



Bachelor Degree in Industrial Technologies

CURRENT CONTROL IN A CC-CA CONVERTER VIA MICROCONTROLLER

Bachelor Thesis

Author:

Jorge Suárez Porras

Supervisor:

Aurelio García Cerrada

Madrid

January 2020

¹English version updated on December 2020.

This english version has not been corrected.

*To religion, life and science,
that encompasses and floods everything.*

Intentional white page

Summary

The current energy system for the production, distribution and consumption of electricity from power plants mainly uses non-renewable resources as raw materials. These plants produce and distribute energy in alternating current through a complex mesh that interconnects consumers and producers. The control in the production and management of energy in these non-renewable systems is widely extended and is widely known, so its control is, today, safe. However, there is a strong trend towards obtaining energy from renewable raw materials, such as solar, hydraulic, wave energy, wind energy ... etc. These new energy sources are being implemented in the form of microgrids (*micro-grids*) [1], so that they can meet certain demands for electrical energy locally, for example, to industrial complexes, residential neighborhoods, hospitals ... etc [2]. Since they are directly connected to the large distribution network, the operating modes are severely restricted based on the state of the network. For this reason, if the distribution network stops supplying power to consumers, these renewable sources of energy are disconnected from the network for safety reasons[3], and do not help to alleviate the incident. Further development of better technologies for these connections with the distribution network, together with development of batteries and energy storage, will significantly help to establish more renewable sources of energy in the electric system, making it gradually, evolve to a more ecological, efficient and green energy model. One of the main technologies used to carry out a connection of renewable energy sources to the large distribution network are the electronic power converters DC - AC voltage source (or *Voltage Source Converter* and *Current Source Converter*

Given the nature of the project (intended for teaching), the low voltages that are operated and due to the ease of handling and programming of the control and its subsequent implementation, it was decided to use a VSC that has GaN MOS-FETS as a particularity. Gallium nitride or GaN is currently used [4] for circuit breakers intended for certain applications [5]. These switches fall within the so-called “forbidden band ” or “ broadband ” (*wide-bandgap semiconductors* or *WBG semiconductors*) and with them it is intended to improve the switching frequency (increasing it), reduce conduction losses and achieve higher blocking voltages than those obtained with Si (Silicon) switches (which is theoretically limited to about 6.5kV) [6]. In this document, first of all, it will be described what is the main objec-

tive of the project. The mathematical approach used to control VSCs is discussed below. Next, the fundamentals of this type of converters will be described; what they are, how they are used and controlled, what they are made of ... etc. Later, the electronic converter that has been used in the project will be shown, as well as the microcontroller and the components that have been used to build the prototype. Finally, a series of controls will be studied, showing simulations in continuous and discrete times, and their experimental validation in the prototype.

Keywords

Inverter, VSC, Current Control, Park Transform, LAUNCHPADXL - F28379D, TMS320F28379D, BOOSTXL - 3PhGaNInv.

Contents

| | |
|--|-----------|
| Summary | I |
| Keywords | II |
| 1 Introduction | 1 |
| 1.1 Project Motivation | 1 |
| 1.2 Objectives | 2 |
| 2 Voltage Source Converters | 3 |
| 2.1 Brief description of the PWM | 4 |
| 2.2 Power Transistors | 5 |
| 2.3 VSCs with PWM | 10 |
| 2.4 VSC Description | 11 |
| 3 Connection of electronic voltage converters to the grid | 16 |
| 3.1 Grid synchronization | 19 |
| 3.2 Some simple filters | 21 |
| 4 Device Control | 23 |
| 4.1 System equations | 23 |
| 4.2 Three-phase variables and reference systems | 25 |
| 4.3 Model in d-q axes of a three-phase inductance | 29 |
| 4.4 Base Magnitudes | 31 |
| 5 Design, simulation and implementation of P - PI controllers | 32 |
| 5.1 Design of a P - PI control in continuous time | 34 |
| 5.2 Description of the simulator | 39 |
| 5.3 P Control | 42 |
| 5.4 PI Control | 50 |
| 6 Main Components of the Prototype | 58 |
| 6.1 Microprocessors and microcontrollers | 58 |
| 6.2 VSC BOOSTXL-3PhGaNInv | 68 |
| 6.3 Prototype Software Configuration | 80 |

| | | |
|----------|---|------------|
| 7 | Experimental Results | 99 |
| 8 | Budget | 108 |
| 8.1 | Materials Budget | 108 |
| 8.2 | Labor Budget | 109 |
| 8.3 | Total Budget | 109 |
| | Bibliography | 110 |
| A | Installing the TI C2000 Package for Matlab | 115 |
| B | Simulator Interface | 123 |
| C | Files | 128 |
| D | Sustainable Development Goals | 129 |

List of Figures

| | | |
|------|--|----|
| 2.1 | Schematic of a three-phase VSC. V_{dc} is the direct input voltage, V_{a0} , V_{b0} and V_{c0} are the alternating output voltages. | 3 |
| 2.2 | PWM generation by comparing with a triangular [11] | 4 |
| 2.3 | Space Vector Voltages [10] | 4 |
| 2.4 | Diagram of a MOSFET transistor a) channel n and b) channel p. [13] | 5 |
| 2.5 | a) Diagram of a channel n and b) MOSFET transistor Diagram of a channel MOSFET transistor. [13] | 6 |
| 2.6 | Diagram of a BJT transistor in direct conduction [15] | 6 |
| 2.7 | IGBT diagram as a a) BJT b) MOSFET | 7 |
| 2.8 | Conduction band of semiconductor materials [13]. | 8 |
| 2.9 | Hypothetical short circuit of phase A. | 8 |
| 2.10 | On and off time of a semiconductor | 9 |
| 2.11 | Time when a branch is short-circuited. | 9 |
| 2.12 | Analog circuit made to satisfy the required dead time. | 10 |
| 2.13 | Service factor definition | 11 |
| 2.14 | Scheme of a VSC feeding a load. V_{dc} is the controllable DC input voltage. T1 to T6 are the switches triggered with the microprocessor. V_{a0} , V_{b0} and V_{c0} are the alternating output voltages and i_a , i_b and i_c the currents through the load. L and R is the load that is fed. . . | 12 |
| 2.15 | Prototype functional diagram. | 13 |
| 2.16 | Synchronization of the ADC with PWM with a single shot per cycle. | 14 |
| 2.17 | ADC synchronization with PWM with double shot per cycle. | 15 |
| 3.1 | VSC connected in <i>grid-forming</i> format [22]. | 16 |
| 3.2 | VSC connected in <i>grid-feeding</i> format [22] | 17 |
| 3.3 | Diagram of a converter in <i>grid-supporting-forming</i> format with voltage source model [23]. | 17 |
| 3.4 | Diagram of a converter in <i>grid-supporting-feeding</i> format with current source model [23]. | 18 |
| 3.5 | PLL block diagram [26]. | 19 |
| 3.6 | PLL linearized model [25]. | 19 |
| 3.7 | MSRF-PLL linearized model [27]. | 20 |
| 3.8 | Filter L. | 21 |

| | | |
|------|---|----|
| 3.9 | Filter LC. | 22 |
| 3.10 | Filter LCL. | 22 |
| 3.11 | Scheme of three-phase inverter with RL load. | 22 |
| 4.1 | Circuit vector scheme. V_{a0} , V_{b0} and V_{c0} are the inverter output voltages. V_a , V_b and V_c the voltages determined by the control and i_a , i_b and i_c the measured currents. | 23 |
| 4.2 | Clarke transformation. | 26 |
| 4.3 | Transformada de Park. | 27 |
| 4.4 | Current control diagram in d-q axes [25] | 29 |
| 5.1 | General control scheme block diagram. | 33 |
| 5.2 | Block diagram of a continuous time control[42]. | 34 |
| 5.3 | Block diagram of a PI control in parallel format. | 35 |
| 5.4 | Simulink to obtain the plant in continuous time. | 36 |
| 5.5 | Block diagram of a discrete time control. | 37 |
| 5.6 | PID block for Software and Hardware implementation. | 39 |
| 5.7 | Interface for simulator configuration. Identical to that used in the prototype hardware. The development is described in Appendix B | 40 |
| 5.8 | Simulator parameter settings. | 41 |
| 5.9 | Time response of the d axis of a P Control in continuous time. Id pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s. | 42 |
| 5.10 | Time response of the q axis of a P Control in continuous time. Pulse of Id = 1 in 0.3s. Pulse of I_q = 1 in 0.8s. | 43 |
| 5.11 | Open-loop frequency response of a P-Control in continuous time. | 43 |
| 5.12 | Configuration of a simulator with a control P. Identical to the one used in the simulator. The development is described in Appendix B. | 46 |
| 5.13 | Scope of a simulator with a P control. The simulink PID block is implemented in discrete time. The simulation control values are listed in Table 5.2 | 47 |
| 5.14 | Graph of the reference following error of the d axis current with a Control P. Hardware implementation. The y-axis data is scaled 0.1 of the pu value. | 49 |
| 5.15 | Id temporal response with constant phase margin with PI control in continuous time. Id pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s. | 51 |
| 5.16 | Temporal response of I_q with constant phase margin with PI control in continuous time. Id pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s. | 51 |
| 5.17 | Id temporal response with constant crossover pulsation with PI control in continuous time. Id pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s. | 52 |
| 5.18 | Temporal response of I_q with constant crossover pulsation with PI control in continuous time. Id pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s. | 52 |

| | | |
|------|---|----|
| 5.19 | Open-loop response with constant phase margin with PI control in continuous time. | 53 |
| 5.20 | Open-loop response with constant crossover pulsation with continuous-time PI control. | 53 |
| 5.21 | Configuration applied to the simulator with a PI Control. Identical to the one used in the simulator. The development is described in Appendix B | 54 |
| 5.22 | Scope of a simulator with a PI control. The PID block is implemented in discrete time. It can be observed in the jumps that occur in the dq axes when their reference is zero, that the variations in the q axis are disturbances in the d axis and vice versa. The simulation control values are listed in Table 5.4 | 55 |
| 5.23 | d -axis current reference tracking error plot with a PI Control. Hardware implementation. The axis values are scaled to 0.1 of the actual value in pu. It is observed that the coil saturates for values higher than 3pu (Approx. 6.1A). | 57 |
| 6.1 | Scheme of a computer. | 58 |
| 6.2 | Scheme of a microcontroller. | 59 |
| 6.3 | Evolution of the TMS320 family [44]. | 60 |
| 6.4 | LaunchPad F28379D Top view. | 61 |
| 6.5 | LaunchPad F28379D bottom view. | 61 |
| 6.6 | LAUNCHXL - F28379D [45] | 62 |
| 6.7 | LAUNCHXL - F28379D detailed top view[45] | 62 |
| 6.8 | LAUNCHXL - F28379D Pinout [46]. | 62 |
| 6.9 | LAUNCHXL - F28379D Pinout left side in a detailed view [46]. The DAC outputs for which there is a block in the Simulink toolbox are outlined. | 63 |
| 6.10 | LAUNCHXL - F28379D Pinout right side in a detailed view [46]. Pins for which there is no specific Simulink block and cannot be used with said library as DAC are outlined. | 64 |
| 6.11 | Texas Instruments C2000 Processor Toolbox. [47] | 66 |
| 6.12 | Texas Instruments LaunchPad F2837xD Toolbox. [47] | 67 |
| 6.13 | BOOSTXL-3PhGaNIInv top view. Resistances changed are boxed. | 68 |
| 6.14 | BOOSTXL-3PhGaNIInv bottom view. | 69 |
| 6.15 | BOOSTXL-3PhGaNIInv function diagram [48]. | 70 |
| 6.16 | LMG5200 in a) off mode and b) on mode [50]. | 71 |
| 6.17 | INA240 functional diagram [48] | 71 |
| 6.18 | INA240 types based on their gain [51] | 71 |
| 6.19 | INA240 electronic circuit. [49] | 72 |
| 6.20 | BOOSTXL - 3PhGaNIInv left side connections (J1) [52]. | 73 |
| 6.21 | BOOSTXL - 3PhGaNIInv right side connections (J2) [52]. | 73 |
| 6.22 | LaunchPad set with VSC with the connection on top. | 74 |
| 6.23 | LaunchPad set with VSC with the connection on bottom. | 74 |

| | | |
|------|---|----|
| 6.24 | Conjunto LaunchPad con VSC con la conexión en la parte inferior. Vista lateral. | 76 |
| 6.25 | Conjunto LaunchPad con VSC con la conexión en la parte inferior. Vista en perspectiva. | 76 |
| 6.26 | Encapsulated VSC and Launchpad. DC voltage input. | 77 |
| 6.27 | Encapsulated VSC and Launchpad. AC voltage outputs and USB connection. | 78 |
| 6.28 | Encapsulated VSC and Launchpad. Oscilloscope pins. | 79 |
| 6.29 | Simulink ePWM block. TI C2000 library. | 80 |
| 6.30 | PWM generation form options in Simulink's ePWM block [47] . . . | 82 |
| 6.31 | Simulink ePWM block. TI C2000 library. | 83 |
| 6.32 | Diagram of the PWM nomenclature in Simulink for the VSC connected to the LaunchPad at the bottom (Section 6.2.2). | 84 |
| 6.33 | ePWMxA configuration. | 85 |
| 6.34 | Final appearance of the ePWM Block. The input is the duty cycle in seconds or in clock cycles. | 85 |
| 6.35 | <i>Counter Compare</i> configuration tab. | 86 |
| 6.36 | ePWMxB Configuration. | 86 |
| 6.37 | ePWMxB configuration from <i>DeadBand Unit</i> tab. | 87 |
| 6.38 | Configuration of ADC startup synchronized with ePWM. | 87 |
| 6.39 | VSC PWM enable configuration. | 88 |
| 6.40 | <i>ePWMB</i> tab configuration to obtain signals by PWM. | 89 |
| 6.41 | Configuration tab <i>Counter Compare</i> to obtain signals by PWM. Counters <i>CMPA</i> and <i>CMPB</i> | 89 |
| 6.42 | Configuration tab <i>Counter Compare</i> to obtain signals by PWM. Counters <i>CMPA</i> and <i>CMPB</i> | 90 |
| 6.43 | <i>Deadband Unit</i> tab configuration to obtain signals through PWM. . . | 90 |
| 6.44 | Final appearance of the ePWM Block to obtain signals. The input signals are the duty cycle. | 90 |
| 6.45 | Simulink ADC block. TI C2000 Library. | 91 |
| 6.46 | ADC Block Configuration. | 92 |
| 6.47 | DAC simulink block. TI C2000 Library. | 93 |
| 6.48 | Difference in PWM - DAC output measurements. | 94 |
| 6.49 | Current through one phase, output through the DAC, through the PWM and measured with a current clamp. | 94 |
| 6.50 | Block diagram to obtain a signal through the DAC or PWM. Actual 1.5V average value may vary. | 95 |
| 6.51 | Saturated signal through DAC output port. | 96 |
| 6.52 | <i>Solver</i> tab configuration for prototype implementation. | 97 |
| 6.53 | <i>Hardware Implementation</i> configuration tab for prototype implementation. | 98 |
| 7.1 | Laboratory set up. | 99 |

| | | |
|------|--|-----|
| 7.2 | Three-phase currents through the coils. The identification of the signals is shown in detail in the figure. See table 7.1. | 100 |
| 7.3 | Three-phase currents through the saturated coils. | 101 |
| 7.4 | Three-phase currents through the inductances with d-axis reference current in the form of pulses. Reference $I_q = 0$. The identification of the signals is shown in detail in the figure. | 102 |
| 7.5 | Three-phase currents through the inductances with d-axis current reference in sinusoidal shape of 4Hz frequency. The identification of the signals are shown in detail in the figure. I_q reference = 0. | 103 |
| 7.6 | Detail of the three-phase currents through the coils with a 4Hz frequency sinusoidal d-axis current reference. The identification of the signals are shown in detail in the figure. I_q reference = 0. | 104 |
| 7.7 | Three-phase currents through the inductances. Time response of d-axis and q-axis current. Application of references simultaneously. The identification of the signals are shown in detail in the figure. | 105 |
| 7.8 | Three-phase currents through the inductanced. Time response of d-axis and q-axis current inversely applied. The identification of the signals are shown in detail in the figure. | 106 |
| 7.9 | Three-phase currents through the inductances. Time response of q-axis current. I_d reference = 0. The identification of the signals are shown in detail in the figure. | 107 |
| A.1 | Matlab Add-Ons menu. | 116 |
| A.2 | Library Configuration window 1 <i>C2000 Processors</i> | 116 |
| A.3 | Library Configuration window 2 <i>C2000 Processors</i> | 117 |
| A.4 | Library Configuration window 3 <i>C2000 Processors</i> | 117 |
| A.5 | Library Configuration window 4 <i>C2000 Processors</i> | 118 |
| A.6 | Library Configuration window 5 <i>C2000 Processors</i> | 118 |
| A.7 | <i>TI controlSUITE</i> downloading page. | 120 |
| A.8 | <i>TI Code Composer Studio</i> downloading page.. . . . | 120 |
| A.9 | <i>TI C2000Ware</i> downloading page.. . . . | 121 |
| A.10 | <i>TI controlSUITE</i> download location window. | 122 |
| A.11 | <i>TI controlSUITE</i> download location window. | 122 |
| A.12 | <i>TI controlSUITE</i> download location window. | 122 |
| B.1 | Creation of variables for the interface. | 124 |
| B.2 | Simulink mask main window. | 124 |
| B.3 | <i>Parameters and Dialog</i> tab settings and descriptions. | 125 |
| B.4 | <i>Popup Options</i> tab settings and descriptions. | 126 |
| B.5 | <i>Initialization</i> tab settings and descriptions.. . . . | 127 |

Chapter 1

Introduction

This project consists of the development and control of a prototype of a three-phase voltage source DC-AC electronic converter (*Voltage Source Converter* or *VSC*) for teaching. VSCs were traditionally as “inverters”, because the energy ranged from the continuous (DC) part to the alternate part (AC), unlike in rectifiers, however this nomenclature is now somewhat obsolete, since the new technologies of VSCs allow them to be completely reversible devices. This type of device is of special interest in fields such as the injection of power to the electrical network, for example from a solar farm, or for the control of electrical machines.

The tasks that are developed will be:

1. Choice of components.
2. Modeling of the system to be controlled.
3. Simulation of the complete system.
4. Simulation in continuous time with a proportional regulator.
5. Continuous time simulation with a proportional-integral regulator.
6. Discrete-time simulation of the previous regulators with as many details as possible that reflect the implementation to be carried out.
7. Implementation with the above regulators on a TMS320F28739D microcontroller evaluation board.
8. Prototype test and results analysis.

1.1 Project Motivation

The social transition towards a more sustainable development, must necessarily go through a greater solidity in the implementation of renewable energies in the energy

system. This implementation must be done, nowadays, through electronic power converters. Any advance, analysis, design, test or development that contributes to the understanding of these systems will facilitate this necessary transition of the energy model. This project has been written having in mind that, as Carl Sagan says in *The Demon-Haunted World* [7], “the method of science, as stodgy and grumpy as it may seem, is far more important than the findings of science.”.

1.2 Objectives

The objective of this project is to carry out a closed-loop current control of a three-phase inductance to illustrate the connection of a VSC (Voltage Source Converter) to the electrical network in the application of renewable energies. The result obtained in this project is intended to be used for teaching in various subjects related to undergraduate and master’s power electronics at the Universidad Pontificia Comillas.

Chapter 2

Voltage Source Converters

The voltage source converters are power electronic devices, which are based on the switching of power semiconductor devices, which transform a direct input voltage into an alternating output voltage.

Electronic voltage source power converters allow, if the input has a variable voltage, at the output a variable voltage can be obtained through the system gain. However, if there is a constant voltage at the input, a variable voltage can also be obtained at the output if the VSC is controlled by pulse width modulation [8], being this a DC-AC converter such as developed in this project.

The basic scheme of the system is shown in Fig. 2.1.

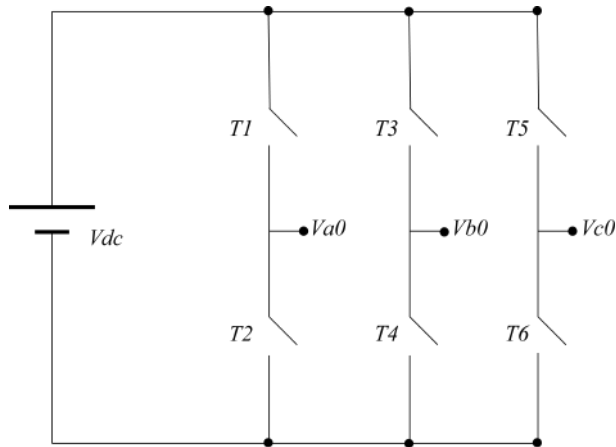


Figure 2.1: Schematic of a three-phase VSC. V_{dc} is the direct input voltage, V_{a0} , V_{b0} and V_{c0} are the alternating output voltages.

2.1 Brief description of the PWM

The VSC control technique by means of pulse width modulation (PWM), allows obtaining a voltage whose fundamental harmonic will be of variable amplitude (alternating), starting from a fixed input voltage (continuous). A PWM output consists of pulses of switching frequency and fixed magnitude, but variable pulse width [9]. However, there are cases in which the switching frequency can be variable, as in the mitigation of audible noise in certain power electronics applications. A PWM output can be generated by two main techniques, comparison of a sinusoidal with a triangular one, as in Fig. 2.2 and by means of PWM in spatial vectors (Space Vector Pulse Width Modulation or *SVPWM*), represented in Fig. 2.3. This last technique, SVPWM, is adapted to the switching of power transistors (see Section 2.2), depending on the load to which the VSC [10] is connected.

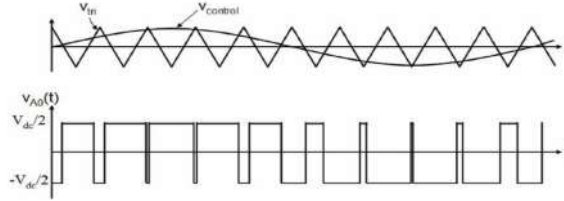


Figure 2.2: PWM generation by comparing with a triangular [11]

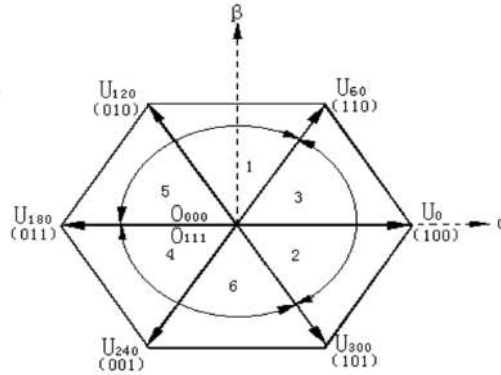


Figure 2.3: Space Vector Voltages [10]

2.2 Power Transistors

Power transistors are switches that can be turned on (shorted) or off (open circuit) using a control signal. Nowadays, given the great evolution of these devices in the industry, switching between the active and inactive states at speeds much higher frequencies is easier than before, which facilitates the incorporation of transistors in electronic converters of power DC - AC, as well as AC - CC [8]. Three of the major groups into which the vast world of power transistors can be divided are described below.

2.2.1 MOSFET Transistors

MOSFET type transistors (*Metal Oxide Semiconductor Field Effect Transistor*) are voltage controlled switches, requiring an input current at the control terminal (G, in Fig. 2.4) only in the first moments of the switch [8]. They allow very high switching frequencies [12], thus facilitating their immersion in electronic power converters. In Fig. 2.4 [13] you can see a diagram of a MOSFET in n-type channel configuration (Fig. 2.4 -a) and p-type channel configuration (Fig. 2.4 -b)

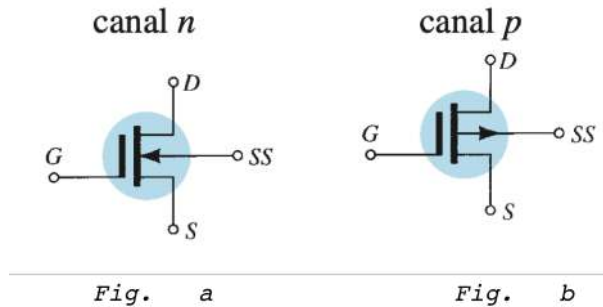


Figure 2.4: Diagram of a MOSFET transistor a) channel n and b) channel p. [13]

In the event that the substrate is not accessible (and this always happens in power electronics), Fig. 2.5 shows its symbology.

2.2.2 BJT Transistors

Bipolar Junction Transistors or BTJs (*Bipolar Junction Transistor*) are devices that are controlled by a current at the gate. These require a current flow through the base (control terminal, B (for Base) in Fig. 2.6), so that, by creating a potential difference between the base and the emitter, the flow of current through the collector is allowed. In these devices, the current gain is highly dependent on temperature

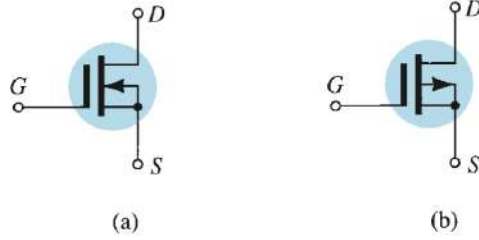


Figure 2.5: a) Diagram of a channel n and b) MOSFET transistor Diagram of a channel MOSFET transistor. [13]

[14]. In Fig. 2.6 [15] a transistor of this type is shown in direct conduction. A BJT transistor needs current in B as long as it has to be active (conducting).

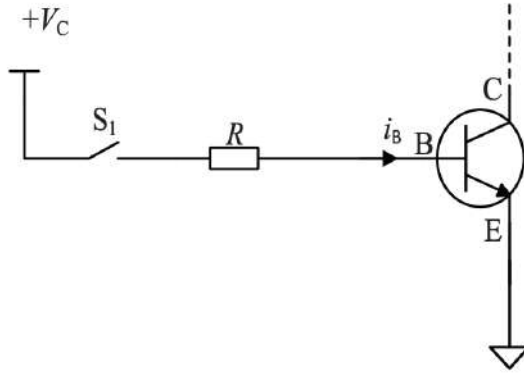


Figure 2.6: Diagram of a BJT transistor in direct conduction [15]

BJTs were popular in DC-AC converters in the 1975-1985 decade, but have been superseded by MOSFETs in low-voltage, low-medium power applications or by IGBTs in medium-high voltage and medium-high power applications, or by other switches in very high voltage and power applications [8].

2.2.3 IGBT Transistors

IGBT power transistors also allow switching between active and inactive states at high frequencies [16], but the maximum switching frequency is noticeably lower in

IGBTs than in MOSFETs. An IGBT transistor has a high input impedance and low conduction losses when active [14]. By behaving in a similar way to a combination between a BJT and a MOSFET, its representation can be made as BJT (Fig. 2.7-a) or as MOSFET (Fig. 2.7-b) .

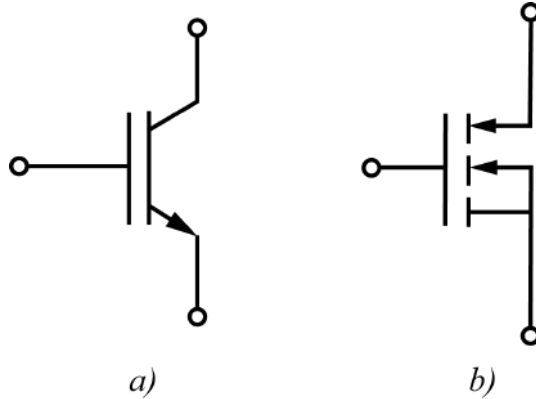


Figure 2.7: IGBT diagram as a) BJT b) MOSFET

2.2.4 Semiconductor materials

Materials with which power transistors are manufactured have undergone great evolutions in recent years thanks to the development and implementation of power electronics in fields such as energy storage, the rise of electric vehicles, the development of so-called microgrids... etc. Typically, the most widely used semiconductor materials are germanium (Ge), silicon (Si), and gallium arsenide (GaAs). There are, within these, intrinsic and extrinsic materials. An intrinsic material is one that has been refined in such a way that it has a low impurity index, while an extrinsic material is one that has been doped with impurities. Within the latter, there are extrinsic n-channel and p-channel materials. An extrinsic n-type material is formed by adding atoms that have five valence electrons (arsenic, phosphor ...), while an extrinsic p-channel material is formed by adding atoms that have three valence electrons (boron, gallium, indium ...). In either type, the activation or deactivation of the semiconductor material occurs due to flows of electrons or holes of these [13].

Semiconductor materials can be controlled to switch between active or inactive mode. The transition between these two levels take place when enough energy is applied to the electrons to be able to overcome the so-called *forbidden band* or *bandgap*, as can be seen in the Fig. 2.8.

Today gallium nitride (GaN) or silicon carbide (SiC) are being used to build switches with advantages over silicon switches: less switching losses, lower channel resistance in MOSFETs, higher blocking voltagesetc. [17].

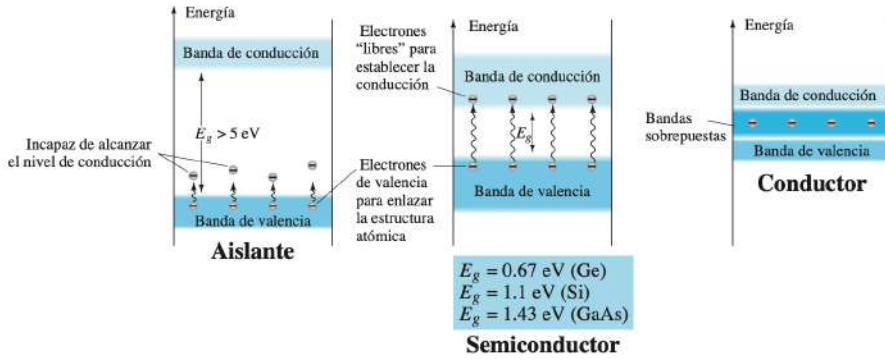


Figure 2.8: Conduction band of semiconductor materials [13].

2.2.5 Dead Time

In order to avoid a short circuit in the voltage source of the electronic converter, it is necessary for the switches of the same branch to change their state from t_{on} to t_{off} and from t_{off} to t_{on} so that they do not match in active mode at the same time. See Fig. 2.9. Semiconductors, as a general rule, can go from an *off* state to an *on* state in a very short period of time. However, the time it takes for semiconductors to go from an *on* state to a *off* state is considerably longer [13], Fig. 2.10.

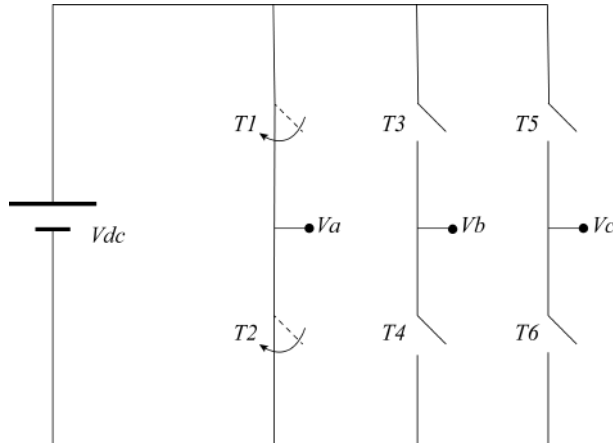


Figure 2.9: Hypothetical short circuit of phase A.

Therefore, since the switches do not turn on and off instantaneously, the turning on of each switch in each branch must be delayed, so that the complementary

semiconductor has time to completely turn off, as can be seen in Fig. 2.11. This prevents two switches on the same branch conducting simultaneously. This effect is called *dead time*

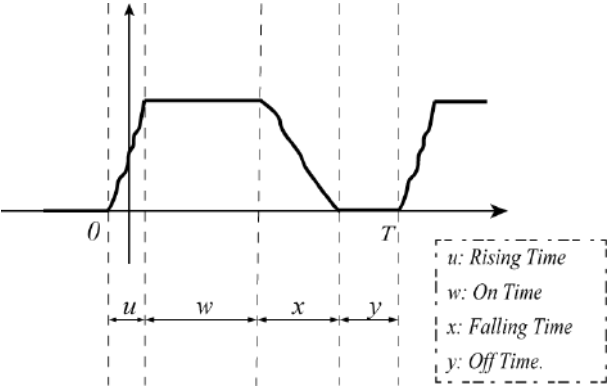


Figure 2.10: On and off time of a semiconductor

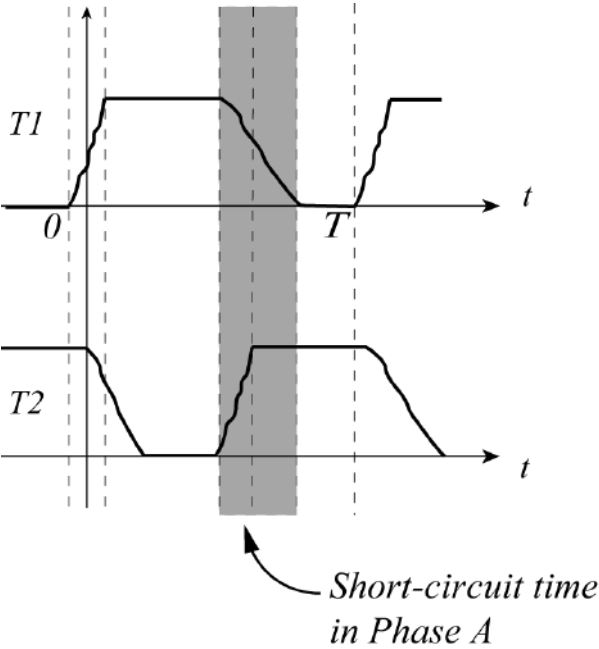


Figure 2.11: Time when a branch is short-circuited.

2.2.6 Analog Method

Analogously, the phase shift necessary to cover the Dead Time described in Section 2.2.5, can be carried out by means of a simple analog circuit as shown in Fig. 2.12.

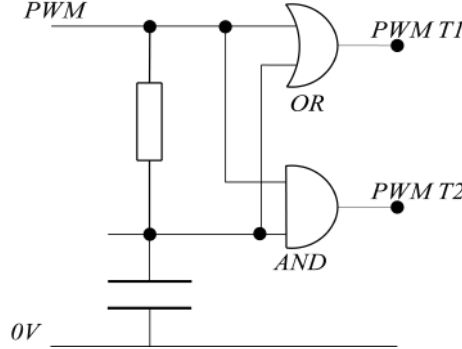


Figure 2.12: Analog circuit made to satisfy the required dead time.

2.2.7 Digital Method

Depending on the architecture and the language in which the calculations for firing the semiconductors are programmed, the methodology may vary. However, it typically consists of delaying the switching on of the complementary semiconductor for a certain phase, so that in no case the power supply is short-circuited.

2.3 VSCs with PWM

MOSFETs, as described in Section 2.2, are controlled by a voltage between gate and source. Once conduction is activated by the MOSFET, it is not necessary to continue supplying current to it.

