete-time Proportional-Integral (PI) controller block and feed-forward compensation. The implemented PI blocks for the simulated controller application can be found in the appendix section, in Figures A.12 and A.13.

The current controller's output is a direct-axis and quadrature-axis stator voltage reference. On the next stage, a voltage limiter system saturates the magnitude of the reference stator voltage direct and quadrature components to be kept below its maximum allowed as in equation 3.31. The axis priority is given to the direct-axis voltage when it is limited. The reason, is because U_d voltage generates the quadrature-axis current i_{qs}, involved in torque generation in a greater measure than i_{ds}.
As a result, U_d and U_q voltage references are bounded by equations 4.3 and 4.4.

$$-U_{\max}^{ph} \leq U_d \leq U_{\max}^{ph} \tag{4.3}$$

$$U_q \leq \sqrt{{U_{\max}^{ph}}^2 - U_d^2} \tag{4.4}$$

Next, the obtained U_d and U_q saturated components are transformed into a three-phase voltage reference. The three-phase reference goes through a SVPWM stage, as seen in figure 4.22, which generates 6 voltage pulses that are mapped to the DO module of the OP5700, and looped back into the DI module. As a result, the switching voltage drives the gates of the IGBTs in the inverter model, completing the control loop.

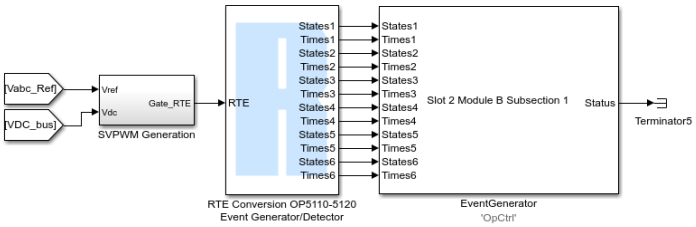


Figure 4.22: SVPWM generation blocks and routing to digital outputs.

The generation of a SVPWM signal from a three-phase voltage reference is implemented as follows. First, knowing that the peak value of the stator voltage is given by equation 3.31, the three-phase reference can be normalised and scaled into a duty cycle signal between zero and one.

Using RT-Events library's block-set as figure A.14 shows, the generated duty cycle and the desired switching frequency, are inputs of the SVPWM block. This block generates six pulses, the first three for the inverter's upper leg and the second set for the inverter's lower leg of IGBTs. These generated signals are rearranged in the same order as previously specified in table B.4 and then mapped to the DO module.

4.6. Hardware in the Loop (HIL) Simulation

In contrast to the real-time powertrain model implemented in the previous section. The hardware in the loop application transfers the controller algorithm to an external controller board. The external controller is a Digital Signal Processor (DSP) LaunchXL-F28379D from Texas Instruments. Alongside, the MCU340 board is a generic Power Electronic (PE) control board based on the LaunchXL with a TMS320F28379D microcontroller from TI. It provides an interface between six PE modules and the LaunchXL as described in figure 4.23.

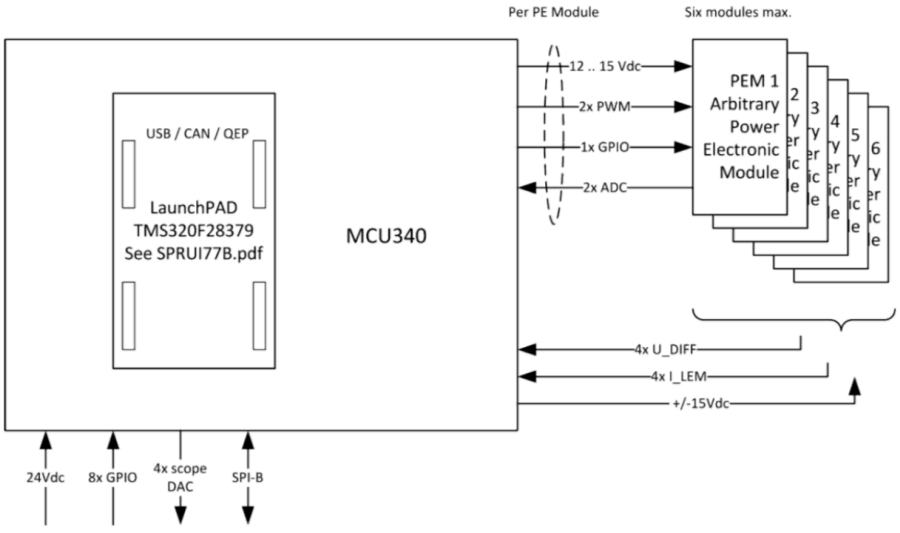


Figure 4.23: Inputs and output of the MCU340 board.

Each of the PE modules has two PWM outputs and two differential amplifier input pins, U_1 and U_2, as seen in figure 4.24. The output voltage U_0, is between $\pm$ 6V and is the input of the 12-bit ADC

module.
For the present project, the gain of the differential input for a maximum of ± 15 V range is:

$$\frac{R_2}{R_1} = 0.402$$

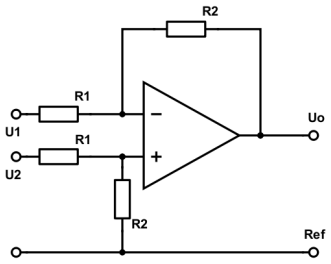


Figure 4.24: MCU340 board's differential input.

In the implementation of the HIL simulation, some of the powertrain signals will be mapped to the OP5700's AO module, then sampled by the external controller. Consequently, the controller generates a command in the form of a PWM signal. This signal, is delivered to the simulated powertrain model through simulator's the DI module. The whole process is summarised in Fig. 4.25, where a block schematic integrates all of the different stages of the project. The figure describes the different parts of the EV powertrain model, simulated either on the CPU or the FPGA of the OP5700 and how the model is interfaced to the external controller board that hosts the control algorithm detailed in previous sections.

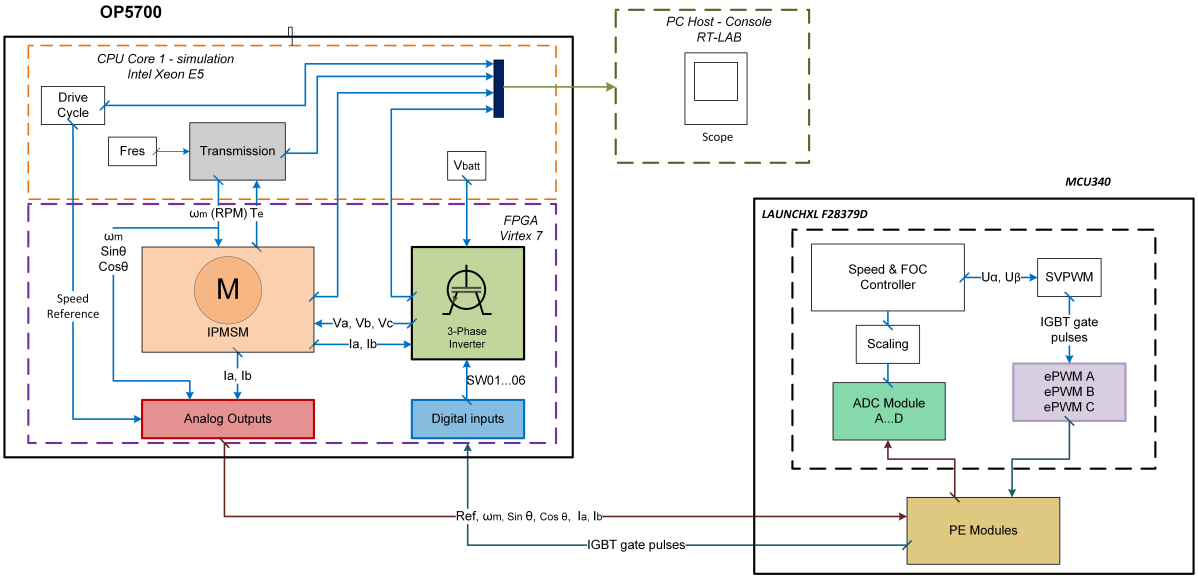


Figure 4.25: Diagram of HIL Simulation of the EV Powertrain.

4.6.1. Powertrain Model

In Figure 4.27 the EV powertrain's RT-Lab model is shown. As it can be observed, it includes the inverter, PMSM and vehicle transmission models. A number of six powertrain signals are scaled and mapped to the OP5700 AO module. The list is shown in table 4.3. Additionally, the mapping and scaling procedure is described in section B.2.

Both 'eHS Gen3' block for the three-phase inverter model and the 'PMSM Motors VDQ' block are the same and maintain their configuration as detailed in section 4.5.

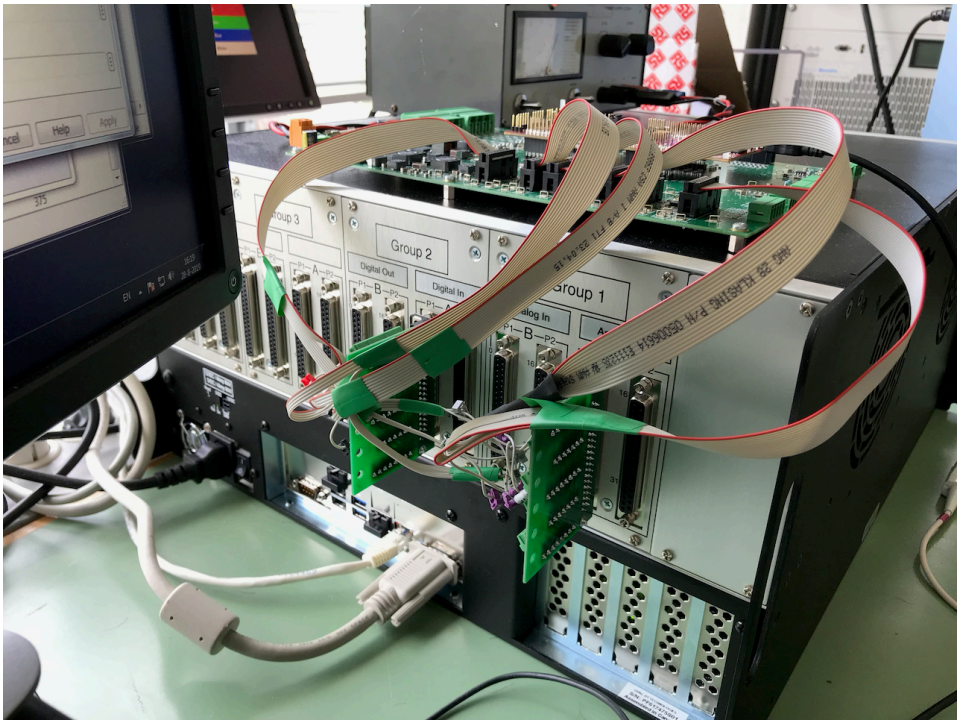


Figure 4.26: OP5700 Analog outputs and Digital Inputs connected to MCU340 Power Electronic modules.

The PMSM currents I_A and I_B are interfaced to the controller board as analog voltage signals generated in the FPGA motor model. The reference and measured vehicle speed, are as well sampled by the controller from the AO module. At the same time, the inverter model receives six PWM gate signals from the PE modules of the MCU340 board.

It was initially intended to map as an AO the rotor position angle to the external controller board. Nevertheless, in practice, this approach was not appropriate. The rotor position signal goes from 0 to 360 degrees and turns back to zero abruptly as a sawtooth signal. Because the angle signal is later used in the controller algorithm to perform the Park transformation, it was observed that this abrupt change introduced spikes of significant amplitude in the transformed I_D and I_Q feedback currents. As a result, undesirable noise appears in the whole control loop and the motor response.

The corrective measure, was to output two signals that represented the sine and cosine of the rotor angle. No matter the amplitude on how the signals are scaled to the AO, this analog voltage always represents a sinusoidal with an amplitude of ± 1 V. Consequently, the sine and cosine signals sampled

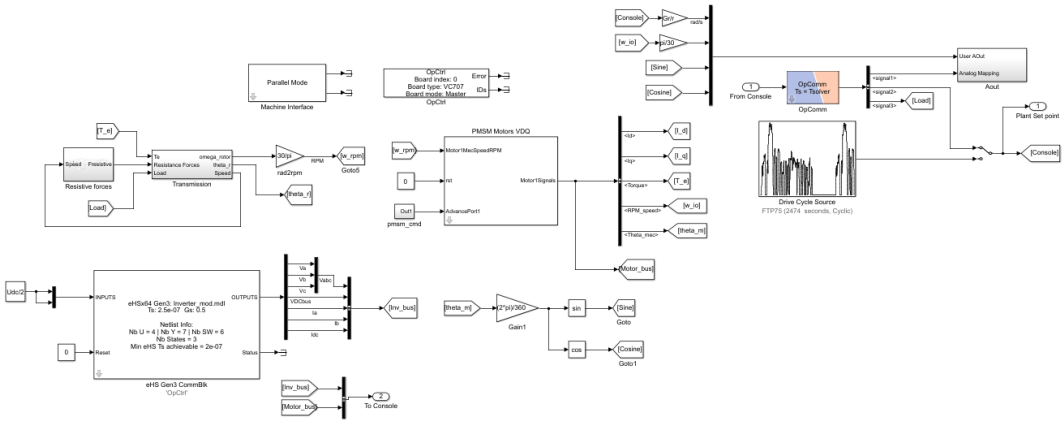


Figure 4.27: EV powertrain model in RT-LAB for HIL application.

