

Received November 13, 2021, accepted December 4, 2021, date of publication December 27, 2021, date of current version January 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3138705

# Study of the Open-Source Arduino DUE Board as Digital Control Platform for Three-Phase Power Converters

PEDRO MELÍN<sup>1</sup>, (Member, IEEE), CARLOS BAIER<sup>2</sup>, (Senior Member, IEEE),  
EDUARDO ESPINOSA<sup>3</sup>, (Member, IEEE), JAVIER RIEDEMANN<sup>4</sup>, (Member, IEEE),  
JOSE ESPINOZA<sup>5</sup>, (Senior Member, IEEE), AND RUBEN PEÑA<sup>5</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Electrical and Electronic Engineering, Universidad del Bio-Bio, Concepción 4051381, Chile

<sup>2</sup>Department of Electrical Engineering, Universidad de Talca, Curicó 3340000, Chile

<sup>3</sup>Department of Electrical Engineering, Engineering Faculty, Universidad Católica de la Santísima Concepción, Concepción 4090541, Chile

<sup>4</sup>Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S102TN, U.K.

<sup>5</sup>Department of Electrical Engineering, Universidad de Concepción, Concepción 4070409, Chile

Corresponding author: Pedro Melín (pemelin@ubiobio.cl)

This work was supported by the Chilean Government under Research Project ANID/FONDECYT/1201683 and Project ANID/FONDECYT/1201308.

**ABSTRACT** This work deals with the study of the open-source Arduino DUE board as a digital control platform for three-phase two-level Voltage-Source Converters (VSC) and Current-Source Converters (CSC), including (i) the description of the power topologies, its connection to the load or the ac grid, and the electrical signals required for sensors and power valves, (ii) the description of the Arduino Due board, its features, and its connection with the sensors and power valves in the converter, and (iii) evaluation of the Arduino DUE's processing time required for typical power converter algorithms, such as modulation, mathematical transforms, and linear controllers typically used in these converters. Experimental results are presented to validate the study, showing that the Arduino DUE is a feasible digital control platform for this type of power converter.

**INDEX TERMS** Pulse-width modulation converter, pulse-width modulation inverter, digital control.

## I. INTRODUCTION

Power electronics converters based on fully controlled switches have been widely studied, implemented, and tested by engineering students, engineers, and researchers owing to their key role in electrical power applications, such as motor drives [1], [2], renewable energy systems and their electrical network integration [3], [4], and electric vehicles [5], [6], among others. The study and implementation of power converters require the design and testing of (i) a power circuit, including passive components such as inductors, capacitors, and power valves with their respective gating drivers; (ii) electronic instrumentation to measure different electrical and/or mechanical variables (e.g., voltage, current, angular velocity, etc.), and (iii) modulation and control strategies that

The associate editor coordinating the review of this manuscript and approving it for publication was Atif Iqbal<sup>1</sup>.

allow regulating the converter switches appropriately for the required application [7].

When the power circuit and measurement system are designed and implemented, the development efforts should implement the modulation technique and control strategies. This implementation is typically carried out in a digital microprocessor-based platform due to its versatility, flexibility, and the possibility of a straightforward modification of the control algorithm and its parameters [8],[9]. Typically, the sequential execution of the controller on a digital platform follows the flowchart shown in FIGURE 1, where the algorithm is triggered by an interrupt defining the sampling time. In every execution of the control algorithm, the state of the power switches is updated, the electrical signals from the sensors are acquired and converted to a digital register, and the controller and modulation strategy defines the states of the power valves for the next sampling period.

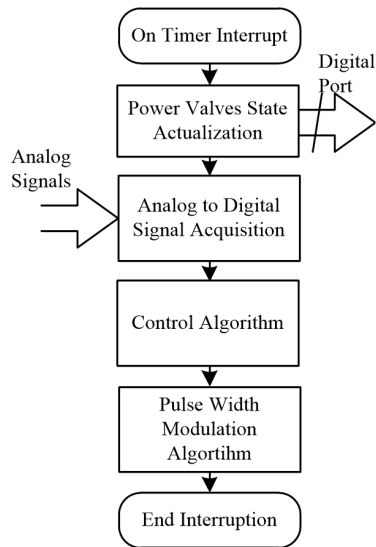


FIGURE 1. Flowchart of the digital implementation of a controller for power converters.

Some typical digital platforms used to control power converters are Digital Signal Processors (DSP) based platforms [10]–[14]. Simultaneously, dSPACE systems are popular to implement new control strategies because they allow straightforward programming [15]–[19]. At the same time, Field Programmable Gate Array (FPGA) platforms are also an attractive alternative because of their parallel processing capabilities and low cost [20]–[25]. These digital platforms present some drawbacks, such as a steep learning curve of the digital device and its development environment, in the case of DSPs and especially in the case of FPGAs, and the equipment cost, in the case of the dSPACE systems. On the other hand, Arduino [26] is an open-source electronics platform based on easy-to-use hardware and software.