These voltages will have a square waveform whose service factor (*Duty Cycle*) is defined as the time in which the wave is at high level (t_{on}) divided by the total period waveform (T), as can be seen in (2.1) and in Fig. 2.13. Since the output is to have a sinusoidal waveform, the service factors must vary sinusoidal.

These pulses that turn the semiconductors on and off are made individually for each phase, and specifically, for each semiconductor. Since it is intended to perform a VSC with the AC output in three-phase, a total of 6 different PWM signals will be computed and produced, that is, six different waves. However, there is a certain relationship between the semiconductors acting on a phase. This is because two semiconductors on the same branch cannot be turned on at the same time, to avoid a short circuit in the DC voltage source. More details in Section 2.2.5.

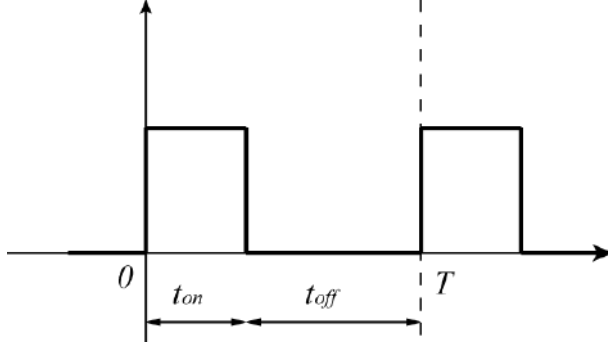


Figure 2.13: Service factor definition

$$D = t_{on}/T \quad (2.1)$$

With:

$$T = t_{on} + t_{off} \quad (2.2)$$

2.4 VSC Description

A VSC DC-AC voltage source consists of a controllable direct voltage source, a series of switches controlled by a microprocessor, an alternating voltage output that arises from the switching of the switches, and a load to which it is connected (In the present project is a three-phase inductance.)

The control system consists of [18]:

1. The initial position of the counter outputs for PWM is set.
2. The ADCs are fired and we wait for them to finish converting.
3. The voltage to be applied to the load is calculated.
4. This voltage value is converted into a service factor for each branch.
5. PWM generation is updated (if applicable).

A model of VSC feeding a load is illustrated in Fig. 2.14. While in Fig. 2.15 the functional diagram of the VSC used in the prototype is shown.

In this prototype, since it consists of a current control, the currents flowing through the load are sampled (i_a , i_b and i_c in Fig. 2.14), used to control the voltage that supplies the load and thus follow the reference imposed in the control.

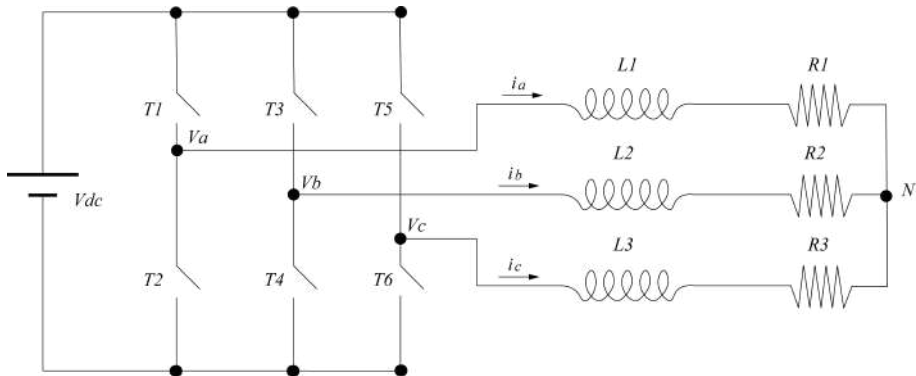


Figure 2.14: Scheme of a VSC feeding a load. V_{dc} is the controllable DC input voltage. T1 to T6 are the switches triggered with the microprocessor. V_{a0} , V_{b0} and V_{c0} are the alternating output voltages and i_a , i_b and i_c the currents through the load. L and R is the load that is fed.

2.4.1 Signals sampling

The ADC of the microcontrollers sample the output signal (from the voltage and current probes) between an instant k and a consecutive instant $k + 1$. It will be strictly necessary for the ADCs to start their sampling at k and it is an indispensable condition that the PWM levels of the signals of the three inverter phases are simultaneously at a high or low level. As can be seen in Fig. 2.16.

Likewise, to achieve the previous condition, different methods can be used. The method to be used in the prototype is to divide the pulse at the high level of the PWM of each cycle into two equal parts, that is, to achieve the same service factor D in a cycle of period T (not necessarily equal to the time between k and $k + 1$), by dividing into two pulses of value $D/2$ for the same period T , as shown in Fig. 2.17. This makes it safer for all three PWMs to be at high level when converting the ADC.

Given the great power of nowadays microcontrollers, the necessary calculations can be made to find the value of the PWM service factor to obtain the control variable as a result, in a short period of time. This period of time for calculations will be computed between k and $k + 1$. For this reason, the need or not to execute the order to update the value of the new service factor before the end of the cycle between k and $k + 1$ must be analyzed, or otherwise, wait for the cycle to be completed and update the value in $k + 1$. Given the extension and objective of this project, the second option is chosen: wait for the cycle to complete.

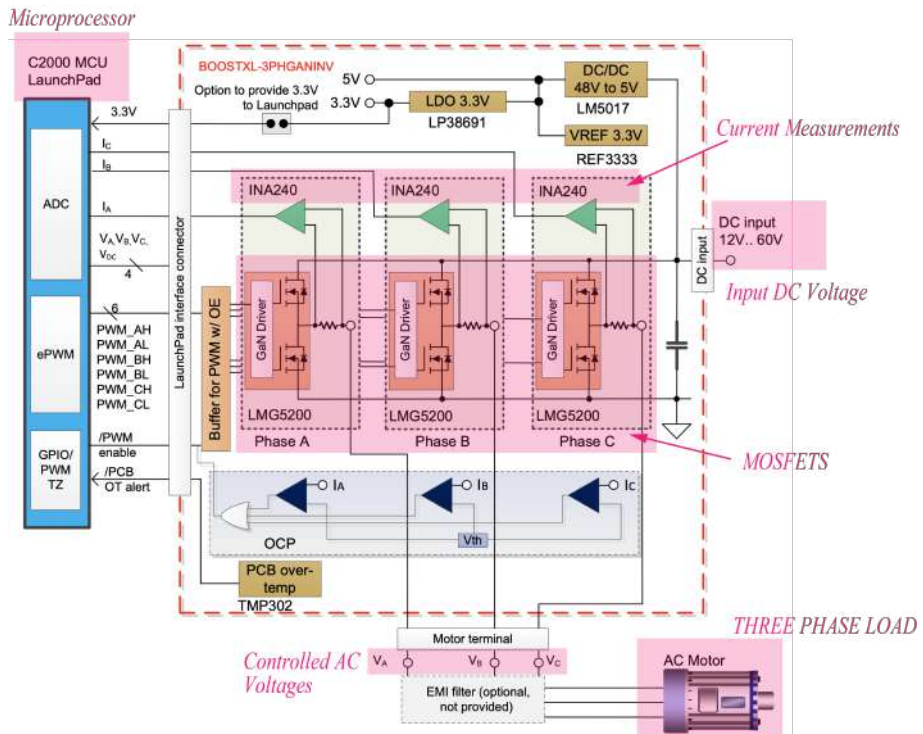


Figure 2.15: Prototype functional diagram.

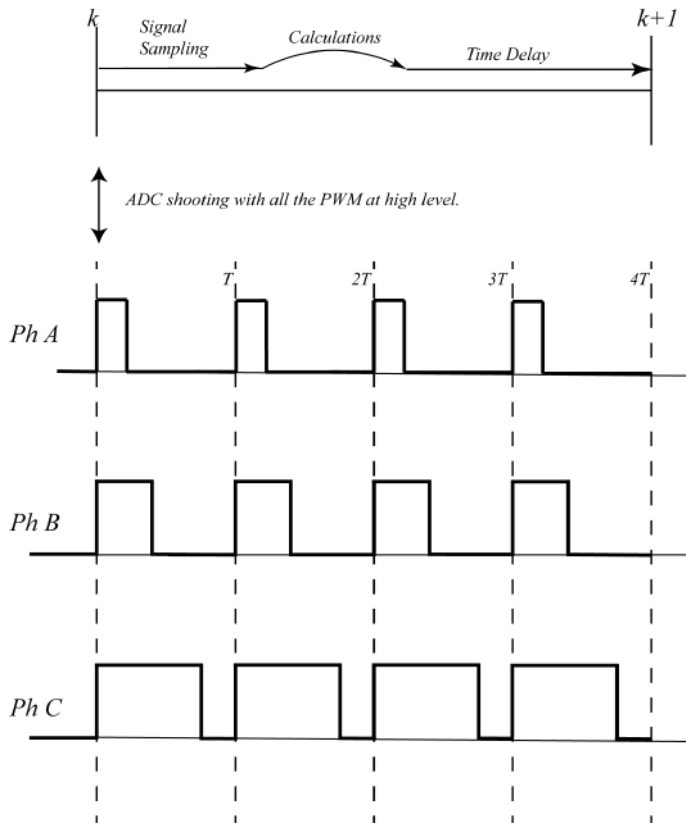


Figure 2.16: Synchronization of the ADC with PWM with a single shot per cycle.

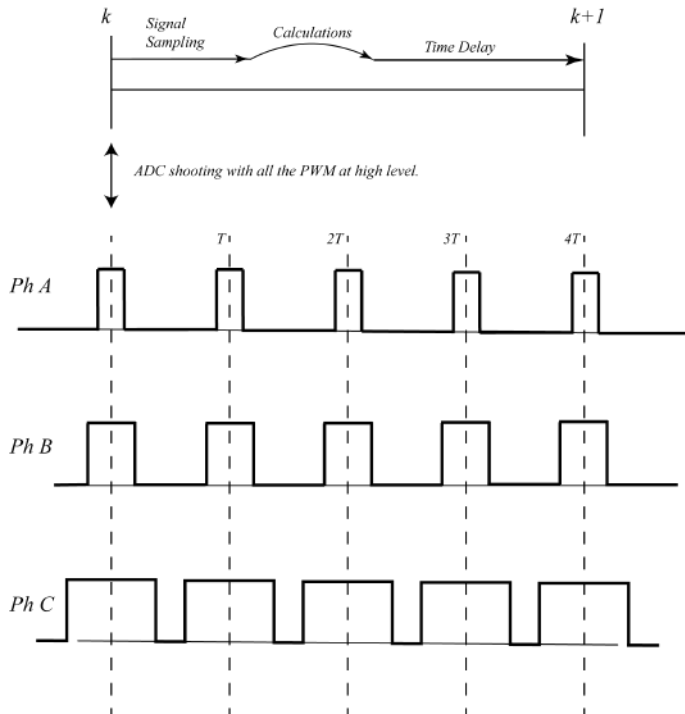


Figure 2.17: ADC synchronization with PWM with double shot per cycle.

Chapter 3

Connection of electronic voltage converters to the grid

The DC - AC electronic converters can be connected to the grid in three ways, in *grid-forming* format, in *grid-supporting* format and in *grid-feeding* [19] format .

VSCs used in *grid-forming* format can resemble an ideal AC voltage source model, with a constant amplitude and frequency, having a very small output impedance, as shown in Fig. 3.1. The amplitude and frequency of the voltage is determined by the local network to which is going to be connected [19] or, failing that, in isolated mode operation [20], it must follow a fixed reference [21].

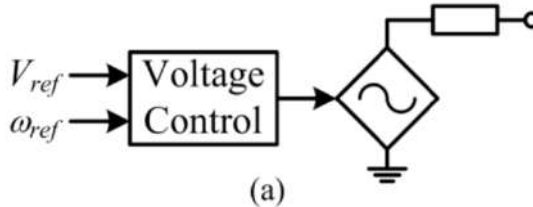


Figure 3.1: VSC connected in *grid-forming* format [22].

The *grid-feeding* format is used to control the power that circulate through the grid, consuming or generating active or reactive power, in a fixed or variable way [19]. It must be perfectly synchronized with the local network to which is desired to connect [21]. It is modelled as a current source connected in parallel with a large impedance, Fig. 3.2.

Finally, the *grid-supporting* format can control the frequency and amplitude of the output voltage, using a model based on an AC voltage source in series with an

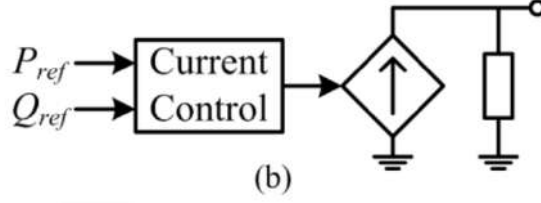


Figure 3.2: VSC connected in *grid-feeding* format [22]

impedance, as shown in Fig. 3.3, or control the power of the network by modeling a current source in parallel with a load, as shown in Fig. 3.4. The *grid-supporting* format consists of acting either on the injected or consumed power using the grid voltage and frequency measurements (*grid-supporting-feeding*), or on the voltage and frequency based on the active and reactive power measurements of the network (*grid-supporting-forming*) [23]. Active power is strongly related to the grid frequency, while reactive power is related to the voltage profile [24].

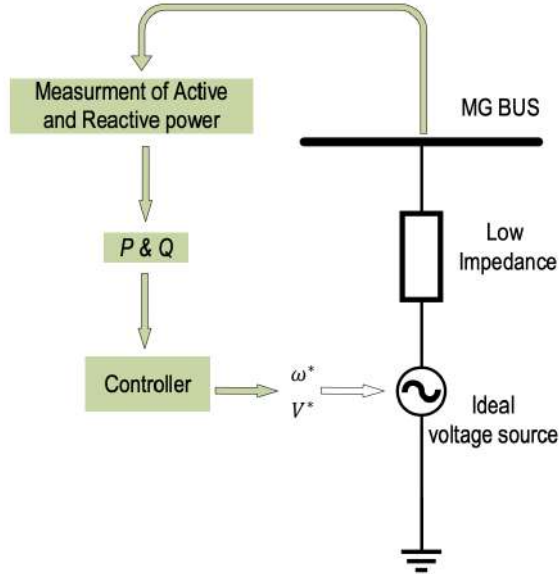


Figure 3.3: Diagram of a converter in *grid-supporting-forming* format with voltage source model [23].

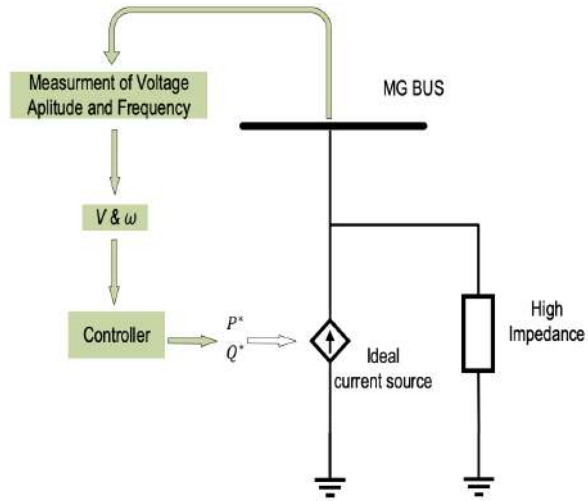


Figure 3.4: Diagram of a converter in *grid-supporting-feeding* format with current source model [23].

3.1 Grid synchronization

The synchronization of the DC - AC electronic converters with the network is essential for their correct operation. This synchronization depends primarily on the VSC operating format. When the VSC is connected to the grid, the operating frequency is determined by the grid frequency, instead, operating in island or isolated mode, the operating frequency will be a reference imposed by the VSC operator. In case of being connected to the network, the synchronization is done through a synchronization block *PLL* [19] (*Phase-Locked Loop*), see Fig. 3.5. This block consists of a sinusoidal reference follower, which reproduces the frequency of the electrical network at the output, in Fig. 3.5, the angle θ . It can be seen in Fig. 3.6 how a *PLL* would be used in a VSC, operating in *grid-forming* format [20], [25].

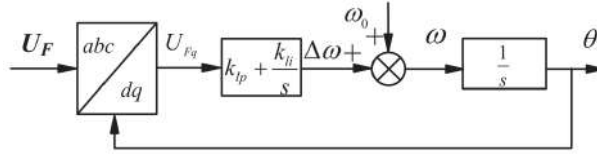


Figure 3.5: *PLL* block diagram [26].

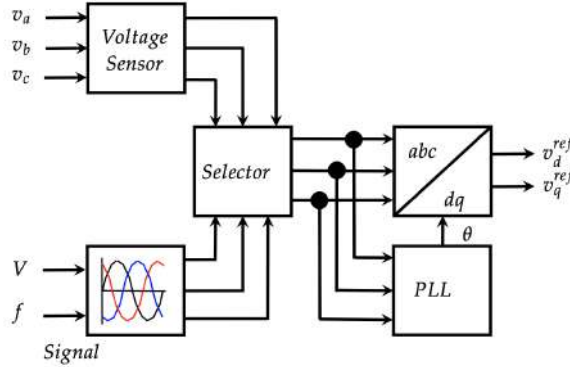


Figure 3.6: *PLL* linearized model [25].

In the case it is desired to synchronize a three-phase alternating network, the block *SRF-PLL* [27] (*Synchronous Reference Frame Phase-Locked Loop*) is used, see Fig. 3.7)

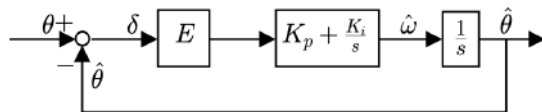


Figure 3.7: MSRF-PLL linearized model [27].

3.2 Some simple filters

Today there is a strong trend in the application of VSC when integrating renewable energy sources to the electrical system, which operates in alternating magnitudes. Such connection to the electrical network requires the implementation of filters to be able to generate triphasic waves that are synchronized with the network and that suppress the harmonics [28]. Filters range from the simplest such as L, LC and LCL filters [20], [29], [30], to filters of much greater complexity. In the present prototype an L filter will be used. Furthermore, it will be merely experimental, without connecting the converter to the electrical network.

However, the use of VSC is also extended to the control of electric motors, which, due to their construction, are strongly inductive loads. This makes it unnecessary to install filters at the VSC outlet in many occasions.

The modeling of the electrical network is carried out assuming that it is equivalent to a three-phase load. The application of connecting the VSC to the network does require the use of filters, the ones listed below being the simplest,

1. Filter L. When supplying a passive load with a filter L like the one in Fig. 3.8 at the inverter output, the current through each of the phases is controlled.
2. Filter LC. When supplying a passive load with an LC filter like the one in Fig. 3.9 at the inverter output, the three-phase voltage that supplies the load is controlled.
3. Filter LCL. When feeding a passive load with an LCL filter like the one in Fig. 3.10 at the inverter output, the current of each of the phases flowing through the coil connected to the load is controlled. [31], [32].

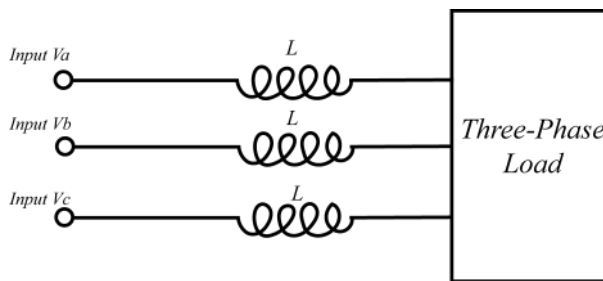


Figure 3.8: Filter L.

The prototypes will be as can be seen in Fig. 3.11. The series resistance displayed in each phase represents the resistance of the coil wire.

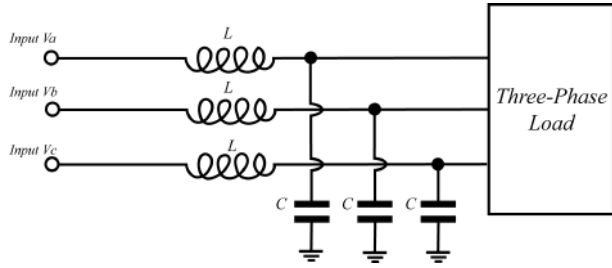


Figure 3.9: Filter LC.

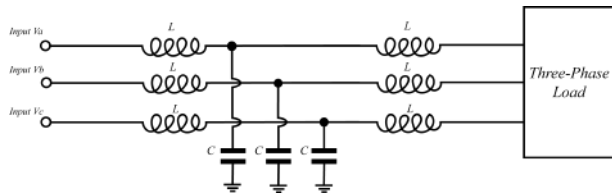


Figure 3.10: Filter LCL.

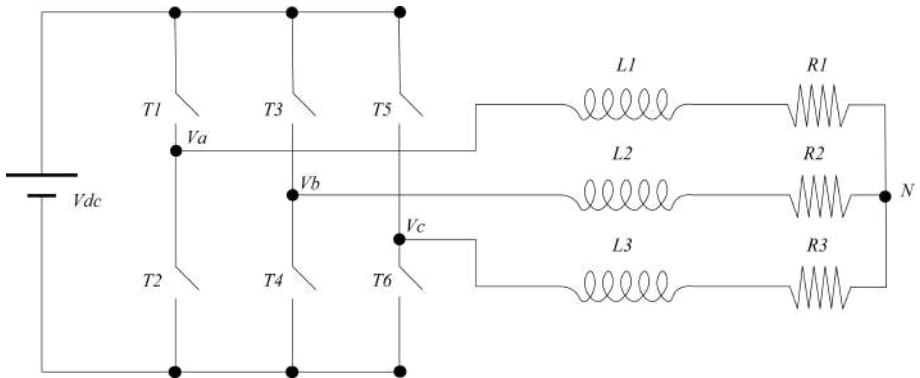


Figure 3.11: Scheme of three-phase inverter with RL load.

Chapter 4

Device Control

4.1 System equations

Let the system of voltages and currents be like the one in Fig. 4.1

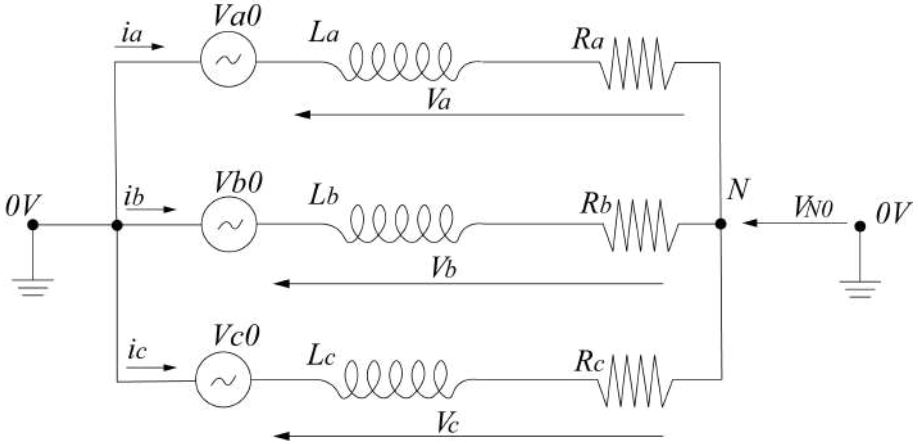


Figure 4.1: Circuit vector scheme. V_{a0} , V_{b0} and V_{c0} are the inverter output voltages. V_a , V_b and V_c the voltages determined by the control and i_a , i_b and i_c the measured currents.

The control to be designed will calculate the voltage to be applied to the load, in Fig. 4.1, the voltages V_a , V_b and V_c . In (4.1) the system equations is shown. In them, V_{a0} , V_{b0} and V_{c0} are the output voltages of the VSC.

$$\begin{cases} V_{a0} = V_a + V_{N0} \rightarrow V_a = V_{a0} - V_{N0} \\ V_{b0} = V_b + V_{N0} \rightarrow V_b = V_{b0} - V_{N0} \\ V_{c0} = V_c + V_{N0} \rightarrow V_c = V_{c0} - V_{N0} \end{cases} \quad (4.1)$$

Adding the previous equations, it results:

$$(V_a + V_b + V_c) = (V_{a0} + V_{b0} + V_{c0}) - 3 \cdot V_{N0} \quad (4.2)$$

It will be assumed that the values of the inductances and resistance of the cable of the three coils will be equal, in such a way that the system is balanced,

$$\begin{cases} L_a = L_b = L_c = L \\ R_a = R_b = R_c = R \end{cases} \quad (4.3)$$

Likewise, the voltage that must fall in each of the coils and their resistance, being the impedance the sum of the inductance and the resistance of the cable:

$$\begin{cases} V_a = L \cdot \frac{di_a}{dt} + R \cdot i_a \\ V_b = L \cdot \frac{di_b}{dt} + R \cdot i_b \\ V_c = L \cdot \frac{di_c}{dt} + R \cdot i_c \end{cases} \quad (4.4)$$

Therefore, assuming a balanced three-phase system, Kirchhoff's law applies that the sum of currents in a node must equal zero [33]:

$$V_a + V_b + V_c = 0 \quad (4.5)$$

Y operando(4.2) en (4.5), resulta:

$$V_{N0} = \frac{1}{3} \cdot (V_{a0} + V_{b0} + V_{c0}) \quad (4.6)$$

The waveforms of DC voltage and the fundamental harmonic of V_{a0} , V_{b0} and V_{c0} will be ¹ In the form (4.7), with m_a the amplitude modulation index, which will vary depending on the voltage applied to the load,

$$\begin{cases} V_{a0} = m_a \frac{V_{dc}}{2} + m_a \cdot \frac{V_{dc}}{2} \cdot \sin(\omega t) \\ V_{b0} = m_a \frac{V_{dc}}{2} + m_a \cdot \frac{V_{dc}}{2} \cdot \sin(\omega t + 120) \\ V_{c0} = m_a \frac{V_{dc}}{2} + m_a \cdot \frac{V_{dc}}{2} \cdot \sin(\omega t - 120) \end{cases} \quad (4.7)$$

¹Obtained from the notes of the Power Electronics Applications course. Last access: 12/18/2018. Accessible to students enrolled in the subject at the Universidad Pontificia de Comillas.

And substituting and operating (4.7) in (4.6):

$$V_{N0} = \frac{1}{3} \cdot \left(3 \cdot \frac{m_a \cdot V_{dc}}{2} \right) = \frac{m_a \cdot V_{dc}}{2} \quad (4.8)$$

4.2 Three-phase variables and reference systems

The variables that want to be controlled, in any case, are triphasic and will be of the form:

$$v_i = \hat{V}_i \sin(\omega t + \varphi_i) \quad (4.9)$$

Being:

1. v_i the voltage of phase i at a given instant of time.
2. \hat{V}_i phase i peak voltage.
3. ω the pulsation of the output sine wave .
4. φ_i phase i initial phase lag or phase lead.

Controlling variables in the previous way is tedious because it contains sinusoidal functions. In addition, being triphasic magnitudes forces to control each branch separately. The solution to such an arduous task is to take advantage of mathematical models such as the Clarke transform and the Park transform.

4.2.1 Clarke Transformation

The Clarke transform consists of the projection of three coplanar axes, not necessarily orthogonal to each other, of components a , b and c , onto a Cartesian system of orthonormal axes, of components α , β and 0 [34], with 0 generally being the zero sequence component of the transformation. In case of having a three-phase system in which the vectors are out of phase with each other 120° , we have the system represented in Fig. 4.2. If the system does not have a fourth wire that connects the neutral of the load to ground, the homopolar component of the current (0 component), is zero, which happens in almost all cases and in the prototype it will be. From now on, the zero sequence component will be ignored.

$$U_{\alpha,\beta,0} = CU_{a,b,c} \quad (4.10)$$

The base change matrix that results from Fig. 4.2, again ignoring the zero sequence component, is

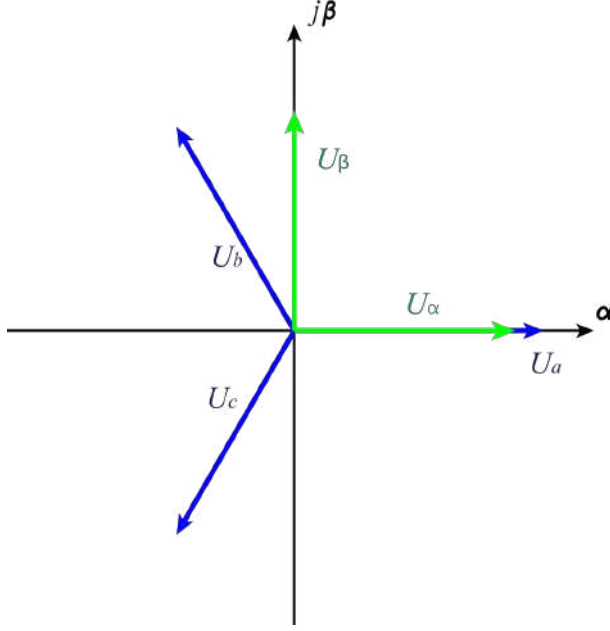


Figure 4.2: Clarke transformation.

$$C_{a,b,c \rightarrow \alpha, \beta} = \begin{pmatrix} \cos(0) & \cos(2\pi/3) & \cos(-2\pi/3) \\ \sin(0) & \sin(2\pi/3) & \sin(-2\pi/3) \end{pmatrix} \quad (4.11)$$

$$C_{a,b,c \rightarrow \alpha, \beta} = \begin{pmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{pmatrix} \quad (4.12)$$

and vice versa,

$$C_{a,b,c \rightarrow \alpha, \beta}^t = C_{\alpha, \beta \rightarrow a, b, c} = \begin{pmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{pmatrix} \quad (4.13)$$

4.2.2 Park Transformation

The Park transform [35] consists of applying to the Clarke transformation, to the orthonormal axes α, β a second transformation in the form of rotation around the center of the coordinate axes, taking them to new axes d, q [34]. This rotation can

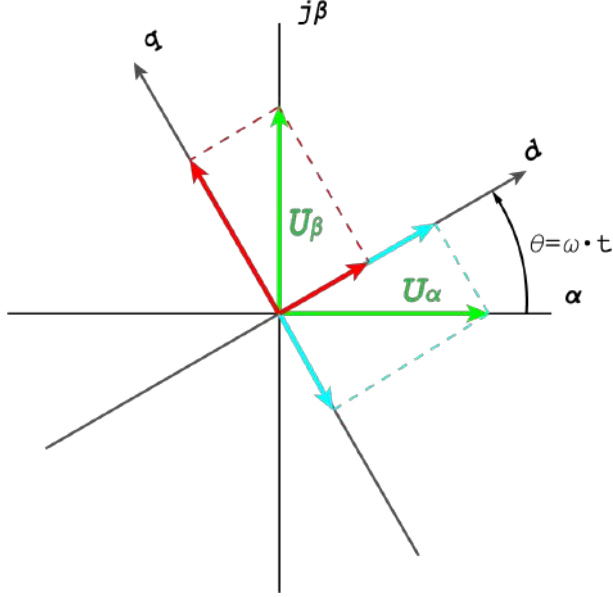


Figure 4.3: Transformada de Park.

be fixed, an angle θ , or, if the rotation is constant, the axes can rotate at angular speed ω . The graphical representation can be seen in Fig. 4.3.

The Park transformation that translates the orthonormal axes α, β to axes dq can be expressed in the forms

1. If the axes rotate an angle θ :

$$\begin{pmatrix} u_d \\ u_q \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} u_\alpha \\ u_\beta \end{pmatrix} \quad (4.14)$$

2. If the axes rotate at constant rate ω :

$$\begin{pmatrix} u_d \\ u_q \end{pmatrix} = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} \begin{pmatrix} u_\alpha \\ u_\beta \end{pmatrix} \quad (4.15)$$

And the step matrix that relates the initial coplanar axes A, B and C with the new axes d, q is:

$$P_{a,b,c \rightarrow d,q} = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} C_{a,b,c \rightarrow \alpha,\beta} \quad (4.16)$$

And substituting (4.12) into (4.16)

$$P_{a,b,c \rightarrow d,q} = \begin{pmatrix} \left(\frac{2}{3}\right) \cos(\omega t) & \left(\frac{2}{3}\right) \cos(\omega t - 2\pi/3) & \left(\frac{2}{3}\right) \cos(\omega t + 2\pi/3) \\ -\left(\frac{2}{3}\right) \sin(\omega t) & -\left(\frac{2}{3}\right) \sin(\omega t - 2\pi/3) & -\left(\frac{2}{3}\right) \sin(\omega t + 2\pi/3) \end{pmatrix} \quad (4.17)$$

Since the coplanar vectors in A , B , C may not have the same magnitude as in the dq axes, a homothety is applied to (4.17), also including the component homopolar:

$$P_{a,b,c \rightarrow d,q,0} = K \begin{pmatrix} \left(\frac{2}{3}\right) \cos(\omega t) & \left(\frac{2}{3}\right) \cos(\omega t - 2\pi/3) & \left(\frac{2}{3}\right) \cos(\omega t + 2\pi/3) \\ -\left(\frac{2}{3}\right) \sin(\omega t) & -\left(\frac{2}{3}\right) \sin(\omega t - 2\pi/3) & -\left(\frac{2}{3}\right) \sin(\omega t + 2\pi/3) \\ a & a & a \end{pmatrix} \quad (4.18)$$

With:

$$P_{d,q \rightarrow a,b,c} = P_{a,b,c \rightarrow d,q,0}^{-1} = \frac{(Adj(P_{a,b,c \rightarrow d,q}))^t}{|P_{a,b,c \rightarrow d,q}|} \quad (4.19)$$

More representative, transforming a vector (x_a, x_b, x_c) into one (x_d, x_q, x_0) would be of the form:

$$\begin{pmatrix} x_d \\ x_q \\ x_0 \end{pmatrix} = K \begin{pmatrix} \left(\frac{2}{3}\right) \cos(\omega t) & \left(\frac{2}{3}\right) \cos(\omega t - 2\pi/3) & \left(\frac{2}{3}\right) \cos(\omega t + 2\pi/3) \\ -\left(\frac{2}{3}\right) \sin(\omega t) & -\left(\frac{2}{3}\right) \sin(\omega t - 2\pi/3) & -\left(\frac{2}{3}\right) \sin(\omega t + 2\pi/3) \\ a & a & a \end{pmatrix} \begin{pmatrix} x_a \\ x_b \\ x_c \end{pmatrix} \quad (4.20)$$

In this way, different cases can be distinguished depending on the value of K in (4.18).

- $K = 1$ and $a = 1/3$: It is called invariant in module. The module of the complex number $u_d + ju_q$ is equal to the amplitude of the triphasic variables.
- $K = 1/\sqrt{2}$ y $a = 1/3$: It is called invariant in effective value. The modulus of the complex number $u_d + ju_q$ is equal to the effective value (RMS) of the three-phase variables.
- $K = \sqrt{3}/2$ y $a = \sqrt{2}/3$: It is called power invariant. Prototype operating mode. In this case, with respect to the instantaneous power, $p(t) = v_a(t) \cdot i_a(t) + v_b(t) \cdot i_b(t) + v_c(t) \cdot i_c(t) = v_d(t) \cdot i_d(t) + v_q(t) \cdot i_q(t) + v_0(t) \cdot i_0(t)$

In this way, having the previous matrices, three-phase models can be written in a simple way in only two Cartesian coordinates (omitting the homopolar sequence). Thus, a vector that rotates at a constant speed around a fixed point in a static coordinate system can be converted into a fixed vector in another reference system that rotates at said constant speed around the same fixed point.