Signal	Type	Specs.	Description	Source	Range
IA	Analog output	± 15 V	Feedback signal from model to controller	PMSM eFPGAsim block	± 900 A
IB	Analog output	± 15 V	Feedback signal from model to controller	PMSM eFPGAsim block	± 900 A
Set-point/ Reference	Analog output	± 15 V	For generating Id and Iq current references	Master or console subsystem in the real-time model.	0 to 1050 rad/s
Speed measurement	Analog output	± 15 V	Feedback signal for speed controller testing	PMSM eFPGAsim block	0 to 1050 rad/s
Rotor angle Sine	Analog output	± 14.93 V	Feedback signal from model to controller	Master subsystem in the real-time model.	± 1
Rotor angle Cosine	Analog output	± 14.93 V	Feedback signal from model to controller	Master subsystem in the real-time model.	± 1
IGBT T1	Digital input	5 kHz	Gate pulse	PE module in MCU340	low-high
IGBT T2	Digital input	5 kHz	Gate pulse	PE module in MCU340	low-high
IGBT T3	Digital input	5 kHz	Gate pulse	PE module in MCU340	low-high
IGBT T4	Digital input	5 kHz	Gate pulse	PE module in MCU340	low-high
IGBT T5	Digital input	5 kHz	Gate pulse	PE module in MCU340	low-high
IGBT T6	Digital input	5 kHz	Gate pulse	PE module in MCU340	low-high

Table 4.3: List of Analog outputs and Digital inputs in EV powertrain HIL real-time simulation

by the controller, are used to obtain the rotor position angle according to equation 4.5.

$$\theta_{rotor} = \arctan\left(\tan\theta_{rotor} = \frac{\sin\theta_{rotor}}{\cos\theta_{rotor}}\right) \tag{4.5}$$

4.6.2. HIL Controller

The LaunchXL-F28379D DSP board from Texas Instruments is mounted on the MCU340 board, as shown in figure 4.28. It provides a scaling and filtering stage to the incoming AO from the OP5700.

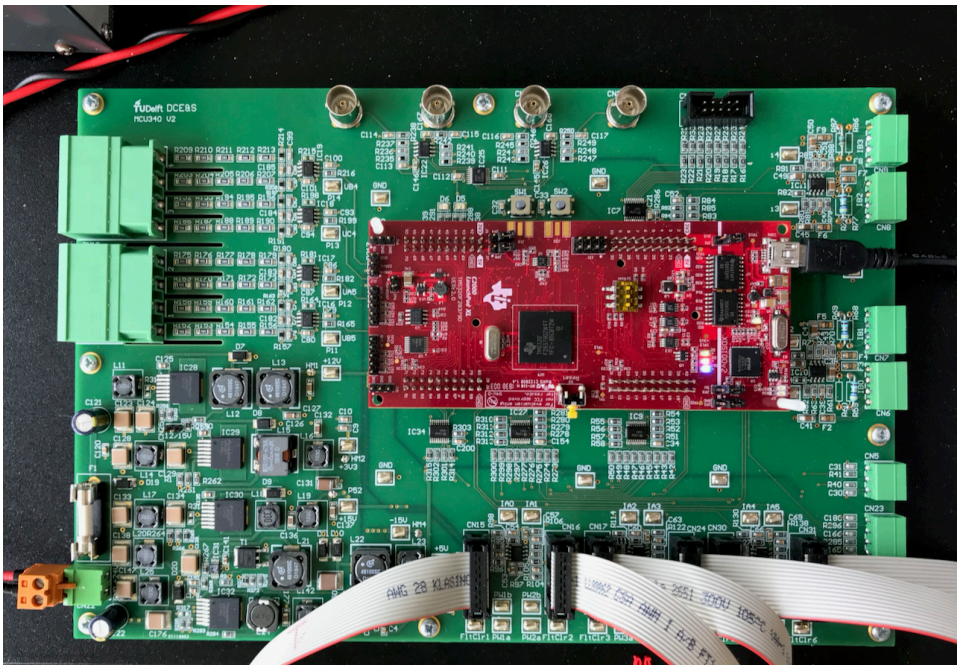


Figure 4.28: LaunchXL TMS320F28379D mounted on the MCU340 board.

The speed and FOC controller are programmed in the TMS320F28379D microcontroller using Simulink with the Embedded Coder Support Package for Texas Instruments C2000 Processors.

The controller subsystems, the ADC configuration blocks and PWM generation stage are executed asynchronously each time a hardware interrupt is triggered by the ADC module. This is accomplished using the 'C28x Hardware Interrupt' block as seen in Figure 4.29. The block's output schedules the execution of the connected subsystem (controller subsystem) when the micro-controller's CPUs receives an interrupt signal from the ADC.

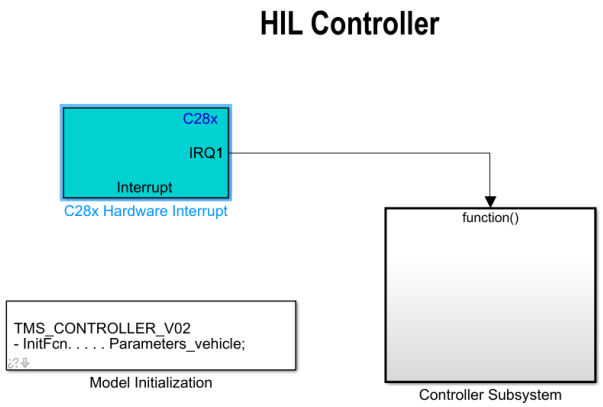


Figure 4.29: Simulink model of the HIL controller algorithm, programmed in the LaunchXL-F28379D DSP board.

In order to set-up the interrupt generation from the ADC module, the configuration of the hardware interrupt block is displayed in table C.1 in the appendix section C.

4.6.3. ADC modules and Input Scaling

The analog to digital conversion (ADC) configuration blocks, are necessary inside the controller program to set-up the ADC module and to provide access to the sampled signals. As seen from figure 4.30, there is an ADC block for each analog input signal that corresponds to the analog outputs of the OP5700, listed in table 4.3. Each one of the sampled signals are converted to a simple-precision floating-point data type and then transformed into a signal between -1 and 1. Finally, the signal is multiplied by a scaling factor and therefore, brought back to their original range as in the powertrain model.

The reason why signals are converted to single floating-point is because the TMS320F28379D is a 32-bit microprocessor and Matlab programs often change between single or double data types during a simulation. Although, the blocks from the TI Embedded Coder Support Package run on fixed-point 32-bits because of the computing efficiency that fixed-point data types offer, it is also possible to build the control algorithm using Simulink blocks and upload the software to the DSP.

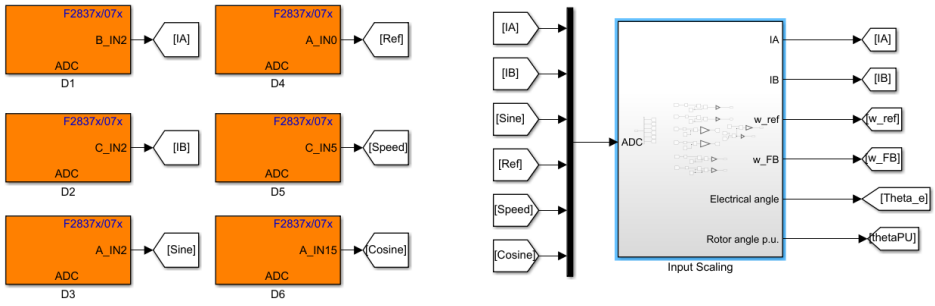


Figure 4.30: ADC blocks for each 12-bit analog input of the LaunchXL-F28379D.

The configuration of the ADC modules and the consequent scaling of the sampled signals is detailed in the appendix sextion C.

4.6.4. Speed, MTPA and FW Control Implementation

The speed controller and FOC control strategies stay the same as the controller described in section 4.5. Inside the speed controller subsystem in figure 4.31 the obtained speed measurement is again compared against to the speed reference. The output of the speed controller is the desired current vector's magnitude, and again, is decomposed into its direct-axis and quadrature-axis current components according to the MTPA equations.

In figure 4.31, it can also be seen the flux-weakening algorithm programmed in the micro-controller. As before, the I_{dx} and I_{qx} current references stay the same except if the PMSM model is running below the base speed. Otherwise, the stator voltage's magnitude is over the modulation index threshold. In that scenario, the current vector I_s is shifted towards the negative direct-axis, as described in section 4.5.3.

4.6.5. Current Controller and PWM Implementation

The current controller is based in the PI control method used in the simulated controller application. Shown in figure 4.32, the current control subsystem, receives as inputs the I_A and I_B measured currents to obtain I_D and I_Q signals. The direct and quadrature-axis current set-points come from the flux-weakening stage. Additionally, the sampled electrical angle and motor angular speed are as well inputs of the subsystem.

The components of the 'Current Control' subsystem are shown in figure 4.33. The implemented FOC algorithm consists of a 'PI Control' subsystem, equal to the implemented in section 4.5. Similarly, the output U_D and U_Q voltage reference components are limited, giving priority to the direct-axis component. In the end, the U_D and U_Q voltages are transformed into the alpha and beta frame of reference before being used to generate a desired space vector.

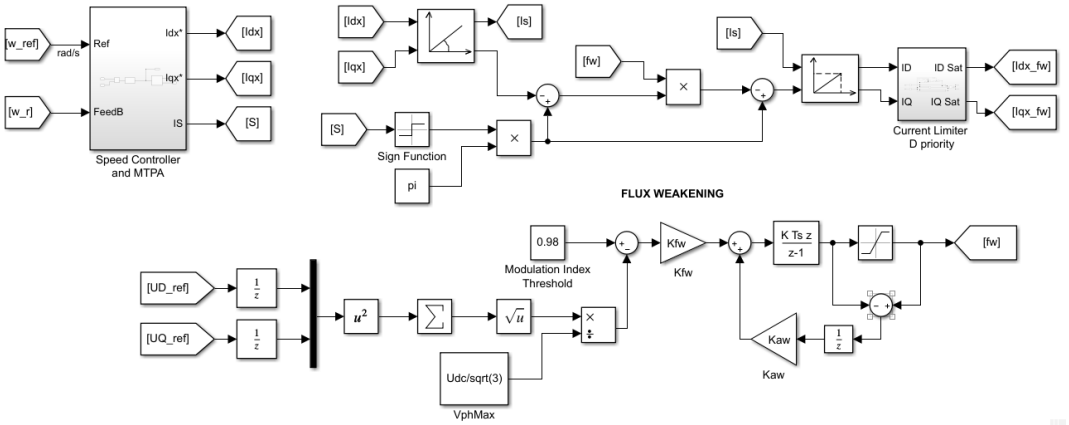


Figure 4.31: Speed, MTPA and FW stages for HIL controller.

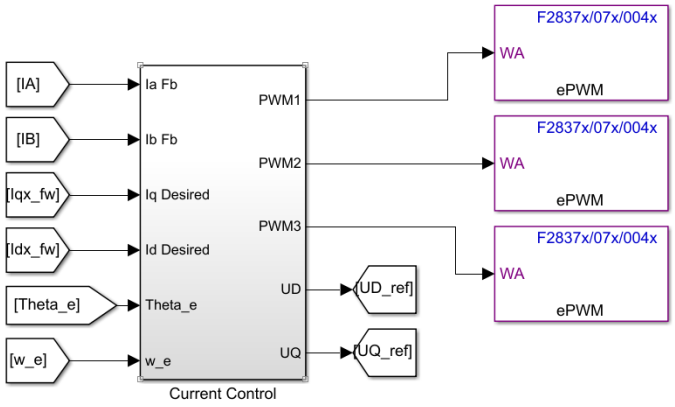


Figure 4.32: Current control subsystem for HIL controller and ePWM blocks.

The ‘SVGen’ subsystem in figure 4.33 generates a set of three duty cycle signals which are inputs of the ePWM modules observed in figure 4.21. The ePWM blocks, output the generated SVPWM signal to the six PWM physical ports of the PE modules in the MCU340 board. These are the PWM signals that will drive the the three-phase inverter in the EV powertrain simulation, as shown in figure 4.25.

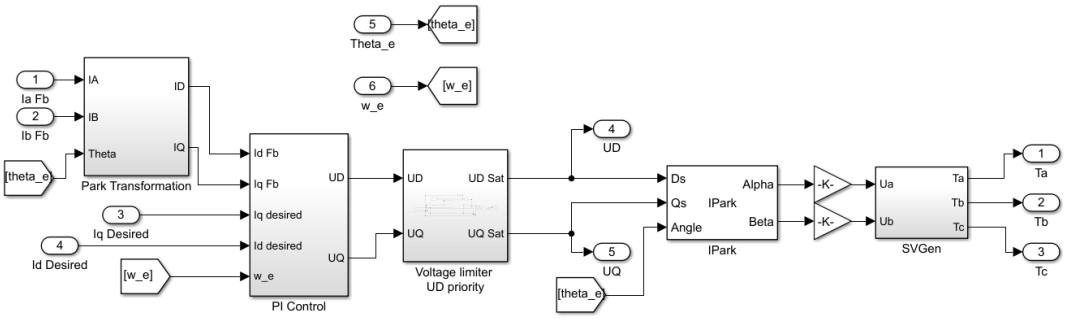


Figure 4.33: Current Control Subsystem components.