The first version of the Arduino board was launched in 2005 and currently offers several boards. This year, the board’s cost varies between 103 USD (Arduino Portenta H7, released in 2020 [27]) to 20 USD (Arduino NANO 33, released in 2008). About the most popular boards, the Arduino UNO Rev.3 (released in 2010, [28]) uses an ATmega328 microcontroller, while Arduino MEGA2560 (released in 2010 [29]) uses an ATmega2560 microcontroller. Both boards are based on an 8-bits Reduced Instruction Set Computer (RISC) processor, use a 16 MHz clock, and currently cost 23 USD and 40 USD, respectively. Their main difference is that the Arduino MEGA2560 board has more physical I/O ports available than the Arduino UNO version, among other features.

The Arduino DUE board was launched in October 2012 and costs 40 USD [30]. The main difference between the Arduino DUE and the other Arduino boards is the use of a 32-bits SAM3X8E ARM Cortex-M3 [14], a RISC processor with more resources than those used in the Arduino UNO and Arduino MEGA. Moreover, compared with Arduino Portenta H7, the last Arduino board launched, which is based on a dual Cortex-M7+ M4 processor, its main attraction is the lower cost and physical ports disposition.

Arduino boards can be programmed by the Arduino Integrated Development Environment (IDE), a free and open-source software available on the Arduino project web page. The set of instructions is compatible with different Arduino boards, allowing a fast and easy board change. The above reduces the time required to learn the Arduino DUE if the developer has previous knowledge about Arduino boards or C programming language. Therefore, due to its low cost and large technical support, compared with other digital control platforms, the Arduino DUE is a suitable alternative for students, academics, and engineers who want

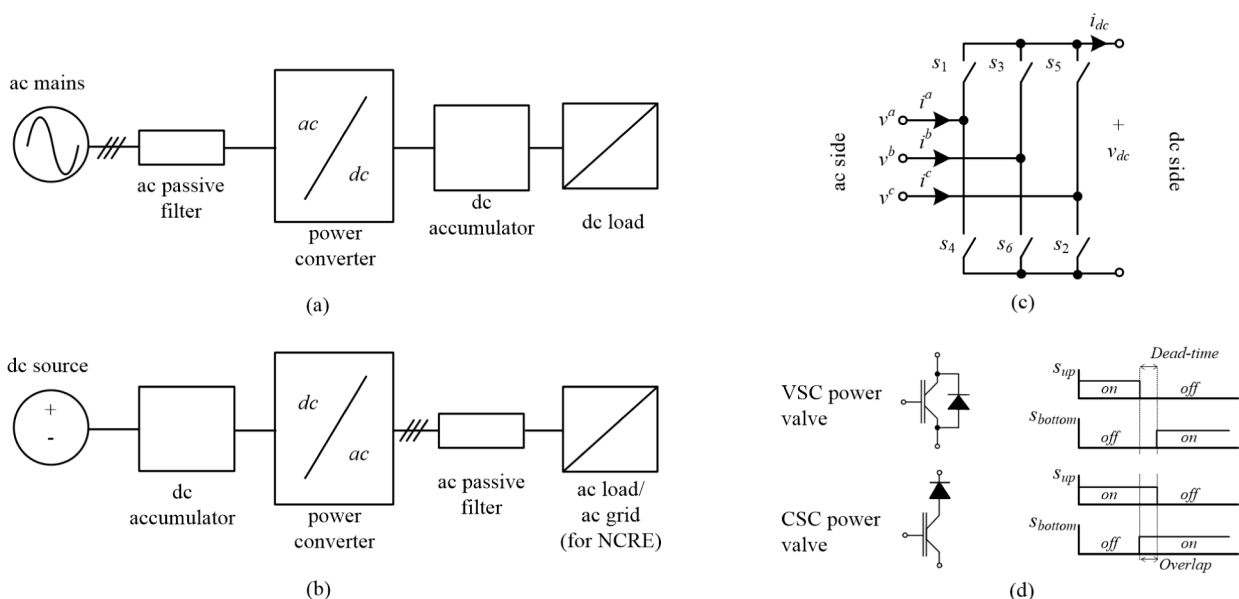


FIGURE 2. Power topologies and their electrical signals (a) rectifier configuration, (b) inverter configuration, (c) six power valves three-phase power topology, (d) power valves for VSC and CSC and their commutations consideration for each leg.

to test modulation and control strategies for power electronic applications.