The prototype that is designed and manufactured in this project is the one that consists of a VSC connected to a three-phase coil. The variable to control will be

the current through the phases, which, taken to axes $d - q$, consists of the control of the variables i_d and i_q , thus making a current control design [36].

With the power-invariant Park transform, the instantaneous active and reactive powers are [37]:

$$p = v_d \cdot i_d + v_q \cdot i_q \quad (4.21)$$

$$q = v_d \cdot i_q - v_q \cdot i_d \quad (4.22)$$

4.3 Model in d-q axes of a three-phase inductance

Due to the transformations of three-phase, alternating time-varying quantities into constant quantities, it is necessary to analyze the behavior of the components in the new axes after the transformation [30], [31], [38]. In particular, the coils on the new axes d and q will have a peculiar model, in which the currents and voltages of the new axes are related to each other, as seen in the block diagram in Fig. 4.4, which is also called “coupling”.

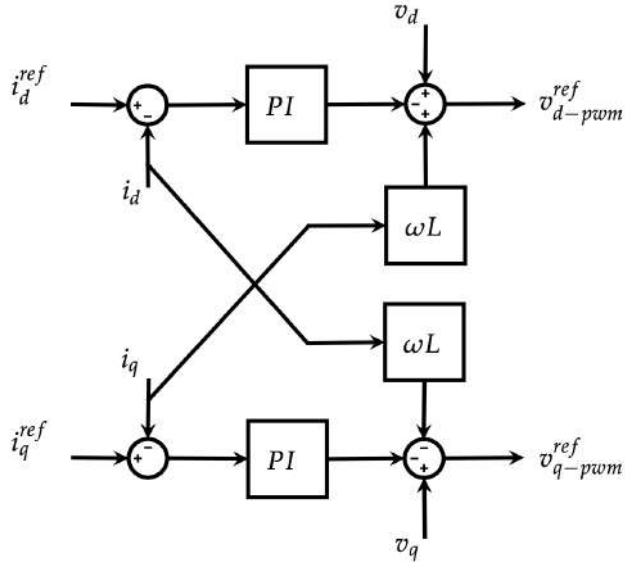


Figure 4.4: Current control diagram in d-q axes [25]

Let (4.23) be the equation that governs the behavior of the coil voltages on the abc axes. v_a, v_b and v_c being the voltages to which the coils are subjected in each

phase.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + L \cdot \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (4.23)$$

Applying the transformation matrix (4.19), results (4.24) (for simplicity $P_{d,q \rightarrow a,b,c} = P$)

$$P \cdot \begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = R \cdot P \cdot \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} + L \cdot P \cdot \frac{d}{dt} \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} \quad (4.24)$$

ac clearing voltage matrices in (4.24):

$$\underbrace{P^{-1} \cdot P}_{\mathbf{I}_{33}} \cdot \begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = P^{-1} \cdot R \cdot P \cdot \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} + P^{-1} \cdot L \cdot P \cdot \frac{d}{dt} \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} \quad (4.25)$$

Operating the matrix operations and solving the voltages in (4.25):

$$\begin{cases} v_d = R \cdot i_d + \frac{d(L \cdot i_d)}{dt} - \omega_1 \cdot L \cdot i_q \\ v_q = R \cdot i_q + \frac{d(L \cdot i_q)}{dt} + \omega_1 \cdot L \cdot i_d \end{cases} \quad (4.26)$$

The equations of (4.26) can be written in values per unit, defining:

$$\begin{cases} v_{pu} = \frac{v}{U_b}, \quad i_{pu} = \frac{i}{I_b}, \quad S_b = 3U_b I_b \\ L_{pu} = \frac{L}{L_b}, \quad R_{pu} = \frac{R}{Z_b}, \quad Z_b = 3 \frac{U_b^2}{S_b} \\ Z_b = L_b \omega_b, \quad \omega_{1,pu} = \frac{\omega_1}{\omega_b} \end{cases} \quad (4.27)$$

So the voltage equations of the inductance and series resistance are obtained in the following values per unit:

$$\begin{cases} \omega_b \cdot v_{d,pu} = \omega_b \cdot (R_{pu} \cdot i_{d,pu}) + \frac{d(L_{pu} \cdot i_{d,pu})}{dt} - \omega_b (\omega_1 \cdot L_{pu} \cdot i_{q,pu}) \\ \omega_b \cdot v_{q,pu} = \omega_b \cdot (R_{pu} \cdot i_{q,pu}) + \frac{d(L_{pu} \cdot i_{q,pu})}{dt} + \omega_b (\omega_1 \cdot L_{pu} \cdot i_{d,pu}) \end{cases} \quad (4.28)$$

Observing the behavior of (4.26), it is verified that in the transient regime, the magnitudes of axes d and q are influenced by each other. However, in steady state the d axis components only depend on the q axis components and the q axis components only depend on the d axis components [39].

$$v_d = -\omega_1 \cdot L \cdot i_q \quad (4.29)$$

$$v_q = \omega_1 \cdot L \cdot i_d \quad (4.30)$$

4.4 Base Magnitudes

It is necessary to determine a series of base electrical quantities, since later on, a PID regulator will be designed both in continuous time and in discrete time. Using the technical documentation of the [40] components, we know that the nominal voltage of the inverter, at its direct voltage input, is $V_{dc} = 48V$. The maximum current of the VSC is $I_{max}^{inverter} = 10A$. The inductances admit a maximum current of $I_{max}^L = 5A$. The inductance of the coils (per phase) are $L = 5.881mH$. The resistance of the inductance (per phase) are $R_L = 1.1 \Omega$.

Likewise, since it consists of a prototype for teaching, it will work with a maximum intensity per phase of 2.5A. Also, making use of (4.31), substituting in this, for safety, a maximum amplitude modulation index of $m_a = 1$, we will have a base voltage magnitude for the phase voltage, that is, voltage between two phases,

$$V_{p-p} \approx 0.61 \cdot m_a \cdot V_{dc} \approx 0.61 \cdot 1 \cdot 48 = 29.28V \quad (4.31)$$

And the line voltage per inductance to the neutral point,

$$V_{p-n} = \frac{V_{p-p}}{\sqrt{3}}$$

The base magnitudes are,

$$\begin{aligned} V_{dc} &= 48V \\ V_b &= V_{p-p} = 29.28V \\ I_b &= 2.5A \\ S_b &= \sqrt{3} \cdot V_b \cdot I_b \\ Z_b &= \frac{V_b}{I_b} \\ \omega_b &= 2 \cdot \pi \cdot f = 2 \cdot \pi \cdot 50 \\ L_b &= \frac{Z_b}{\omega_b} \end{aligned}$$

Chapter 5

Design, simulation and implementation of P - PI controllers

Controls on variables have been used for centuries to control all kinds of variables. The first controls were carried out in an analog way, using mechanisms [41]. Today, the vast majority of controls are implemented digitally. The implementation of a control digitally requires a computer (or microcomputer) to perform all the pertinent operations. (See Section 6.1). There are many types of variable controls, from proportional, integral, and differential (PID) controls, or state-space controls, using state feedback to more complex control topologies.

In this project, a PI control will be carried out on the current of the inductances, however, it is possible to carry out complete PID controls [36], even though these systems can be very noisy due to the commutation of the semiconductors (harmful using a differential control). The design of a PI control in a rotational axis system $d - q$ provides a null error in permanent regime in reference tracking [31] under constant disturbances.

The transfer function between reference (R) and output (Y) of every system is defined as in (5.1) in continuous time using the Laplace variable (s).

$$F(s) = \frac{R(s)}{Y(s)} \quad (5.1)$$

Let be the block diagram of a continuous time system represented in Fig. 5.1, in which, the designed control is $C(s)$, the plant of the system is $P(s)$ and the feedback is $H(s)$. Likewise:

- $r(t)$: is the reference signal of the control.

- $e(t)$: is the error between the measurement of the output ($y_m(t)$) and the output ($y(t)$).
- $u(t)$: is the command that is applied to the plant.
- $y(t)$: is the system output.
- $y_m(t)$: is the measure of the output.

The transfer function between reference and open-loop output is called $G(s)$, see (5.2). While the transfer function between reference and output in closed loop is called $F(s)$, see (5.3).

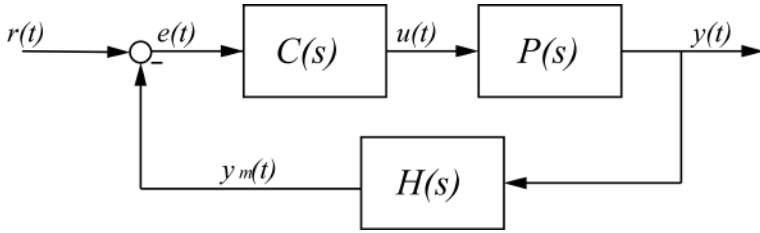


Figure 5.1: General control scheme block diagram.

$$G(s) = C(s) \cdot P(s) \cdot H(s) \quad (5.2)$$

$$F(s) = C(s) \cdot \frac{P(s)}{1 + G(s)} \quad (5.3)$$

The design of PID controls is done by temporal response or frequency response. The response of both controls depending on the values of the parameters of the designed control. Among the main questions to take into account will be, regarding its temporal response for a PI control, analyzing its response to a step in reference in closed loop:

- Tracking error.
- Overshoot.

And regarding its frequency response, analyzing the open-loop Black diagram:

- Gain margin.
- Oscillating pulsation.

The frequency response design is related to its temporal response by:

$$\text{Damping} \Longleftrightarrow \text{Gain margin}$$

$$\text{Speed} \Longleftrightarrow \text{Crossover pulsation}$$

5.1 Design of a P - PI control in continuous time

First, a continuous time control will be designed in the form of Fig. 5.2.

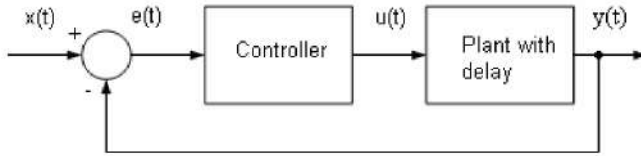


Figure 5.2: Block diagram of a continuous time control[42].

The delay on the plant is carried out by multiplying it by $e^{-\frac{T_s}{2} \cdot s}$ [42], thus emulating the delay of the sampling in continuous time.

The design of a continuous time control requires a variable that varies with time and represents it. The Laplace variable (s) will be used for the continuous time design of the control.

Its design can be done in both serial and parallel format. The parameters in parallel format are:

- K : Parameter corresponding to the proportional part of the control.
- T_i : Parameter corresponding to the integral part of the control.
- b : Weighting parameter of the reference corresponding to the integral part of the control.

The block diagram of the PI control in parallel format can be seen in Fig. 5.3.

The transfer function, between reference and output, of a PI control in parallel format and continuous time will have, as the transfer function of the control in (5.4) in the control of the prototype:

$$C_{parallel}(s) = K \cdot \left(b + \frac{1}{T_i \cdot s} \right) \quad (5.4)$$

The parameters in serial format are:

- K_p : Parameter corresponding to the proportional part of the control.

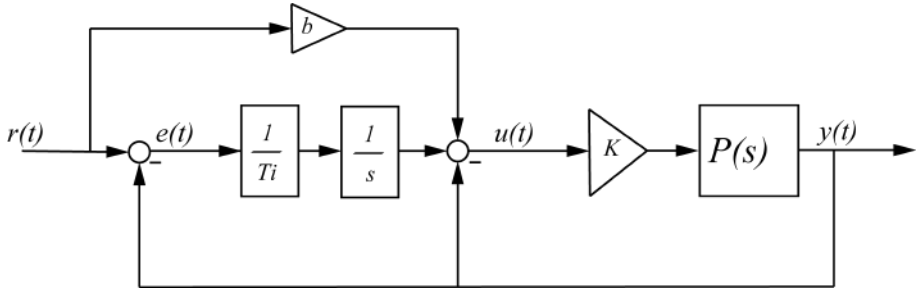


Figure 5.3: Block diagram of a PI control in parallel format.

- I : Parameter corresponding to the integral part of the control.

The control transfer function will have the form of (5.5);

$$C_{serial}(s) = K_p \cdot \frac{1 + I \cdot s}{I \cdot s} \quad (5.5)$$

The equation that relates both formats is:

$$\mu = 1$$

$$K = \mu \cdot K_p$$

$$T_i = \mu \cdot I$$

Obtaining the transfer functions in open loop and closed loop between reference and output will be carried out by executing a simulink file, represented in Fig. 5.4, in which a simulink model will be implemented a coupling and decoupling of the inductances in dq axes. In this way, when executing the Matlab script that compiles the simulink file, the transfer functions in said axes will be obtained, as well as the plant in continuous time. However, it is necessary to determine some base quantities (see Section 4.4) for the components, such as inductances, resistors, voltages, currents, etc.

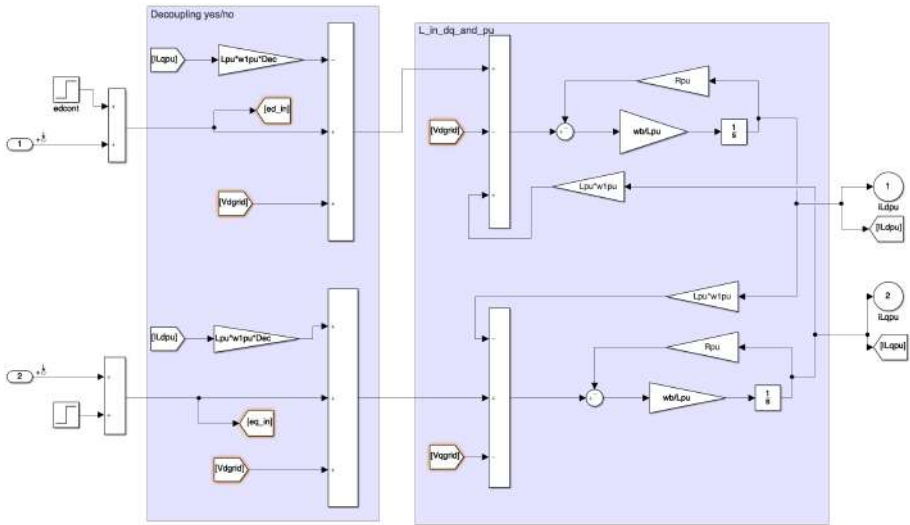


Figure 5.4: Simulink to obtain the plant in continuous time.

5.1.1 Design of a PI control in discrete time

The design of a PI control in discrete time can be done in different ways:

- Through the discretization of a control in continuous time.
- Through integral design in discrete time.

5.1.2 Discretization of a continuous time control

The discretization of a system in continuous time to discrete time requires a variable in discrete time and a specified sampling period. Using the variable in discrete time z and a sampling period T_s , an equation can be found that relates the poles and zeros of a system in continuous time with the corresponding ones in discrete time, see (5.6).

$$z = e^{s \cdot T_s} \quad (5.6)$$

Likewise, it is possible to discretize the control using other methods, such as:

- Approximation of the derivative or the integral: Using a transformation from continuous time to discrete time using the variables s and z . Different methods can be used, such the backward Euler, forward Euler and trapezoidal approximations.
- Invariant transformation: Assuming that there is a hold at the input, for example, of order 0, so that the discretization maintains a constant value during the sampling period.

The block diagram of a discrete time control can be seen in Fig. 5.5. It is necessary to implement two blocks that allow the passage of continuous time to discrete and vice versa. These blocks are:

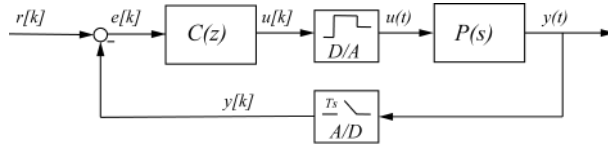


Figure 5.5: Block diagram of a discrete time control.

- D / A: It is the retainer block. It consists of maintaining a certain value during the sampling time, so that the output “sees ” a continuous signal.
- A / D: It is the sampler block, it allows converting the continuous time values of the output to discrete values, each sampling period.

The discretization method used will be that of the derivative in advance of the integral (forward Euler), whose definition is shown in (5.7) ¹, where k represents the current moment and $k - 1$ the moment a sampling period (T_s) behind.

$$y[k] = \int_0^{k \cdot T_s} u(\tau) d\tau \simeq \int_0^{(k-1) \cdot T_s} u(\tau) d\tau + u \cdot [k - 1] \cdot T_s \quad (5.7)$$

Being,

$$\int_0^{(k-1) \cdot T_s} u(\tau) d\tau = y \cdot [k - 1] \quad (5.8)$$

And applying the Z transform results in the equality of the equation,

$$\frac{1}{s} = \frac{z^{-1} \cdot T_s}{1 - z^{-1}} = \frac{T_s}{z - 1} \quad (5.9)$$

In this way, the applied control, using (5.9), we discretize the control in z .

Determining the sampling period is not trivial, since, as explained in Section 2.4.1, the sampling period needs to be greater than the period of a PWM cycle. Therefore, if we want the inverter to commute at a frequency f_c , that is, for the commutation period to be $T_c = \frac{1}{f_c}$, the sampling period must be:

$$T_s = k \cdot T_c \quad (5.10)$$

To avoid unwanted effects in the sampling, working below the nyquist frequency, thus avoiding aliasing, we will calculate the sampling period of (5.10) with $k = 3$, so that the ADC starts the conversion of values every three periods of PWM switching, therefore we will have that the sampling period will be,

$$T_s = k \cdot T_c = 3 \cdot \frac{1}{f_c} = 3 \cdot \frac{1}{10kHz} = 0.3ms \quad (5.11)$$

Considering therefore a sampling period of $T_s = 0.3ms$ and a crossover pulse of $\omega_o = 500rad/s$ (used in the simulations, this value may be different),

$$\omega_o \cdot T_s = 500rad/s \cdot 0.0003s = 0.15rad$$

Which is considered a median sampling period, being between $[0,1]$, following a classification criterion of the sampling period ¹. Being a medium sampling period, the delay produced by the integral action must be between 5° and 30° . The phase delay of the integral control is $\phi_{pi} = 15.3428^\circ$ for the previously set values, thus verifying the validity of the control. By discretizing the integral action with a median sampling period, the discretization method used is correct.

¹Obtained from the notes of the Digital Control course. Last access: 06/20/2019. Material accessible to students enrolled in the subject of Digital Control at the Universidad Pontificia de Comillas.

The application of (5.9) to the transfer function in continuous time of the PI control developed in Section 5.1, that is to say, discretizing the control of (5.5) on page 35, results:

$$C(z) = Kp \cdot \frac{I + \frac{z-1}{T_s}}{I \cdot \frac{z-1}{T_s}} = Kp \cdot \frac{I + \frac{T_s}{z-1}}{I} \quad (5.12)$$

The Simulink block that allows the software and hardware implementation of the PID control is shown in Fig. 5.6. Such figure shows how the value of the integral has to be entered in such a way that it responds to the behavior of the equations described above.

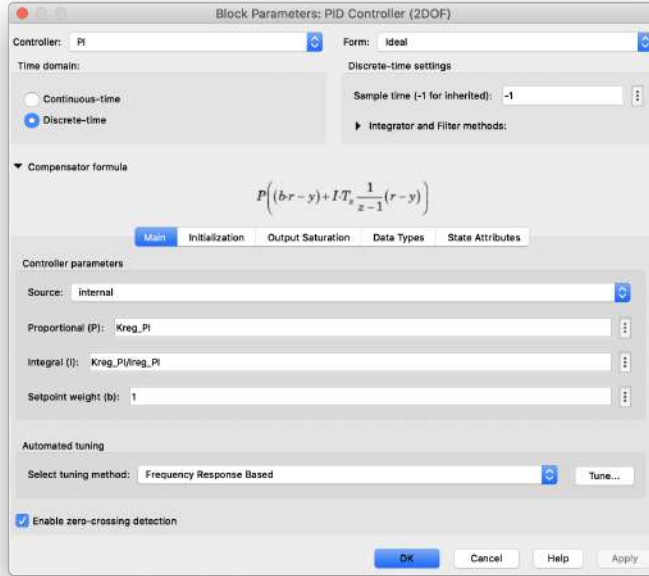


Figure 5.6: PID block for Software and Hardware implementation.

5.2 Description of the simulator

The simulator designed in Simulink allows you to configure:

- Outputs of the scopes. Both in the format of theoretical quantities and those that would be obtained by the oscilloscope when executing the program on the LaunchPad.

- References in d and q axes.
- Open loop, closed loop and coupling
- Type of control, P or PI.

In this simulator, the base quantities described in Section 4.4 have been introduced. The way to enter the options described above is done through an interface (identical to the one used for the hardware in the prototype) such as those shown in Fig. 5.7.

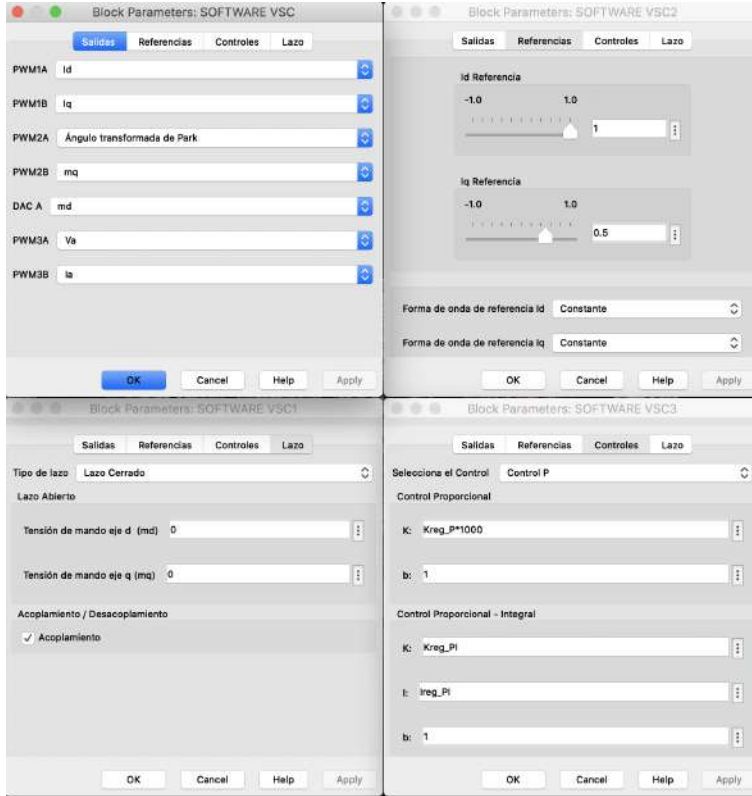


Figure 5.7: Interface for simulator configuration. Identical to that used in the prototype hardware. The development is described in Appendix B

The configuration of the Simulink parameters will be done as shown in Fig. 5.8.

It should be noted that for the simulation to run correctly, the value of the *Fixed Step* must necessarily be a value smaller than the shortest period the system has and must be a sub-multiple divisor of its own. Since the smallest period is the switching period, of $1e-4s$, the *Fixed Step* time that simulates the system correctly, with a good quality in the outputs, with a reasonable simulation time, of $1e-6s$.

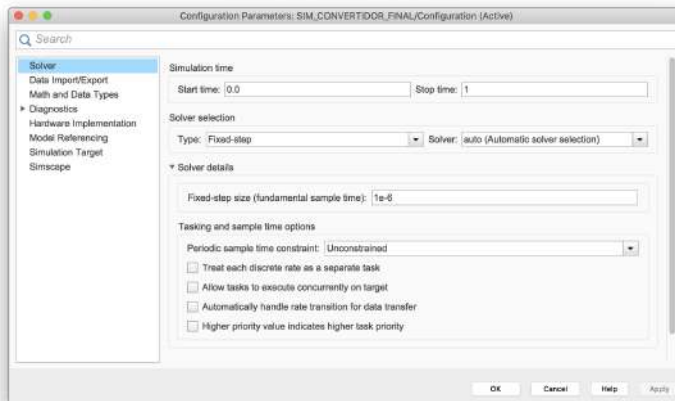


Figure 5.8: Simulator parameter settings.

5.3 P Control

5.3.1 Design of a control P. Continuous time.

The design equations of a continuous time P control by frequency response are described below.

$$\omega_0 = -180 + \phi_m - \angle P \quad (5.13)$$

$$K = \frac{1}{|P(\omega_0)|} \quad (5.14)$$

$$(5.15)$$

Since the proportional control is a gain K:

$$C(s) = C(z) = K \quad (5.16)$$

Testing different values for the phase margin of (5.14), we obtain the different temporal and stability responses in open loop of Fig. 5.9, 5.10 and 5.11.

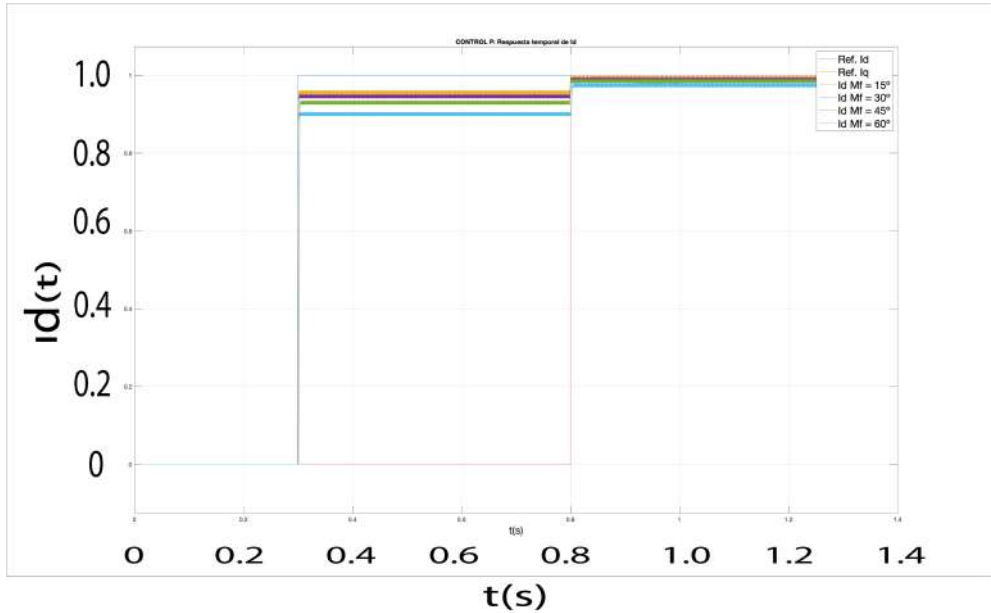


Figure 5.9: Time response of the d axis of a P Control in continuous time. Id pulse = 1 in 0.3s. Pulse of Iq = 1 in 0.8s.

The results of the phase and profit margins are shown in Table5.1:

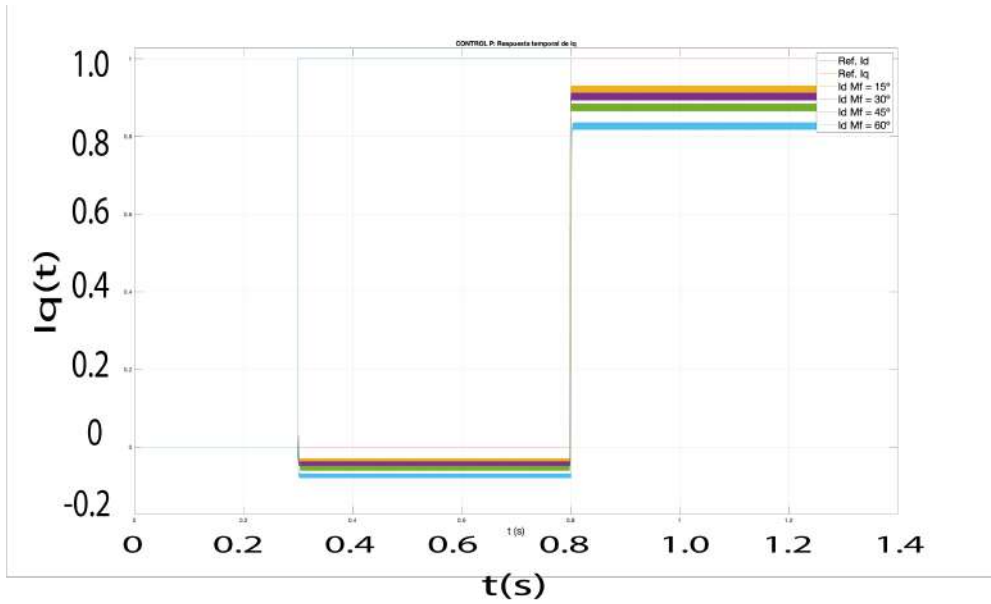


Figure 5.10: Time response of the q axis of a P Control in continuous time. Pulse of $I_d = 1$ in 0.3s. Pulse of $I_q = 1$ in 0.8s.

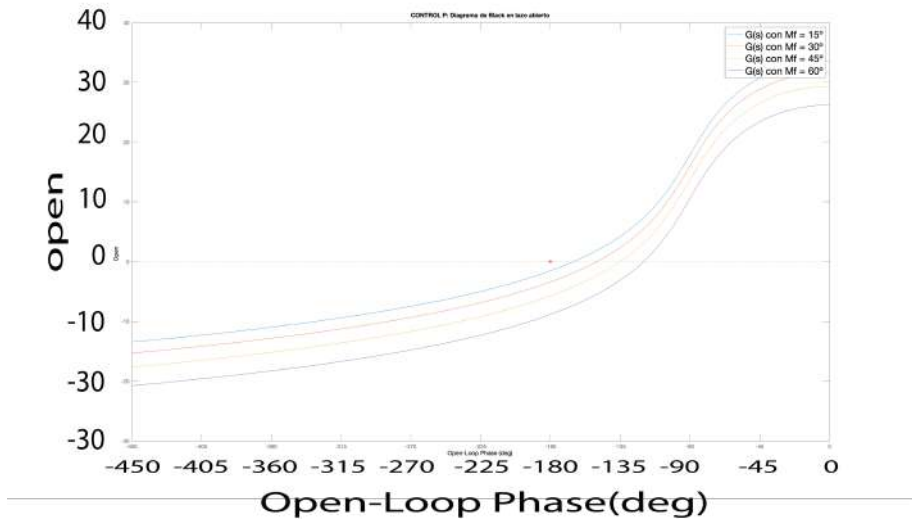


Figure 5.11: Open-loop frequency response of a P-Control in continuous time.

| Phase Margin (ϕ_m) | Crossover Pulsation(ω_0) | P control Gain (K) |
|---|---|--|
| 15 ^o | 8867 rad/s | 7.7138 |
| 30 ^o | 7155.3 rad/s | 6.2255 |
| 45 ^o | 5463.8 rad/s | 4.755 |
| 60 ^o | 3816.7 rad/s | 3.3238 |

Table 5.1: Table with the data of a Control P.

5.3.2 Control P Simulator. Discrete Time.

Applying a configuration like the one that can be seen in Fig. 5.12, the outputs that can be seen in Fig. 5.13 are obtained. Table 5.2 shows the data of the control applied to the simulator. It can be seen that there is an error in the permanent reference tracking regime in $d - q$ axes because it is a proportional control, without an integrator.

| Variable | Value |
|-----------------------------|------------|
| Phase margin (ϕ_m) | 30° |
| Reference weighting (b) | 1 |
| Proportional Gain (K) | 0.5 |
| Modulation Index (m) | 0.5 |

Table 5.2: Table with the values of a Control P applied to the simulator.

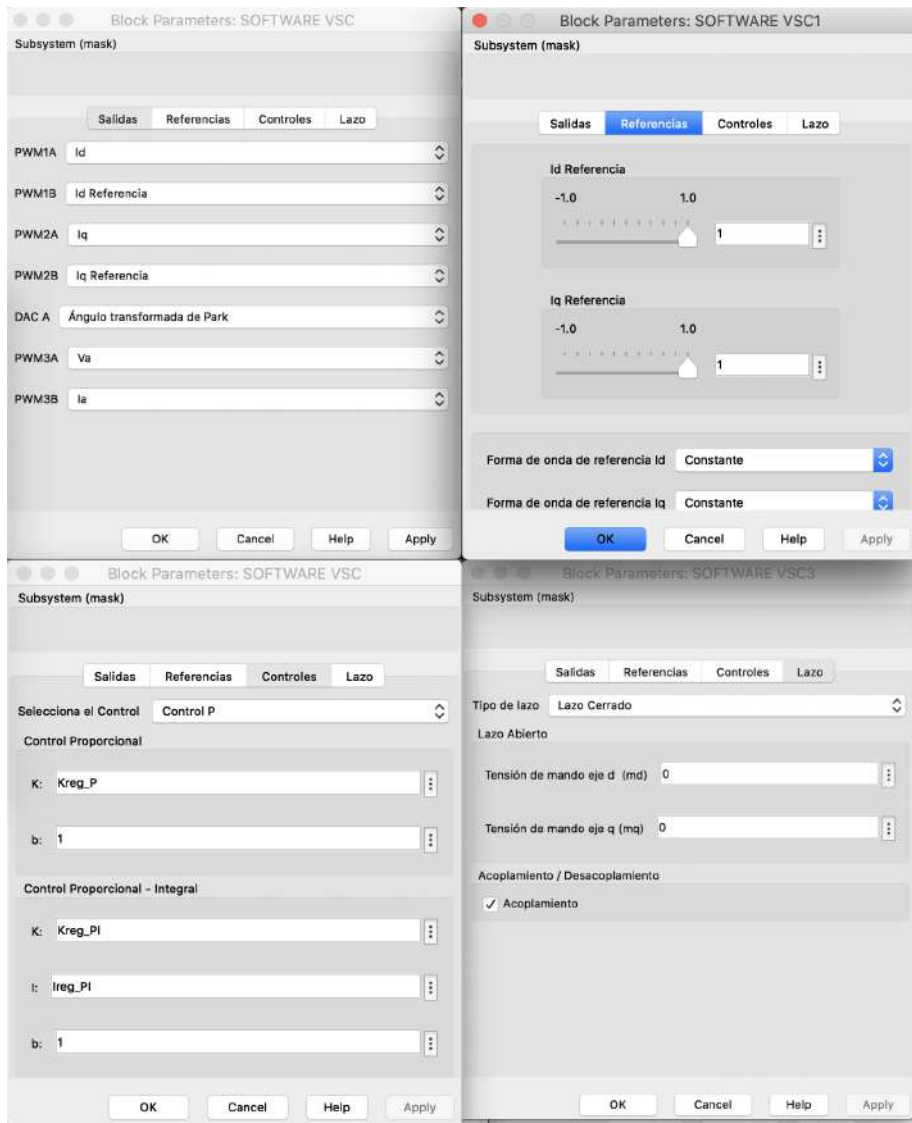


Figure 5.12: Configuration of a simulator with a control P. Identical to the one used in the simulator. The development is described in Appendix B.