5

Results and Discussion

5.1. Preliminary Results

5.1.1. Speed Estimation

The Nissan Leaf motor's base speed is documented as 2100 RPM. For reference, the speed point is converted from angular speed to linear speed using equations 2.9 and 2.8.

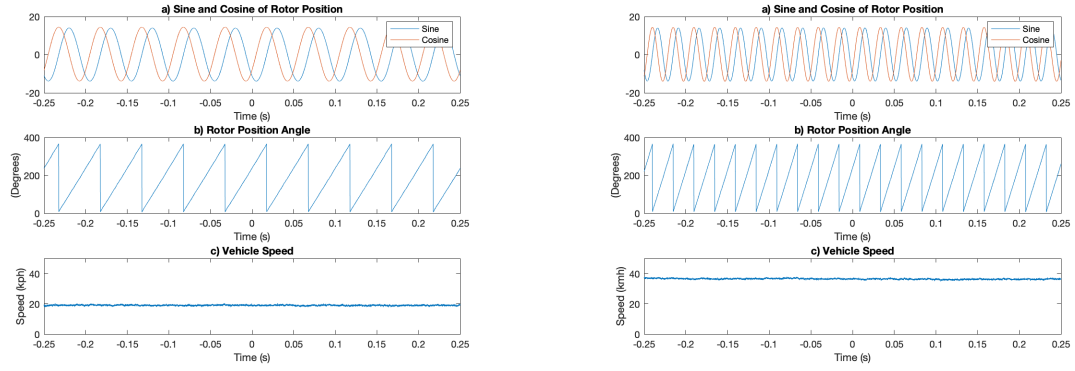
$$2100\,RPM * \frac{\pi}{30} * \left(\frac{r_{wheel}}{G_r}\right) * 3.6 = 31.51\,{}^{km}/_{h} \tag{5.1}$$

5.1.2. Maximum Torque

The MTPA curve in figure 3.10 intersects with the stator current's limit circle in point $i_{sd} = -363.20$ A, $i_{sq} = 477.58$ A of the D-Q current plane. Therefore, replacing i_{sd} and i_{sq} in equation 3.15, the result is given below:

$$T_{max} = 458.88Nm \tag{5.2}$$

5.1.3. Rotor position measurement for HIL simulation



(a) Rotor position signals at 18 kmh.

(b) Rotor position signals at 36 kmh.

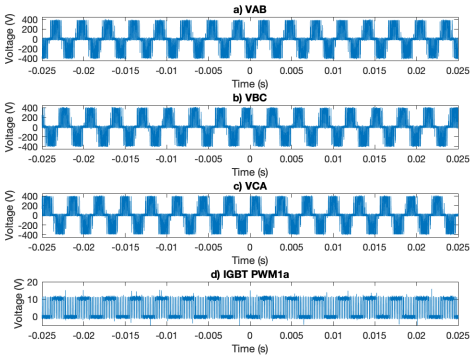
Figure 5.1: Rotor position signal and measured vehicle speed.

Figure 5.1 shows the output sine and cosine signals of the rotor position measurement from the PMSM model. As described in section 4.6.1, the HIL controller calculates the rotor position angle using equation 4.5. From the rotor position, the controller obtains the speed of the motor and therefore, the speed of the vehicle. The validity of the implemented approach is confirmed with figure 5.1.(a). The

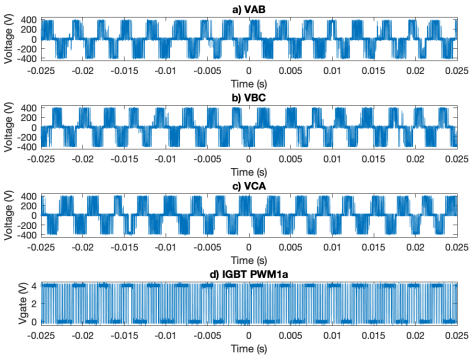
vehicle's speed is 18 kmh or 10 m/s, equivalent to 251.2 rad/s from the motor's side of the transmission. The angular speed is equal to a frequency of 40 Hz, the same frequency of the rotor position signal in 5.1.(a).

5.2. Inverter Model

To confirm the validity of the implemented real-time three-phase inverter model, the line to line output voltages were scaled and measured from the AO module of the OP5700 using an oscilloscope.



(a) SIL controller simulation.



(b) HIL controller simulation.

Figure 5.2: Three-phase inverter model's line to line output voltages for both SIL and HIL applications.

In figure 5.2, the inverter output voltage's waveforms are shown for a constant vehicle speed of 72 kmh. The figure shows the expected behaviour of a three-phase inverter.The line to line voltage is within the $\pm U_{DC}$ voltage range, where U_{DC} is equal to the 375 V of the EV's battery.

The difference between figures 5.2.(a) and 5.2.(b) is the source of the SVPWM signal that drives the inverter IGBTs. Although in both applications the inverter is simulated in the OP5700's FPGA at a 250 ns time-step, the HIL controller (see Fig.5.2.(b)), samples inputs and generates a SVPWM command at a 5 kHz frequency (every 200 μs). On the other hand, the SIL controller samples the powertrain signals and generates a reference voltage vector at a 5 kHz frequency, but as described in figure 4.14, the SVPWM signal is generated in the simulator's CPU at 10 μs, a much smaller time-step. Therefore, the SVPWM command is generated at a faster rate in the SIL application.

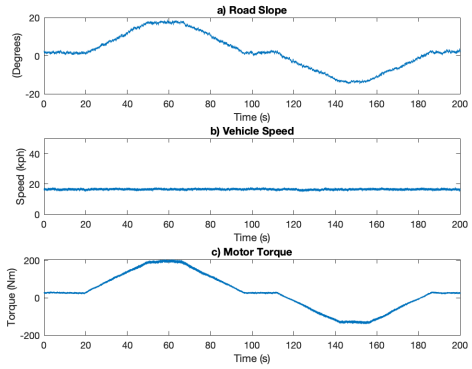
The motive for generating the space vector PWM command at a different time-step can be seen in figure 4.22. The 'EventGenerator' block that routes the PWM signal to the DO module, can only be executed at the same time-step of the CPU simulation. As a result, the voltage waveforms look 'cleaner' in Fig.5.2.(a).

Additionally, in the last row of figures 5.2.(a) and 5.2.(b) the switching gate voltage is shown for one IGBT. As can be seen, the amplitude is higher in 5.2.(a) because the OP5700 allows regulating the output voltage of the DO module using an external DC source, unlike the LaunchXL-F28379D board.

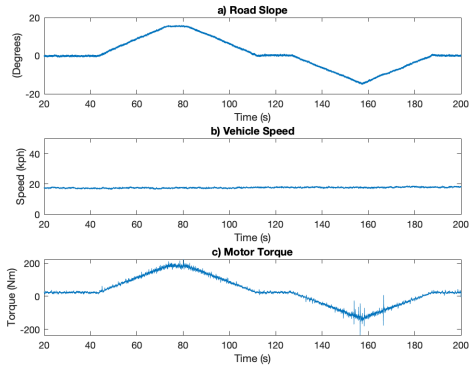
In any way, the response of the inverter model is similar for both SIL and HIL applications.

5.3. Gear Transmission and Resistance Force Model

Figure 5.3 makes a comparison between the behaviour of the SIL and HIL EV powertrain simulation when the road's slope angle is modified. The maximum slope change tested is ± 15 degrees. Initially, the slope remains unchanged at zero degrees, the speed is constant at 18 kmh, and the output torque is equal to the transferred load from the chassis to the motor's shaft through the transmission. As previously described in chapter 2, the observed load torque is a result of the rolling resistance and the air-drag force over the vehicle, and this steady-state load is always present as long as the car is moving.



(a) SIL controller simulation.



(b) HIL controller simulation.

Figure 5.3: Motor response for a change of 15 degrees in the road's slope angle at 18 kmh.

When the slope is gradually increased, the gradient resistance force adds to the vehicle's load. Therefore, the motor increases the torque at the same rate to maintain the vehicle's speed until the slope stops rising. When the slope angle, returns to zero, the motor torque goes back to its steady-state load torque.

In the same way, when the slope angle becomes negative, it means the vehicle is moving downhill. As a result, the gradient force will accelerate the vehicle downwards. Consequently, the controller commands a negative output torque to the motor to brake the vehicle and maintain the speed. During this stage, the motor's output power is negative ($P_{out} = T_e\omega_m$) because it is transforming the vehicle's kinetic energy into electric energy. If the generated energy is used to charge the vehicle's battery, the process is called regenerative braking.
Comparing figures 5.3.(a) and 5.3.(b) it can be mentioned that the EV powertrain simulation behaves as expected during the variation of the resistance forces. In the same manner, the response of the HIL controller is similar to the SIL application and can drive the electric motor into the desired behaviour.

5.4. Speed and FOC control strategies

The waveforms in figure 5.4 are meant to show the speed controller response given a speed command. For both SIL and HIL applications, the speed of the vehicle can rise from 0 kmh to 97 kmh in less than 9.9 seconds as claimed by the manufacturer [14]. Nevertheless, the response of the vehicle for the HIL simulation is slightly slower but still within the range.

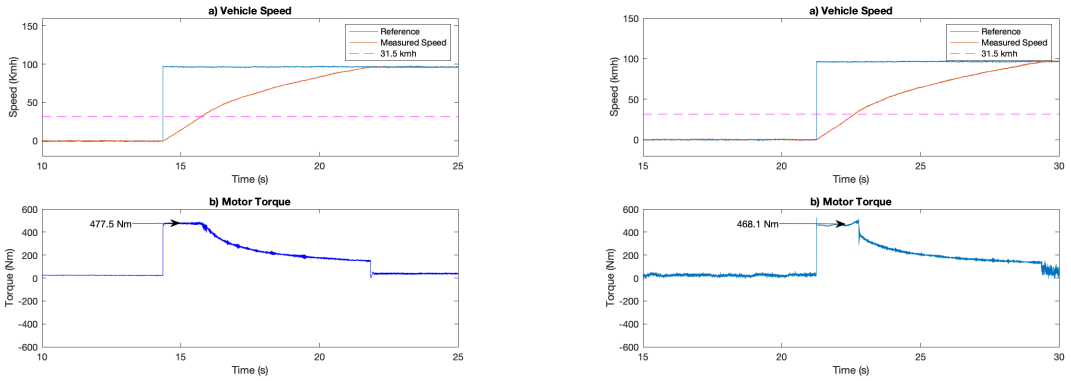
One reason why the HIL simulation's controller response is different, might be related to the different latencies of both systems. As previously described, the PWM signal that drives the inverter is generated at different time-steps for SIL and HIL simulations. In addition, because the SOC of the ADC modules of the hardware controller occurs at a 5 kHz frequency, the motor currents are sampled at that frequency. On the other hand, in the SIL application, the motor currents are sampled by the AI module of the OP5700 at a much faster rate, managed by an FPGA card. As a result, the external controller might not sample changes in the motor currents that occur faster than the sampling rate, whereas the SIL controller probably can.

As a result, the flux-weakening controller response is not the same. The response of the HIL controller appears to be slower, but still acceptable. This is why figure 5.4.(b) presents a drop of the maximum torque once it reaches the base speed when compared to 5.4.(a), where the torque decreases more smoothly.

For figures 5.4.(a) and 5.4.(b), it can be noticed that as the speed rises, the permanent magnet motor enters the flux-weakening region when it reaches the base speed of 2100 RPM (see figure 2.6) or 31.5 kmh (vehicle's speed). At base speed, the flow of power into the motor drops as a consequence

of the rising BEMF. However, the task of the FW controller is to counteract the BEMF and delay the moment where the motor cannot generate more mechanical power. Consequently, it is expected that the motor's output torque starts to decrease gradually from the second 19.7 and 22.6 for the SIL and HIL simulations, respectively. Therefore, the vehicle develops less acceleration but manages to reach the speed reference.

Similarly, the observed torque response is as expected. The validity of the MTPA control strategy is verified in figure 5.4. The MTPA control strategy must generate the maximum torque output as soon as the speed command is given. The theoretical T_{max} has a magnitude of 458.87 Nm. In comparison, figure 5.4 shows maximum torque measurements of 477.5 Nm in the SIL simulation and 468.1 Nm for the HIL simulation.



(a) SIL controller simulation.

(b) HIL controller simulation.

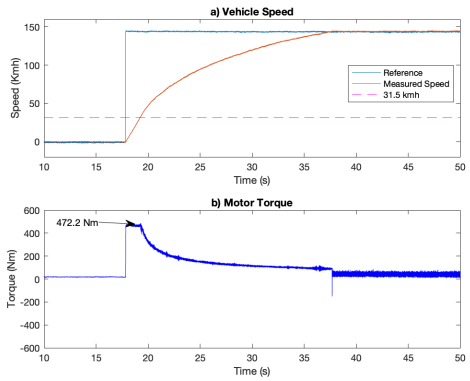
Figure 5.4: a) Reference and measured linear speed of the vehicle, 0 to 97kmh. b) Torque response of the motor

In comparison, figure 5.5 presents the vehicle response to a 144 kmh speed set-point. According to Nissan [15], the maximum vehicle speed is 144 kmh. In figures 5.5.(a) and 5.5.(b) it can be observed that the speed controller can drive the vehicle model up to the speed reference. The waveforms of figures 5.5.(a) and 5.5.(b), show initial high torque as a result of the MTPA control strategy. 472.2 Nm and 474.3 Nm for the SIL and HIL simulations respectively. The measured maximum torques are within a 3% difference from the theoretical T_{max}, which validates the implemented MTPA strategy.