This paper presents the study of the Arduino DUE board as a digital electronic control platform for power converters, focusing on its application to two-level three-phase Voltage-Source Converters (VSC) and three-phase Current-Source Converters (CSC). An extensive explanation of the board features with its performance to execute modulations techniques, linear controller, and phase-locked loop locked algorithms have been selected because they are widely used in several three-phase power converter applications. The work aims to be a reference for engineers and students interested in developing power electronic applications using this kind of open-source technology. The paper is organized as follows: Chapter II deals with the analysis of VSC and CSC topologies, the digital signals required to turn ON/OFF the power valve, and the requirements of the control platform to acquire analog signals from the power converters instrumentation. Chapter III presents an analysis of the Arduino DUE capabilities to implement the control tasks required to operate a VSC or a CSC, including analog to digital converters, digital inputs and outputs, and the conditioning circuits. Chapter IV reviews the Arduino IDE and the programming for the Arduino DUE to carry out the data acquisition and the pulse-width modulation. Finally, Chapter V shows the results of the experimental application, and Chapter VI presents the conclusions of the study.

## II. CONSIDERATION IN THE CONTROL OF THREE-PHASE POWER CONVERTERS

### A. DESCRIPTIONS OF THE POWER TOPOLOGIES

A power converter is an array of power valves that must be driven with a defined pattern to achieve the desired power conversion. The three-phase Voltage-Source Converter (VSC) and the Current-Source Converter (CSC) are two common and widely used six-valve power electronic converters – see FIGURE 2 -. Both can be used as a rectifier– to convert the ac voltage/current to dc voltage/current, see FIGURE 2a - or as an inverter – to convert the dc voltage/current to ac voltage/current, see FIGURE 2b-. Hence, the term converter is used to remark the power bidirectionality of the equipment. On the other hand, power valves are based on semiconductor devices, typically MOSFETs, IGBTs, or IGCTs, depending on the converter voltage/current ratings and the required switching frequency. Moreover, the limitations of these devices must be considered in the operation of the converter and its control system design.

Three-phase two-level VSCs are typically implemented using IGBTs or MOSFETs with an anti-parallel diode to provide a path for the reverse current – see FIGURE 2c and FIGURE 2d– and a capacitor at the dc side which operates as a voltage source for the VSC. The power valves states and their relationship with the ac voltages and the dc current are shown in Table 1, where it can be noticed that the relation between the upper and lower switches for each converter leg

TABLE 1. VSC ac voltages and dc current as function of the gating signals.

$v^{ab}$	$v^{bc}$	$v^{ca}$	$i_{dc}$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	HEX B[s6...s1]
$v_{dc}$	$-v_{dc}$	0	$i^a$	1	0	0	0	1	1	Ox31
0	$v_{dc}$	$-v_{dc}$	$i^b$	1	1	1	0	0	0	Ox07
$-v_{dc}$	0	$v_{dc}$	$i^c$	0	0	1	1	1	0	Ox1C
$-v_{dc}$	$v_{dc}$	0	$-i^a$	0	1	1	1	0	0	Ox0E
0	$-v_{dc}$	$v_{dc}$	$-i^b$	0	0	0	1	1	1	Ox38
$v_{dc}$	0	$-v_{dc}$	$-i^c$	1	1	0	0	0	1	Ox23
0	0	0	0	1	0	1	0	1	0	Ox15
0	0	0	0	0	1	0	1	0	1	Ox2A

TABLE 2. CSC ac currents and dc voltage as function of the gating signals.

$i^a$	$i^b$	$i^c$	$v_{dc}$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	HEX B[s6...s1]
$i_{dc}$	$-i_{dc}$	0	$v^a - v^b$	1	0	0	0	0	1	Ox21
0	$i_{dc}$	$-i_{dc}$	$v^b - v^c$	0	1	1	0	0	0	Ox06
$-i_{dc}$	0	$i_{dc}$	$v^c - v^a$	0	0	0	1	1	0	Ox18
$-i_{dc}$	$i_{dc}$	0	$-(v^a - v^b)$	0	0	1	1	0	0	Ox0C
0	$-i_{dc}$	$i_{dc}$	$-(v^b - v^c)$	0	0	0	0	1	1	Ox30
$i_{dc}$	0	$-i_{dc}$	$-(v^c - v^a)$	1	1	0	0	0	0	Ox03
0	0	0	0	1	0	0	1	0	0	Ox09
0	0	0	0	0	0	1	0	0	1	Ox24
0	0	0	0	0	1	0	0	1	0	Ox12

is the logic inverted signal. Additionally, when the states in the power valves in a leg are changed, it is necessary to ensure that both switches are OFF during a period – typically some nanoseconds or microseconds – which is called dead-time and is shown in FIGURE 2d. The objective of the dead-time is to avoid the natural delay during the semiconductor commutation which could lead to a short-circuit of the dc source through the converter leg.

The VSC operating as an inverter can be connected directly to the ac load or through a filter to reduce the harmonics due to the switching, but when a VSC is connected to the ac grid, a passive filter is mandatory. For these filters, a first-order inductive filter can be used to reduce the injected current distortion, while for the NCRE application, the third-order LCL filter has become very popular because of its higher harmonic attenuation compared with a first-order filter [32],[33]. If the application requires the generation of a sinusoidal voltage source [34]–[36], a second-order LC filter is required between the VSC and the load.