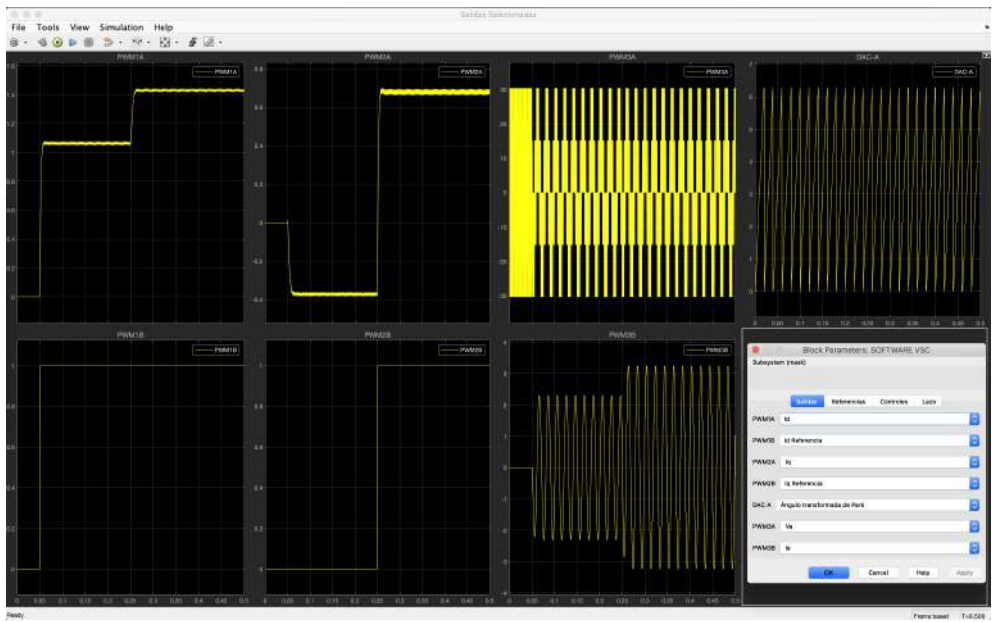


Figure 5.13: Scope of a simulator with a P control. The simulink PID block is implemented in discrete time. The simulation control values are listed in Table 5.2

5.3.3 Implementation of a Control P in the prototype.

The implementation of a P Control necessarily implies the permanent existence of a reference tracking error. D-axis tracking error data was taken experimentally, with different d-axis reference values and zero q-axis reference. The prototype operating conditions are recorded in Table 5.3. In all of them, a phase margin $\phi_m = 30^\circ$ has been used. The decoupling explained in Section 4.3 of the page 29, makes that the current of axis q does not vary with the reference of axis d, in this way, the output of Iq will always be 0 in permanent regime when the reference of Iq is 0. However, since the d axis represents a disturbance in the q axis and vice versa, and applying a P control, a following error is produced and therefore the output of Iq is different 0 in steady state. Likewise, the supply voltage of the VSC was 48V.

| Id Reference (pu) | Id Output (pu) | Error Id(pu) |
|--------------------------|-----------------------|---------------------|
| 0.02 | 0.016 | 0.004 |
| 0.08 | 0.03733 | 0.0427 |
| 0.1 | 0.044 | 0.056 |
| 0.12 | 0.0533 | 0.0667 |
| 0.14 | 0.06 | 0.08 |
| 0.16 | 0.0667 | 0.0933 |

Table 5.3: Table with the data of a Control P. Hardware implementation.

In Fig. 5.14 the data from Table 5.3 is shown in a graph.

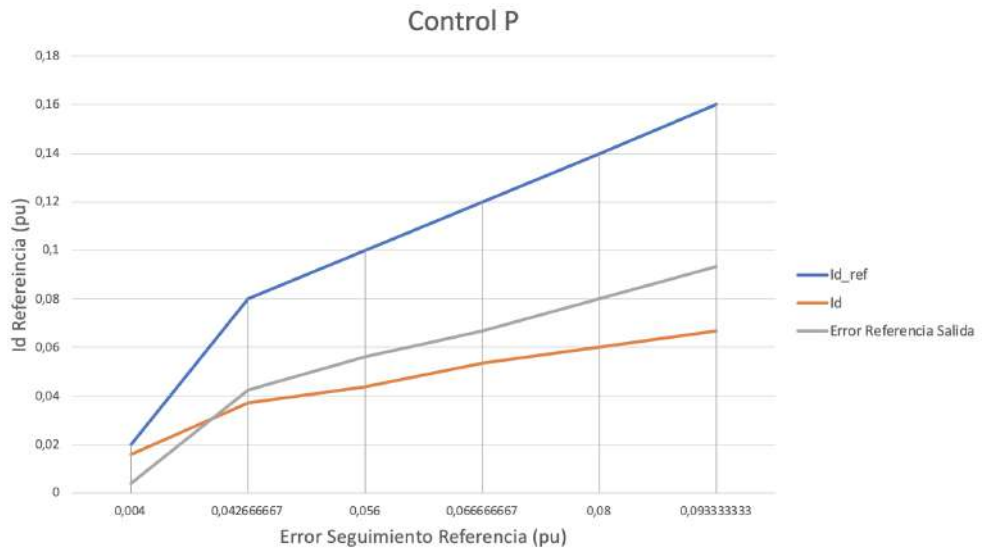


Figure 5.14: Graph of the reference following error of the d axis current with a Control P. Hardware implementation. The y-axis data is scaled 0.1 of the pu value.

5.4 PI Control

5.4.1 PI Control Design

The control will be designed through its frequency response, using the continuous-time plant obtained in Section 5.1. The frequency response design equations for a PI control will be:

$$\phi_c = -180 + \phi_m - \angle P(\omega_0) \quad (5.17)$$

$$Ac = \frac{1}{|P(\omega_0)|} \quad (5.18)$$

$$I = \tan\left(\frac{90 + \phi_c}{\omega_0}\right) \quad (5.19)$$

$$K = Ac \cdot \frac{I \cdot \omega_0}{\sqrt{1 + (I \cdot \omega_0)^2}} \quad (5.20)$$

$$(5.21)$$

The design of the integral proportional control, that is, of the variables of the control transfer function in (5.22) will be designed by adjusting the crossover pulsation and phase margin parameters. The procedure to follow will be iteration to adjust the crossover pulsation and phase margin values to the optimal values.

$$C_{serial}(s) = K_p \cdot \frac{1 + I \cdot s}{I \cdot s} \quad (5.22)$$

Since there are two values that can vary, in the first instance the phase margin value will be varied. Fig. 5.15, 5.16, 5.17 and 5.18, show the different temporal responses in continuous time that we can have, taking, for example, four different values for the phase margin (keeping the crossover pulsation constant) and another four different values for the crossover pulsation (keeping the phase margin constant).

Likewise, Fig. 5.19 and 5.20 are Black diagrams that allow analyzing the stability of the open loop system.

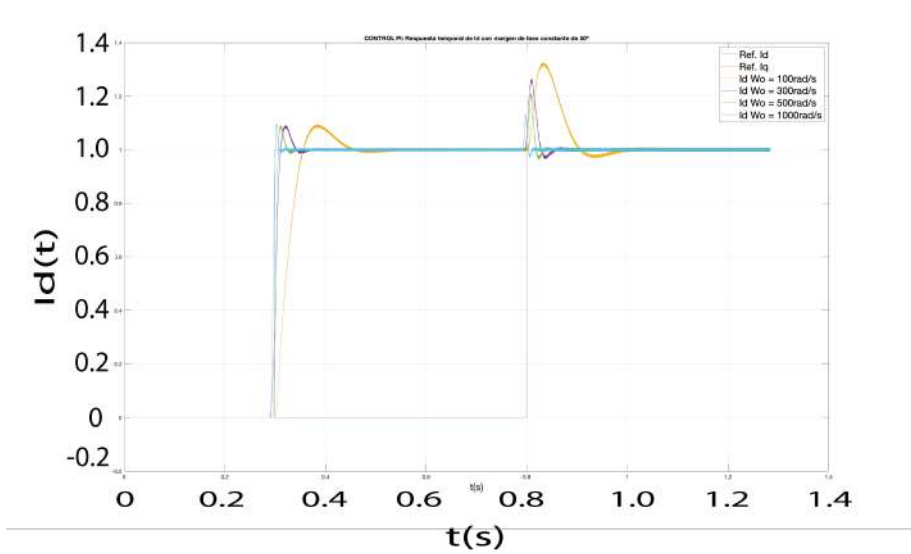


Figure 5.15: I_d temporal response with constant phase margin with PI control in continuous time. I_d pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s.

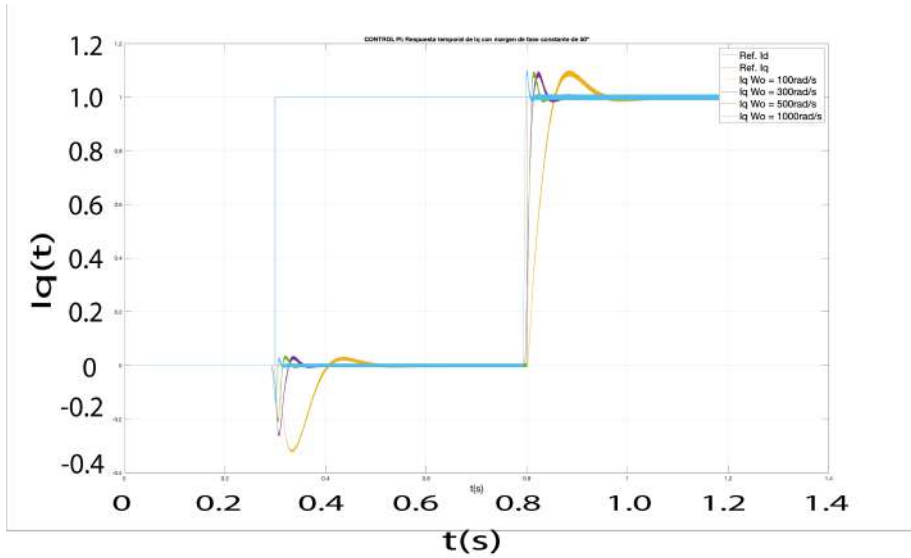


Figure 5.16: Temporal response of I_q with constant phase margin with PI control in continuous time. I_d pulse = 1 in 0.3s. Pulse of I_q = 1 in 0.8s.

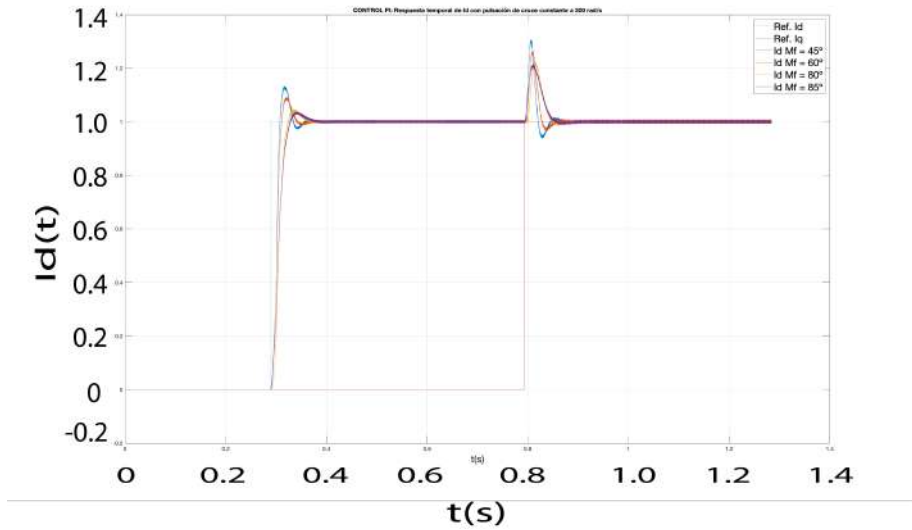


Figure 5.17: Id temporal response with constant crossover pulsation with PI control in continuous time. Id pulse = 1 in 0.3s. Pulse of Iq = 1 in 0.8s.

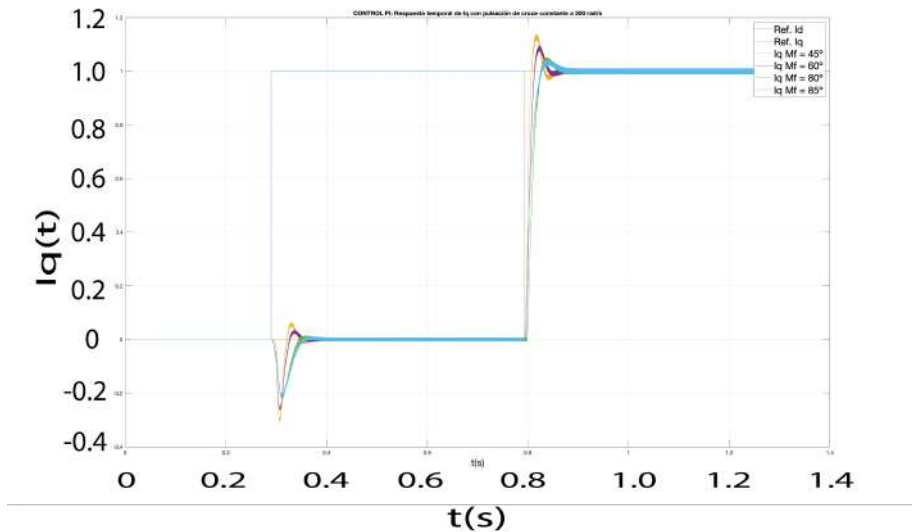


Figure 5.18: Temporal response of Iq with constant crossover pulsation with PI control in continuous time. Id pulse = 1 in 0.3s. Pulse of Iq = 1 in 0.8s.

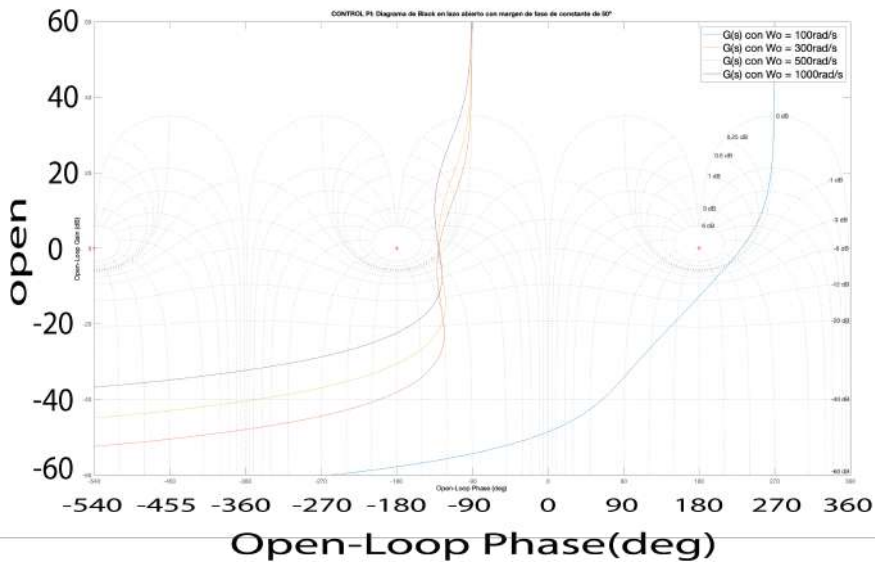


Figure 5.19: Open-loop response with constant phase margin with PI control in continuous time.

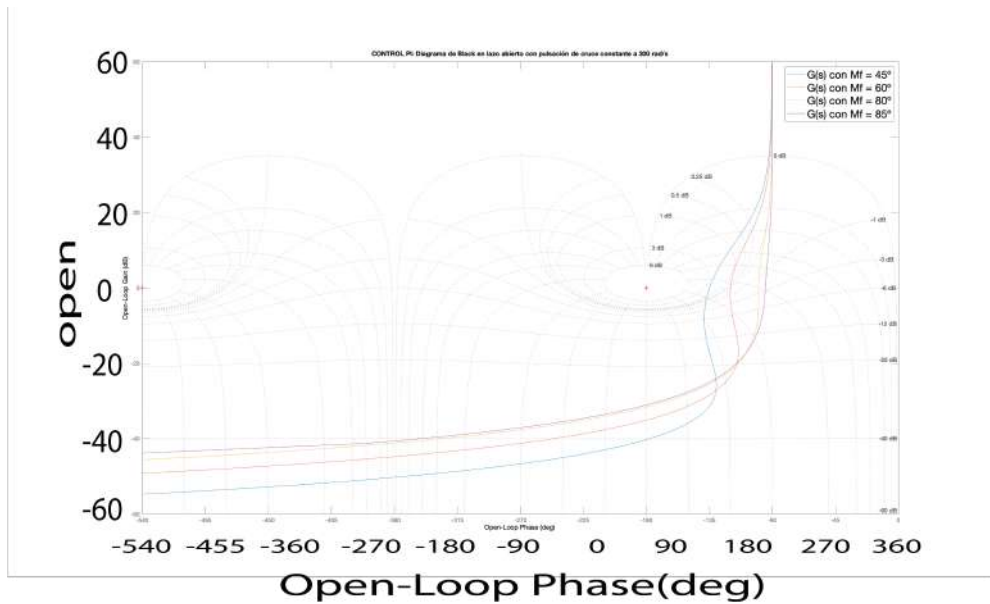


Figure 5.20: Open-loop response with constant crossover pulsation with continuous-time PI control.

5.4.2 PI Control Simulator

For the simulation of the PI control, a configuration like the one in Fig. 5.21 will be applied to the interface created for the simulator.

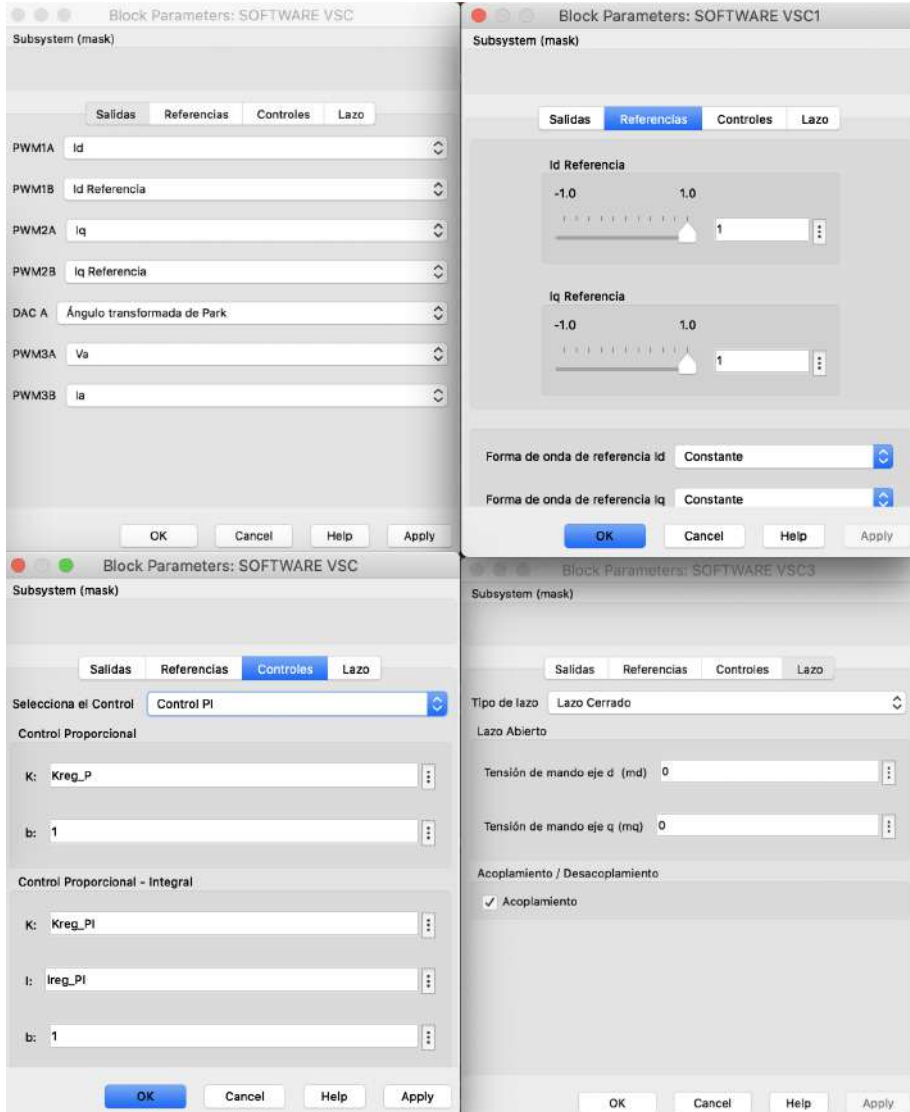


Figure 5.21: Configuration applied to the simulator with a PI Control. Identical to the one used in the simulator. The development is described in Appendix B

The simulator output, applying the configuration of Fig. 5.21 is the one that

can be observed in Fig. 5.22. It can be observed in the jumps that occur in the axes when their reference is zero, that the variations in the q axis are disturbances in the d axis and vice versa. The values of the PI control applied to the simulator are those described in Table 5.4:

| Variable | Value |
|------------------------------------|--------|
| Phase Marging (ϕ_m) | 50 |
| Crossover pulsation (ω_0) | 500 |
| Reference weighting (b) | 1 |
| Proportional gain (K) | 0.2582 |
| Integral constant (I) | 0.0013 |
| Modulation index (m) | 0.5 |

Table 5.4: Table with the values of a PI Control applied to the simulator.

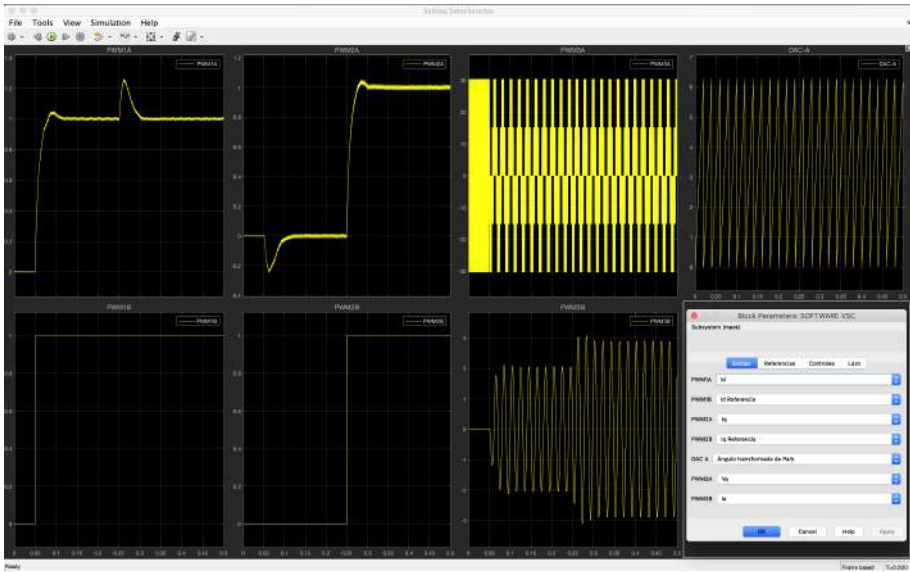


Figure 5.22: Scope of a simulator with a PI control. The PID block is implemented in discrete time. It can be observed in the jumps that occur in the dq axes when their reference is zero, that the variations in the q axis are disturbances in the d axis and vice versa. The simulation control values are listed in Table 5.4

5.4.3 Implementation of a PI Control in the prototype.

All control designed in continuous time, necessarily, is executed in discrete time in the microcontroller. Next, Table 5.5 shows the conditions in which the PI control was implemented in the prototype. In all of them, a nominal supply voltage of 48V was applied, and a q-axis reference current of 0. In all cases a phase margin of 50 is applied, a crossover pulsation of $500rad/s$, and a modulation index of 0.5.

| Id Reference (pu) | Id Output (pu) | Error Id(pu) |
|--------------------------|-----------------------|---------------------|
| 0.02 | 0.0253 | -0.0053 |
| 0.04 | 0.04 | 0 |
| 0.08 | 0.08 | 0 |
| 0.1 | 0.0933 | 0.0067 |
| 0.12 | 0.12 | 0 |
| 0.14 | 0.144 | 0.004 |
| 0.16 | 0.16 | 0 |
| 0.2 | 0.2053 | -0.00533 |
| 0.25 | 0.2533 | -0.0033 |
| 0.3 | 0.3067 | -0.0067 |
| 0.35 | 0.3333 | 0.0167 |
| 0.4 | 0.3227 | 0.0773 |
| 0.5 | 0.2493 | 0.2507 |

Table 5.5: Table with the data of a PI Control. Hardware implementation.

In Fig. 5.23 a graph is shown with the data from Table 5.5.

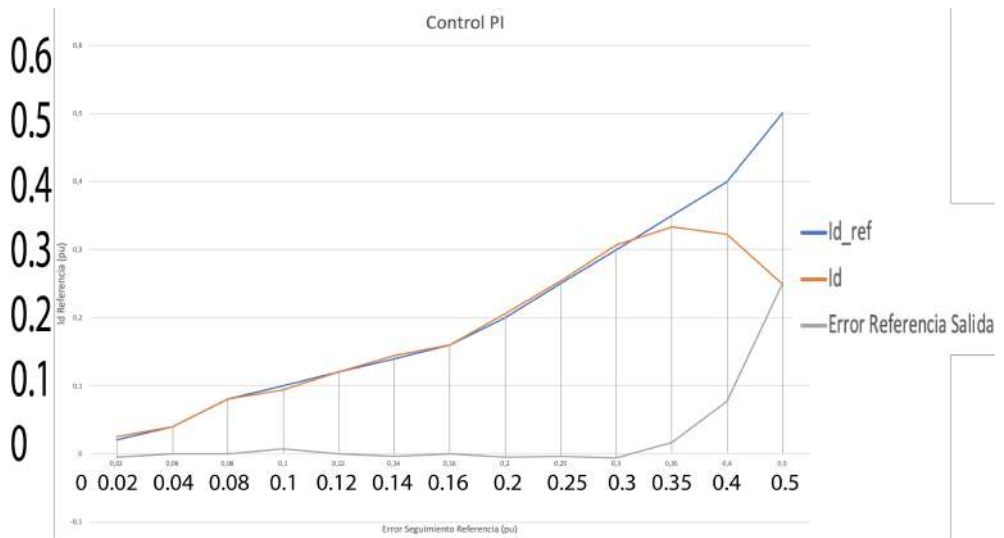


Figure 5.23: d-axis current reference tracking error plot with a PI Control. Hardware implementation. The axis values are scaled to 0.1 of the actual value in pu. It is observed that the coil saturates for values higher than 3pu (Approx. 6.1A).

Chapter 6

Main Components of the Prototype

6.1 Microprocessors and microcontrollers

The architecture of every microcomputer can be summarized as shown in Fig. 6.1, according to the Harvard model ¹. The microprocessor or CPU that can be seen in such figure, is the unit where all the instructions that come from the rest of the peripherals are processed, analyzed and executed.

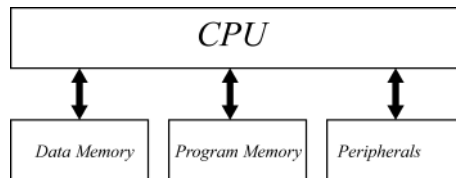


Figure 6.1: Scheme of a computer.

A microcontroller is a microcomputer or CPU compressed all into a single physical unit or integrated circuit, as in Fig. 6.2. For the development of this project, it has been necessary to distinguish between different microcontrollers for the correct operation of the VSC. The RaspberryPi 3B + was considered first. However, it was decided not to use this module because a higher switching frequencies was desired than could be achieved with available peripherals. Likewise, communications with

¹Obtained from the notes of the Microprocessors course. Last access: 01/09/2020. Accessible material for students enrolled in the Microprocessors subject at the Universidad Pontificia de Comillas.

peripherals had to be done through i2c, which increased the complexity of Matlab / Simulink diagrams.

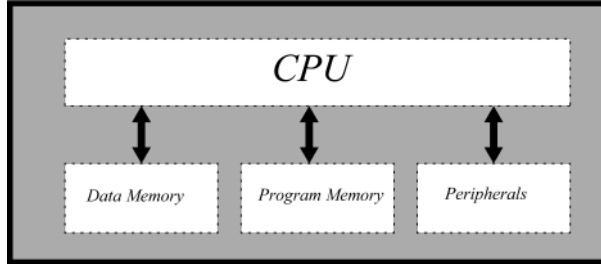


Figure 6.2: Scheme of a microcontroller.

It will be in the microcontroller where the designed controls will be run in real time. (See Section 5). These controls will be designed in Matlab - Simulink.

6.1.1 TMS320 Microcontrollers

The TMS320 microcontrollers appeared on the market in 1982 with the TMS32010 microcontroller, 16-bit fixed point, from the manufacturer Texas Instruments. Since then, the DSP family (*Digital Signal Processor*) has spread worldwide in a multitude of variants and subfamilies [43].

The TMS320 family consists of 16-bit fixed point, 32-bit floating point, and 64-bit multiprocessor DSPs [44]. The characteristics of this family of *chips* make it possible to use these microcontrollers in various sectors,

- Flexible instruction set.
- Operational flexibility.
- High processing speed.
- Parallel architecture.
- High cost efficiency.

In Fig. 6.3 you can see the various families of controllers of the TMS320 family until nowadays.

6.1.2 TMS320F28379D

The microcontroller used in the project is from the manufacturer Texas Instruments, model TMS320F28379D. Analyzing the nomenclature of the microcontroller [44],

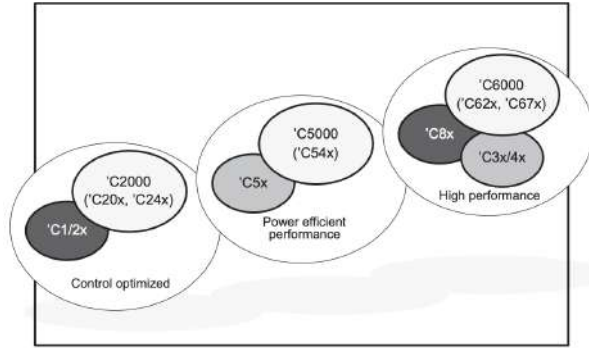


Figure 6.3: Evolution of the TMS320 family [44].

- TMS: Qualified device.
- 320: TMS320 family.
- F: CMOS with flash memory.
- 28739: generation of the microprocessor.
- D: Double core cite seventeen.

Among its main operating characteristics, the following [40] should be highlighted:

- 32-bit dual core.
- Internal clock at 200MHz.
- 512Kb or 1Mb internal flash memory.
- Two internal 10 MHz oscillators.
- Up to four channels of ADC conversion with up to twenty-four differential ports.
- Three 12-bit DAC channels.
- Twenty-four PWM ports. Sixteen of them high resolution.

However, the microcontroller has limitations when it comes to incorporating it into the card that incorporates it, the LAUNCHPADXL - F28379D, as discussed in Section 6.1.4.

6.1.3 LAUNCHPADXL - F28379D

The LAUNCHPADXL - F28379D used in the project is shown in Fig. 6.4 and 6.5, from the manufacturer Texas Instruments, a model of the microcontroller that incorporates, the TMS320F28379D, shown assembled in the LaunchPad in Fig. 6.6 and 6.7, within the C2000 Microcontrollers series.



Figure 6.4: LaunchPad F28379D Top view.

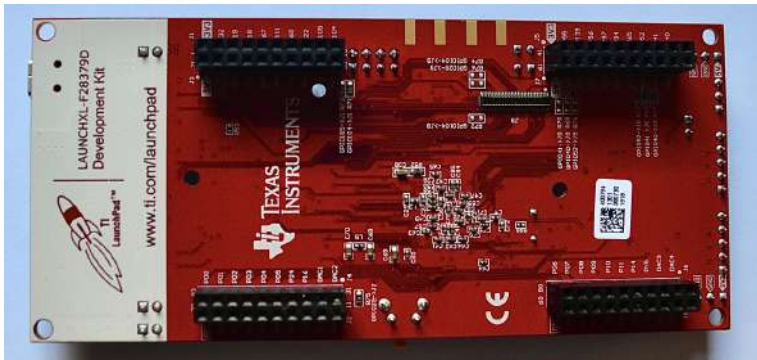


Figure 6.5: LaunchPad F28379D bottom view.

The pins are located in such a way that the LaunchPad board attaches directly to the VSC BOOSTXL-3PhGaNInv, and programming is greatly simplified.

The input and output pins of the microcontroller are shown in Fig. 6.8, 6.9 and 6.10.

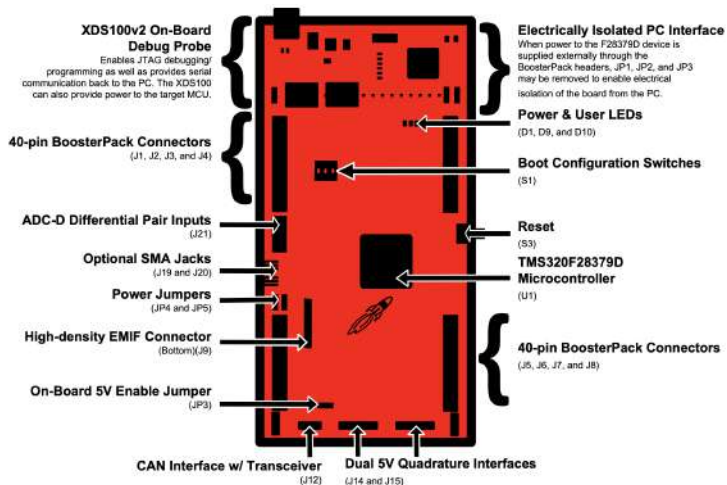


Figure 6.6: LAUNCHXL - F28379D [45]

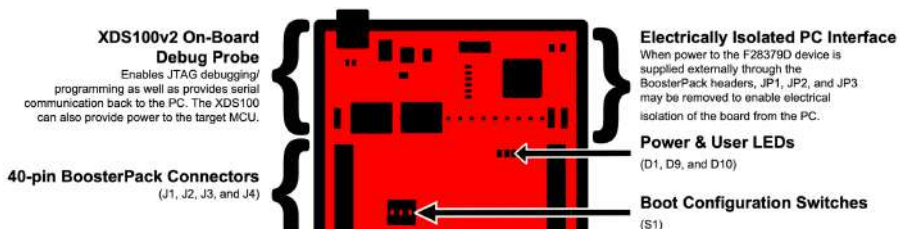


Figure 6.7: LAUNCHXL - F28379D detailed top view[45]

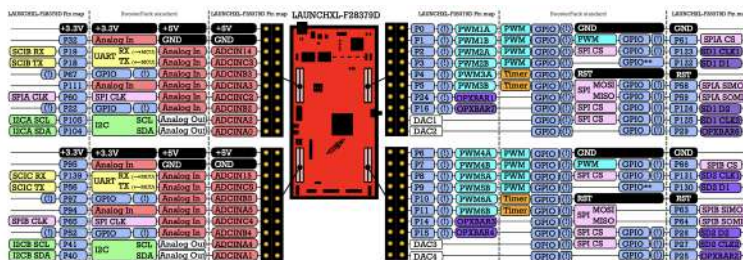


Figure 6.8: LAUNCHXL - F28379D Pinout [46].