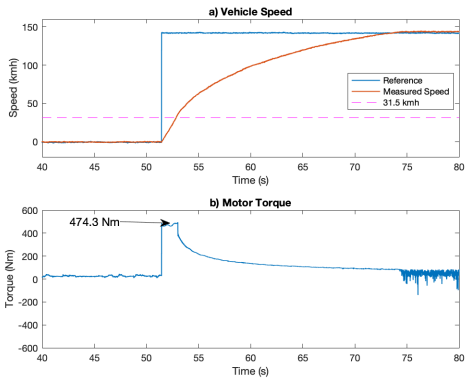
Again, the vehicle speed starts to decrease when it reaches the base speed of 31.52 kmh, entering the flux-weakening region. As figure 5.5 shows, the torque starts to decrease. However, the flux-weakening controller is able to extend the speed range and the vehicle reaches the maximum speed possible with almost no torque left. Comparing figures 5.5.(a) and 5.5.(b) it can be noticed as well the drop in the torque of the HIL simulation. The response of the FW controller can be considered to be better in the SIL simulation because the electric torque decreases slower and in a smoother shape, meaning the FW algorithm is reacting faster. As a result, the SIL simulation's vehicle can reach the speed set-points faster.

On the other hand, in figure 5.6, the importance of the FW controller is exposed. It shows the operation of the EV powertrain HIL simulation without the FW algorithm running inside the controller. As can be seen, the vehicle's motor loses power significantly as the speed is growing. The generated torque has dropped in figure 5.6.(a) where the speed controller manages to barely reach the set-point after a longer time. In figure 5.6.(b) for a higher speed set-point, the available torque the machine can develop has decreased to the point the speed stalls and the vehicle never reaches the desired speed. Therefore, it can be said it is an appropriate strategy to implement a FW controller in this application.

One last thing to remark from the previous tests, is the observed torque ripple at the end of the cycles in figures 5.4.(b) and 5.5.(b). The ripple appears when the speed of the simulated vehicle is high and only in the HIL simulation. One reason for this could be the source of the rotor position signal. In section 4.6.1 it is described how the controller board samples the sine and cosine of the rotor position

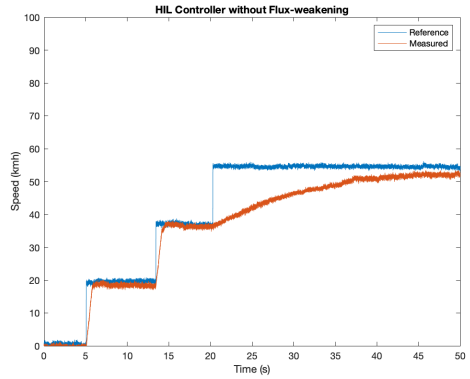


(a) SIL controller simulation.

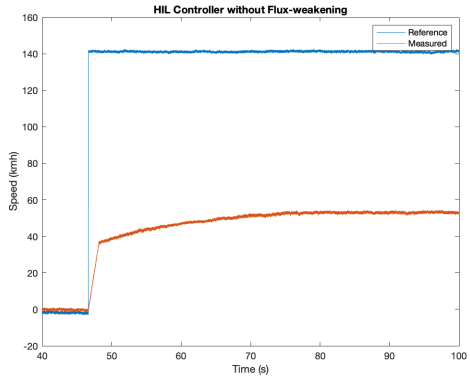


(b) HIL controller simulation.

Figure 5.5: a) Reference and measured speed of the vehicle, 0 to 144kmh. b) Torque response of the motor



(a)



(b)

Figure 5.6: HIL simulation of EV without flux-weakening control

from the OP5700's AO module. Next, the controller obtains the rotor position signal using equation 4.5. As a result, the rotor position measurement has a frequency that rises with the speed, as documented in figure 5.1. Therefore, the rotor position signal used by the HIL controller, could be the origin of the ripple at high speeds.

5.5. Drive Cycles

In this case, the EV powertrain real-time model is tested by setting a drive cycle as the speed reference. The new european driving cycle (NEDC) is a standardised option provided within the 'Drive Cycle Source' block in Simulink. In previous cases, the powertrain model was tested using a user input speed reference. However, it is an extreme scenario for testing since a vehicle is rarely required to reach its maximum speed abruptly.

In figure 5.7, fractions of a drive cycle for the SIL simulation are shown. The simulated EV powertrain can follow the drive cycle. Both figures are displaying the behaviours of the torque, direct-axis and quadrature-axis currents. The motor magnitudes are considerably smaller when compared to the previous experiment and is because the NEDC shows slower-growing speeds and shorter acceleration intervals, therefore, lower torque requirements, which are more realistic test conditions for vehicles.

During the deceleration periods, the response of the speed controller is to generate a negative I_s

current vector. As a result, during deceleration, the motor will generate a negative torque to slow the vehicle down even though the speed of the car or the rotation of the PMSM is still positive. Therefore, the output power ($P_{out} = T_e\omega_m$) is negative, meaning the motor is acting as a generator during deceleration (T_e is the motor's torque and ω_m the motor's angular speed). During braking or deceleration stages, the electric motor generates energy that can be used to charge the batteries and extend the driving range.

Next, the HIL simulation of the EV powertrain is tested with the same drive cycle. As figure 5.8 shows, the response of the powertrain model and the controller itself is very similar to the behaviour observed in the SIL simulation. In figures 5.8.(a) and 5.8.(b) the quadrature-axis current changes between positive and negative values depending on the torque requirement. Looking back at the electric torque equation in Eq.3.15, i_{sq} determines the sign of the torque. The simulated powertrain behaviour shows that when the speed of the vehicle is rising, i_{sq} is positive, and consequently, the motor outputs a positive torque. Conversely, when the vehicle decelerates, i_{sq} and the motor torque T_e are negative, in fact i_{sq} and T_m have the same shape.

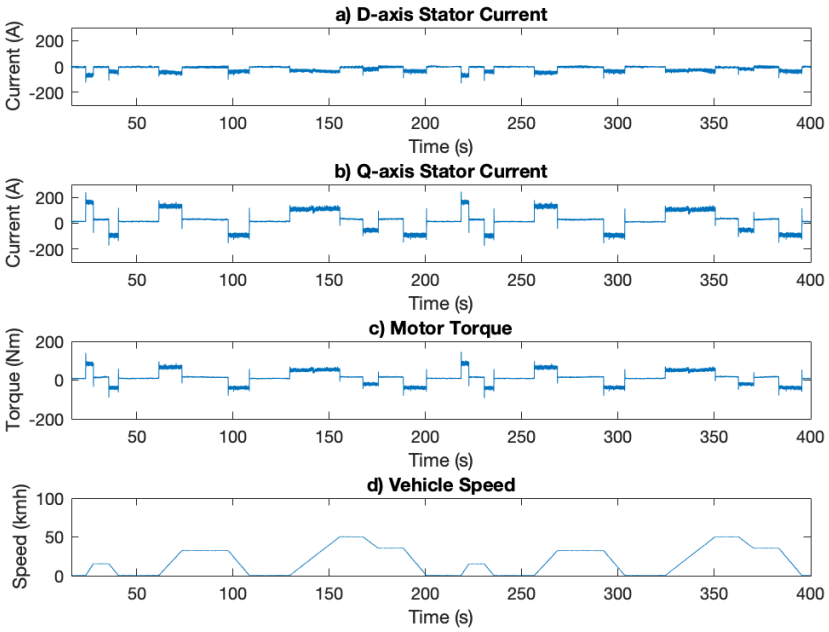
On the other hand, i_{sd} remains negative throughout the drive cycle, which is expected. In figure 3.10 it can be seen that the MTPA curve is placed on the second quadrant of the D-Q plane meaning the MTPA control strategy, will have the current controller to drive the direct-axis current to be always negative. If the required torque is negative (during deceleration), then i_{sd} and i_{sq} are negative. Therefore, it can be said that the operation of the current controller is as expected because the motor stator's currents follow the set-points of the implemented FOC strategy.

Nevertheless, when comparing figures 5.7 and 5.8 it can be seen that torque and current signals of the HIL simulation have noise. As described in the previous section, much of this ripple is originated from the rotor position measurement by the HIL controller. The ripple increases during high speed operation and it is observed in the currents and torque, because the rotor position signal is used in the park transformation of the phase currents from the motor simulation.

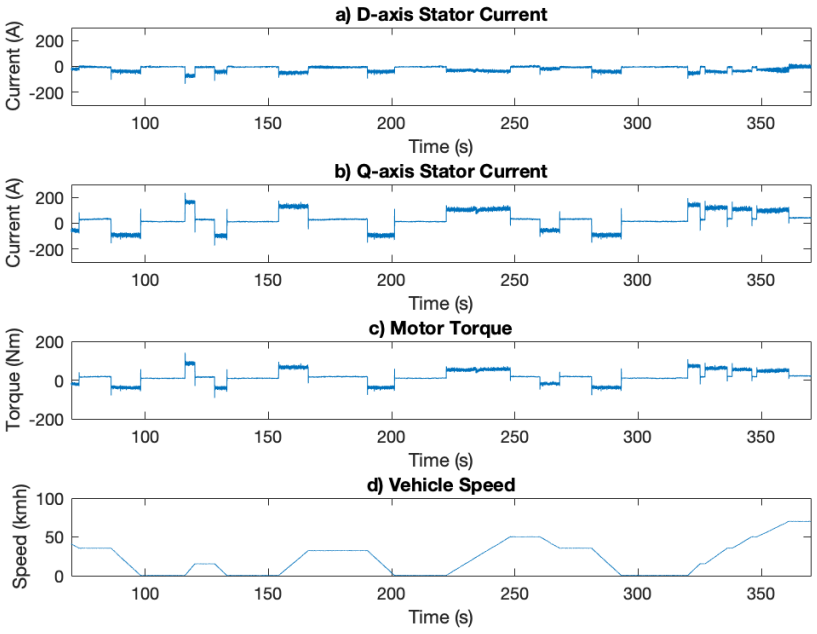
Another reason can be attributed to the analog voltage signals the HIL controller samples from the OP5700's AO module. First of all, the HIL controller samples the stator currents, and the rotor position signals. Noise can be induced to the measurement from the wire connections.

In the same way, the implemented flux-weakening controller is another factor to be mentioned. The response of the implemented FW algorithm depends on how well-tuned are the integral and anti-windup gains of the controller. The process is challenging and time-consuming because it is not systematic, and instead, based on trial and error.

In addition, the ripple in the measured i_{sd} and i_{sq}, introduced by the rotor position measurement, directly affects the performance of the flux-weakening algorithm, since it relies in actively modifying the stator current vector at high speeds. Overall, the response of the FW controller, the noise from the analog inputs, and the rotor position signal, affects the performance of the HIL controller specially for high speeds (FW region).

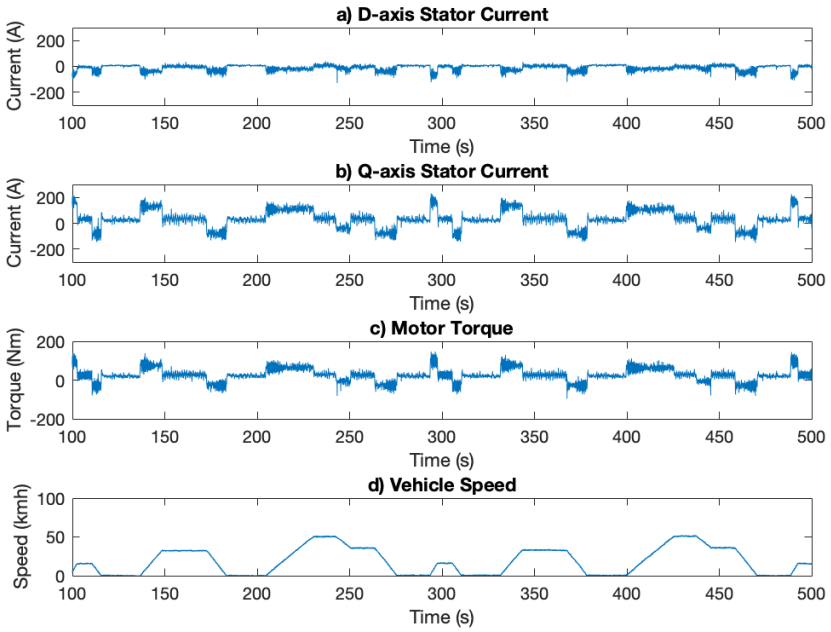


(a)

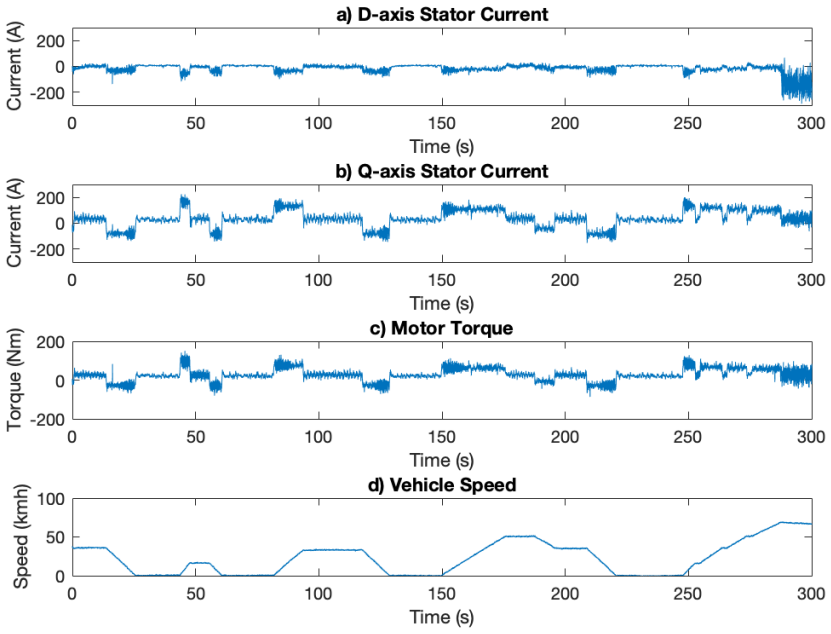


(b)

Figure 5.7: SIL simulation of an EV drive cycle. a) D-axis measured stator current . b) Q-axis measured stator current c) Motor Torque. d) Vehicle measured speed.