In a three-phase two-level CSC – see FIGURE 2c-, the power valve must provide reverse voltage blocking capability, which can be achieved with a diode in series to the IGBT or the MOSFET– see FIGURE 2d- or by using a power semiconductor with reverse blocking capability –, such as GTO, SGCT or RB-IGBT-. On the dc side, an inductor is required to implement the current source for the converter [38]. The power valve states and their relationship with the ac voltages and the dc current are shown in Table 2, where it can be observed that CSCs do not have a logical relation between power valves. Additionally, during the state changes in the power valves, it is mandatory always to be at least one upper and a lower switch are in ON state during some time – typically some microseconds – which is called overlap and is shown in FIGURE 2d. -. The overlap’s objective is to avoid leading the charged dc inductor to open-circuit during the state change.

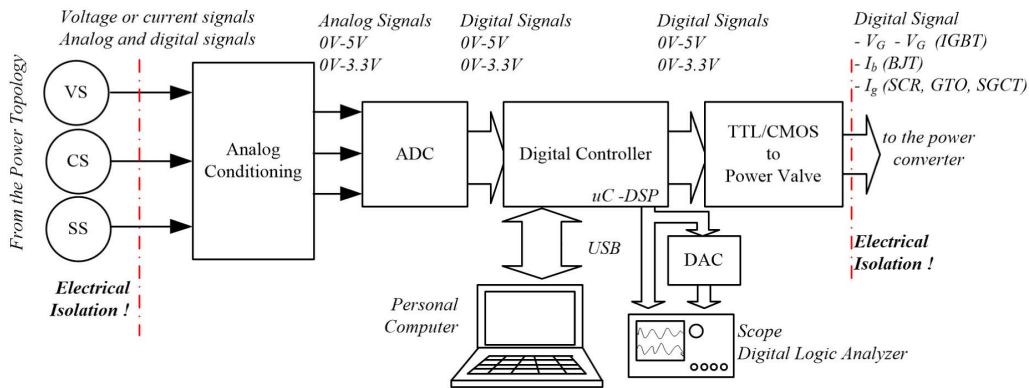


FIGURE 3. Electrical signals and digital electronic controller architecture.

For inverter applications, the CSC must be connected to the load using a first-order capacitive filter – if the ac load is inductive, such as an ac motor – or by a second-order  $LC$  filter - to connect the CSC to the electrical grid [39], [40]. For renewable energy applications, an  $LC$  filter is also typically used.

The power converter interacts with an electronic controller through electrical signals. The signals are classified as (i) signals generated by sensors in the application and (ii) gating signals generated by the controller to drive the power valves, as is shown in FIGURE 3. Electrical isolation is mandatory because of the different voltage levels between the power topology and the electronic controllers.

### B. SENSORS SIGNALS

For power electronics applications, the sensed variables can be electrical, such as voltages and currents, or non-electrical, depending on the application, for example, angular position and speed in a motor drive or temperature in a controlled furnace (see FIGURE 3). Independent of the variable, they are measured with special transducers whose output can be analog or digital signals. If the output of the transducer is an analog signal, it cannot be directly processed by the digital control platform, requiring it to be converted to a digital value. Two alternatives are possible for this task: (i) the use of a comparator and an analog reference to detect if the variable is higher or lower than a reference, or (ii) the use of an Analog-to-Digital Converter (ADC) that converts the analog value to a digital representation of  $n$  bits. On the other hand, the digital outputs in power electronic applications could be generated by on/off sensors to define alarms or operational modes.

The number of transducers required depends on the application and its characteristics. For example, if three-phase balanced voltage or currents are to be measured, only two transducers are needed, as the third signal is obtained as a linear combination of the first two; also, there are control strategies and algorithms that could even require sensing only one phase voltage/current. The number of transducers is typically equal to the number of analog inputs required in the digital controller.

### C. GATING SIGNALS

Gating signals are required to drive power valves in the converter. In each VSC power leg, the gating signal of the upper power valve is the logical complement of the lower one (Table 1). With the above, three signals are required to drive the converter if the dead-time is implemented by hardware, or six signals are required if the dead-time is implemented in the digital control platform. On the other hand, CSC always requires six signals because there is no relation between the states of the upper and lower valves in a leg (Table 2). The overlaps can be implemented in external hardware or implemented in the digital control platform.

Each type of power valve requires different electrical signals according to its technology to change its state. At the same time, the control platform generates digital signal levels according to its electronic digital technology. Thus, they must be electrically conditioned to the levels required by the power semiconductor devices.