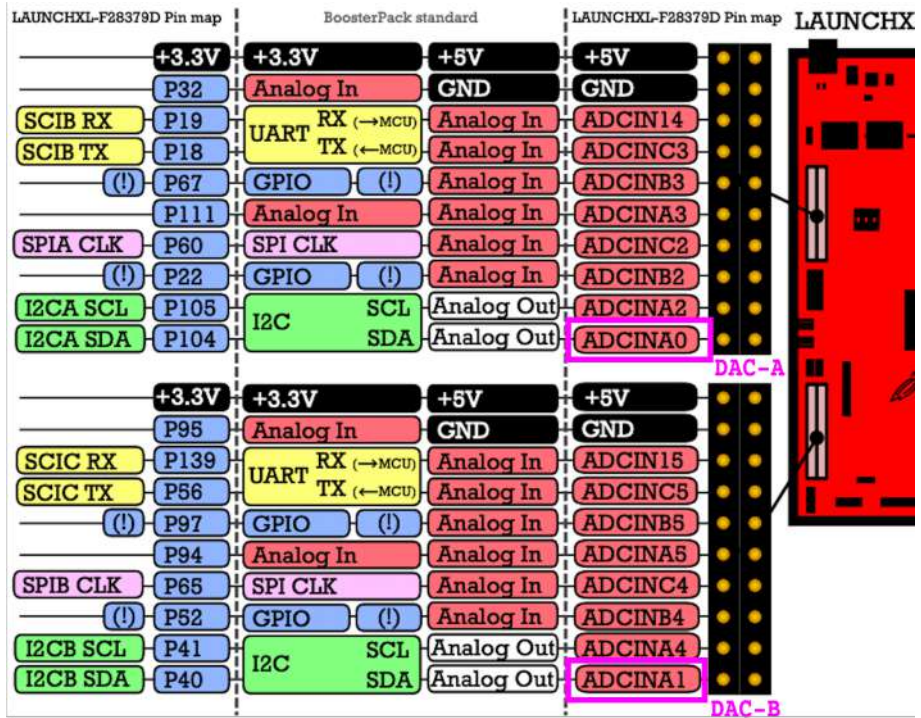


Figure 6.9: LAUNCHXL - F28379D Pinout left side in a detailed view [46]. The DAC outputs for which there is a block in the Simulink toolbox are outlined.

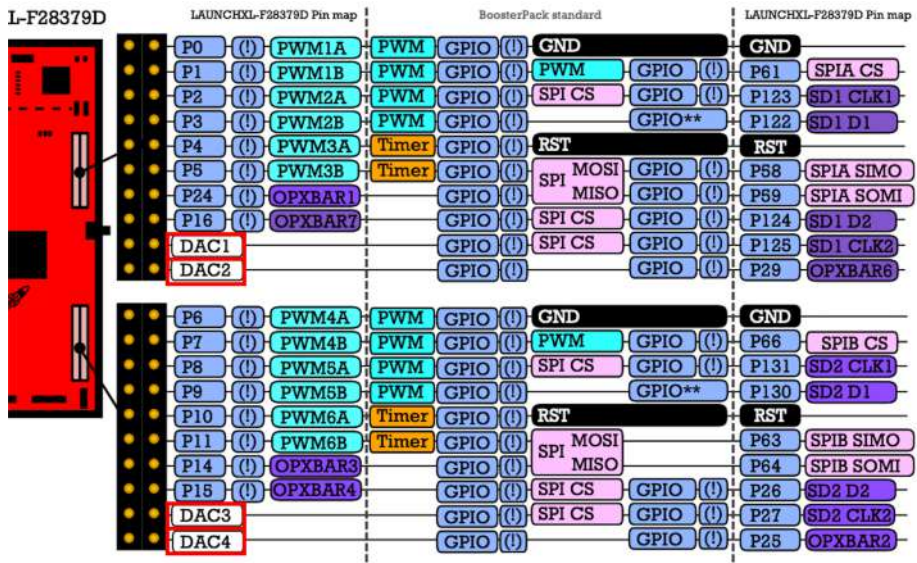


Figure 6.10: LAUNCHXL - F28379D Pinout right side in a detailed view [46]. Pins for which there is no specific Simulink block and cannot be used with said library as DAC are outlined.

6.1.4 Pinout Rectification

Fig. 6.8, 6.9 and 6.10 show the generic pins for the TMS320F28379D microcontroller, however the LAUNCHPADXL - F28379D has some differences in its PINOUT. There are certain modifications that have been found when developing the project, listed below:

1. The ADCINA0 pin can behave as an ADC or as a DAC-A (boxed in Fig. 6.9).
2. The ADCINAq pin can behave as an ADC or as a DAC-B (boxed in Fig. 6.9).
3. The DAC1, DAC2, DAC3 and DAC4 pins are not accessible from a Simulink library block (boxed in the Fig. 6.10).

6.1.5 Matlab and Simulink

The software used to control inverter variables will be Matlab in version R2019. Within it, scripts will be developed to:

- The calculation of variables for the follow-up controls.
- Initialization of the variables of the real values of the model.
- The calculation of transfer functions that may be relevant for further analysis.
- Initialization of the communication protocols with the different sensors and probes found in the VSC.
- The programming of the PWMs that trigger the semiconductors of the VSC.

Next, Simulink module will be used, belonging to Matlab, which will be implemented in the microcontroller and will run in real time on it.

There are specific libraries and files for this board (and others from the same manufacturer) available in a *toolbox* in SIMULINK. The different Simulink blocks for the C2000 LaunchPad series processors are shown in Fig. 6.11. Within these, the prototype is developed with the LaunchPad F28379D, therefore, the *F28379D* library is used, displayed in Fig. 6.12. When compiling the Matlab and Simulink files, a code is generated using the manufacturer's own compiler. This code is what runs on the LaunchPad. To install the necessary *toolbox*, see Appendix A.

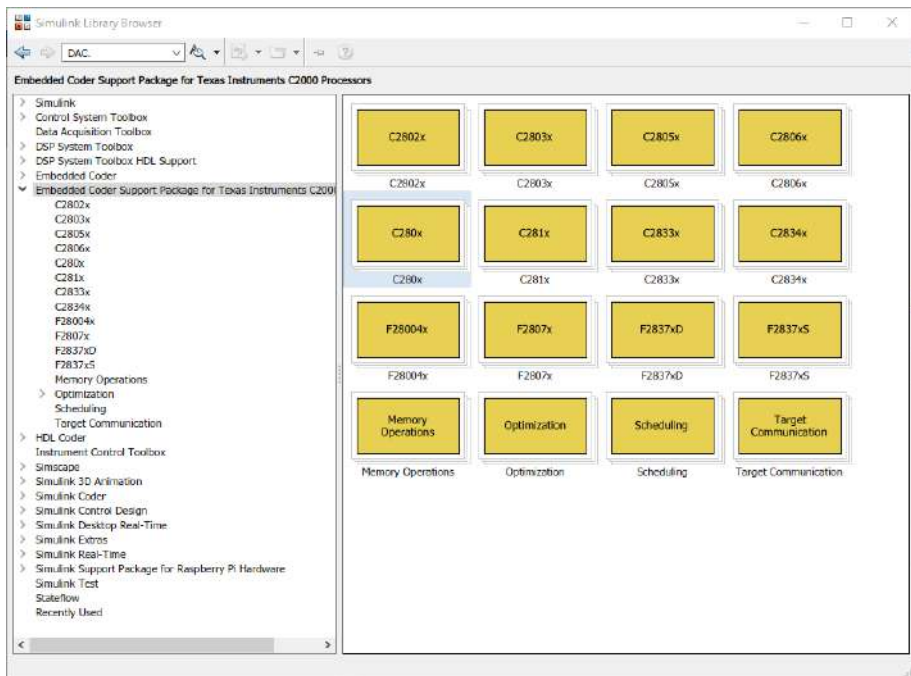


Figure 6.11: Texas Instruments C2000 Processor Toolbox. [47]

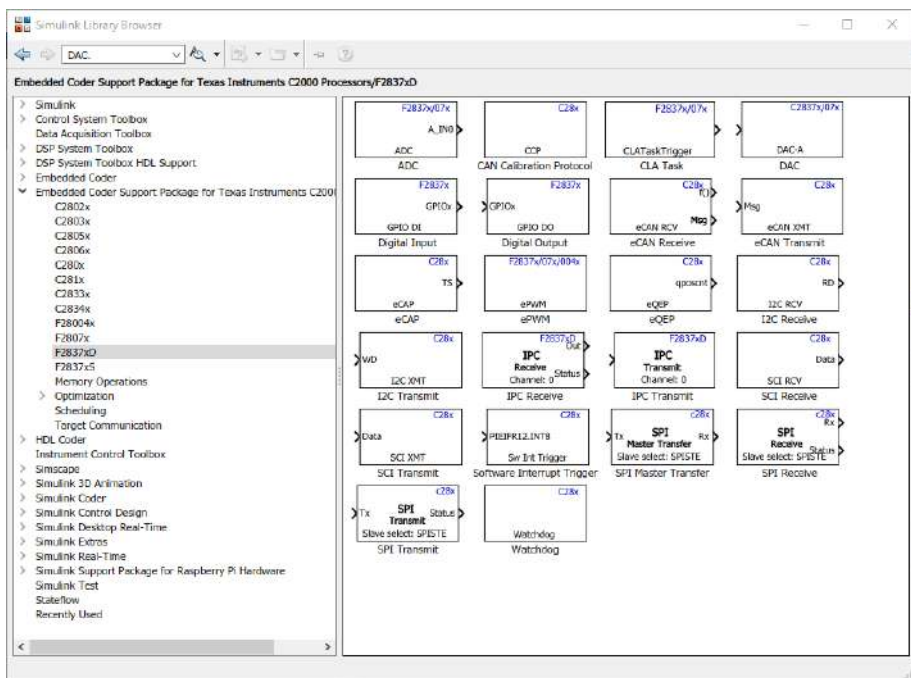


Figure 6.12: Texas Instruments LaunchPad F2837xD Toolbox. [47]

6.2 VSC BOOSTXL-3PhGaNInv

The VSC that will be used in the project is from the manufacturer Texas Instruments, model BOOSTXL-3PhGaNInv, shown in Fig. 6.13 and 6.14.

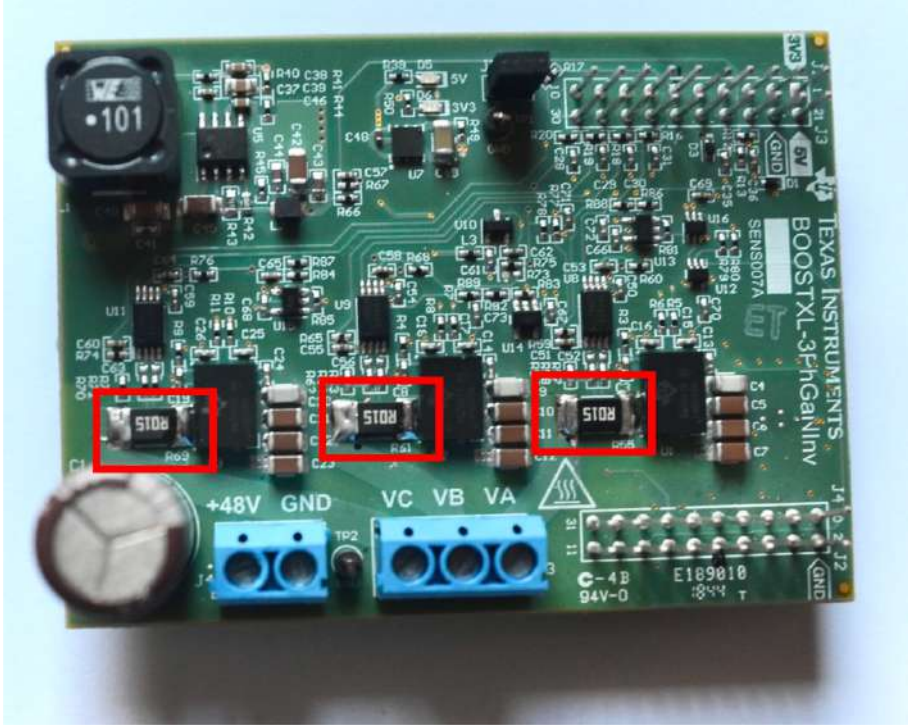


Figure 6.13: BOOSTXL-3PhGaNInv top view. Resistances changed are boxed.

It is a three-phase DC - AC converter. The VSC operation diagram is shown in Fig. 6.15.

This VSC has the main features [48]:

- Supply voltage between 12V and 60V. The nominal supply voltage is 48V.
- Semiconductors MOSFETs of gallium nitride.
- Current sensors per phase with peak currents between $\pm 16.5A$. Three resistors, which were 5m *Omega*, have been changed for 15m *Omega*, to increase the precision of current measurement, with the three changed resistors in Fig. 6.13.
- Compatible with C2000 MCU (MicroController Unit) series evaluation boards.

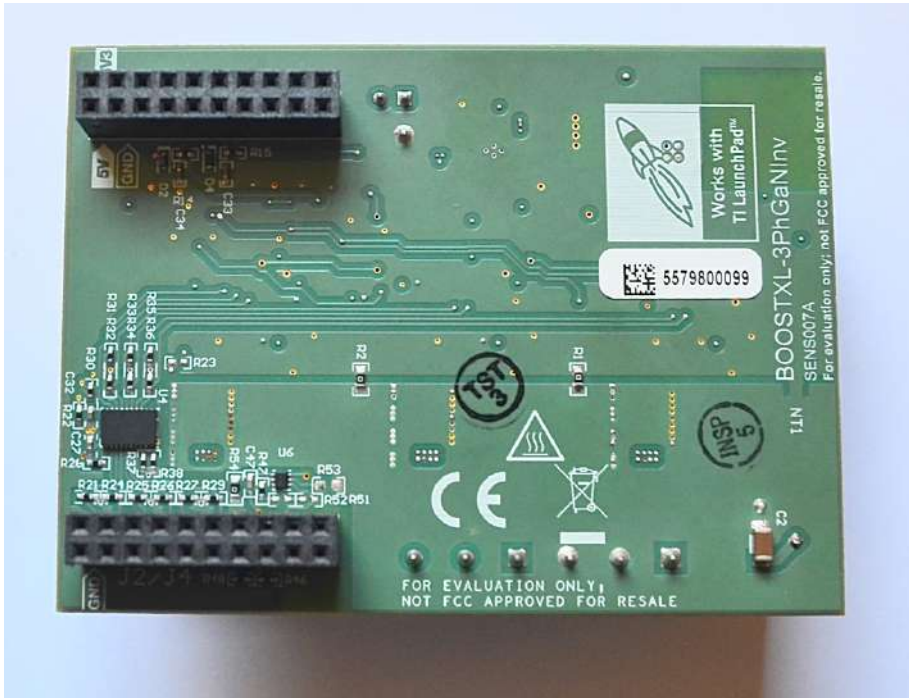


Figure 6.14: BOOSTXL-3PhGaNInv bottom view.

The components that appear in Fig. 6.15 and that need to be explained are the *LMG5200* and the *INA240*.

The *LMG5200* is a device that contains the semiconductors that want to be controlled. Each device contains two FETS (Section 2.2). Each of the FETS has the ability to withstand continuous voltage up to 80V and voltage spikes up to 100V. Likewise, it is capable of conducting up to 10A of current. All technical information can be found in [49]. Regarding the turn-off and turn-on time of semiconductors, explained in Section 2.2.5, the behavior of semiconductor activation and deactivation can be observed in Fig. 6.16-a. Then, Fig. 6.16-b represents the activation zone in detail.

The semiconductors contained in the device are controlled by a driver built into it. It can be read in the *LMG5200* datasheet [49] that the driver that controls the on and off of the semiconductors is capable of generating a dead time (Section 2.2.5) of 2ns. This great speed in semiconductors is due to the fact that they are made of gallium nitride (GaN), which, as described in Section 2.2.4, is substantially faster in conducting electrons than the most common ones such as silicon cite seven.

The *INA240* is an electronic device that amplifies the current flowing through

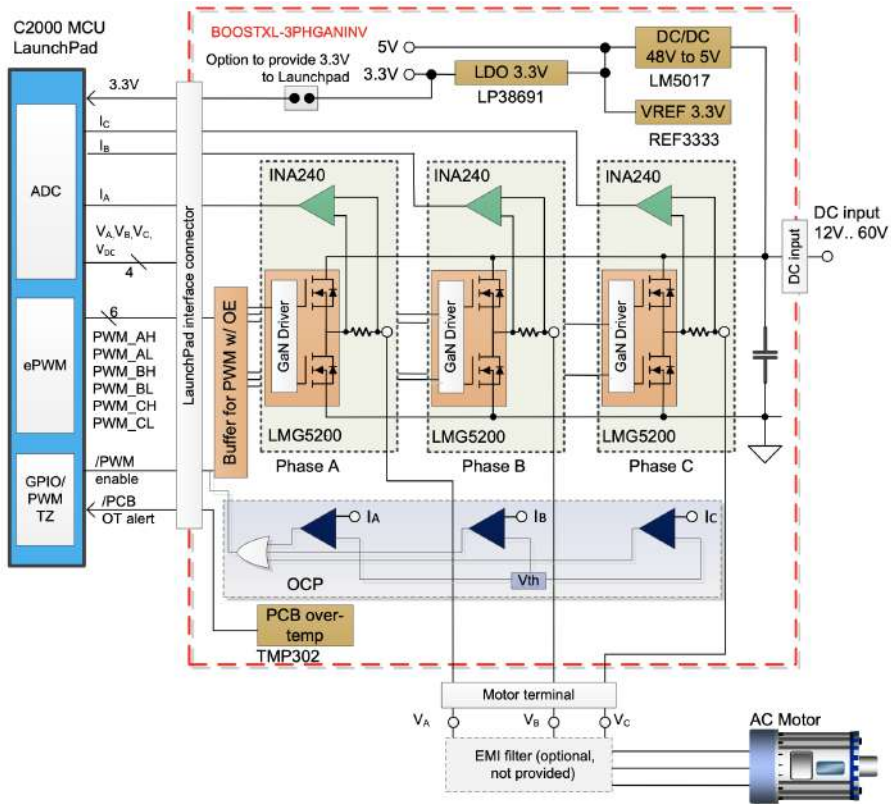


Figure 6.15: BOOSTXL-3PhGaNInv function diagram [48].

the phases in output voltage format, with properties such as rejection of noise from PWM switching [51]. A functional diagram of the system is shown in Fig. 6.17.

This device comes by default with a measurement resistance (R_s in (6.1)) of $5m\Omega$. However, if the currents to be measured are too small, the precision of the measurement can be increased by modifying this resistance for another of higher value, so that at the same value of current through said resistance, a higher voltage is obtained at the output. In this project, the resistance of $5m\Omega$ is changed to a resistance of $15m\Omega$ (boxed in Fig. 6.13), increasing the value of the measurement by three. Likewise, the gain value of the INA240 integrated in the inverter, of $20V/V$, makes the INA240 that is used the INA240A1, as can be seen in Fig. 6.18.

Let the electronic circuit of the INA240 be parametric in Fig. 6.19.

The relationship between the differential input $V_1 - V_2$ and the output V_o is

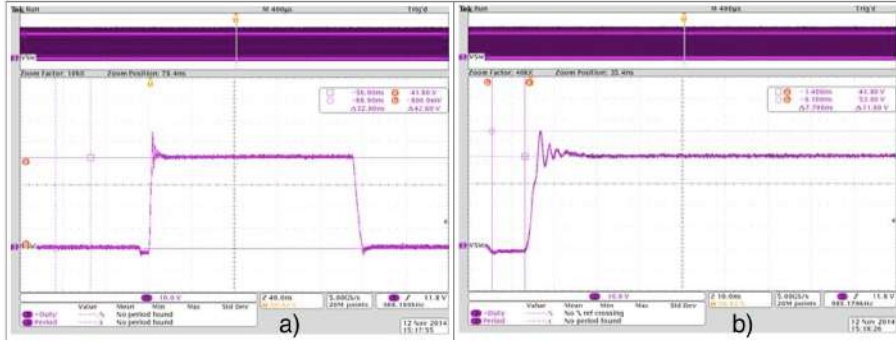


Figure 6.16: LMG5200 in a) off mode and b) on mode [50].

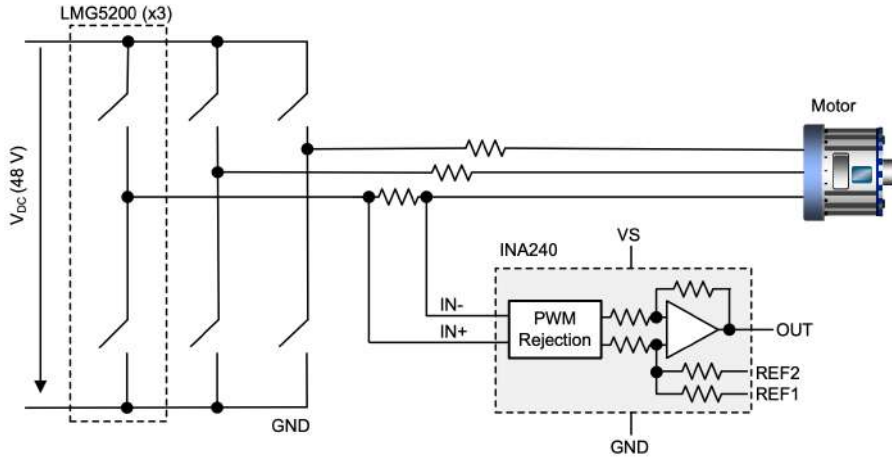


Figure 6.17: INA240 functional diagram [48]

| PRODUCT | GAIN (V/V) |
|----------|------------|
| INA240A1 | 20 |
| INA240A2 | 50 |
| INA240A3 | 100 |
| INA240A4 | 200 |

Figure 6.18: INA240 types based on their gain [51]

shown in (6.2), in which:

$$V_{in} = V_1 - V_2 = \text{SampledCurrent} \cdot R_s \quad (6.1)$$

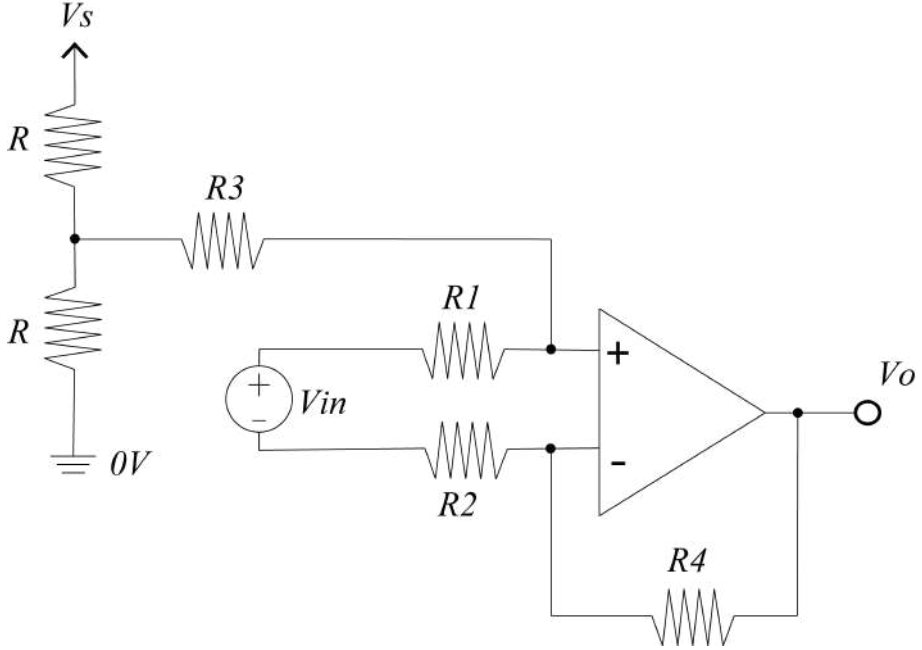


Figure 6.19: INA240 electronic circuit. [49]

$$V_A = V_S \frac{R}{R + R} = \frac{V_S}{2}$$

$$V_o = V_1 \frac{R_2 + R_4}{R_2} \frac{R_3}{R_1 + R_3} - V_2 \frac{R_4}{R_2} + V_A \frac{R_1}{R_1 + R_3} \frac{R_2 + R_4}{R_2} \quad (6.2)$$

In (6.2) it can be observed that the value of the output voltage will consist of a positive mean value, as well as a relationship between the resistances that multiply V_1 and V_2 such that the gain depends of the INA240 used, shown in Fig. 6.18.

6.2.1 BOOSTXL - 3PhGaNInv Pinout

BOOSTXL - 3PhGaNInv pinout is shown in Fig. 6.20 and 6.21.

| PIN | SIGNAL | I/O (3.3 V) | PIN | SIGNAL | I/O (3.3 V) |
|-------|-------------------------|-----------------------------|-------|--------|---------------------------------|
| J1-1 | 3.3-V supply (optional) | O or NC (jumper selectable) | J1-2 | NC | — |
| J1-3 | NC | — | J1-4 | GND | GND |
| J1-5 | NC | — | J1-6 | VDC | O (0 to 3.3 V) ⁽¹⁾ |
| J1-7 | NC | — | J1-8 | VA | O (0 to 3.3 V) ⁽¹⁾ |
| J1-9 | NC | — | J1-10 | VB | O (0 to 3.3 V) ⁽¹⁾ |
| J1-11 | VREF | O (3.3 V, 10 ppm) | J1-12 | VC | O (0 V to 3.3 V) ⁽¹⁾ |
| J1-13 | NC | — | J1-14 | IA | O (0 to 3.3 V) |
| J1-15 | NC | — | J1-16 | IB | O (0 to 3.3 V) |
| J1-17 | NC | — | J1-18 | IC | O (0 to 3.3 V) |
| J1-19 | NC | — | J1-20 | VREF | O (3.3 V, 10 ppm/K) |

⁽¹⁾ Overvoltage protection with Schottky diodes ensures output voltage remains below 3.6 V.

Figure 6.20: BOOSTXL - 3PhGaNInv left side connections (J1) [52].

| PIN | SIGNAL | I/O (3.3 V) | PIN | SIGNAL | I/O (3.3 V) |
|-------|-------------------|------------------------|-------|--------------------------|-------------|
| J2-1 | PWM A (high-side) | I (10k PD) | J2-2 | GND | GND |
| J2-3 | PWM A (low-side) | I (10k PD) | J2-4 | NC | — |
| J2-5 | PWM B (high-side) | I (10k PD) | J2-6 | NC | — |
| J2-7 | PWM B (low-side) | I (10k PD) | J2-8 | NC | — |
| J2-9 | PWM C (high-side) | I (10k PD) | J2-10 | NC | — |
| J2-11 | PWM C (low-side) | I (10k PD) | J2-12 | NC | — |
| J2-13 | /PCB OT alert | O (open drain, 10k PU) | J2-14 | NC | — |
| J2-15 | NC | — | J2-16 | /PWM enable (active low) | I (10k PU) |
| J2-17 | NC | — | J2-18 | NC | — |
| J2-19 | NC | — | J2-20 | NC | — |

Figure 6.21: BOOSTXL - 3PhGaNInv right side connections (J2) [52].

The connections of the BOOSTXL - 3PhGaNInv (VSC) with the LaunchPad F28379D are made in such a way that they coincide:

- VSC measurements with LaunchPad ADCs.
- The VSC PWMs are triggered from the LaunchPad PWM ports.

6.2.2 LaunchPad - VSC interconnection

There are two ways to connect the LAUNCHPADXL F28379D with the BOOSTXL-3PhGaNIv, shown in Fig. 6.22 and 6.23.



Figure 6.22: LaunchPad set with VSC with the connection on top.

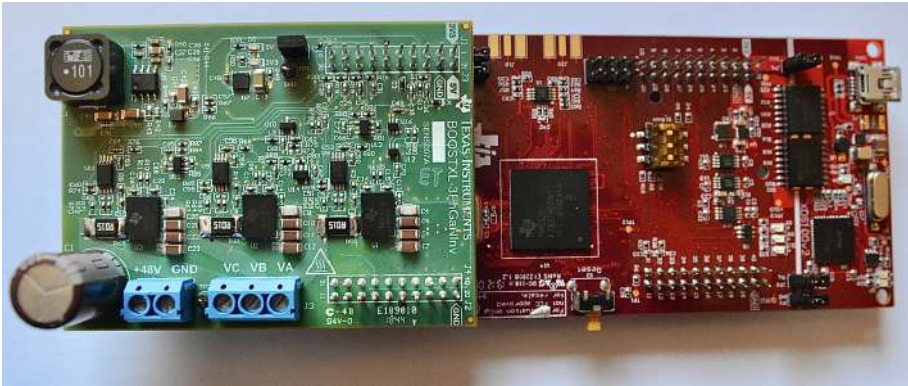


Figure 6.23: LaunchPad set with VSC with the connection on bottom.

Table 6.1 ² must be taken into account the main connections between the LaunchPad F28379D and the BOOSTXL - 3PhGaNIv when it is connected in

²The limitations described in Section 6.1.4

the upper or lower part ³.

| Launchpad F28379D Upper | Launchpad F28379D Lower | BOOSTXL |
|----------------------------|----------------------------|---------------------|
| ADCIN14 | ADCIN15 | V_{DC} |
| ADCINC3 | ADCINC5 | V_a |
| ADCINB3 | ADCINB5 | V_b |
| ADCINA3 | ADCINA5 | V_c |
| ADCINC2 | ADCINC4 | I_a |
| ADCINB2 | ADCINB4 | I_b |
| ADCINA2 | ADCINA4 | I_c |
| ADCINA0 | ADCINA1 | V_{ref} |
| PWM1A | PWM4A | PWMA (High Side) |
| PWM1B | PWM4B | PWMA (Low Side) |
| PWM2A | PWM5A | PWMB (High Side) |
| PWM2B | PWM5B | PWMB (Low Side) |
| PWM3A | PWM6A | PWMC (High Side) |
| PWM3B | PWM6B | PWMC (Low Side) |
| GPIO 124 | - | PWM Enable |
| - | GPIO 26 | PWM Enable |
| ADCINA0 | - | DAC-A |
| - | ADCINA1 | DAC-B |

Table 6.1: Connections between the LaunchPad F28379D and the BOOSTXL-3PhGaNInv depending on the Upper-Lower connection.

The format used in this project is to connect the LaunchPad with the VSC at the bottom to facilitate later its encapsulation and access to output terminals to obtain measurements. Figures 6.24 and 6.25 show more views of the assembly used in the prototype.

³It is taken as a reference to say *Upper* or *Lower*: Upper, the VSC immediately next to the input microusb cable; Bottom, VSC away further microusb cable input.

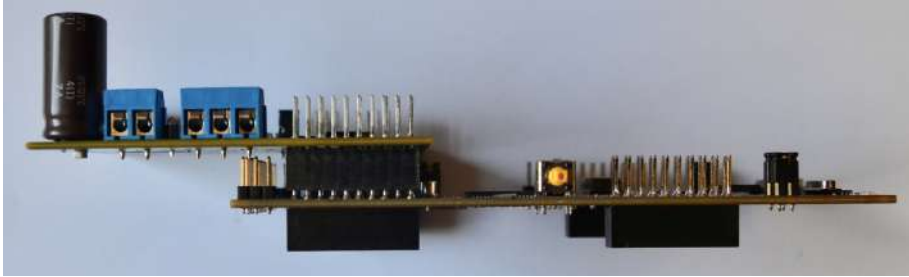


Figure 6.24: Conjunto LaunchPad con VSC con la conexión en la parte inferior. Vista lateral.

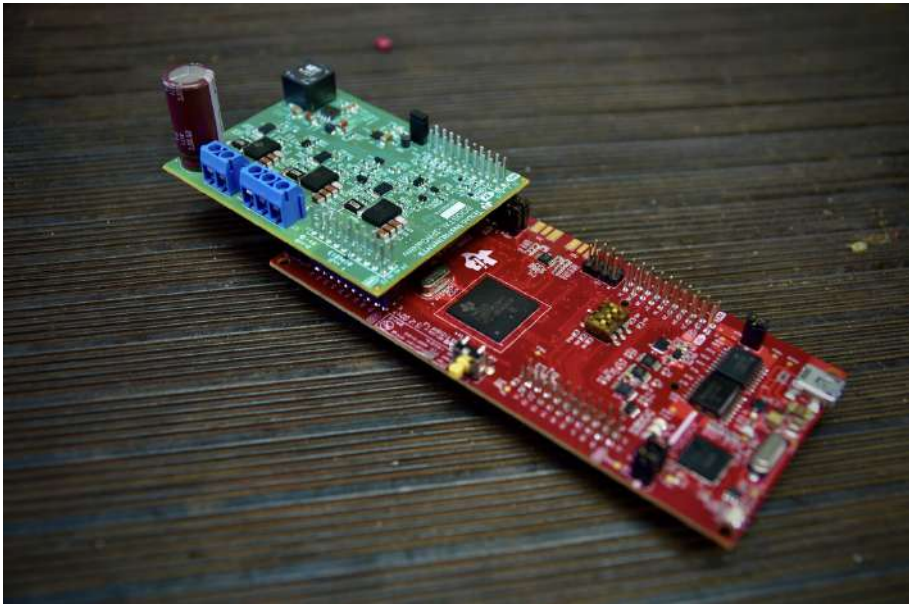


Figure 6.25: Conjunto LaunchPad con VSC con la conexión en la parte inferior. Vista en perspectiva.

The prototype of the VSC along with the Launchpad was encapsulated in such a way that all measurements could be accessible. Likewise, since it is a teaching prototype, safety bananas were used to avoid having access to high voltages and / or currents, shown in Figs. 6.26 and 6.27. Pins in Fig. 6.28 are voltages between 0 and 3.3 V. Wires between pins are GNDs (*Grounds*) connected to the same GND of the LaunchPad and VSC.

LED lights were also incorporated to monitor the status of the system, being in open or closed loop, the type of control implemented and if the coils are coupled on $d - q$ axes.



Figure 6.26: Encapsulated VSC and Launchpad. DC voltage input.