(a)



(b)

Figure 5.8: HIL simulation of an EV drive cycle. a) D-axis measured stator current. b) Q-axis measured stator current. c) Motor Torque. d) Vehicle measured speed.

6

Conclusions and Recommendations

6.1. Conclusions

- As described in chapter 3, a field oriented controller is applied. Together with MTPA, the flux-weakening control strategy is implemented to expand the speed range of the PMSM in the powertrain model. The results show that the implemented speed and field-oriented controller operate as desired. The speed controller can drive the vehicle to the speed set-points. In addition, the MTPA control strategy results in a maximum torque output from the PMSM, when high acceleration is desired as in figures 5.4 and 5.5. In the same way, the experiments demonstrate that the flux-weakening algorithm can extend the speed range of the electric motor model in both SIL and HIL simulations. As a result, the simulated vehicle can reach speeds above the base speed and up to 144 kmh. Figure 5.6 shows limited operation of the powertrain model, when the flux-weakening control is not implemented. Therefore, it confirms the chosen strategy as appropriate.

- The initial SIL simulation can be considered as an appropriate starting point to test the performance of the control algorithm. The results show how the response of the powertrain simulation is not only as expected but very similar for both SIL and HIL applications. Since the SIL simulation results are less prone to be influenced by external noise, and it is easier to implement, the results of the HIL simulation can be compared against them. Therefore, the SIL simulation is a useful tool for testing the powertrain model and controller before using any external hardware.

- In a similar way, the powertrain model is validated by the observed response of the motor currents and torque output to a speed reference imposed by the drive cycle as in figures 5.4 and 5.5. In the same manner, the expected behaviour of transmission and load forces model is validated by testing the response of the proposed models when a variation of the resistance forces is enforced, as in figure 5.3.

- The proposed methodology to achieve a real-time simulation of the powertrain components, is described in section 4.2. The eHS and eFPGASIM libraries are employed to build models of the three-phase inverter and PMSM, respectively. As a result, they are simulated in the FPGA of the OP5700. A simulation time-step of 250 ns is set in order to fulfil the low latency and high sampling resolution requirements, because the inverter and PMSM models interact with the external controller hardware. Additionally, the EV transmission and load force model are simulated in the OP5700's CPU at a 10 μs time-step. The CPU and FPGA models can be simulated in real-time successfully, interact between them and with the external LaunchXL-F28379D controller board. In the end, the proposed HIL simulation can reproduce the expected behaviour of the EV powertrain components during drive cycle tests and the variation of the vehicle's resistance forces as shown in figures 5.8 and 5.3.(b), respectively.

- As it could be observed, the implemented flux weakening algorithm (see figure 4.31) was able to extend the speed range of the EV model. However, the dynamic response is limited. This was attributed to the latency of the system and how the external controller acquired the rotor position signal from PMSM model. The latter, introduced a noticeable amount of ripple in the currents

and torque waveforms at high speeds, making the HIL controller prone to be less reliable at high speed. Nevertheless, for comparison or testing purposes the algorithm can accomplish the initial objective.

- The developed HIL simulation of an EV powertrain can be considered as a basis for future electric vehicle research or controller testing. The results validate the capability of the simulator to represent the expected behaviour of the inverter and PMSM. The real-time simulation allows the powertrain model to interact with external hardware and the LaunchXL-F28379D DSP has been found a suitable device to execute the required control algorithms and command the simulated plant.

6.2. Recommendations

- Another method can be used to communicate the rotor position signal from the PMSM model. A good option is to digitalise the angle signal and use the DO module of the OP5700 to emulate a digital encoder. In this manner, the introduced noise when using an analog signal is avoided. Additionally, if the rotor position signal is shaped as sawtooth waveform, the controller introduces errors into all the related computations (e.g. park transformation) because of the sudden changes, causing instability. Therefore, using a digital rotor position measurement is recommended.

- It is recommended to do a SIL powertrain simulation as a starting point before implementing external hardware. Nevertheless, is better not to use the signal loop-back kit of the OP5700 and instead do a full software simulation. This will avoid signal scaling/de-scaling, configuring the IO modules and unnecessary time-step differences.Thus, the controller testing process is less complex and time consuming.

- For future work, the proposed EV powertrain model can be improved or extended by adding a battery pack model, a motor's thermal model, and a loss model to estimate the consumed energy.

- To improve the effectiveness of the proposed FOC strategy, the flux-weakening algorithm can be replaced by look-up tables that store the desired current set-points for the current controller. The look-up table data can be generated from the information given in figure 3.12, table 2.2, and table 2.1. As a result, given the desired speed or torque reference, the tables return a current set-point that follows both MTPA and FW control strategies.

- If it is desired to build a more accurate PMSM model that takes into account the effects of saturation and current changes, tables with the inductance and magnet flux linkage data can be uploaded into the PMSM eFPGASIM block. Consequently, MTPA and FW look-up tables that consider the inductance changes should be generated as well.

A

Powertrain model in RT-LAB

The following models were implemented using Matlab R2017a, and the eFPGASim library in RT-LAB version software.

A.1. eHS Inverter Model

The ‘eHS Gen3 CommBlk’ block needs to be added to the same RT-LAB project window where the ‘OpCtrl’ block is added, as shown in figure A.1. The ‘OpCtrl’ block is where the BitStream File name is specified. This Bitstream File, is provided by OPAL-RT with the acquisition of the OP5700 device in the laboratory and is inside the firmware documentation where the specifications of the physical ‘IO’ of the device are enlisted. The inverter circuit is built in a separate Simulink file with a ‘.mdl’ file extension. The inverter model file, must be saved in the same path as the RT-LAB project model. Next, the inverter’s file name, must be written inside the ‘Circuit file name’ box inside the eHS Gen3 block.

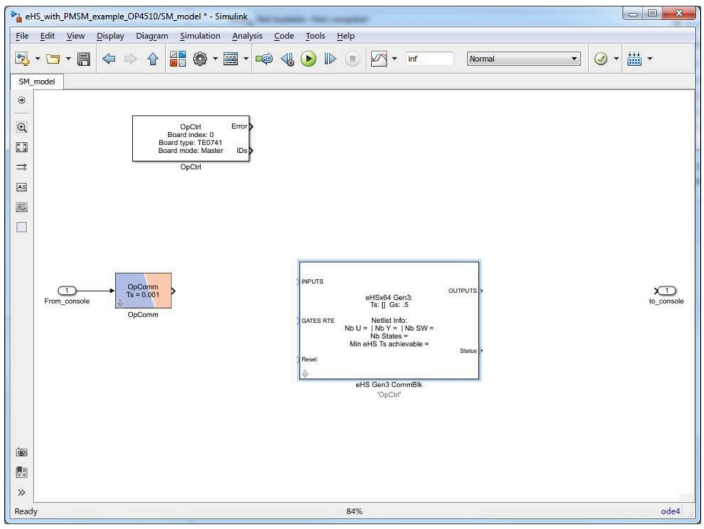


Figure A.1: eHS Gen3 and OpCtrl blocks in RT-Lab project.

The 3-phase IGBT inverter circuit, is modelled according to the element naming instructions specified in ‘eHS User Guide’. This document must be followed for any electrical circuit real-time simulation using the software [17]. Conveniently, the IGBT bride can be easily modelled using Simulink’s Universal Bridge block from the SimPowerSystems Elements library. Nevertheless, the name of the bridge block must be done using the ‘SW01-06’ nomenclature specified in the eHS guide [17].

Going back to the RT-LAB model and inside the eHS Gen3 CommBlk block’s mask, under the ‘Infos’ tab, it is possible to observe how the information of the inverter circuit is gathered and shown in the block if the inverter’s circuit file has been loaded correctly, as it is shown in Figure A.2.

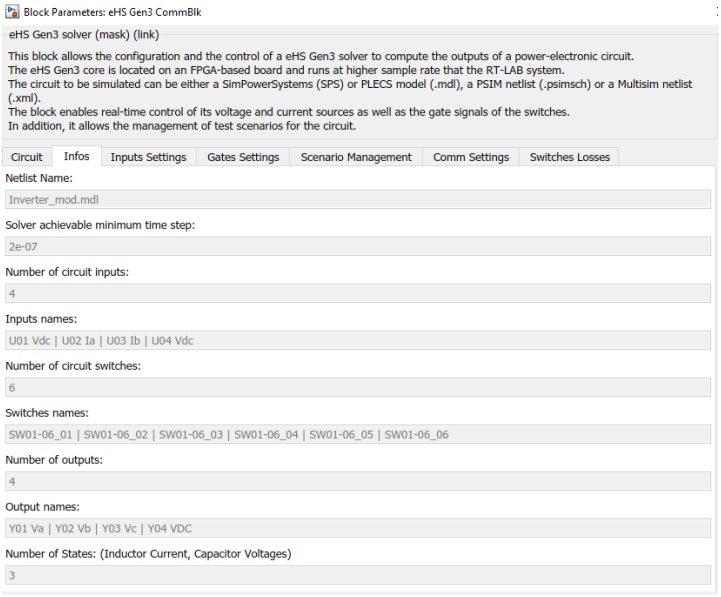


Figure A.2: Inverter's circuit information in eHS Gen3 block.

The mapping of the input signals to the inverter circuit from the RT-LAB model, is done inside of tab 'Inputs Settings', which is shown in Figure A.3. The 'Voltage/current source control enumeration' box, is filled according to the eFPGASIM manual [18], so that inputs 1 and 4 come from the DC-Link voltage signal generated from the RT-LAB project running in the OP5700's CPU to the eHS inverter circuit running in the OP5700's FPGA. On the other hand, inputs 2 and 3, are routed from the PMSM model block corresponding to the phase A and B currents.

Table A.1 lists the previously mentioned configuration in order to couple the inverter circuit simulated by the eHS Gen3 block, with the PMSM Motors VDQ block. This table was obtained from several test done for the VC707_2-EX-0001-3_1_10_49-eHSx64m_C3C1FBCEC3C1FBCE-01-02.bin firmware at the EEMCS laboratory.

Source Type	Source Channel	Description
0		Input from RT CPU Model
1		Input from Analog Input
2	0	PMSM1 (VDQ) Ia
2	1	PMSM1 (VDQ) Ib
2	2	PMSM1 (VDQ) Ic

Table A.1: Motor-to-eHS mapping signal list of the OP5700 example firmware

The modulated voltage signal's source that will drive the IGBTs in the inverter model, is mapped in the 'Gates Settings' tab as displayed in Figure A.4. The configuration is implemented so that the source of the IGBT's gate pulses is read directly from the first 6 digital inputs of the OP5700 device.

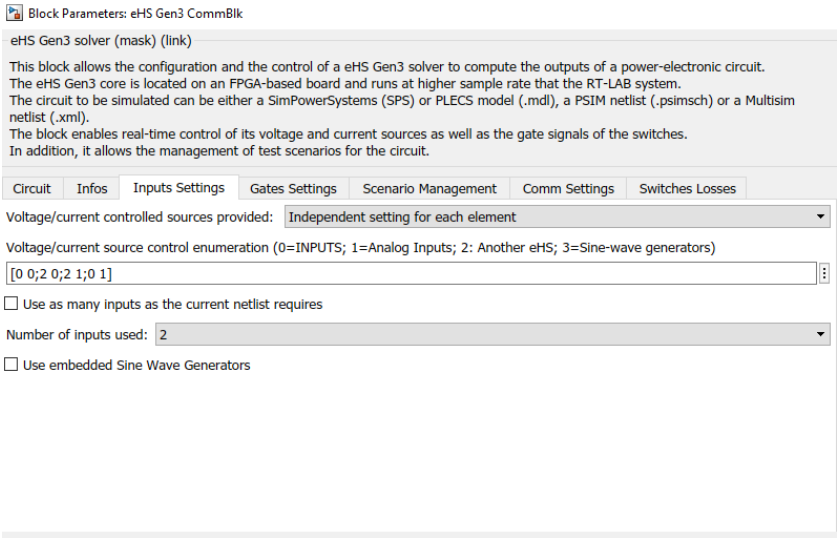


Figure A.3: Inverter's circuit input signals setting.