## III. ARDUINO DUE RESOURCES TO CONTROL POWER CONVERTERS

The Arduino DUE uses a SAM3X8E microcontroller at 84 MHz, which is based on an ARM®Cortex®-M3 processor; SAM3X8E includes, among other features,

- 9-channel 32-bit Timer Counter (TC) for capture, compare, and PWM mode.
- Sixteen channels, 12-bits Successive-Approximation (SAR) ADC at 1 MSPs (multiplexed), one of them reserved for the internal temperature sensor.
- Six 32-bit Parallel Input/Output Controllers
- Eight-channel, 16-bit PWM with Complementary Output (PWMC) and 12-bit Dead Time Generator.
- Two channels, 12-bit, 1 MBPS Digital-to-Analog Converter (multiplexed).

The Arduino DUE limits the access to the SAM3X8E I/O ports to 69 of them, and the individual pins of each port are not typically physically close to the others.

### A. PROCESSOR AND MATHEMATICAL OPERATION

The SAM3X8E is based on an ARM®Cortex®-M3 processor, a deterministic 32-bits Harvard architecture

TABLE 3. Mathematical operations.

Numerical Format	int a,b,c;					float a,b,c;				
	c=a+b;	c=a-b;	c=a*b;	c=a/b;	float(a)	c=a+b;	c=a-b;	c=a*b;	c=a/b;	int(a)
Operation	0.08	0.14	0.08	0.08	0.74	1.23-2.75	1.23-3	0.69-2.7	2.73-13.2	0.47
Time (μs)										

TABLE 4. Arduino IDE instruction.

Instruction	Time (μs)	Instruction	Time (μs)	Instruction	Time (μs)
Vin = analogRead(Ax);	4.8	digitalWrite(pin, value)	1.5	Vint = int(Vfloat) + b;	0.36
Vfloat = m * float(Vin) + b;	2.8	PIOC->PIO_ODSR= int_value	0,05	analogWrite(DACx, Vint)	3.7

processor. It supports Nested Vectored Interrupt Controller, with 16 different priority levels and dynamically re-prioritized. About mathematical operation, Cortex®-M3 included a 32-bits integer Arithmetic Logic Unit (ALU) with hardware support for multiplications and divisions. This means that the Arduino DUE can compute arithmetic operations using 32-bits integer values, including signed integer, and results of 32-bits for and 64-bits for multiplications and divisions. On the other hand, Arduino IDE allows defining and operating single and double floating-point operations, synthesized using the integer operation in the SAM3X8E.

To test the execution time for mathematical operation in the SAM3X8E, they were implemented in Arduino IDE for integer and floating-point values (see Table 3), using the methodology presented in [41]. Also, conversion times between the two formats were measured. The results show the differences in the execution time between integer and floating-point values, the last ones with variable executing time, which take less than 3 μs for sum, subtraction, and multiplication. In comparison, a division could take approximately 14 μs. Based on these results, it can be concluded that it is possible to use floating-point values in the Arduino DUE within the algorithm to control and modulate a power converter. Nevertheless, it is necessary to verify that the algorithm’s execution time is lower than the sampling time required for the application.

**B. TIMERS**

Nine timers were available in the SAM3X8E. They can be configured using the individual timers register or installing the DueTimer library available in the Arduino IDE platform. With the library, the following instruction set and starts the timer

```
Timerx.attachInterrupt( Ctrl_Function );
Timerx.setPeriod( TimerValue );
Timerx.start();
```

where Timerx is the timer to be configured (from Timer0 to Timer8), Ctrl\_Function is the function to be called in the timer interruption, TimerValue is the time (in microseconds) to set the timer interrupt, and Timerx.start() starts the Timerx count. At least one timer could be required for control applications to handle the

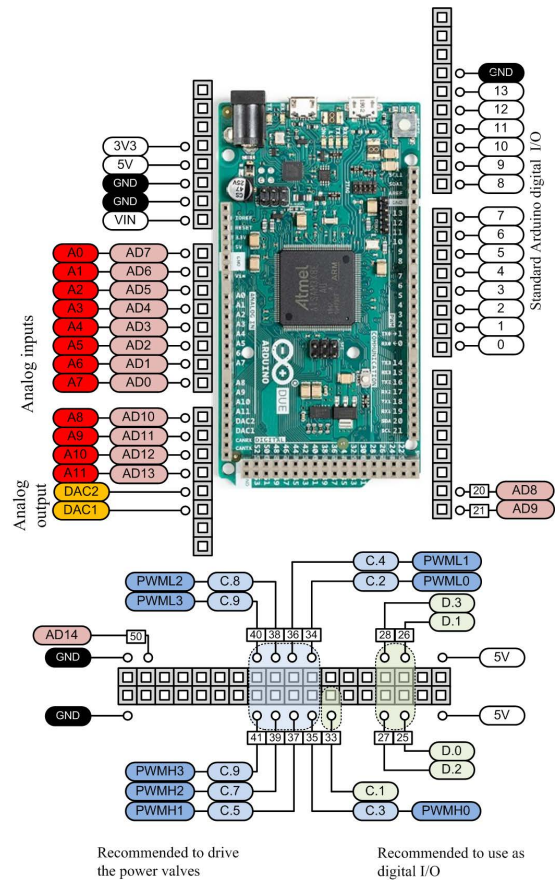
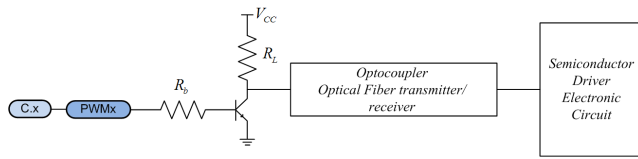