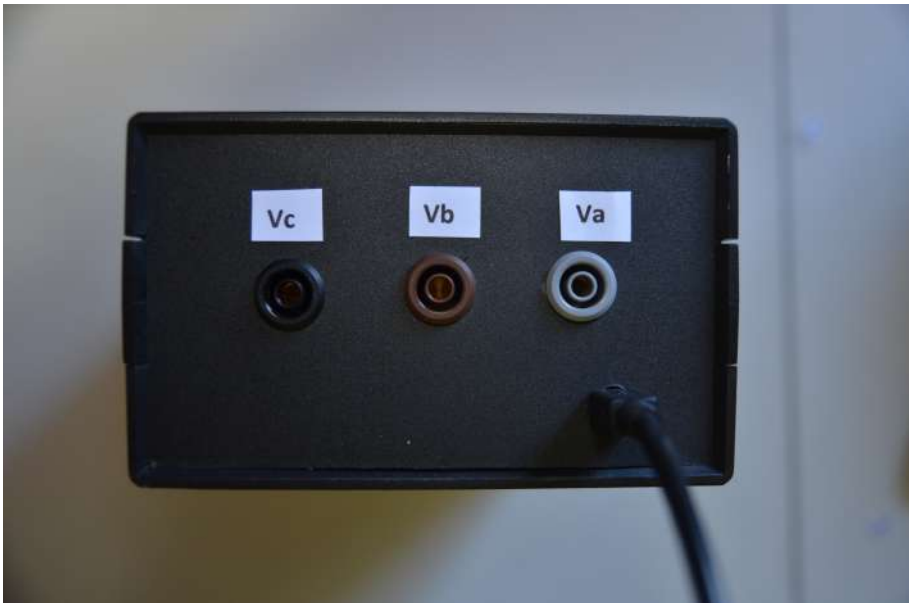


Figure 6.27: Encapsulated VSC and Launchpad. AC voltage outputs and USB connection.

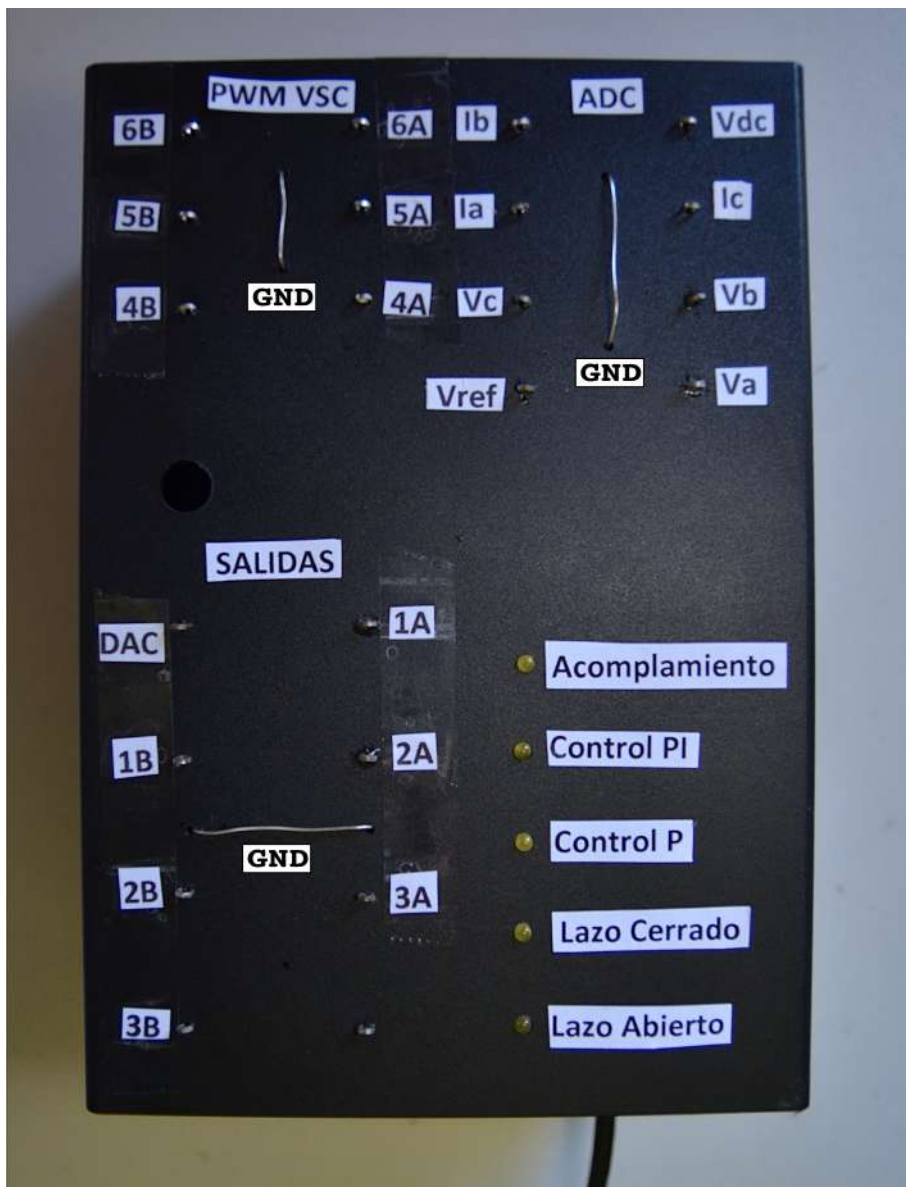


Figure 6.28: Encapsulated VSC and Launchpad. Oscilloscope pins.

6.3 Prototype Software Configuration

This section deals with the different modules that make up the prototype in the Simulink ⁴. Everything needed to install in the Matlab package is described in Appendix A.

6.3.1 PWM Module

Simulink ePWM block, as it is called for the LaunchPad F28379D, is shown in Fig. 6.29. This module allows to configure the activation and deactivation of the semiconductors of the VSC BOOSTXL - 3PhGaNInv directly, either each transistor separately, or to program the single-phase transistors so that one is complementary to the other.



Figure 6.29: Simulink ePWM block. TI C2000 library.

The PWM works with an internal counter, so that while the counter does not reach the service factor (this in clock cycles as well), the PWM output is high (t_{on}) and the rest of the switching period, the output is low (t_{off}). First of all, the setting of the PWM switching frequency is done taking into account the frequency of the internal clock and pre-scaled of the microcontroller. This switching frequency must be entered in the switching period format, in clock cycles or in seconds. If it is entered in seconds, the microcontroller converts said value to clock cycles, so that to reduce the computational load when compiling the code, it will be entered in the form of clock cycles, according to (6.3):

$$TimerPeriod = \frac{FrequencyClock}{FrequencyConm \cdot Pre - scale} \quad (6.3)$$

Where,

- TimerPeriod: Switching period in clock cycles.
- FrequencyClock: Internal clock frequency. The internal clock frequency of the TMS320F28379D is 200MHz [40].

⁴All help and description of the blocks is available at url: <https://es.mathworks.com/help/supportpkg/texasinstrumentsc2000/modeling.html>. Last access 01/01/2020.

- FrequencyComm: Switching frequency in Hz.
- Pre-scale: Pre-scale of the microcontroller. On the LaunchPad F28379D it is 4.

Thus, looking for a switching frequency of $10kHz$, knowing that we are working with a prescaling of 4:

$$TimerPeriod = \frac{200MHz}{10kHz \cdot 4} = 5000$$

The activation method described in Section 2.4.1, can be configured with the options *up count*, *down count* or *up-down count* shown in Fig. 6.30.

Since we want to divide the service factor (Section 2.1, page 11), into two equal parts in a switching period, the form *up- down count*.

The main settings box, from the *General* tab, looks like Fig. 6.31.

Given the way to implant the VSC BOOSTXL - 3PhGaNInv with the LaunchPad F28379D in the prototype, the configuration of the PWMs is shown in Fig. 6.32.

The ePWM block allows you to configure the way to activate and deactivate the fets of the same phase. To configure the transistors so that one of the transistors (the lower one) is complementary to the other (the upper one), it must:

1. Activate the upper PWM.
2. Do not activate the lower PWM.
3. In the Dead Band Unit tab, configure the lower PWM so that it is the reciprocal of the upper one.

Likewise, a waiting zone can be configured between activations and deactivations of the transistors, to satisfy the on time and dead time, so that the voltage source is not short-circuited (see Section 2.2.5). The upper PWM (ePWMxA) is configured in this way, such that, as long as the internal counter (CMP) does not reach the value of the service factor (introduced by an external signal, the result of the designed control, in the *Counter Compare*), the output is high. The output is low from when the duty factor to the switching period value in clock cycles. The configuration can be seen in Fig. 6.33.

Thus, the appearance of the ePWM block is as shown in Fig. 6.34.

The PWM block works by means of an internal counter. The trigger properties are configured in the *Counter Compare* tab. In this tab, the counter A (CMPA) is configured as in Fig. 6.35. Counters C (CMPC) and D (CMPD) must be configured as counter B (CMPB).

The configuration of the lower ePWM (ePWMxB) is done, firstly, not activating it from the *ePWMB* tab, as in Fig. 6.36. Secondly, accessing the *DeadBand Unit* tab, the dead time option is activated for the lower transistor, so that it is the inverse of the upper PWM. In the BOOSTXL - 3PhGaNInv datasheet it is described that

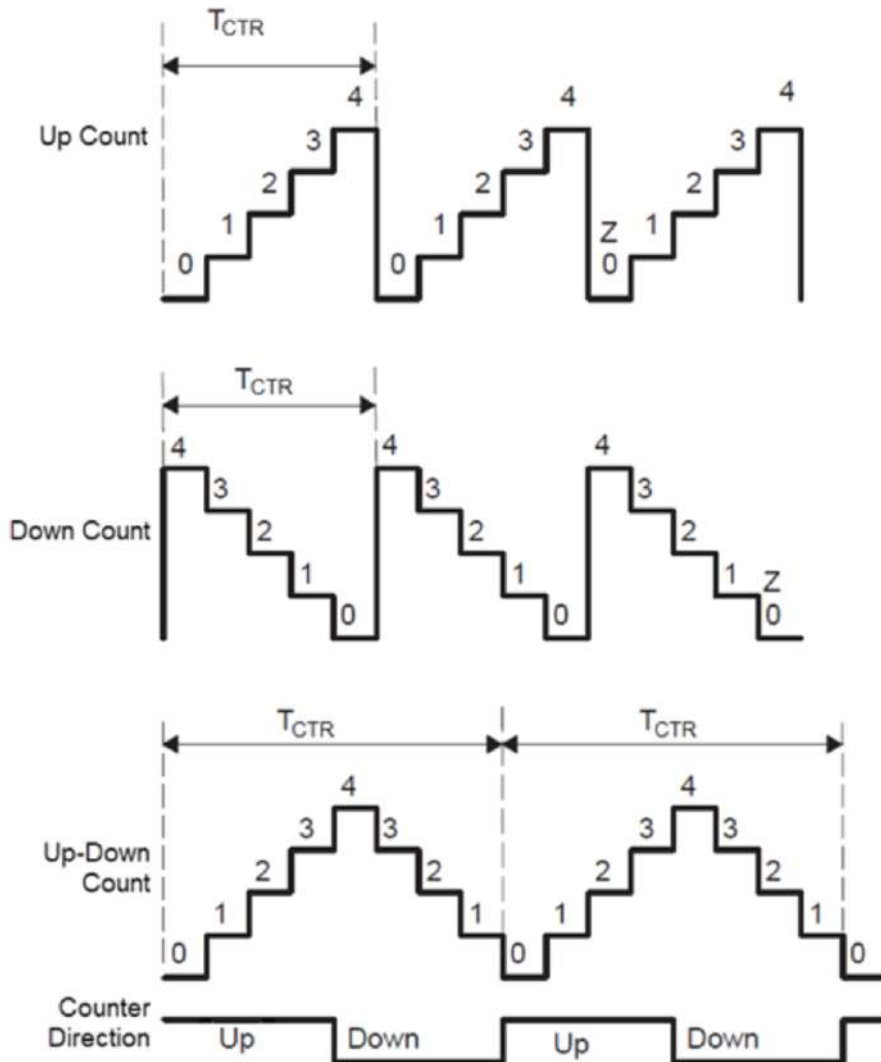


Figure 6.30: PWM generation form options in Simulink's ePWM block [47]

the dead time of gallium nitride semiconductors is 12.5ns, therefore, translating this value into clock cycles, if 5000 clock cycles correspond to a frequency of 10kHz (0.1ms), 12.5ns would correspond to 0.625 clock cycles, and since the minimum value that can be entered is 1, the value is left as 1. Thus, the lower transistor of

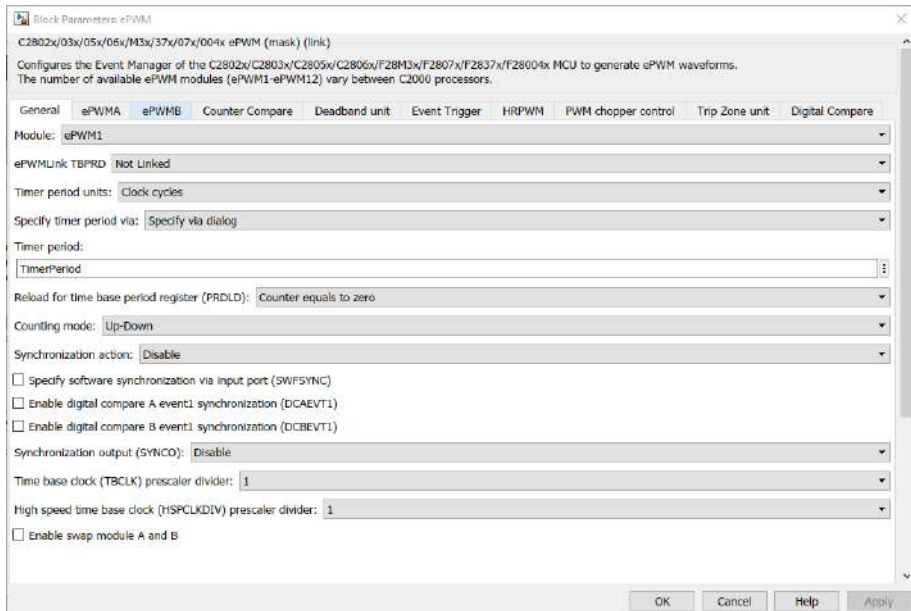


Figure 6.31: Simulink ePWM block. TI C2000 library.

one phase is configured as in Fig. 6.36 and 6.37.

As mentioned in Section 2.4.1 and is reflected in (5.11) of page 38, the ADC is triggered every three PWM switching periods. Accessing the *Event Trigger* tab, we configure the ADC conversion start every three PWM events. We configure each event in such a way that each event (interruption) is activated each switching period, see Fig. 6.38.

The prototype that is designed and built in this project is three-phase. The PWM settings are identical for all three phases.

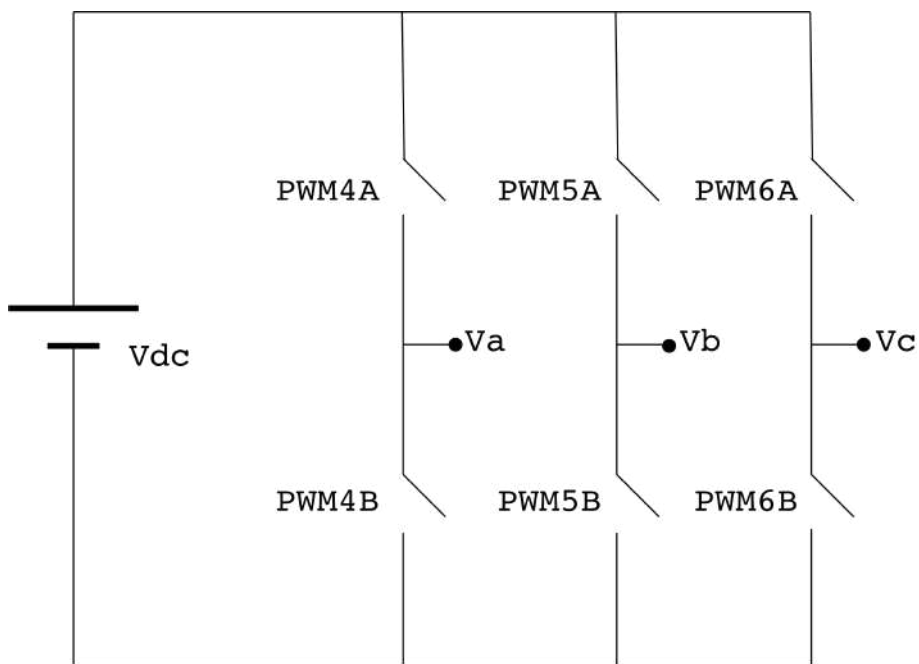


Figure 6.32: Diagram of the PWM nomenclature in Simulink for the VSC connected to the LaunchPad at the bottom (Section 6.2.2).

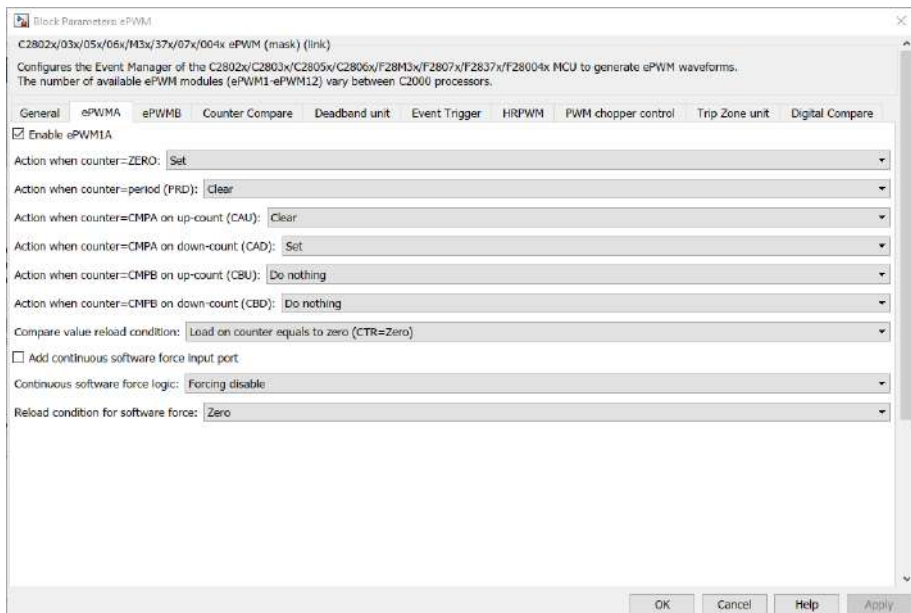


Figure 6.33: ePWMxA configuration.



Figure 6.34: Final appearance of the ePWM Block. The input is the duty cycle in seconds or in clock cycles.

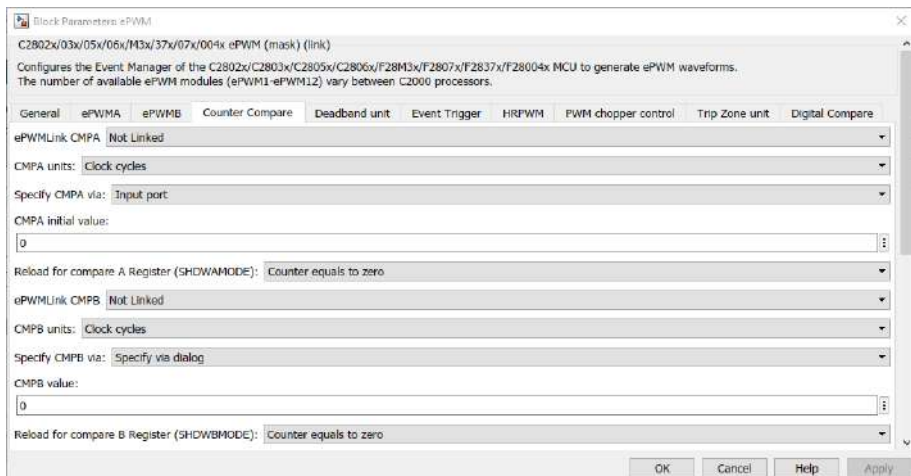


Figure 6.35: *Counter Compare* configuration tab.



Figure 6.36: ePWMxB Configuration.

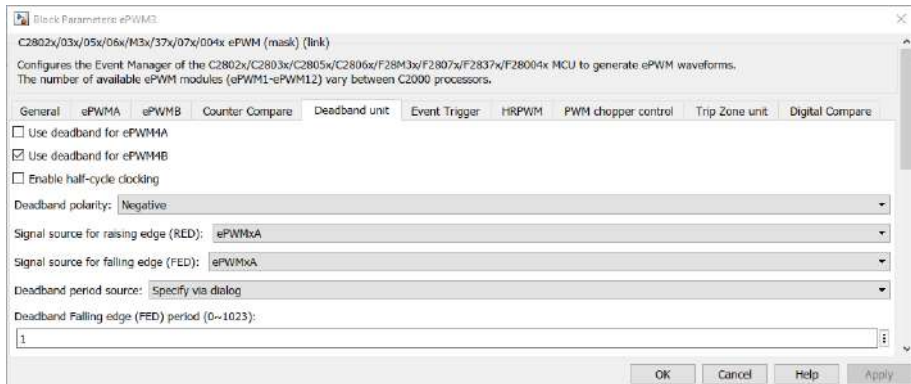


Figure 6.37: ePWMxB configuration from *DeadBand Unit* tab.

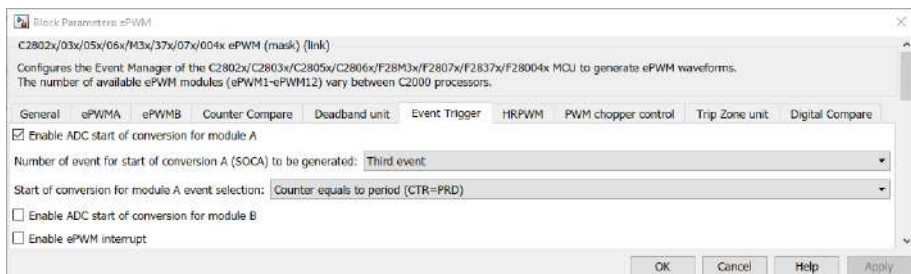


Figure 6.38: Configuration of ADC startup synchronized with ePWM.

These shots do not take effect on the BOOSTXL-3PhGaNInv board until PWM is enabled. The PWMs are activated low on port J2-16 (See Fig. 6.21 on page 73). This pin corresponds to the GPIO 124 or GPIO 26 port of the microcontroller, depending on the PWM that want to be used by the LAUNCHPADXL - F28379D, depending on the relative connection between the microcontroller and the inverter (See Section 6.2.2). In the simulink diagram, the system of Fig. 6.39 is implemented to activate the PWM of the VSC.

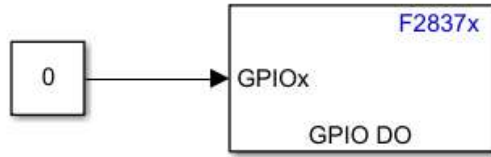


Figure 6.39: VSC PWM enable configuration.

Using the coupling method described in Section 6.2.2 on page 74, the VSC PWMs are activated from GPIO port 26, by docking the VSC at the bottom of the LaunchPad, i.e. the one used in this project.

6.3.2 Output Signals through PWM ports

The LaunchPad F28379D has a total of twelve PWM ports. The BOOSTXL - 3PhGaNInv requires six of them to function properly, allowing the remaining six to be used to output signals from the system. The service factor of the PWMs will be the value of the signal in question that wants to be obtained through the PWM port. Said service factor must be strictly positive and be between 0 and 1. The configuration of the PWM ports, unlike the configuration explained previously, lies in configuring the ePWMxB port as the ePWMxA port; activating the ePWMxB port and using its CMPB counter, as can be seen in Fig. 6.40. The ePWMxA port is configured as before, available in Fig. 6.33.

The internal counter of the ePWMxB will follow a service factor imposed by the internal signal that it wants to be obtained. The *Counter Compare* tab is configured as in Fig. 6.41 and 6.42.

Likewise, in the *Deadband Unit* tab, all options are disabled, as can be seen in Fig. 6.43.

In this way, the ePWM block configured to obtain outputs is shown in Fig. 6.44.



Figure 6.40: *ePWMB* tab configuration to obtain signals by PWM.

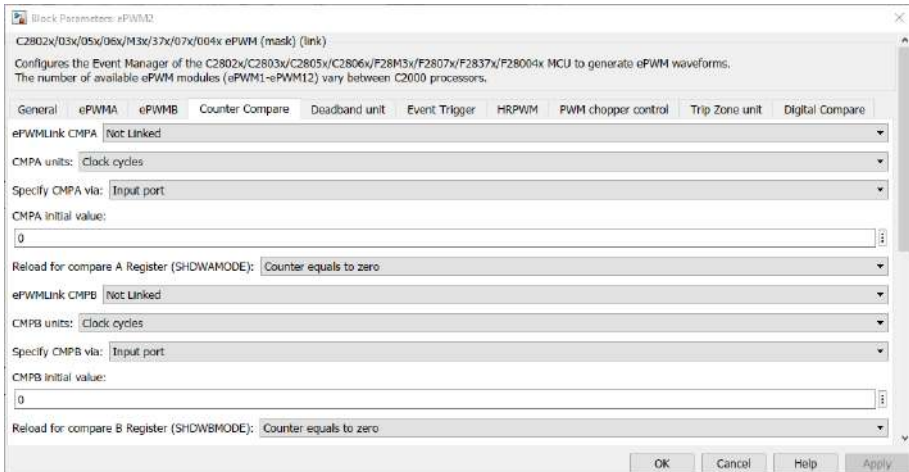


Figure 6.41: Configuration tab *Counter Compare* to obtain signals by PWM. Counters *CMPA* and *CMPB*.

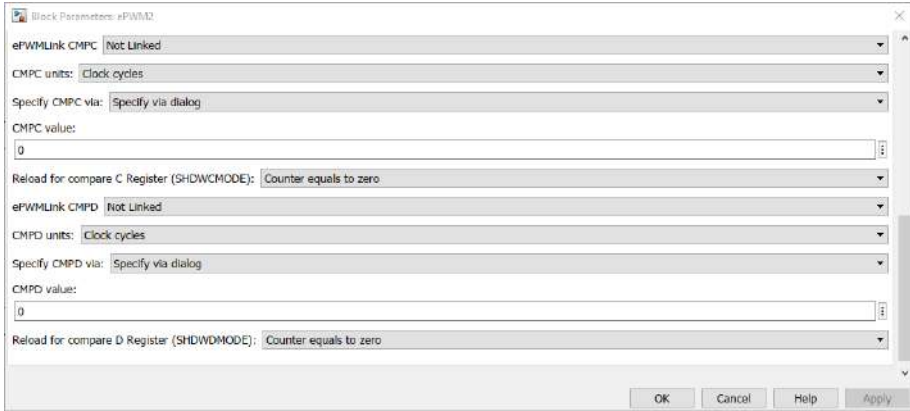


Figure 6.42: Configuration tab *Counter Compare* to obtain signals by PWM. Counters *CMPA* and *CMPB*.

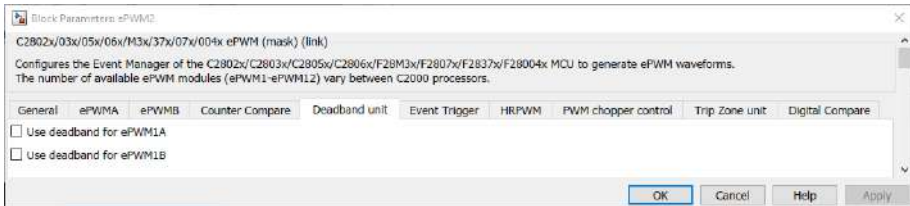


Figure 6.43: *Deadband Unit* tab configuration to obtain signals through PWM.

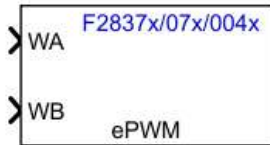


Figure 6.44: Final appearance of the ePWM Block to obtain signals. The input signals are the duty cycle.

6.3.3 ADC Module

In the present project, the ADC converters sample the currents through the phases. The ADC block of the Simulink library for the TMS320F28379D microcontroller is shown in Fig. 6.45.

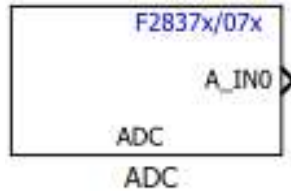


Figure 6.45: Simulink ADC block. TI C2000 Library.

Once the starting conversion procedure from the PWM block has been configured, the ADC block is configured. It has two tabs: *SOC Trigger* and *Input Channels*. In the *SOC Trigger* menu we select the ADC module that will convert the signal. This module must be chosen according to Table 6.1 on page 75.

The ADC resolution, given the way the data is collected, will be 12-bits. The number at the beginning of the ADCs (*SOC Number* will have to be different among all the ADCs (if not, a compilation error will occur). The *SOCx Acquisition Window* is a value between 0 and 15. This value is, proportionally between 0 and 15, the amount of time that the *Sample and Hold* of the ADC remains open reading the measurement (0 not open, 15 fully open). The start of the ADC is caused by an interruption of the PWM of that phase. In this way, Select in the *SOCx trigger source* dialog the *ePWMx ADCSOCA* option (see Fig. 6.38). The ADC does not trigger any action. Likewise, the ADC sampling time will be the same as the rest of the blocks of the program (*Sample Time* = -1), that is, inherited from the Simulink parameter settings (the switching period, see Section 6.3.8). The data type used and that comes by default is *uint16*, because then, the signal will be scaled and corrected in the simulink, due to the fact that this signal is strictly positive. Finally, the configuration of the three ADCs needed to sample the three currents from an triphasic electronic converter, is shown in Fig. 6.46. Finally, select the channel within the ADC module selected in the *Input Channels* tab, listed in Table 6.1 on page 75.

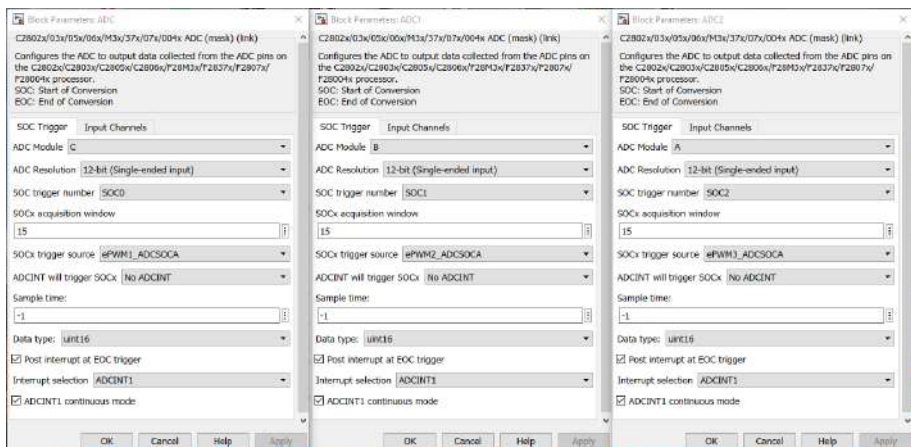


Figure 6.46: ADC Block Configuration.

6.3.4 DAC Module

The DAC module of the microcontroller allows an analog signal to be obtained through a terminal. The Simulink DAC module is the one shown in Fig. 6.47.

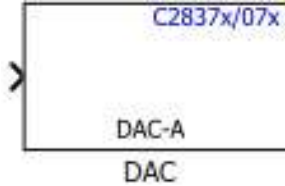


Figure 6.47: DAC simulink block. TI C2000 Library.

Although the DAC configuration allows a choice between DAC-A, DAC-B, or DAC-C, the DAC-C cannot be used on the LaunchPad F28379D. Furthermore in Section 6.1.4 it was explained that one of the DACs would necessarily coincide with the V_{ref} output of the VSC BOOSTXL-3PhGaNIInv. Therefore, only a single DAC will be available. The DAC, by default, uses an average value of 1.5V, so that the output signal is strictly positive.

6.3.5 Difference of obtaining outputs through PWM or DAC

The LAUNCHPADXL F28379D has a total of twelve PWM ports, these can be used to obtain control outputs. The average value of the signal, obtained by the PWM port after being filtered by a low pass filter, will be the value of the analog (proportional) signal that it is desired to be obtained. Also using a filter on the oscilloscope measurement of around 500Hz cut-off frequency, a continuous signal very similar to that of the DAC can be obtained. And if the same 500 Hz filter is applied to the DAC, and to the rest of the measurements, the phase shift of the obtained signals will be constant and the same.

In Fig. 6.48 a graph is shown in which the data from outputting a current (from phase a in this case) is collected, by the PWM module and by the DAC module, obtaining the actual value with a current clamp. It can be verified that the DAC shows a value closer to the real one than the PWM.

However, the phase shift of the PWM output ($610 - 4s$) is less than the DAC output ($7.910 - 3s$). Fig. 6.49 shows the phase current a real, measured with a clamp meter, the output by the DAC and by a PWM. It can be seen that the delay applied by the DAC output is much greater than by the PWM. All measurements were filtered with a 500Hz cut-off frequency low pass filter.

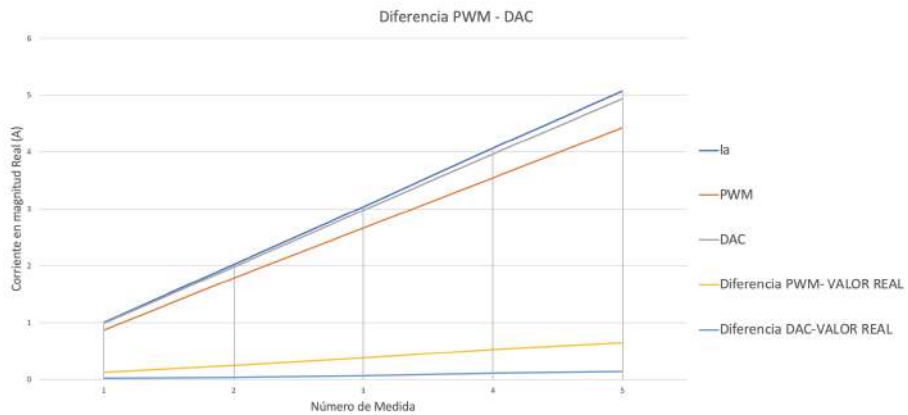


Figure 6.48: Difference in PWM - DAC output measurements.