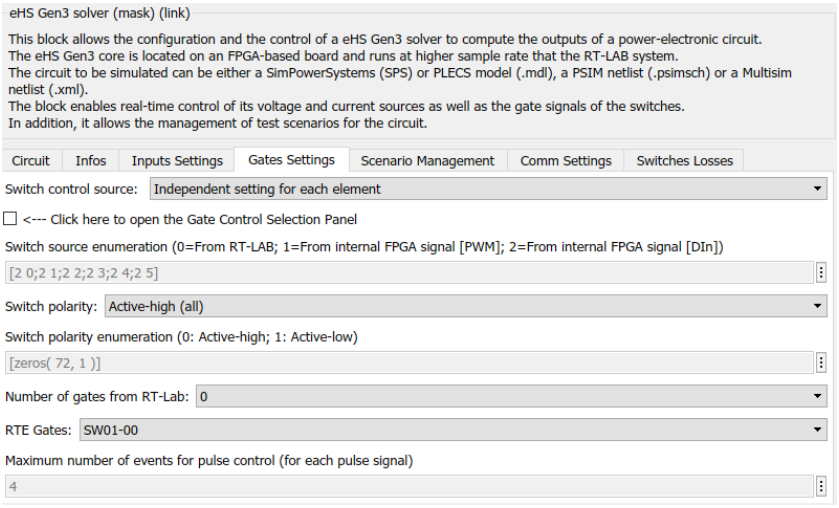


Figure A.4: Inverter's circuit gates setting.

A.2. eFPGASIM PMSM Model

Since the model of the permanent magnet motor is simulated in the FPGA in real-time, the communication ports must be configured according to the OP5700 firmware file, summarised in table A.2 below. The tab 'other', shown in figure A.5, must be filled correctly so that the model performs properly.

	Data In	Load In	Data Out
Port Number	15	8	9

Table A.2: Port configuration for PMSM Motors VDQ block

The information in table A.2 is found inside the OP5700's firmware configuration file. The configuration of the machine interface block (figure is done in order to link the blocks that measured the line voltage inside of the inverter model, to the PMSM model. The configuration is summarised in table A.3

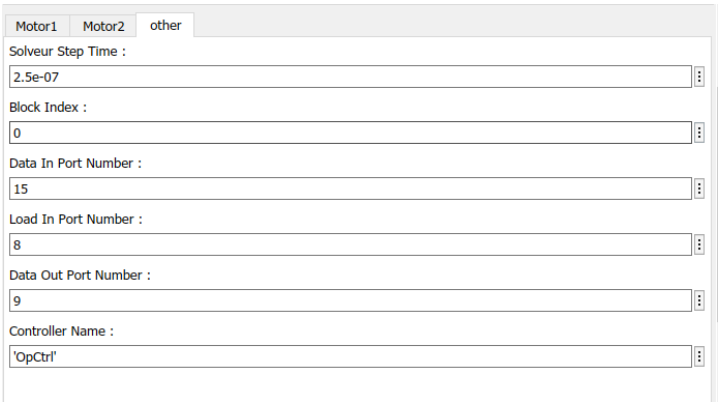


Figure A.5: PMSM Motors VDQ port configuration.

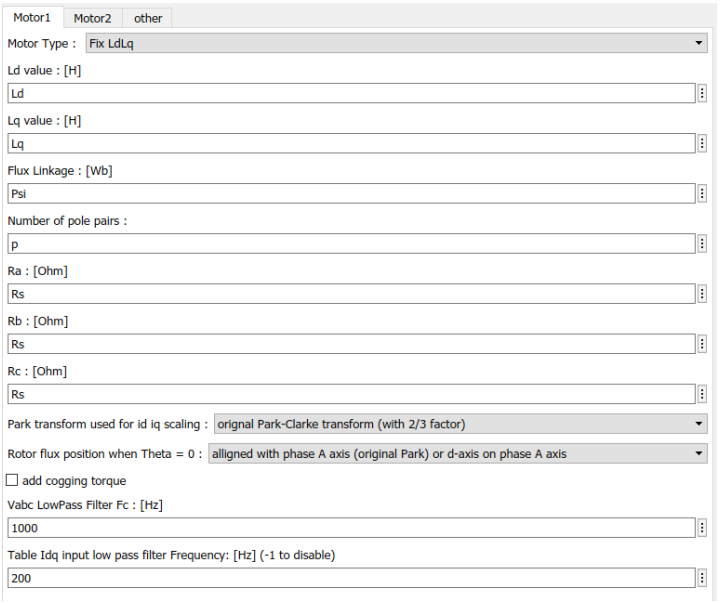


Figure A.6: PMSM and Machine interface blocks.

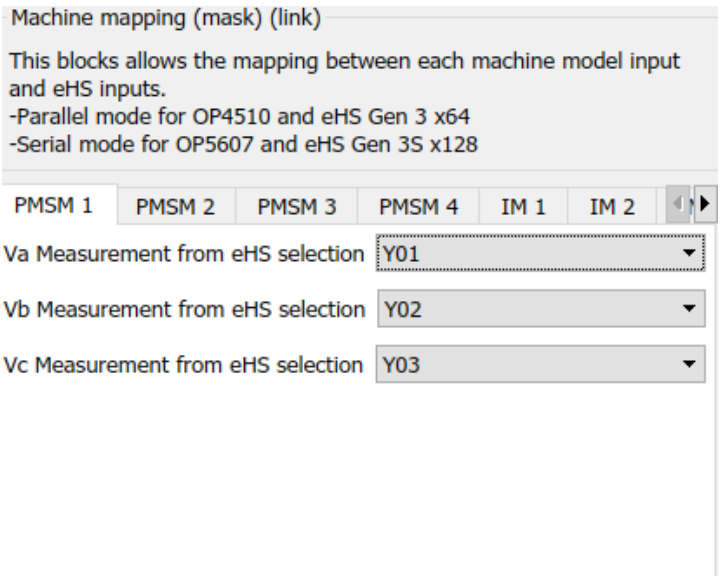


Figure A.7: Machine Interface port configuration.

Signal Name	Port
Va measurement	Y01
Vb measurement	Y02
Vc measurement	Y03
Load In	17

Table A.3: Machine Interface block configuration

The parameters of the PMSM to be simulated, are declared inside the PMSM block (Figure A.6). The machine parameters are found in Table 2.2. Nonetheless, the inductance and flux linkage parameters, can also be sourced from tables in the case they ought to be modelled as current dependant.

A.3. Vehicle Resistance Force Model

During this modelling stage, a particular consideration was taken. The term J_w in equation 2.13, is actually considering the inertia of two wheels and an additional 20% for the rotational inertias added by shafts and attachments. The reason, is to make an approximation of the rotational inertia that would represent the two wheels of a front-axle driven vehicle.
The individual wheel inertia J_{wheel} given in table 2.1 is computed based on an approximation proposed by Metz et al [27] for mounted wheel tyres according to the size of the tyre of the Nissan Leaf in table 2.1.

Therefore, the rotational inertia is:

$$J_w = J_{wheel} * 2 * 1.2 \tag{A.1}$$

Regarding the load forces, or resistance forces. These are modelled from equations 2.2, 2.3 and 2.4 in chapter 2, using Simulink blocks. Finally, the implementation of the resistance forces of the powertrain can be seen in figure A.8.

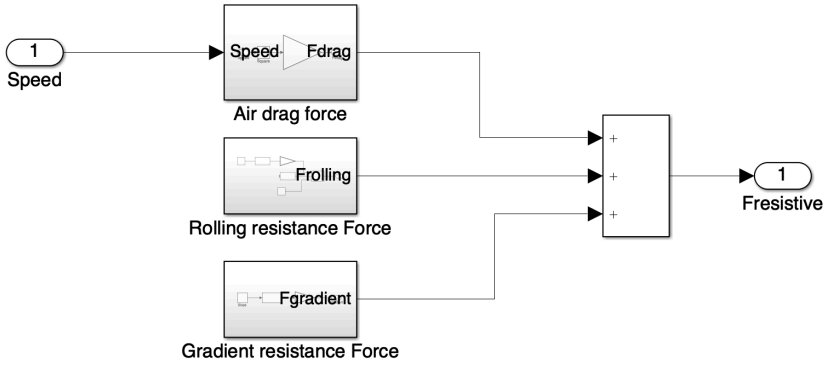


Figure A.8: Powertrain's resistance forces model.

A.4. SIL Powertrain Simulation

A.4.1. Park Transformation model

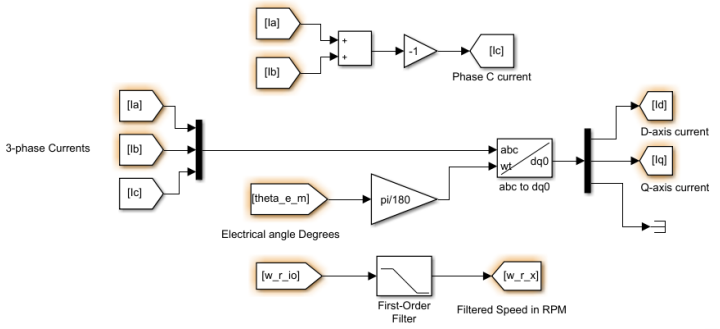


Figure A.9: Park transformation of three-phase currents

A.4.2. Speed Controller model

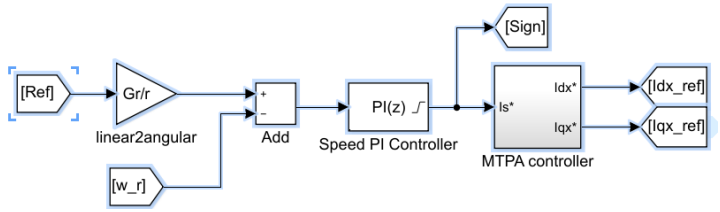


Figure A.10: Speed controller model.

A.4.3. MTPA Control Strategy model

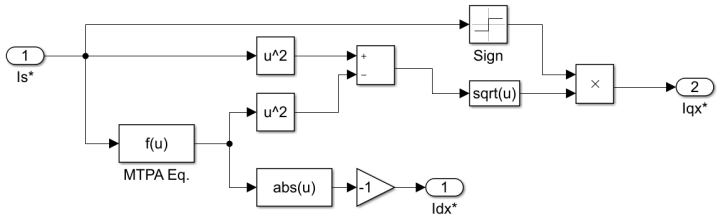


Figure A.11: MTPA equations model.

The I_D and I_Q MTPA current component equations are modelled using the 'Fcn' block and the constant motor parameters from table 2.2. The resulting current pair ${I_{dx}}^*$ and ${I_{qx}}^*$ showed in Figure A.11, will be used as the desired current set-points to the current controller stage when the PMSM is operating below its base speed.

A.4.4. Current Controller Model

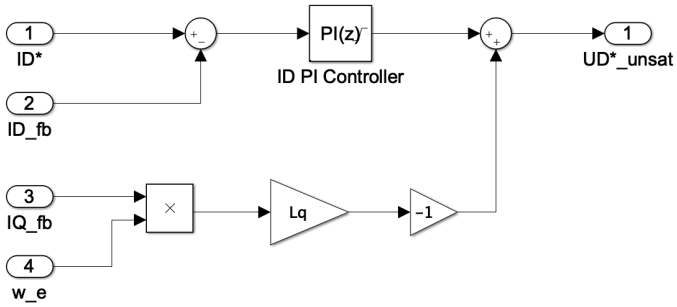


Figure A.12: I_D Control loop and feedforward compensation.

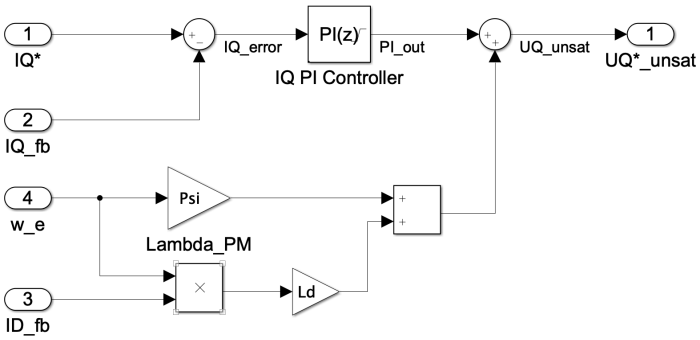


Figure A.13: I_Q Control loop and feedforward compensation.

A.4.5. SVPWM Generation

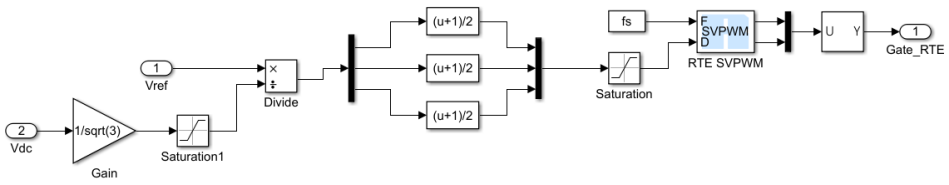


Figure A.14: SVPWM Generation.