FIGURE 4. Arduino due pinout.

control algorithm with a fixed sampling time. In any case, the other timers can be used to manage the modulation algorithm, master control loop over a slave control loop, or diagnostic algorithm, among others.

**C. DIGITAL INPUT/OUTPUT PORTS**

The Arduino DUE has 54 physical digital I/O, and all of them can be configured as 3.3 V CMOS input/output digital ports. Arduino IDE allows the configuration, writing and reading of each Arduino board port with the commands pinMode(pin, OUTPUT), digitalWrite(pin, HIGH/LOW), and digitalRead(pin), where pin is the digital port according to the Arduino board.



**FIGURE 5. Recommended Conditioning Circuit to digital output signal to the semiconductor driver.**

The `digitalWrite` instruction takes  $1.5 \mu\text{s}$  to write one digital output, but it is better to simultaneously write all the outputs for the converter modulation. The recommended alternative is to use ports C.1 to C.9 available in pins 33 to 41 of the Arduino DUE board (see FIGURE 4) because they are close to each other, simplifying the connection. To set these pins as digital outputs, it is necessary to write the SAM3X8E as

```
REG_PIOC_OWER = 0x000001FE;
REG_PIOC_OER = 0x000001FE;
```

while to write the pins, the following command can be used

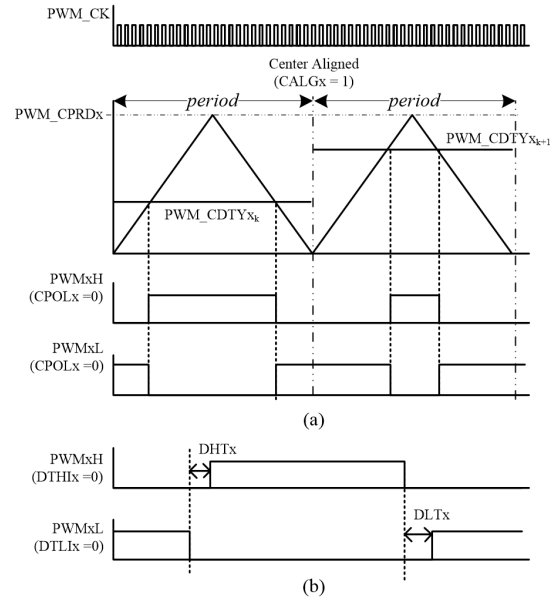
```
PIOC->PIO_ODSR = unsigned_int_value;
```

Arduino IDE supports these instructions. Because C.0 is not available on the Arduino DUE board, the developer must have in mind to make a left-shift to the `int_value` (`int_value << 1`) or write the `int_value` considering that C.0 is not available. To write the whole port using the above command takes  $0.08 \mu\text{s}$ , which is much smaller than the time required by the `digitalWrite` command. This option is highly recommended for modulation techniques that cannot be implemented using the internal PWM hardware, such as Selective Harmonic Elimination (SHE), Sliding Control, or Hysteresis Control. Table 1 and Table 2 show the valid states of VSC and CSC and their hexadecimal equivalents to write the register. Dead-times or overlaps must be implemented during the state change by software.

Each digital output signal must drive a power semiconductor device. A conditioning circuit is proposed for the above (FIGURE 5), where a pull-up configuration is used to drive an optocoupler or an optical fiber transmitter connected to the semiconductor driver’s input. It should be noted that because of the pull-up configuration, there is a logic inverter relation between the digital output and the pull-up output.

**D. PULSE WIDTH MODULATION HARDWARE**

Pulse Width Modulation (PWM) hardware allows the generation of gating signals for a VSC, including dead-time and complementary signals for the lower valves in each leg. Thus, three PWM channels allow driving a three-phase VSC, while an external digital circuit is mandatory for the CSC case to convert the VSC gating signals to CSC gating signals. In Arduino IDE, the instruction `analogWrite(pin, value)` generates a PWM signal with a variable duty cycle into an I/O digital pin of the Arduino board. However, this instruction does not control the frequency nor synchronizes



**FIGURE 6. PWM Waveform and Register Values (a) Triangular waveform (center aligned) compared with the duty cycle, (ii) dead time generation using DHTx and DLTx.**

the three channels among them. The recommendation is to write the PWM register directly.