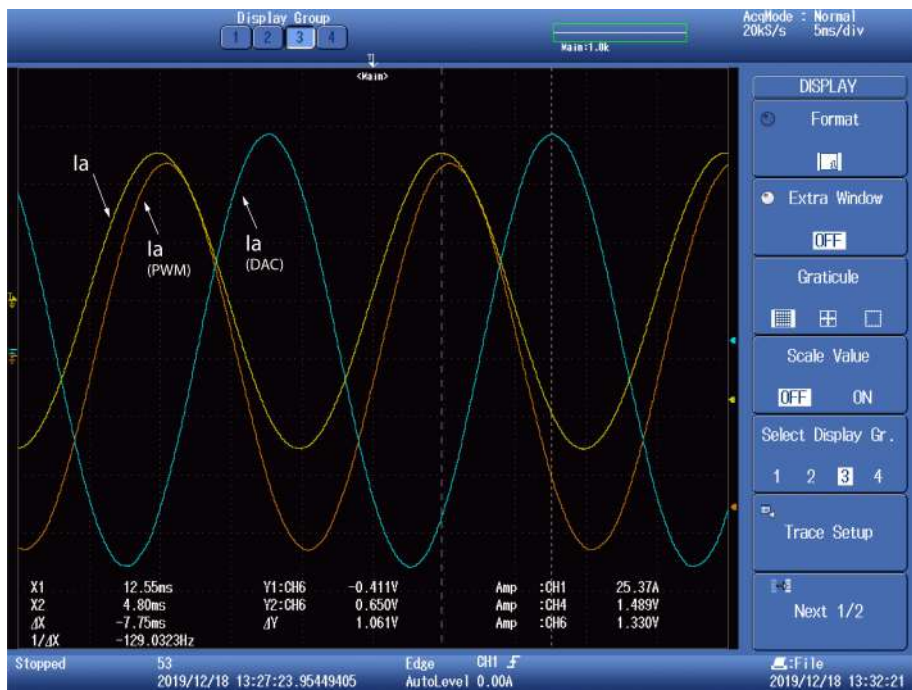


Figure 6.49: Current through one phase, output through the DAC, through the PWM and measured with a current clamp.

6.3.6 Getting a value from a DAC or PWM port

To obtain a signal through a DAC or PWM port, it is necessary to perform a series of operations to ensure that the output signal is displayed correctly.

1. Scale the amplitude of the signal so that it does not exceed the limits of the output. Limits are set between 0V and + 3.3V. Said value to which it is scaled can vary depending on the value of the signal to be obtained.
2. Convert the signal to a value between 0 and 4096. This is done because the Launchpad accepts 12-bit values in *uint* format. Since the most significant bit is the sign bit, this results in a total of $2^{11} = 2048$ values (positive and negative).
3. Set a mean value so that the entire signal is on the positive semi-axis.

A simulink block diagram is shown in Fig. 6.50 that is used in the simulator and prototype software to obtain the output signals.

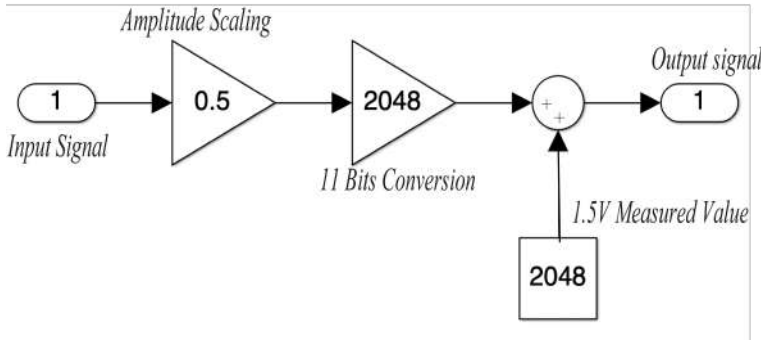


Figure 6.50: Block diagram to obtain a signal through the DAC or PWM. Actual 1.5V average value may vary.

In case the signal saturates in the saturation limits of the output voltage, in this case through a DAC port, a signal could be obtained like the one shown in Fig. 6.51 [53].

6.3.7 Equivalences between oscilloscope value and real value

Obtaining values by the oscilloscope is directly related to the real values of the prototype. The equations show the gains to be applied necessary to obtain the real values from the oscilloscope measurements.

Obtaining the output signal through the DAC or PWM, to obtain the value in *pu*:

$$\text{Value in pu} = \frac{\text{DAC Amplitude in V} - \text{Average DAC Measured Value}^5}{0.75} \quad (6.4)$$

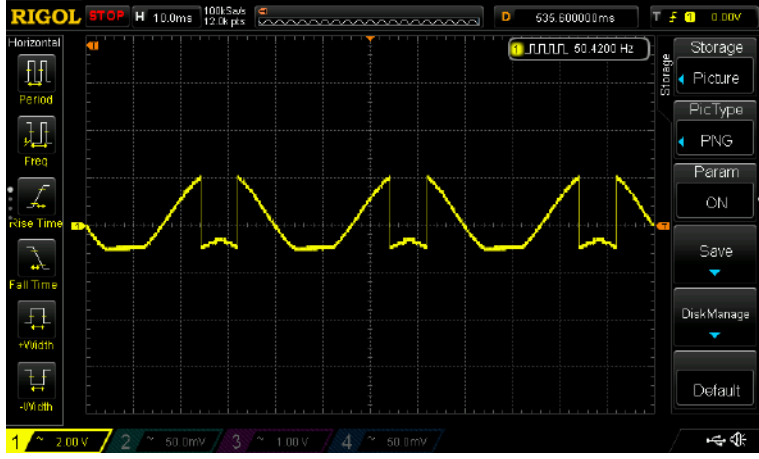


Figure 6.51: Saturated signal through DAC output port.

And to go from a reference in pu in dq axes to real values in currents, knowing that the homothetic constant due to a power-invariant Park transformation is $\sqrt{\frac{3}{2}}$, described in Section 4.2.2:

$$I_a(pu) = I_b(pu) = I_c(pu) = \frac{I_{base}(A) \cdot Id(pu)}{\sqrt{\frac{3}{2}}} \quad (6.5)$$

In this way, if, for example, a d-axis current value of amplitude 1.8V is obtained, a current in pu of:

$$Valueinpu = \frac{1.8 - 1.5}{0.75} = 0.4pu$$

And substituting the previous value in (6.5),

$$I_a(A) = I_b(A) = I_c(A) = \frac{2.5(A) \cdot 0.4(pu)}{\sqrt{\frac{3}{2}}} = 0.81649A$$

⁵This value may vary depending on whether the usb cable is connected to the computer or due to other factors. It is recommended to take the mean value of the signal when the reference is null (and therefore the mean value is constant).

6.3.8 Build program into the Launchpad

Before uploading the code to the LaunchPad, it is necessary that the 48V voltage source is not supplying voltage to the LaunchPad. Otherwise, the code will not be debugged and an error will appear. It will be necessary to determine a value for the *Fixed-Step* of the *Solver* tab. This value must be the smallest time value used in the system. In the prototype, the smallest is the switching period of the PWMs. Therefore, the *Fixed-Step* of the program will be the switching period. In Fig. 6.52, the value of the variable T_{pwm} is $0.1ms$.

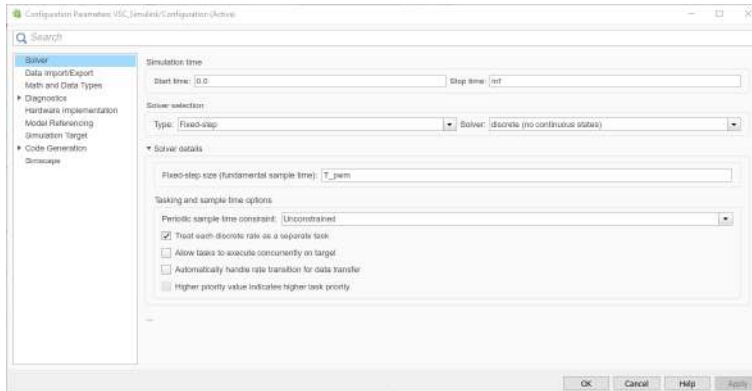


Figure 6.52: *Solver* tab configuration for prototype implementation.

In order to upload the program to the LaunchPad, it is necessary to tell the program which is the LaunchPad. This is done in the *Configuration Properties* > *Hardware Implementation* settings. In this window, the LaunchPad is selected. In this project, it is the *TI Delfino LaunchPad F28379D*.

It is recommended not to load the program into the flash memory of the LaunchPad. In this way, when the power supply is removed from the voltage source and the USB port, the execution stops and when the microcontroller is powered again, the program is not executed. In Fig. 6.53 the configuration of the *Hardware Implementation* tab is shown. It is also recommended to verify that the *Code Generation* tab uses the Texas Instruments compiler in the *Toolchain* dialog box. This compiler must be installed when the *Embebbed Coder Support Package for Texas Instruments C2000 Processors* package is installed, explained in Appendix A.

After the previous steps are completed, click on 'Deploy to Hardware 'or' Build, Deploy & Start' (depending on the Matlab version installed).

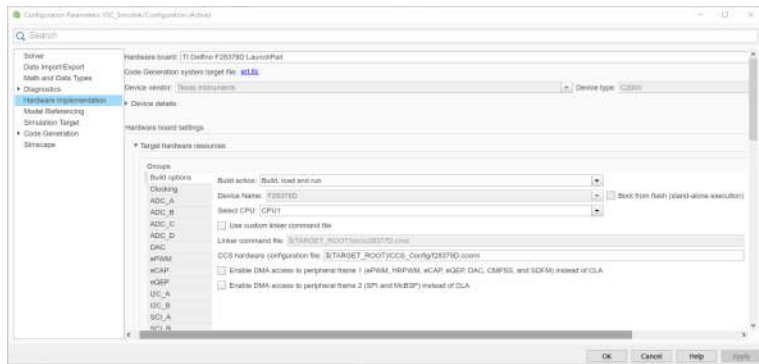


Figure 6.53: *Hardware Implementation* configuration tab for prototype implementation.

Chapter 7

Experimental Results

The working environment is shown in the Fig. 7.1.

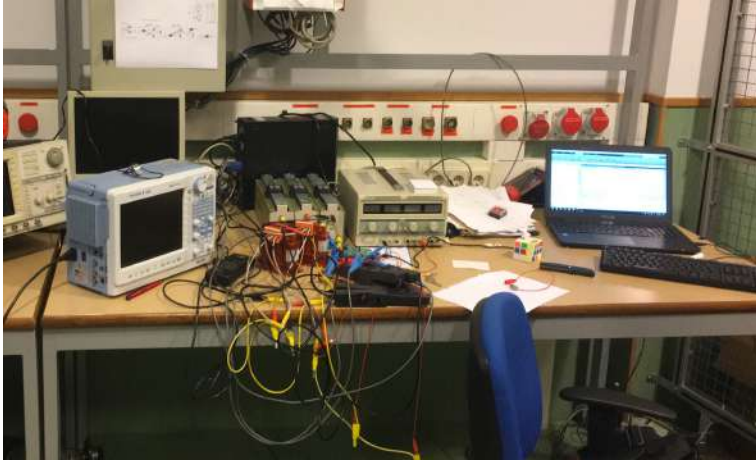


Figure 7.1: Laboratory set up.

Below are snapshots taken on the oscilloscope [54]. All tests were carried out with a supply voltage of 48V.

Fig. 7.2 displays the waveform of three-phase currents in a normal operating mode (rated current through the coils).

In the case that the VSC cannot supply a current or service factor necessary to reach the values needed by the control, it saturates the control integral term and the waveforms obtained are those shown in Fig. 7.3.

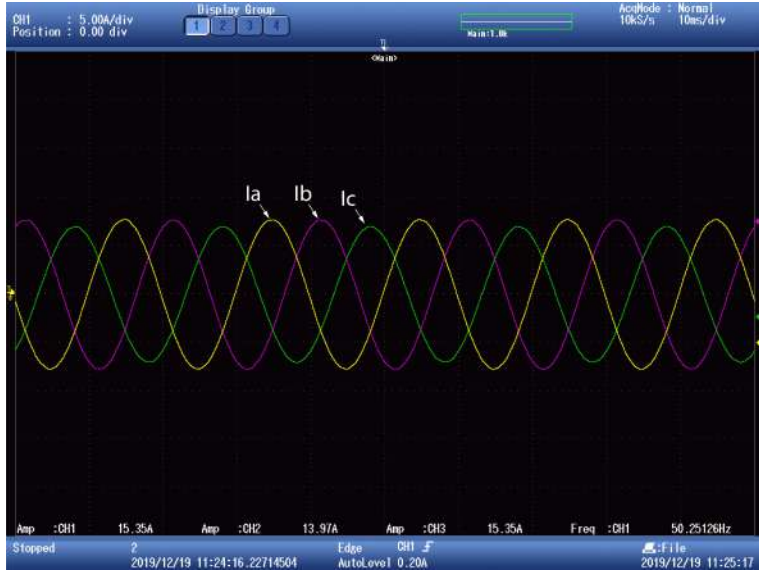


Figure 7.2: Three-phase currents through the coils. The identification of the signals is shown in detail in the figure. See table 7.1.

| Channel | Colour | Element | Amplitude (V) | N ^o Turns | Current (A) |
|---------|--------|---------|---------------|----------------------|-------------|
| 1 | Yellow | I_a | 15.35 | 5 | 3.07 |
| 2 | Green | I_b | 13.97 | 5 | 2.794 |
| 3 | Pink | I_c | 15.35 | 5 | 3.07 |

Table 7.1: Table with the data in Fig. 7.2 with the three-phase currents.

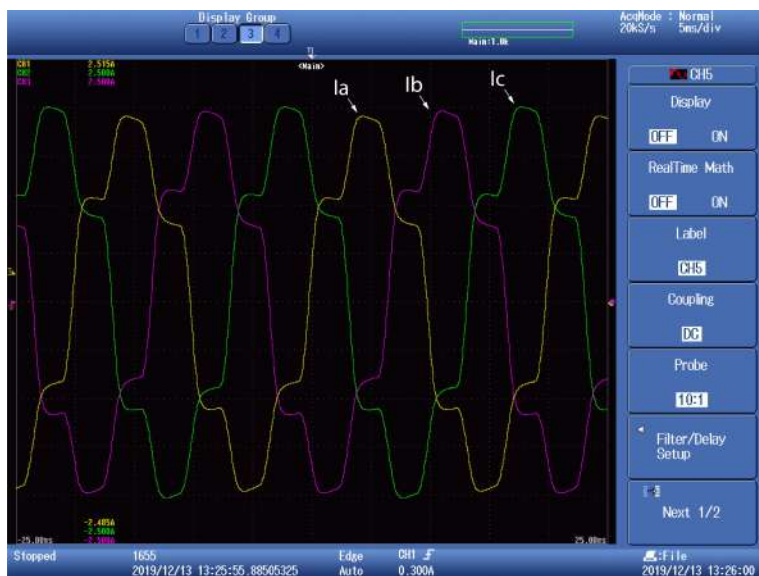


Figure 7.3: Three-phase currents through the saturated coils.

Fig. 7.4 shows an integral control with a pulse-shaped d-axis reference.



Figure 7.4: Three-phase currents through the inductances with d-axis reference current in the form of pulses. Reference $I_q = 0$. The identification of the signals is shown in detail in the figure.

Fig. 7.5 shows an integral control with a 4Hz sinusoidal d-axis reference. Fig. 7.6 shows in detail the zone of change from positive to negative reference of Fig. 7.5.

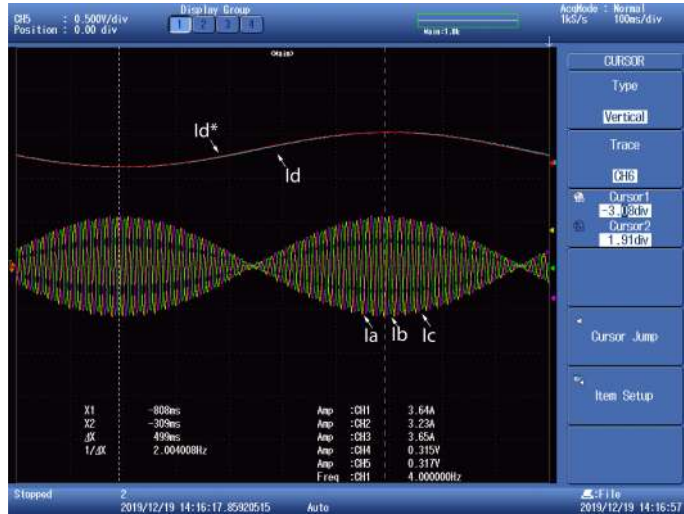


Figure 7.5: Three-phase currents through the inductances with d-axis current reference in sinusoidal shape of 4Hz frequency. The identification of the signals are shown in detail in the figure. I_q reference = 0.

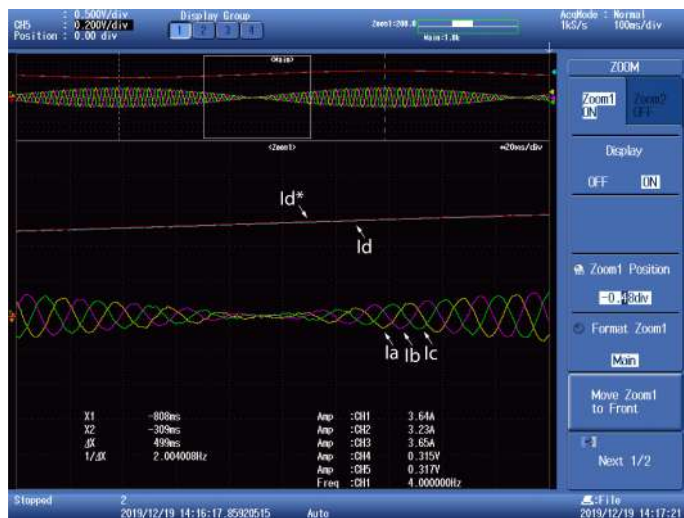


Figure 7.6: Detail of the three-phase currents through the coils with a 4Hz frequency sinusoidal d-axis current reference. The identification of the signals are shown in detail in the figure. I_q reference = 0.

Fig. 7.7 shows the reference tracking temporal response I_d and I_q at the same time. It should be noted that the I_d^* , I_q^* (references) and I_d signals were obtained through a PWM port, while the I_q output was obtained through the DAC port, thus, what is commented in Section 6.3.5 is observed, the DAC port lags more than the PWM, as can be seen in Fig. 7.7. Likewise, the waveform of the currents can be observed when a rising step is applied. The three-phase current shapes appear noisy because the inductances work far below their nominal current (5A).

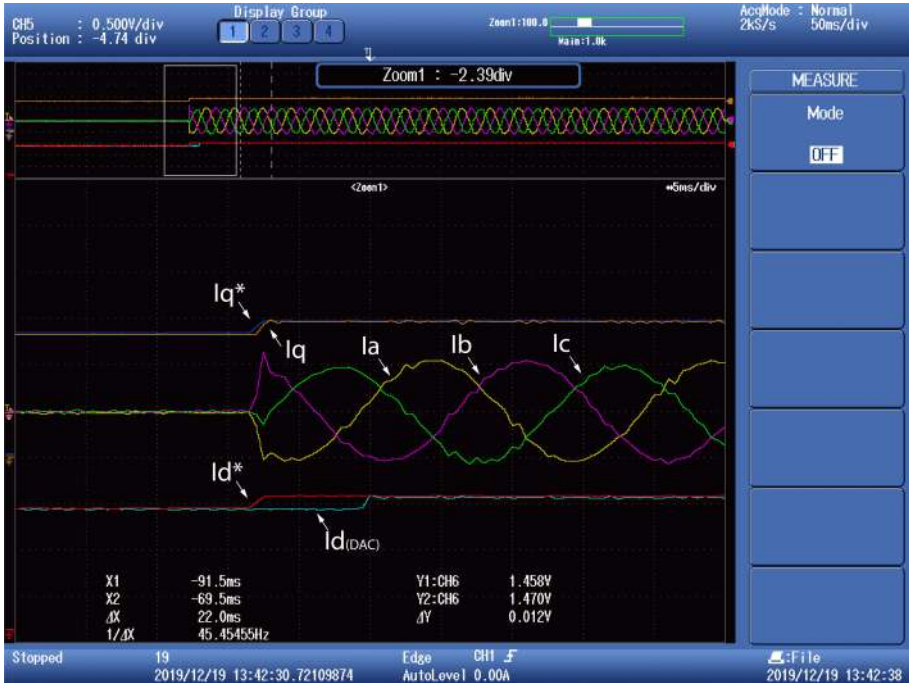


Figure 7.7: Three-phase currents through the inductances. Time response of d-axis and q-axis current. Application of references simultaneously. The identification of the signals are shown in detail in the figure.

Fig. 7.8 shows a PI control to which a rising step and a falling step are applied simultaneously on d and q axes. A complex waveform appears at the point of application of the references.

Fig. 7.9 shows the q-axis current reference tracking time response.

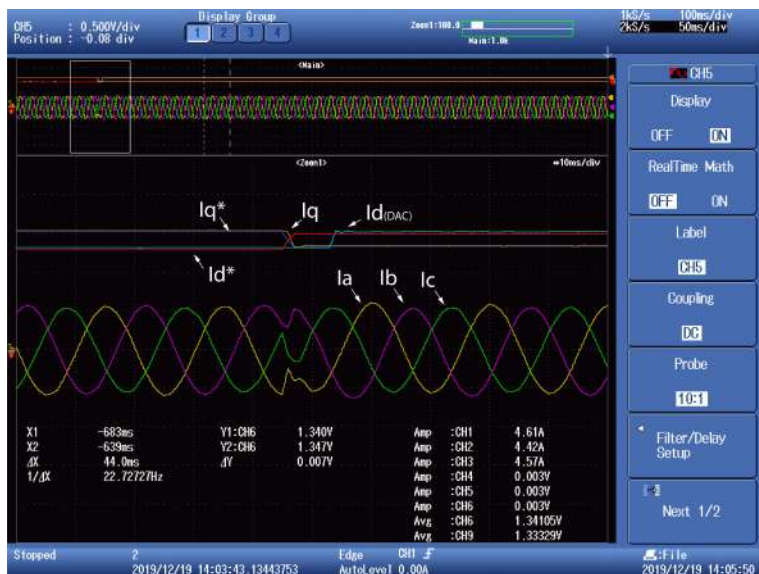


Figure 7.8: Three-phase currents through the inductances. Time response of d-axis and q-axis current inversely applied. The identification of the signals are shown in detail in the figure.

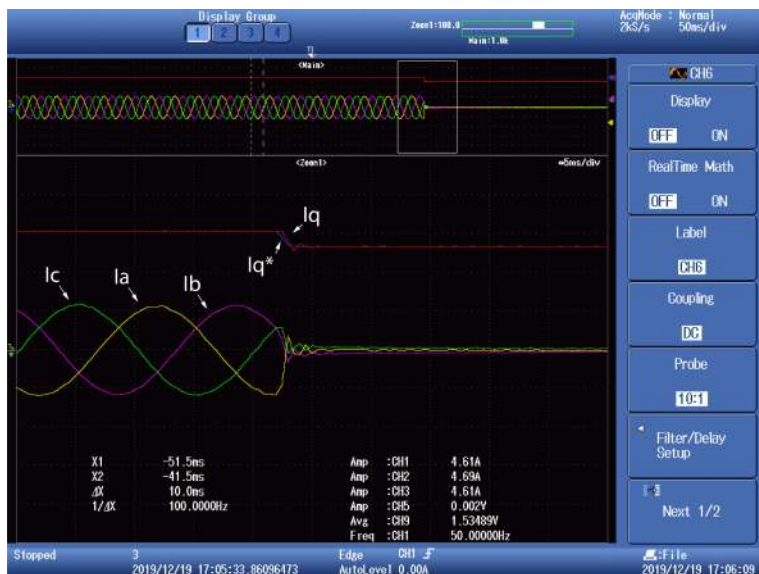


Figure 7.9: Three-phase currents through the inductances. Time response of q-axis current. I_d reference = 0. The identification of the signals are shown in detail in the figure.

Chapter 8

Budget

The cost of the prototype is detailed in Table 8.1. This does not include the materials used for the encapsulation of the system, since it is not required for the correct functioning of the prototype.

8.1 Materials Budget

| Element | Quantity (<i>u</i>) | Unitary Cost (<i>Euros/u</i>) | Cost (<i>Euros</i>) |
|---|--------------------------|------------------------------------|--------------------------|
| LaunchPad F28379D ⁵ | 1 | 34.79 ⁶ | 34.79 |
| BOOSTXL - 3PhGaNInv ₅ | 1 | 50.45 ⁶ | 50.45 |
| Three-Phase Inductance RTLX5. ⁷ | 1 | 65.10 ⁸ | 65.10 |
| Total Materials | - | - | 150.05 |

Table 8.1: Table with the budget of the prototype materials.

¹Manufacturer: *Texas Instruments*.

²Distributor: *Mouser Electronics*; <https://www.mouser.es/>. Last access: 08/01/2020

³Manufacturer: *Polylux*.

⁴Distributor: *Trafo Direct* ; <https://trafo-direct.es>. Last Access: 08/01/2020

⁵Manufacturer: *Texas Instruments*.

⁶Distributor: *Mouser Electronics*; <https://www.mouser.es/>. Last access: 08/01/2020

⁷Manufacturer: *Polylux*.

⁸Distributor: *Trafo Direct* ; <https://trafo-direct.es>. Last Access: 08/01/2020

8.2 Labor Budget

The cost of labor for the development of the project and prototype is detailed in Table 8.2.

| Labor kind | Quantity (<i>h</i>) | Unitary Cost (<i>Euros/h</i>) | Cost (<i>Euros</i>) |
|-------------------------|--------------------------|------------------------------------|--------------------------|
| Engineerl | 150 | 36.00 | 5400 |
| Supervision | 30 | 90 | 2700 |
| Total workforce cost | - | - | 8100 |

Table 8.2: Table with the budget of labor for the construction of the prototype.

8.3 Total Budget

The total budget is detailed in Table 8.3.

| Concept | Cost (<i>Euros</i>) |
|--------------|-----------------------|
| Materials | 150.05 |
| Engineering | 8100 |
| Total budget | 8250.05 |

Table 8.3: Table with the prototype construction budget.

Bibliography

- [1] H. Chen, H. Leng, H. Tang, J. Zhu, H. Gong, and H. Zhong, "Research on model management method for micro-grid," in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (IT-NEC)*, Dec. 2017, pp. 163–166. DOI: [10.1109/ITNEC.2017.8284930](https://doi.org/10.1109/ITNEC.2017.8284930).
- [2] W. Kohn, Z. B. Zabinsky, and A. Nerode, "A micro-grid distributed intelligent control and management system," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2964–2974, Nov. 2015, ISSN: 1949-3061. DOI: [10.1109/TSG.2015.2455512](https://doi.org/10.1109/TSG.2015.2455512).
- [3] M. Gujar, A. Datta, and P. Mohanty, "Smart mini grid: An innovative distributed generation based energy system," in *2013 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, Nov. 2013, pp. 1–5. DOI: [10.1109/ISGT-Asia.2013.6698768](https://doi.org/10.1109/ISGT-Asia.2013.6698768).
- [4] R. M. Kotecha, Y. Zhang, A. Rashid, N. Zhu, T. Vrotsos, and H. A. Mantooth, "A physics-based compact gallium nitride power semiconductor device model for advanced power electronics design," in *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, Mar. 2017, pp. 2685–2691. DOI: [10.1109/APEC.2017.7931078](https://doi.org/10.1109/APEC.2017.7931078).
- [5] I. Omura, W. Saito, T. Domon, and K. Tsuda, "Gallium nitride power hemt for high switching frequency power electronics," in *2007 International Workshop on Physics of Semiconductor Devices*, Dec. 2007, pp. 781–786. DOI: [10.1109/IWPSD.2007.4472634](https://doi.org/10.1109/IWPSD.2007.4472634).
- [6] R. R. Duarte, G. G. Pereira, M. A. Dalla Costal, C. H. Barriquello, and J. M. Alonso, "Off-grid commercial led driver optimization using gan transistors," in *2018 IEEE Industry Applications Society Annual Meeting (IAS)*, Sep. 2018, pp. 1–7. DOI: [10.1109/IAS.2018.8544686](https://doi.org/10.1109/IAS.2018.8544686).
- [7] C. Sagan and D. Udina, *El mundo y sus demonios: la ciencia como una luz en la oscuridad*, ser. Colección la línea del horizonte. Planeta, 1997, ISBN: 9788408020431. [Online]. Available: <https://books.google.es/books?id=R8rYAAACAAJ>.
- [8] M. Rashid, *Electrónica de Potencia: Circuitos, Dispositivos y Aplicaciones*, 2nd ed. Prentice Hall, 1995, ISBN: 968-880-586-6.

- [9] S. K. Sahoo, A. Ramulu, S. Batta, and S. Duggal, "Performance analysis and simulation of three phase voltage source inverter using basic pwm techniques," in *IET Chennai 3rd International on Sustainable Energy and Intelligent Systems (SEISCON 2012)*, Dec. 2012, pp. 1–7. DOI: [10.1049/cp.2012.2223](https://doi.org/10.1049/cp.2012.2223).
- [10] T. Lifang, "Study of the svpwm converter based on tms320f24x," in *2013 Third International Conference on Intelligent System Design and Engineering Applications*, Jan. 2013, pp. 1316–1319. DOI: [10.1109/ISDEA.2012.311](https://doi.org/10.1109/ISDEA.2012.311).
- [11] S. Singh and A. Tiwari, "Simulation and comparison of spwm and svpwm control for two level inverter," Mar. 2017.
- [12] B. J. Baliga, "Power semiconductor device figure of merit for high-frequency applications," *IEEE Electron Device Letters*, vol. 10, no. 10, pp. 455–457, Oct. 1989, ISSN: 1558-0563. DOI: [10.1109/55.43098](https://doi.org/10.1109/55.43098).
- [13] L. Boylestad Robert L. y Nashelsky, *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*, 10th ed. Pearson Education, 2009, ISBN: 978-607-442-292-4.
- [14] E. C. Niehenke, "The evolution of transistors for power amplifiers: 1947 to today," in *2015 IEEE MTT-S International Microwave Symposium*, May 2015, pp. 1–4. DOI: [10.1109/MWSYM.2015.7166768](https://doi.org/10.1109/MWSYM.2015.7166768).
- [15] U. Müter, C. Sammler, S. Fahlbusch, S. Klötzer, and K. F. Hoffmann, "Comparison of driving concepts for silicon carbide bipolar junction transistors," in *2016 18th European Conference on Power Electronics and Applications (EPE'16 ECCE Europe)*, Sep. 2016, pp. 1–9. DOI: [10.1109/EPE.2016.7695636](https://doi.org/10.1109/EPE.2016.7695636).
- [16] A. Mohapatra, S. Bhawal, and K. Hatua, "A si igt and sic mosfet hybrid shunt active filter," in *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, Dec. 2018, pp. 1–6. DOI: [10.1109/PEDES.2018.8707561](https://doi.org/10.1109/PEDES.2018.8707561).
- [17] P. Wellmann, "Power electronic semiconductor materials for automotive and energy saving applications – sic, gan, ga2o3, and diamond," *Zeitschrift für anorganische und allgemeine Chemie*, vol. 643, Oct. 2017. DOI: [10.1002/zaac.201700270](https://doi.org/10.1002/zaac.201700270).
- [18] Tiago Kommers Jappe, S. A. Mussa, and Richard Henry Sutter Rosendo, "Synchronous state machine inner fpga controlling pfc boost converter," in *2010 IEEE International Symposium on Industrial Electronics*, Jul. 2010, pp. 1097–1102. DOI: [10.1109/ISIE.2010.5636865](https://doi.org/10.1109/ISIE.2010.5636865).
- [19] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodríguez, "Control of power converters in ac microgrids," *IEEE Transactions on Power Electronics*, vol. 27, no. 11, pp. 4734–4749, Nov. 2012, ISSN: 1941-0107. DOI: [10.1109/TPEL.2012.2199334](https://doi.org/10.1109/TPEL.2012.2199334).

- [20] Thanh-Vu Tran, Tae-Won Chun, Hong-Hee Lee, Heung-Geun Kim, and Eui-Cheol Nho, "Control for grid-connected and stand-alone operations of three-phase grid-connected inverter," in *2012 International Conference on Renewable Energy Research and Applications (ICRERA)*, Nov. 2012, pp. 1–5. DOI: [10.1109/ICRERA.2012.6477348](https://doi.org/10.1109/ICRERA.2012.6477348).
- [21] S. Reichert, G. Griepentrog, and B. Stickan, "Comparison between grid-feeding and grid-supporting inverters regarding power quality," in *2017 IEEE 8th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Apr. 2017, pp. 1–4. DOI: [10.1109/PEDG.2017.7972536](https://doi.org/10.1109/PEDG.2017.7972536).
- [22] J. Liu, "Studies on improving dynamic performance of microgrids by applying virtual synchronous generator control to distributed generators," Ph.D. dissertation, Osaka University, Jan. 2016, p. 2.
- [23] B. Arbab-Zavar, E. Palacios-Garcia, J. Vasquez, and J. Guerrero, "Smart inverters for microgrid applications: A review," *Energies*, vol. 12, no. 840, p. 11, Mar. 2019.
- [24] H. Niu, M. Jiang, D. Zhang, and J. Fletcher, "Autonomous micro-grid operation by employing weak droop control and pq control," in *2014 Australasian Universities Power Engineering Conference (AUPEC)*, Sep. 2014, pp. 1–5. DOI: [10.1109/AUPEC.2014.6966519](https://doi.org/10.1109/AUPEC.2014.6966519).
- [25] A. Rizqiawan, P. Hadi, and G. Fujita, "Development of grid-connected inverter experiment modules for microgrid learning," *Energies*, vol. 12, no. 476, pp. 2–15, Feb. 2019. DOI: [10.3390/en12030476](https://doi.org/10.3390/en12030476).
- [26] L. Yu, R. Li, and L. Xu, "Distributed pll-based control of offshore wind turbines connected with diode-rectifier-based hvdc systems," *IEEE Transactions on Power Delivery*, vol. 33, no. 3, pp. 1328–1336, Jun. 2018, ISSN: 1937-4208. DOI: [10.1109/TPWRD.2017.2772342](https://doi.org/10.1109/TPWRD.2017.2772342).
- [27] B. Meersman, J. De Kooning, T. Vandoorn, L. Degroote, B. Renders, and L. Vandevelde, "Overview of pll methods for distributed generation units," in *45th International Universities Power Engineering Conference UPEC2010*, Aug. 2010, pp. 1–6.
- [28] A. H. Fahad and M. S. Reza, "Single-phase shunt active power filter using parabolic pwm for current control," in *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, Aug. 2019, pp. 134–138. DOI: [10.1109/SEGE.2019.8859868](https://doi.org/10.1109/SEGE.2019.8859868).
- [29] N. Panten, N. Hoffmann, and F. W. Fuchs, "Finite control set model predictive current control for grid-connected voltage-source converters with lcl filters: A study based on different state feedbacks," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 5189–5200, Jul. 2016, ISSN: 1941-0107. DOI: [10.1109/TPEL.2015.2478862](https://doi.org/10.1109/TPEL.2015.2478862).