B

Configuration of the OP5700 I/O

B.1. Analog Output Configuration

The OP5700 chassis at the laboratory is provided with an OP5330-3 card supporting 16 Analog Output (AO) channels. This section will provide some insight on how to map and rescale as needed the plant variables that the OP5700 will output as analog signals.

Four output signals will be mapped, as Figure 4.13 explains.

- Stator current phase A
- Stator current phase B
- Machine's electrical angle
- Speed reference signal

The first two signals are the PMSM phase currents. As previously stated in Table 4.2, these two signals are generated and measured inside of the FPGA three-phase inverter circuit. Therefore, these two signals are already stamped with the 'eHS' mark in their names $Y05$ and $Y06$. In the case of a software in the loop (SIL) simulation, the same analog outputs are 'looped back' into the OP5700's analog inputs and will be used by the simulated controller running in the simulator's CPU.

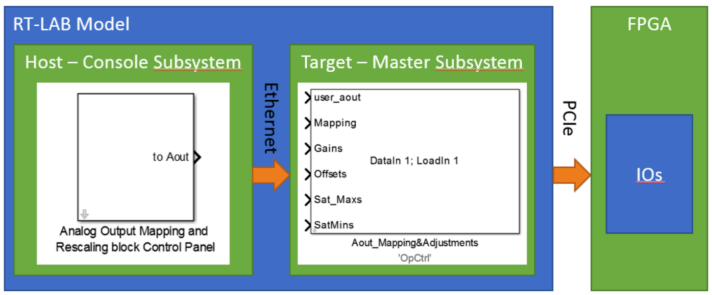


Figure B.1: AOMR communication flow

As it can be seen in figure B.1, the 'Analog Output Mapping and Rescaling' (AOMR) block is used to assign the available plant signals to the desired analog output channel and as well, to scale and calibrate them within the FPGA. The AOMR is normally located within the console subsystem of an RT-LAB project.

The AOMR block configuration is displayed in Table B.1.

On the other hand, the 'Aout_Mapping&Adjustments' block is as well needed. In this block is where the internal communication channels are configured according to the firmware configuration file of the OP5700. Additionally, the input ports of these block must be connected to the respective output of the AOMR block. How ever, the first input 'user_aout' is where all the signals generated in RT-LAB but not

in the FPGA, must be connected.
The set up, for both HIL and SIL applications, is found in tables B.2 and B.3.

Aout Mapping and Adjustment Block		
Parameter	**HIL config.**	**SIL config**
Num. of Channels	16	16
Num. of Aout User Signals	2	4
Block Index	0	0
Data In Port Number	21	21
Load In Port Number	12	12

Table B.1: AOMR block parameter within Master Subsystem

Channel	HIL/SIL	Signal Description HIL	Signal Description SIL
Ch0	eHS.Y05	PMSM phase A current	PMSM phase A current
Ch1	eHS.Y06	PMSM phase B current	PMSM phase B current
Ch2	User 1	Sineθ_r	Mechanical speed
Ch3	User 2	Cosineθ_r	Electrical angle
Ch4	User 3	Speed reference	N/A
Ch5	User 4	Speed measurement	N/A

Table B.2: Analog channels and signal mapping for HIL and SIL simulations.

Signal Scaling					
Signal	**Range**	**Gain**	**Offset**	**Max Volt**	**Min Volt**
PMSM I_A	-800 to 800 A	1/53.6	0	14.925	-14.925
PMSM I_B	-800 to 800 A	1/53.6	0	14.925	-14.925
Speed reference	0 to 1050 rad/s	1/67.335	0	14.925	-14.925
Speed measurement	0 to 10000 RPM	1/670	0	14.925	-14.925
Electrical angle	0 to 360 degrees	1/24.12	0	14.925	-14.925
Cosineθ_r	14	0	14.925	-14.925	
Sineθ_r	14	0	14.925	-14.925	

Table B.3: Analog signals scaling for HIL simulation

As Table B.3 shows, all of the signals that are mapped to a analog output channel are scaled, in other words, they are reduced by a gain factor so that their full amplitude range can be represented within the maximum and minimum output voltages set for the current configuration. These output voltage ranges for the AO module are set to fit the analog inputs in the HIL controller board.

B.2. Analog Input Configuration

The analog inputs (AI) of the OP5700 will only be used in the SIL application of the powertrain. These will receive the same signals mapped to the AO as explained in the previous section. The analog inputs are mapped into the RT-LAB model using the ‘AnalogIn’ block, which is shown in figure B.2, and the respective parameter setting is shown in figure B.3.

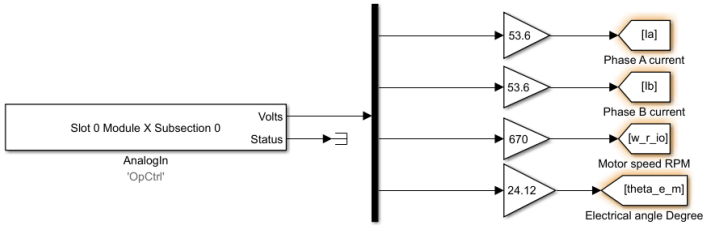


Figure B.2: Analog input block and scaling for SIL simulation

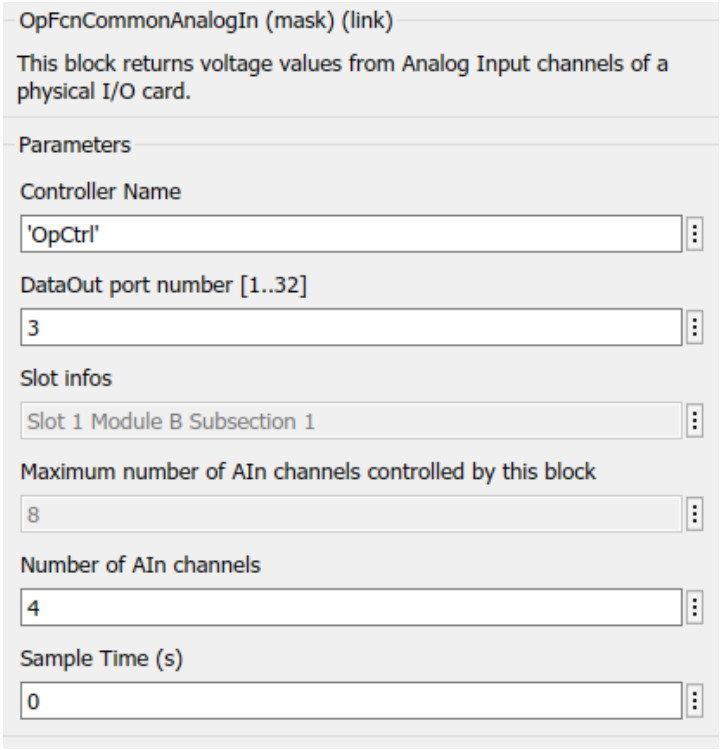


Figure B.3: Analog Input block configuration

Figure B.2 shows that the analog input signals are scaled back to their original values by multiplying them by the same gain they were down-scaled when they were originally mapped to the analog outputs in previous section.

B.3. Digital Input Configuration

As figure A.4 showed previously, the gates of the IGBTs in the inverter FPGA model are driven directly from the digital inputs of the OP5700 using the ‘Gate Control Selection Panel’. This option allows to map an individual digital input channel to a desired IGBT gate as figure B.4. Considering that the switches in the inverted model are arranged according to the Simulink’s Universal Bridge block, the signal mapping in the Gate Control Selection Panel is done as table B.4 shows.

Gate Control Selection Panel				
IGBT	**Switch Name**	**Switch Control Type**	**DI Channel (Index)**	**Switch Position**
Q1	SW01-06_01	From Digital Inputs	0	First leg upper switch
Q2	SW01-06_02	From Digital Inputs	1	First leg lower switch
Q3	SW01-06_03	From Digital Inputs	2	Second leg upper switch
Q4	SW01-06_04	From Digital Inputs	3	Second leg lower switch
Q5	SW01-06_05	From Digital Inputs	4	Third leg upper switch
Q6	SW01-06_06	From Digital Inputs	5	Third leg lower switch

Table B.4: Gate Control Selection Panel Configuration

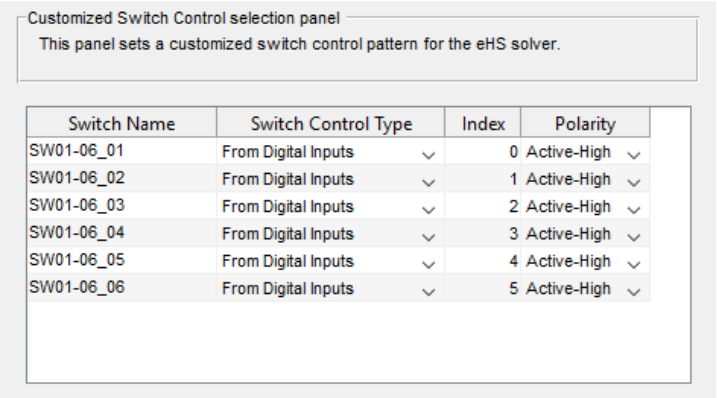


Figure B.4: Gate Control Selection Panel

B.4. Digital Output Configuration

The mapping of the digital outputs is done using the ‘EventGenerator’ block as in figure A.14. During the implementation of the simulated controller, the digital output module of the OP5700 is used to output the 6 pulse signals that are looped back into the the inverter model through the DI module and then drive the gates of the IGBTs.

The configuration of the DO block is shown in table B.5.

	DataIn Port Number	**Number of Channels**	**Number of events per Channel**	**Time Unit**
Value	22	6	10	Time ratio

Table B.5: Event Generator block configuration for Digital Output mapping

C

Hardware in the Loop Controller set-up

C.1. C28x Hardware Interrupt block configuration

Parameter	CPU interrupt number	PIE interrupt number	Simulink task priorities	Preemption flags
Value	1	2	30	0

Table C.1: C28x Hardware Interrupt block configuration

C.2. ADC module configuration

The configuration of each ADC block is listed in table C.2. Each analog input module has a 12-bit ADC, the output of the block is a stream of values between 0 and 4095 with uint16 data type.

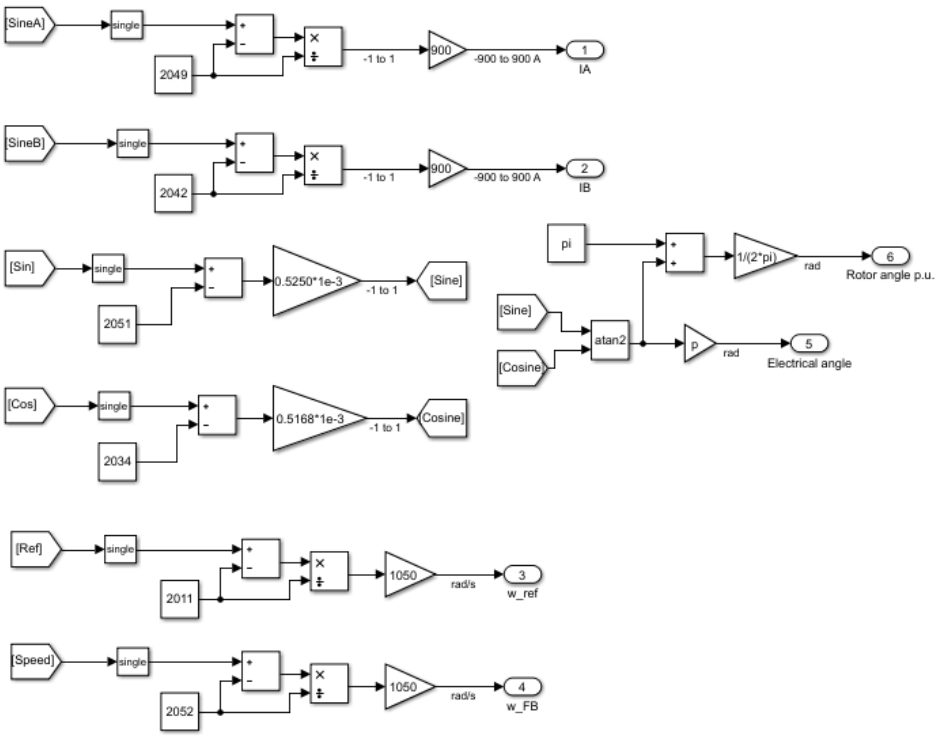


Figure C.1: ADC signals scaling.