PWM channels are based on a 16-bits register. They must be configured to (i) use the SAM3X8E physical output considering the pulses for the upper and lower valves of the converter leg -as a recommendation, PWM0 to PWM3, including their complementary outputs PWMxL and PWMxH are available in the Arduino DUE expansion port, see FIGURE 4-, (ii) have a center-aligned mode and enable dead-times using the PWM\_CMR register, and (iii) set the period, which is a function of the selected clock frequency in the PWM\_CLK register and the PWM\_CDTY register. PWM outputs are defined by comparing the PWM register and the duty cycle register, and the last one must be updated by the sampled value of the modulator signal (FIGURE 6a). The duty cycle register can be updated for each interruption, but the value is applied in the next period; thus, a delay is introduced due to the hardware operation. The dead-time can be configured to set the DTH and DLT bits in the PWM\_CMR register (FIGURE 6b), while its duration must be set in the PWM\_DT register.

Because the PWM output is a digital output of the Arduino DUE, the circuit shown in FIGURE 5 is proposed to convert the SAM3X8E levels to the levels required by the power semiconductor device. Finally, the PWM hardware interruption can be used to trigger the control algorithm instead of using a timer interruption. The above is recommended because it allows synchronization of the control algorithm with the PWM period, ensuring that the computed power valve states will be applied in the next sampling period.

**E. ANALOG INPUTS**

The 15 ADC inputs provided by the SAM3X8E are available on the Arduino DUE board, and 12 of them are located

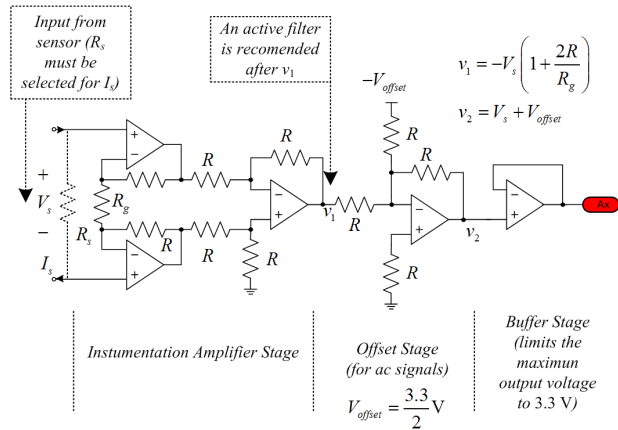


FIGURE 7. Example of a Conditioning Circuit to connect the signal from the sensor to the ADC.

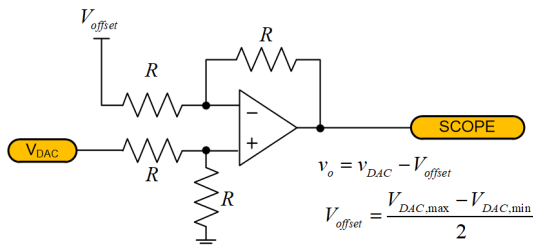


FIGURE 8. Recommended Conditioning Circuit to the DAC output.

closely, as shown in FIGURE 4. By compatibility with other Arduino boards, the ADC is set by default for 10-bits conversion and can be set to 12-bits using the following instruction

```
analogReadResolution(12);
```

The 12-bit ADC allows converting voltage from 0 V to 3.3 V to an integer range from 0 (0 × 000) to 4095 (0xFFFF); thus, the electrical signals from the sensors must be conditioned to this range. FIGURE 7 shows an example of a conditioning circuit that can be used for the above, including an instrumentation amplifier, offset stage–for ac signals with negative values–, and a buffer stage to ensure a voltage between 0 V and 3.3 V.

Analog signal can be easily acquired using the Arduino IDE command

```
Vin = analogRead(Ax);
```

where  $V_{in}$  could be an int or an unsigned int, and  $A_x$  is the ADC channel from A0 to A14. The integer value can be scaled to a float value using the following linear equation

$$V_{float} = m * (float)V_{in} + b;$$

where  $V_{float}$ ,  $m$ , and  $b$  are values in float format. The execution time of the above instructions requires 4.8  $\mu$ s for the ADC conversion – because the Arduino IDE sets the board to use the ADC with polling– and 2.8  $\mu$ s for scaling the acquired value to a floating-point format, as is shown in Table 4. Thus, to acquire an analog signal and scale it to a floating-point value requires 7.6  $\mu$ s.

F. DIGITAL TO ANALOG CONVERTERS

The Arduino DUE includes two channels of 12-bit Digital-to-Analog Converter (DAC) with a conversion range from 0.55 V to 2.75 V. The DAC resolution can be set using the Arduino IDE command

```
analogWriteResolution(RES);
```

where RES is the DAC resolution (12-bits or lesser) while to write a DAC channel, the instruction

```
analogWrite(DACx, value)
```

can be used, where value is an unsigned int, which a maximum value is given by  $2^{RES} - 1$  and take 3.47  $\mu$ s to be executed by Arduino DUE. Although digital-to-analog conversion is not required for power converter control, the two DAC channels can be used – and it is highly recommended to do it – as analog outputs to monitor the control algorithm in a scope. FIGURE 8 shows a recommended conditioning circuit to compensate the offset of the DAC output.