- [30] X. Bao, F. Zhuo, Y. Tian, and P. Tan, "Simplified feedback linearization control of three-phase photovoltaic inverter with an lcl filter," *IEEE Transactions on Power Electronics*, vol. 28, no. 6, pp. 2739–2752, Jun. 2013, ISSN: 1941-0107. DOI: [10.1109/TPEL.2012.2225076](https://doi.org/10.1109/TPEL.2012.2225076).
- [31] N. He, D. Xu, Y. Zhu, J. Zhang, G. Shen, Y. Zhang, J. Ma, and C. Liu, "Weighted average current control in a three-phase grid inverter with an lcl filter," *IEEE Transactions on Power Electronics*, vol. 28, no. 6, pp. 2785–2797, Jun. 2013, ISSN: 1941-0107. DOI: [10.1109/TPEL.2012.2219322](https://doi.org/10.1109/TPEL.2012.2219322).
- [32] H. Niu, M. Jiang, D. Zhang, and J. Fletcher, "Autonomous micro-grid operation by employing weak droop control and pq control," in *2014 Australasian Universities Power Engineering Conference (AUPEC)*, Sep. 2014, pp. 1–5. DOI: [10.1109/AUPEC.2014.6966519](https://doi.org/10.1109/AUPEC.2014.6966519).
- [33] C. Jian-ming, "Study on traffic flow simulation model based on kirchhoff's law," in *2010 WASE International Conference on Information Engineering*, vol. 3, Aug. 2010, pp. 418–421. DOI: [10.1109/ICIE.2010.277](https://doi.org/10.1109/ICIE.2010.277).
- [34] C. J. O'Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley, "A geometric interpretation of reference frames and transformations: Dq0, clarke, and park," *IEEE Transactions on Energy Conversion*, vol. 34, no. 4, pp. 2070–2083, Dec. 2019, ISSN: 1558-0059. DOI: [10.1109/TEC.2019.2941175](https://doi.org/10.1109/TEC.2019.2941175).
- [35] P. Krause, O. Wasynczuk, and S. Sudhoff. John Wiley and Sons, Ltd, 2013, 632 pp., ISBN: 9781118524336.
- [36] M. Murali, A. Gokhale, A. V. Pandey, and E. Sharma, "Modelling, design and comparison of pi and pid controllers for static synchronous compensator (statcom)," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Jul. 2016, pp. 1–6. DOI: [10.1109/ICPEICES.2016.7853563](https://doi.org/10.1109/ICPEICES.2016.7853563).
- [37] T. Suyata and S. Po-Ngam, "Simplified active power and reactive power control with mppt for three-phase grid-connected photovoltaic inverters," in *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, May 2014, pp. 1–4. DOI: [10.1109/ECTICon.2014.6839804](https://doi.org/10.1109/ECTICon.2014.6839804).
- [38] X. Shen, J. Zheng, S. Zhu, and Lian Shu, "D-q axis decoupling parameter identification strategy for the grid-connected inverter of photovoltaic generation system," in *2012 China International Conference on Electricity Distribution*, Sep. 2012, pp. 1–4. DOI: [10.1109/CICED.2012.6508671](https://doi.org/10.1109/CICED.2012.6508671).
- [39] A. Jana, P. K. Gayen, P. K. Dhara, and R. Garai, "Simultaneous active and reactive power control of single-phase grid connected battery storage system," in *2017 Devices for Integrated Circuit (DevIC)*, Mar. 2017, pp. 289–293. DOI: [10.1109/DEVIC.2017.8073954](https://doi.org/10.1109/DEVIC.2017.8073954).
- [40] *Tms320f2837xd dual-core delfino microcontrollers*, English, Texas Instruments, 222 pp., December, 2013.

- [41] S. Bennett, "Development of the pid controller," *IEEE Control Systems Magazine*, vol. 13, no. 6, pp. 58–62, Dec. 1993, ISSN: 1941-000X. DOI: [10.1109/37.248006](https://doi.org/10.1109/37.248006).
- [42] L. Balasevicius, A. Kunickaite, and V. S. Janusevicius, "Discrete-time pid controller design in programmable logical controllers," in *2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Sep. 2007, pp. 86–90. DOI: [10.1109/IDAACS.2007.4488380](https://doi.org/10.1109/IDAACS.2007.4488380).
- [43] P. Papamichalis, *Introduction to the TMS320 Family of Digital Signal Processors*. CRC Press LLC, 1999, ISBN: 978-607-442-292-4.
- [44] *Tms320 dsp development support reference guide*, English, Texas Instruments, 430 pp., May, 1998.
- [45] *Launchxl-f28379d overview*, English, Texas Instruments, 30 pp., August, 2016.
- [46] *Meet the tms320f28379d launchpad development kit*, English, Texas Instruments, 2017, 7 pp., 2017.
- [47] MATLAB, *version 9.6.0.1214997 (R2019a)*. Natick, Massachusetts: The MathWorks Inc., 2020.
- [48] *Boostxl-3phganinv evaluation module*, English, Texas Instruments, 41 pp., June, 2017.
- [49] *Lmg5200 80-v, 10-a gan half-bridge power stage*, English, Texas Instruments, 29 pp., March, 2015.
- [50] *Using the lmg5200: Gan half-bridge power stage evm*, English, Texas Instruments, 41 pp., March, 2015.
- [51] *Ina240 high- and low-side, bidirectional, zero-drift, current-sense amplifier with enhanced pwm rejection*, English, Texas Instruments, 41 pp., July, 2016.
- [52] *48-v three-phase inverter with shunt-based in-line motor phase current sensing reference design*, English, Texas Instruments, 41 pp., November, 2016.
- [53] *Oscilloscopios digitales de la serie ds1000d/e*, Spanish, version UGA07107-1110, RIGOL, 168 pp.
- [54] *Dl850e/dl850ev scopecorder user's manual*, English, 7th, version IM DL850E-02EN, Yokogawa, 210 pp.

Appendix A

Installing the TI C2000 Package for Matlab

Installation of the *Embedded Coder Support Package for Texas Instruments C2000 Processors* requires the following system requirements:

- Have Matlab R2017a or later versions installed, although there are versions of this package that can be run in Matlab from versions 2014a to 2019b.
- The operating system must be Windows, it is not available for MacOS or Linux.

In the pop-up window that appears, find the package: *Embedded Coder Support Package for Texas Instruments C2000 Processors*. In Fig. A.1 the main screen of the package to be installed is shown.

The installation of the package for *C2000 Microcontrollers* requires the previous Matlab packages been installed:

- Simulink.
- Embedded Coder.
- Matlab Coder.
- Simulink Coder.

Fig. A.2, A.3 and A.4 show the following screens in setup. In all the necessary ones, click on *Continue* and *Setup Now*.

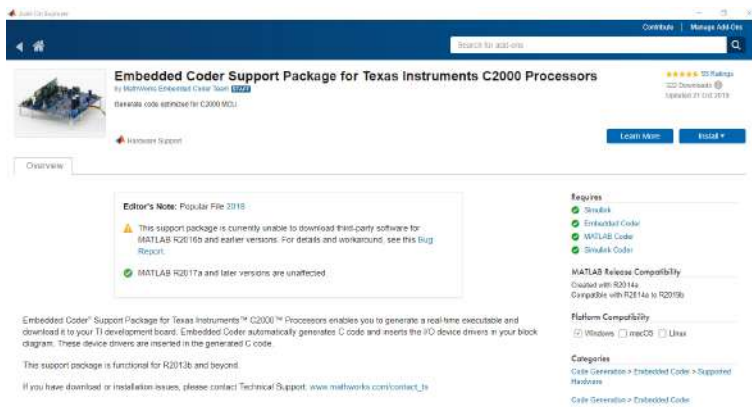


Figure A.1: Matlab Add-Ons menu.

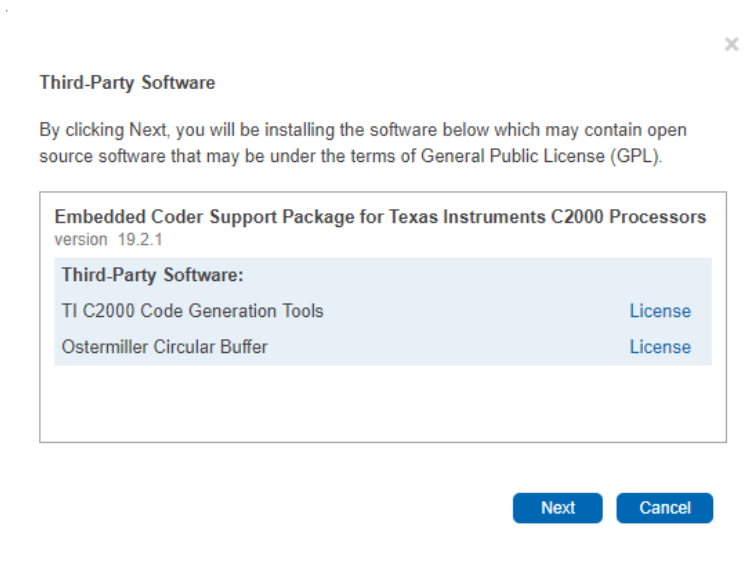


Figure A.2: Library Configuration window 1 *C2000 Processors*.

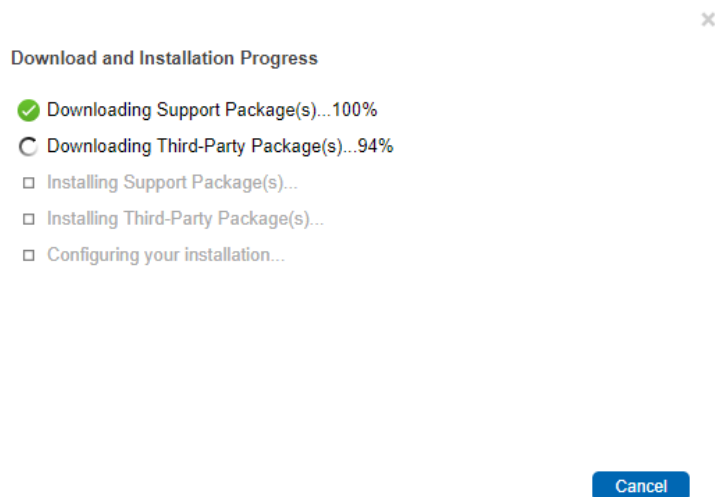


Figure A.3: Library Configuration window 2 *C2000 Processors*.

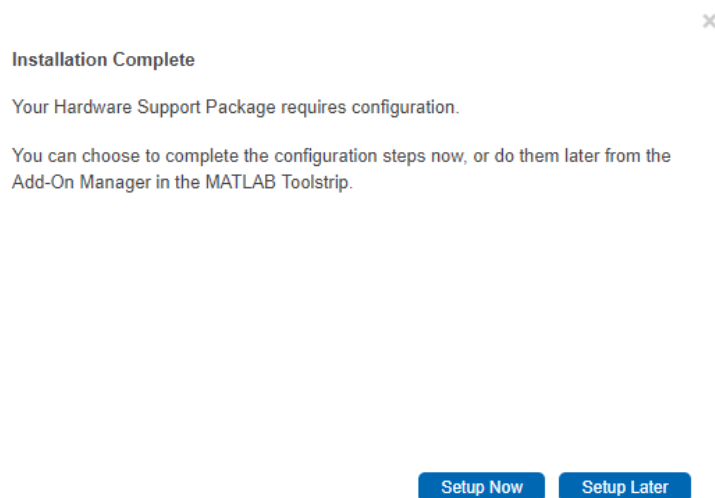


Figure A.4: Library Configuration window 3 *C2000 Processors*.

In Fig. A.5 the screen in which you have to choose the LaunchPad processor of the prototype (or failing that, the one you want to use) is shown, in this case, it is installed for the processor *TI Delfino F2837xD*.

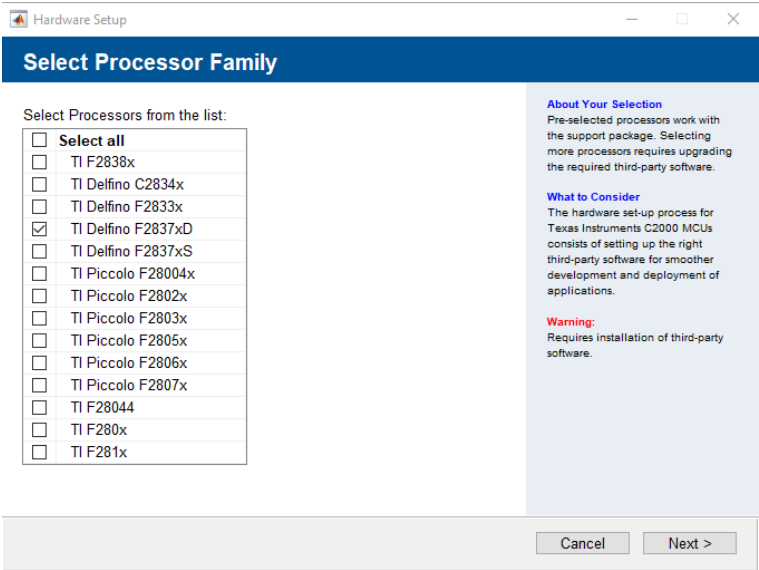


Figure A.5: Library Configuration window 4 *C2000 Processors*.

Once the screen in Fig. A.6 is reached, the components that appear in the window and are described in Table A.1 must be installed.

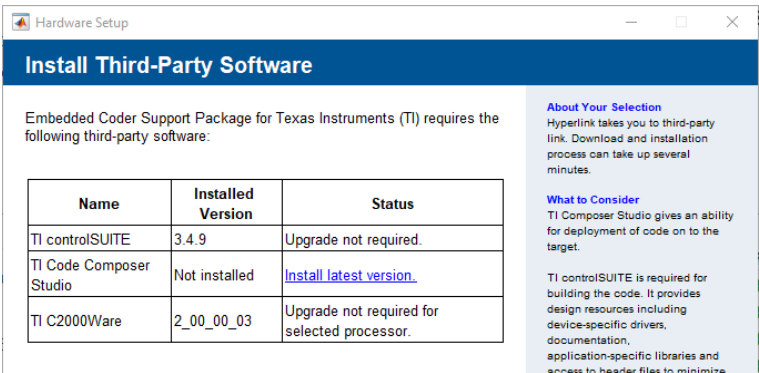


Figure A.6: Library Configuration window 5 *C2000 Processors*.

| Programme | Description |
|--------------------------------------|--|
| TI controlSUITE ¹ | Software and Development Tools |
| TI Code Composer Studio ² | TI Compiler |
| TI C2000Ware ³ | Documentation for C2000 Microcontrollers |


Table A.1: Table with the Texas Instruments components to install.

¹From: <http://ti.com/tool/CONTROLSUITE> . Last access: 03/01/2020.

²From: http://software-dl.ti.com/ccs/esd/documents/ccs_downloads.html. Last access: 03/01/2020.

³From: <http://www.ti.com/tool/download/C2000WARE>. Last Access: 03/01/2020.

Figs. A.7, A.8 and A.9 show the web pages where the programs are downloaded.

 **TEXAS INSTRUMENTS**

[Products](#) [Applications](#) [Design resources](#) [Quality & reliability](#) [Support & training](#) [Order now](#) [About TI](#)

[TI Home](#) [Item controllers](#) [Design resources](#) [controlSUITE™ Software Suite: Software and Development Tools for C2000™ Microcontrollers](#) [Worldwide \(in English\)](#)

controlSUITE™ Software Suite: Software and Development Tools for C2000™ Microcontrollers

CONTROLSUITE

[Description & Features](#)

[Technical Documents](#)

[Support & Training](#)

[Order Now](#)

Important Note

controlSUITE is no longer being updated, but will continue to be available for download. The latest software releases for C2000 are now provided through C2000ware with application examples through the DigitalPower SDK and MotorControl SDK.
[View all related products \(0\)](#)

Order Now

| Part Number | Buy from Texas Instruments or Third Party | Alert Me | Status | Current Version | Version Date | Description |
|--|---|--------------------------|--------|-----------------|--------------|---|
| CONTROLSUITE-ZIP-Offline (ZIP) Installer | Free Download | Alert Me | ACTIVE | v3.4.9 | 26-AUG-2018 | Drivers, libraries, BOMs, demos, schematics & code examples, etc (powerSUITE & DesignORVE software) |
| CONTROLSUITE-Web (WEB) Installer | Free Download | Alert Me | ACTIVE | v3.4.9 | 26-AUG-2018 | Drivers, libraries, BOMs, demos, schematics & code examples, etc (powerSUITE & DesignORVE software) |

4

Figure A.7: TI controlSUITE downloading page.

Code Composer Studio Version 9 Downloads

There are two types of installers:

- On-demand installers** allow you to download only the software components that you require. Formerly known as web installers.
- Single file installers** will download a large compressed file (about 800MB) so you may then uncompress it then select what you require to install. Formerly known as offline installers.

| Release | Build # | Date | Download | Notes |
|---------|-------------|--------------|---|---|
| 9.3.0 | 9.3.0.00012 | Dec 19, 2019 | Single file (offline) installers: Windows 64-bit only MDS Linux MDS - 64-bit only MacOS MDS On-demand (web) installers: Windows 64-bit only MDS Linux MDS - 64-bit only MacOS MDS Manifest | New/Notable In This Release (9.3.0.00012): <ul style="list-style-type: none">Release notesWindows support: Code Composer Studio is now supported only on 64-bit Windows machines.Mac OS installers are now distributed as signed and notarized disk image (DMG) files.Bugfixes for IDE, DVT, DebuggerMCU Compilers LTS 18.12.4C6800 Compiler 8.3.5MSP GCC v8.3.1.25MDCTools 3.00.02Device support updatesTI EMU15_ML1 v8.4.0.00006 |

[Download](#)

[Latest release information](#)

[Things to know before installing](#)

CCS Incremental Update Policy

[Code Composer Studio Version 9 Downloads](#)

[Code Composer Studio Version 8 Downloads](#)

[Code Composer Studio Version 7 Downloads](#)

[Code Composer Studio Version 6 Downloads](#)

[Code Composer Studio Version 5 Downloads](#)


[Code Composer Studio Version 4 Downloads](#)

[Code Composer Studio Version 3 Downloads](#)

[Code Composer Studio Version 2 Downloads](#)

Figure A.8: TI Code Composer Studio downloading page..

⁴It is necessary to provide personal information in order to download.


TEXAS INSTRUMENTS

[Products](#)
[Applications](#)
[Design resources](#)
[Quality & reliability](#)
[Support & training](#)
[Order now](#)
[About TI](#)

[TI Store](#)
[Semiconductors](#)
[Microcontrollers \(MCUs\)](#)
[C2000Ware for C2000 MCUs](#)
[C2000WARE_2.01.00.00](#)

[Back to software product page](#)

C2000Ware for C2000 Microcontrollers


C2000WARE_2.01.00.00

Release Date: 23 Dec 2019

[View release notes](#)
[Supported Platforms](#)
[What's new?](#)
[Release Information](#)

This page contains specific information about C2000Ware for C2000 Microcontrollers release package. Refer to the table below for download links and related content.

Product downloads

 Download requires export approval (7 minute)




| | Title | Version | Description | Size |
|---|--|------------|--|----------|
| C2000Ware Installers | | | | |
|  | Windows Installer for C2000Ware | 2.01.00.00 | Windows Installer for C2000Ware | 248505 K |
|  | macOS Installer for C2000Ware | 2.01.00.00 | macOS Installer for C2000Ware | 240909 K |
|  | Linux Installer for C2000Ware | 2.01.00.00 | Linux Installer for C2000Ware | 244079 K |
| C2000Ware Documentation | | | | |
| | TI Resource Explorer | 2.01.00.00 | C2000Ware in Cloud TI Resource Explorer | |
| | C2000Ware Quick Start Guide | 2.01.00.00 | C2000Ware Quick Start Guide | |
| | ControlSUITE to C2000Ware Transition Guide | 2.01.00.00 | ControlSUITE to C2000Ware Transition Guide | |

Figure A.9: TI C2000Ware downloading page..

Finally, once the three programs have been downloaded, returning to the package installer in the Matlab pop-up window, we locate the installed programs, shown in Fig. A.10, A.11 and A.12.

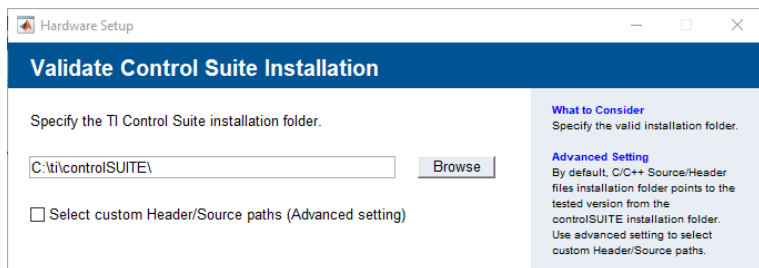


Figure A.10: *TI controlSUITE* download location window.

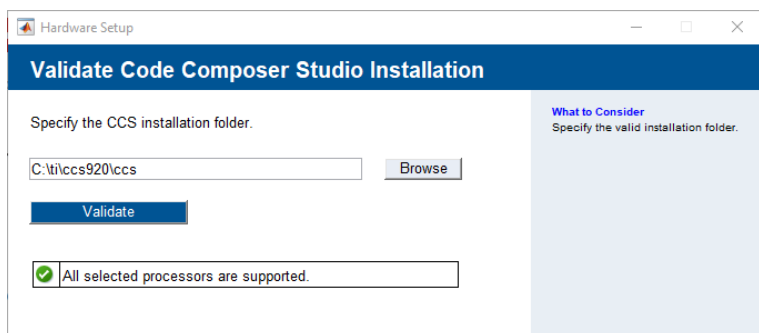


Figure A.11: *TI controlSUITE* download location window.

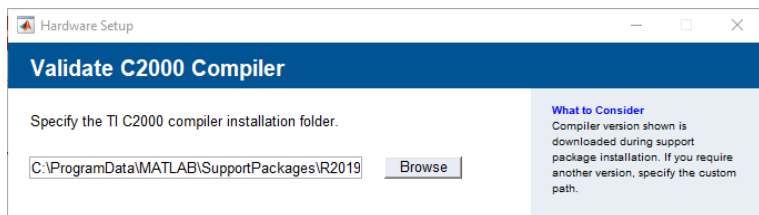


Figure A.12: *TI controlSUITE* download location window.

Appendix B

Simulator Interface

The development of the simulator interface has the objective of facilitating the modification of the environment and simulation conditions by the user. To create the interface, the following steps must be followed:

1. Create the variables used in dialog boxes and dropdowns.
2. Create a subsystem of the entire simulink assembly.
3. Create a subsystem mask.
4. Incorporate the necessary dialog boxes and dropdowns.

First of all, the simulink variables must be created, which will be modified from the interface that will be developed later. To do this, access the *Tools > Model Explorer* tab on the top bar of Simulink. In the window that opens, access the *Model Workspace ** option. In it, show the Data Objects variables. In it, create the variables with the names assigned in the simulink blocks. Select the *Argument* option so that the interface can modify its values after initialization.

Second, a subsystem of the entire simulink diagram set is created: It can be done by selecting everything and pressing ctrl + G or right button, *Create Subsystem from Selection*. Next, a system mask is created: Right-clicking on the subsystem *Mask > Create Mask*. Once *Create Mask* has been clicked, a window will appear with the appearance of Fig. B.2.

The *Parameters and Dialog* tab allows the modification of the tabs, dialog boxes, options, variable names, etc., of the interface box. In Fig. B.3 the parts that make up this tab are detailed.

The variables contained in the interface must be allowed to be modified internally in simulink. This is done by leaving the *Tuneable* option set to *on*. In the *Property Editor > Popup Options* option, the options contained in the dialog box are modified. Each option in a different row, as seen in Fig. B.4. When an option is selected in

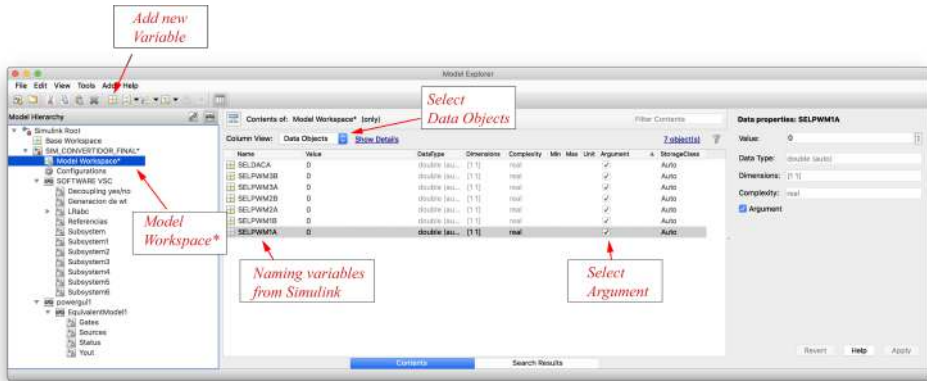


Figure B.1: Creation of variables for the interface.

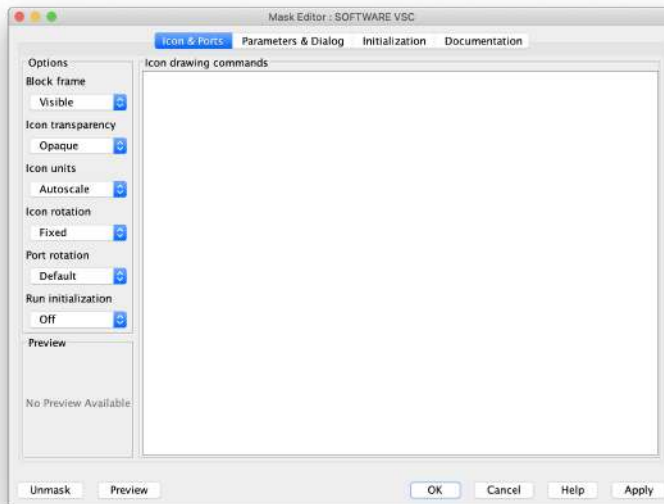


Figure B.2: Simulink mask main window.

the dialog box, the variable associated with that dialog box takes the value of the row number of the *Popup Options*.

Once the interface of the window with the dialog boxes has been created, its behavior is programmed. This is done in the *Initialization* tab. In this tab, you

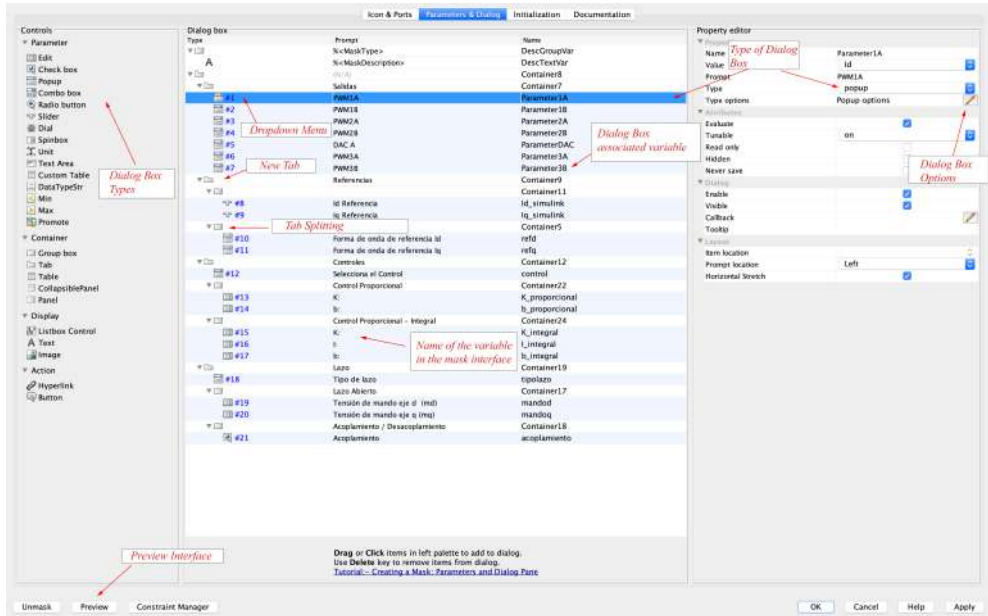


Figure B.3: *Parameters and Dialog* tab settings and descriptions.

program with code in Matlab language, the value that the simulink variables will take when the options in the dialog boxes have been selected. The name of the variables associated with the dialog boxes can be seen on the left side in the *Dialog Variables* section. In Fig. B.5 an example of programming a pop-up dialog box with seven options is shown.

Once the mask interface has been created, the behavior of the prototype simulator and hardware can be modified from the mask interface.

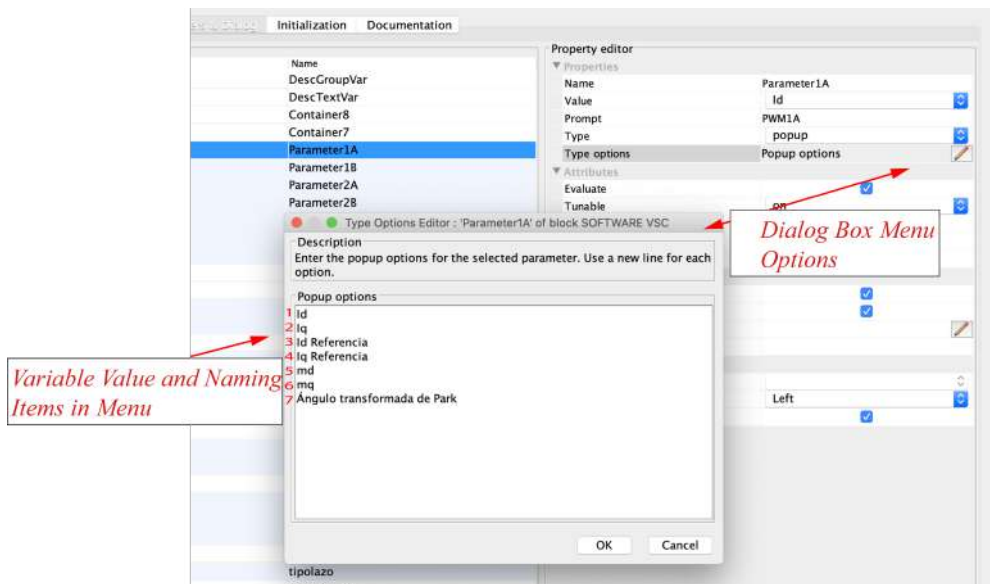


Figure B.4: *Popup Options* tab settings and descriptions.

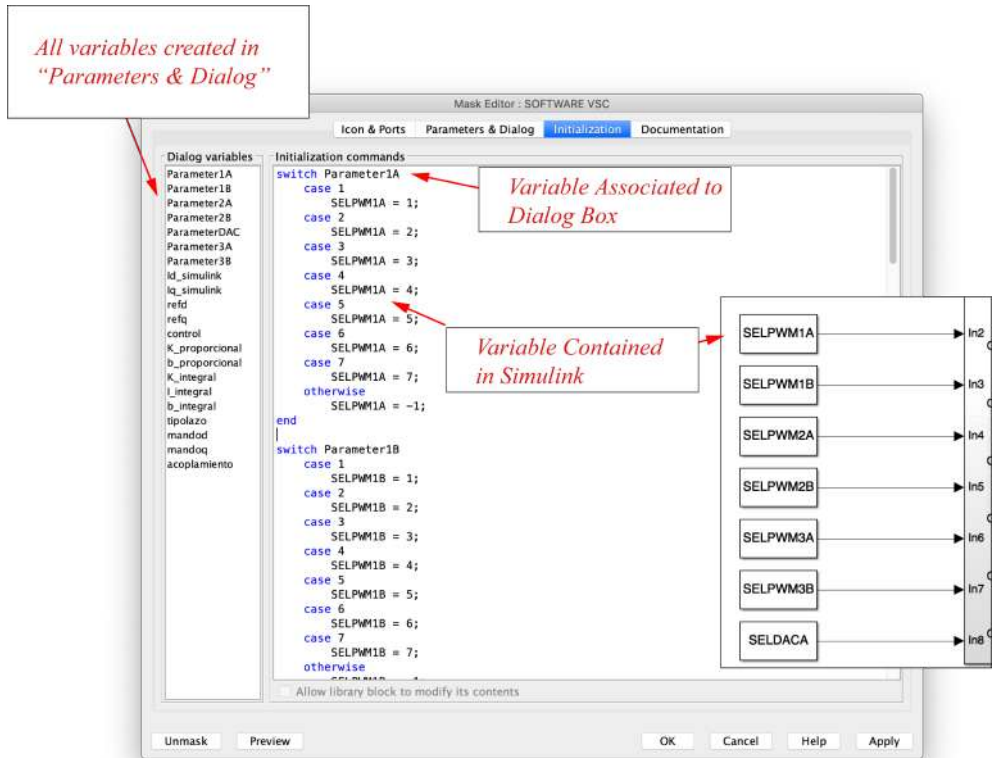


Figure B.5: Initialization tab settings and descriptions..

Appendix C

Files

All files are available in the following GitHub repository.

<https://github.com/JorgeSuarezPorras/ConvertidorElectronicoCC-CA>

Appendix D

Sustainable Development Goals

The social transition towards a more sustainable development, must necessarily go through a greater solidity in the implementation of renewable energies in the energy system. This implementation must be done, nowadays, through electronic power converters. Any advance, analysis, design, test or development that contributes to the understanding of these systems will facilitate said necessary transition of the energy model (Section 1.1 of project motivation).

The sustainable development goals promoted by the United Nations are (some of them) closely linked to this project. The greatest alignment with them can be found in:

1. Goal 7: Affordable and clean energy.
2. Goal 9: Industry, innovation and infrastructure.
3. Goal 11: Sustainable cities and communities.
4. Goal 13: Climate action.

Goal 7: Affordable and clean energy.

The vast majority of renewable sources of electrical energy are connected to the large distribution network through DC-AC converters, voltage sources with PWM. Given that this project consists of the development of a prototype of this type, the dissemination of the project could help facilitate access to documentation and information for research on these devices. In this way, its development and further implementation in the system could occur more quickly. Even more so, being a device specially developed for teaching.

Goal 9: Industry innovation and infrastructure.

Electronic power converters DC-AC voltage sources with PWM are used to connect to the electrical network, energy storage systems such as flywheels or batteries. Likewise, they can be used to optimize vastly extended and relatively inefficient processes of the current energy system, such as the use of PWM to optimize the cooling pumps of thermal power plants, thus favoring the transition to a more efficient and cleaner energy system.

Likewise, the development of new materials in science and industry help advance the technological capacity of these systems, thus increasing their efficiency. Materials such as gallium nitride that today is being widely investigated, and of which the switches (MOSFETs) of the VSC of the prototype of the present project are composed.

Goal 11: Sustainable cities and communities.

DC-AC converters WITH PWM are essential in electric traction, whether for train engines, electric vehicles ... etc. This project opens the door to implement the knowledge acquired in the clean transport of people and goods, favoring the development of the so-called Smart Cities.

Goal 13: Climate Action.

The production of power electronics devices currently produces polluting emissions (because the industry's energy comes mainly from non-renewable sources), in a way that could seem counterproductive. However, the more power electronics are implanted in the system, the less it will be polluted in the future, so that this feedback favors the obtaining, distribution and consumption of clean energy, thus helping to mitigate the effects of known climate change.