	ADC Block					
Parameter	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**
ADC Module	B	C	A	A	C	A
Input Chanel	ADCIN2	ADCIN2	ADCIN2	ADCIN0	ADCIN5	ADCIN15
ADC Resolution	12-bit	12-bit	12-bit	12-bit	12-bit	12-bit
SOC trigger number	SOC0	SOC0	SOC0	SOC1	SOC1	SOC2
SOCx acquisition window	26	26	26	26	26	26
SOCx trigger source	ePWM1 ADCSOCA	ePWM1 ADCSOCA	ePWM1 ADCSOCA	ePWM1 ADCSOCA	ePWM1 ADCSOCA	ePWM1 ADCSOCA
Data type	unint16	unint16	unint16	unint16	unint16	unint16

Table C.2: Simulink ADC block configuration for LaunchXL-F28379D DSP board

C.3. ePWM module configuration

Three space-vector duty cycle signals need to be scaled into microprocessor clock cycles before being used by the ePWM block. Keeping in mind that the clock frequency of the micro-controller is $f_{Clock} = 200$ MHz and the desired switching frequency is $f_{PWM} = 5$ kHz, the scaling is done following equation C.1.

$$TBPRD = \frac{T_{PWM}/_2}{2T_{Clock}} = 10000 \tag{C.1}$$

Equation C.1 shows in number of clock-cycles the needed length for the Time Based Period (TBPRD) in an up-down count mode to obtain a 5 kHz PWM symmetrical waveform. For more information on how to configure the ePWM block for the TMS320F28379D chip, please refer to the manufacturer manual [19],[20].

The input port of the ePWM block, WA, seen in figure 4.21, will be used as the modulating waveform that is compared against the internal counter register TBCTR, of the ePWM module that is counting from zero up to TBPRD, and from TBPRD down to zero every switching period as observed in the example in figure C.2.

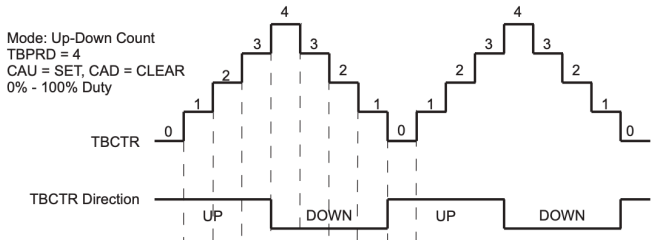


Figure C.2: Internal TBCTR Up-Down counter example for ePWM module [19].

When the signal at the WA port of the ePWM block is equal to the TBCTR counter con the way up, the PWM output port switches on, and when it is equal to TBCTR on the way down, the PWM switches off. It is when the TBPRD and the TBCTR counter match that an interrupt can be generated to trigger the ADC modules Start of Conversion (SOC). Consequently, the SV modulation waveform needs is scaled accordingly to values between 0 to TBPRD as observed in figure C.4.

Each ePWM block is able to generate a EPWMxA signal and its EPWMxB complement. The given configuration to the ePWM1 module of the TMS320F28379D is shown in figure C.5. Following this configuration a PWM signal is generated at both ePWM1A and ePWM1B ports. Likewise, the same parametrisation must be followed for ePWM2 and ePWM3 blocks except the 'Enable ADC start of conversion for module A' checkbox observed in figure C.5e. This option is only reserved for module ePWM1 since is the one that will generate the SOC for all of the ADC blocks as specified in table C.2.

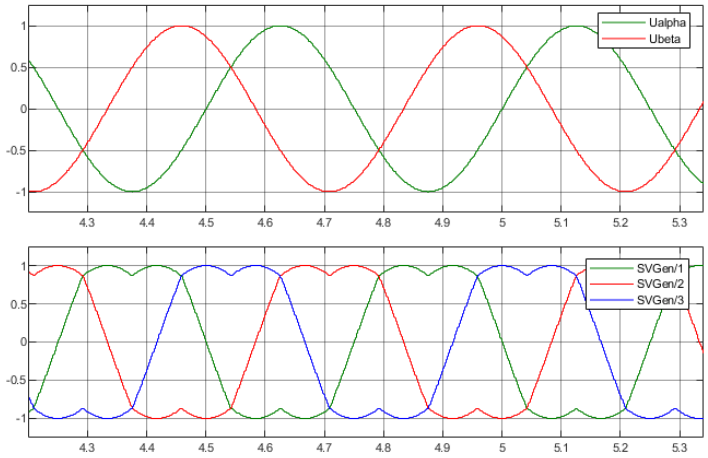
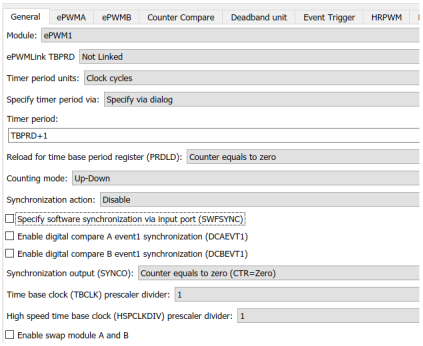
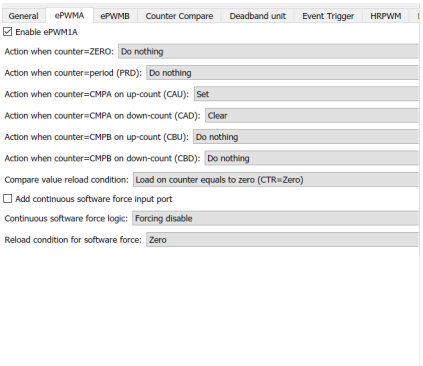


Figure C.3: SV modulating waveform generated from SVGen block in HIL controller.

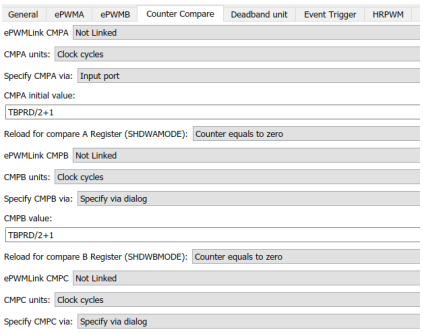
Figure C.4: Scaling of space vector modulation signal.



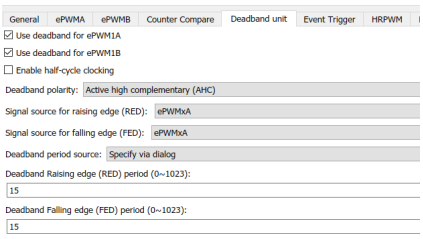
(a) ePWM1 General tab



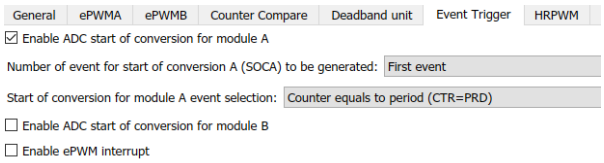
(b) ePWM1A Tab



(c) ePWM1 Counter Compare tab



(d) ePWM1 Dead band unit tab



(e) ePWM1 Event trigger tab

Figure C.5: ePWM1A block configuration

Bibliography

[1] K. Abdallah A. Umut. Aerodynamics concept study of electric vehicles. *Chalmers University of Technology*, 2017.

[2] Ahmed Kotb Abdalla. Validation and enhancement of two-level inverter models for very low time-step real-time applications, 2017.

[3] A. Sathyan B. Bilgin. *Fundamentals of Electric Machines*, chapter 5, pages 107–186. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2015. ISBN 978-1-4665-9770-9.

[4] Tim Burress. Benchmarking of competitive technologies project id: Ape006. *Oak Ridge National Laboratory*, 2012.

[5] Jean Bélanger, P Venne, and Jean-Nicolas Paquin. The what, where and why of real-time simulation. *Planet RT*, pages 37–49, 2010.

[6] Calin Capitan. Torque control in field weakening mode. *Institute of Energy Technology, Aalborg University*, 2009.

[7] Wen-Po Chiang, Dejun Yin, and Hiroshi Shimizu. Slip-based regenerative abs control for in-wheel-motor drive ev. *Journal of the Chinese Institute of Engineers*, 38(2):220–231, 2015. doi: 10.1080/02533839.2014.955974.

[8] Alan Courtay and Bryan Kelly. Development of a pmsm traction drive system for phev, hev, ev using virtual prototyping. *IEEE*, 2018. ISSN 978-1-5386-3048-8.

[9] Rik De Doncker, Duco W.J. Pulle, and André Veltman. *Advanced Electrical Drives: Analysis, Modeling, Control*, volume 52. 01 2011. doi: 10.1007/978-94-007-0181-6.

[10] Qichao Dong, Hongzhong Qi, Xintian Liu, and Yansong Wang. Calibration and optimization of an electric vehicle powertrain system. *Journal of the Chinese Institute of Engineers*, 41(7):539–546, 2018. doi: 10.1080/02533839.2018.1530952.

[11] Christian Dufour, Simon Abourida, and Jean Bélanger. Hardware-in-the-loop simulation of power drives with rt-lab. pages 1646 – 1651, 12 2005. ISBN 0-7803-9296-5. doi: 10.1109/PEDS.2005.1619952.

[12] M Eshani, Y Gao, S E. Gay, and A Emadi. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*. 01 2009.

[13] M. Ferdowsi. Plug-in hybrid vehicles - a vision for the future. In *2007 IEEE Vehicle Power and Propulsion Conference*, pages 457–462, Sep. 2007. doi: 10.1109/VPPC.2007.4544169.

[14] John G. Hayes and Kevin Davis. Simplified electric vehicle powertrain model for range and energy consumption based on epa coast-down parameters and test validation by argonne national lab data on the nissan leaf. *2014 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–6, 2014. doi: 10.1109/itec.2014.6861831.

[15] Izuho Hirano Shigeaki Ishikawa Hirofumi Shimizu, Takahito Okubo and Makoto Abe. Development of an integrated electrified powertrain for a newly developed electric vehicle. *SAE Technical Paper Series*, 2013. doi: 10.4271/2013-01-1759.

[16] Simon Howroyd and Rob Thring. An electric vehicle model and validation using a nissan leaf: A python-based object-oriented programming approach. *Advances in Mechanical Engineering*, 10 (7), 2018. ISSN 1687-8140 1687-8140. doi: 10.1177/1687814018782099.

[17] OPAL-RT Technologies Inc. ehs user guide. 2016.

[18] OPAL-RT Technologies Inc. efpgasim machine models user guide. 2018.

[19] Texas Instruments. Technical reference manual tms320f2837xd dual-core delfino microcontrollers. 2013.

[20] Texas Instruments. Launchxl-f28379d overview. 2016.

[21] Namdoo Kim, Ayman Moawad, and Aymeric Rousseau. Validation of sizing algorithm for several vehicle powertrains. 10 2016.

[22] Sang-Hoon Kim. Chapter 4 - modeling of alternating current motors and reference frame theory. In Sang-Hoon Kim, editor, *Electric Motor Control*, pages 153 – 202. Elsevier, 2017. ISBN 978-0-12-812138-2. doi: 10.1016/B978-0-12-812138-2.00004-0.

[23] G Kohlrusz and D Fodor. Comparison of scalar and vector control strategies of induction motors. *Hungarian Journal of Industry and Chemistry*, 39(2):265–270, 2011.

[24] Ramu Krishnan. *Permanent magnet synchronous and brushless DC motor drives*. CRC press, 2017.

[25] Muyang Li. Flux-weakening control for permanent-magnet synchronous motors based on z-source inverters. *Marquette University*, 2009.

[26] P. Lin and Y. Lai. Voltage control of interior permanent magnet synchronous motor drives to extend dc-link voltage utilization for flux weakening operation. In *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pages 1689–1694, Nov 2010. doi: 10.1109/IECON.2010.5675427.

[27] Akouris C. Agney C. Metz, L. and M. Clark. Moments of inertia of mounted and unmounted passenger car and motorcycle tires. *SAE Technical Paper 900760*, 1990. doi: https://doi.org/10.4271/900760.

[28] Morimoto, Takeda, and Hirasa. Expansion of operating limits for permanent magnet motor by current vector control considering inverter capacity. *IEEE Transactions on Industry Applications*, 26(5), 1990.

[29] Mohan Ned. *Electric Drives An Integrative Approach*. MNPERE, Minneapolis, 2003. ISBN 0971529256.

[30] PD Chandana Perera. *Sensorless control of permanent-magnet synchronous motor drives*. Institute of Energy Technology, Aalborg University, 2002.

[31] Yoshinori Sato, Shigeaki Ishikawa, Takahito Okubo, Makoto Abe, and Katsunori Tamai. Development of high response motor and inverter system for the nissan leaf electric vehicle. *SAE Technical Paper Series*, 2011. doi: 10.4271/2011-01-0350.

[32] Shigeaki Ishikawa Tohru Nakada and Shunji Oki. Development of an electric motor for a newly developed electric vehicle. *SAE Technical Paper Series*, 2014. doi: 10.4271/2014-01-1879.

[33] Jonathan Wai and T. Jahns. A new control technique for achieving wide constant power speed operation with an interior pm alternator machine. *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No.01CH37248)*, 2:807–814 vol.2, 2001.

[34] Guang Wu, Xing Zhang, and Zuomin Dong. Powertrain architectures of electrified vehicles: Review, classification and comparison. *Journal of the Franklin Institute*, 352, 01 2014. doi: 10.1016/j.jfranklin.2014.04.018.

[35] Rong Yang. Electrified vehicle traction machine design with manufacturing considerations, 2016.

[36] Jianfeng Yu, Ting Zhang, and Jianming Qian. 10 - modern control methods for the induction motor. In Jianfeng Yu, Ting Zhang, and Jianming Qian, editors, *Electrical Motor Products*, pages 147 – 172. Woodhead Publishing, 2011. ISBN 978-0-85709-077-5. doi: https://doi.org/10.1533/9780857093813.147.