G. COMMUNICATION PERIPHERAL

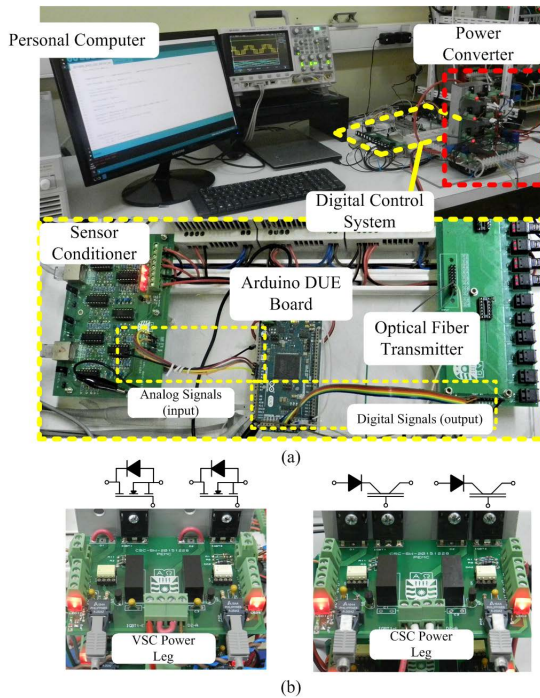
Arduino DUE supports several types of digital communications. Among others, it provides (i) Universal Asynchronous Receiver Transceiver (UART) and Universal Synchronous Asynchronous Receiver Transmitter (USART), (ii) Serial Peripheral Interface, (iii) Synchronous Serial Controller (SSC) and, (iv) Controller Area Network (CAN).

The digital communications allow the integration of additional ADC or DAC devices into the digital control system (using SPI or SSC as I2C), communicate the board to other machines as a personal computer using RS232 with the UART or to a PLC using MODBUS over RS485, or communicate the board with a net of other power converters by CAN.

IV. APPLICATION EXAMPLES

A VSC and a CSC rated at 2 kVA -FIGURE 9a-, including their power valve drivers and instrumentation, were implemented to evaluate the performance of the Arduino DUE as a digital control platform. VSC power valves are based on CREE-C3M0065090D SiC MOSFET, while CSC power valves are based on an IRG4PC40UD Si IGBT in series with an ON-GRG5060 Si hyperfast diode (FIGURE 9b). Six gating signals were required to drive each converter. Each driver uses a 660 nm optical fiber system compatible with the CMOS level of the Arduino DUE. Electrical variables were acquired using Hall-effect sensors LA25-p and LV25-p. At the same time, signal conditioning was implemented to ensure that the sensor signals were in the range from 0 V to 3.3 V. The optical fiber system and Hall-effect sensors isolate the digital controller and the power circuit. Also, key internal variables are obtained using the Arduino DUE DAC, updated them each control interruption.

With the above, two modulation techniques -sinusoidal pulse-width modulation and selective harmonic elimination–two linear controllers – a proportional-resonant controller



**FIGURE 9. Prototype (a) power converter and digital control system, (b) VSC and CSC power legs.**

and a proportional-integral controller - and a phase-locked loop algorithm were programmed in Arduino IDE and loaded into the Arduino DUE board. The computing time for each function is summarized in Table 5, while flowcharts for the linear controller and a phase-locked loop (PLL) are shown in FIGURE 10.

**A. MODULATION FOR A VSC**

The VSC Sinusoidal Pulse Width Modulation (SPWM) strategy was implemented using the Arduino PWM hardware. The SPWM compares three modulator signals with a triangular carrier with a higher frequency. Thus, the three modulator signals can define the converter’s ac fundamental voltage [42].

To implement SPWM with the six gating signals required, three PWM channels are used – one per leg -, with their complementary outputs and the dead-time generator. The

control interrupt is set at the beginning of each carrier period to compute the modulation values and apply them to the next period. Table 5 shows the required time to execute the above routines, where (i) to read the sine-cosine look-up tables takes  $1.2 \mu s$ , (ii) to compute the three references for modulation takes  $13.5 \mu s$ , and (iii) to update the PWM register takes  $13.7 \mu s$  from floating-point references. Finally, other operations – as increase the indexes for the look-up table and the sum of all actions is lesser than  $30 \mu s$ , leading to having a maximum modulation frequency of 33 kHz, which is an acceptable switching frequency range for most of the VSC applications.

The experimental results are shown in FIGURE 11 for a fundamental frequency of 50 Hz, 1050 Hz (21 p.u.) for the carrier frequency, and unitary modulation index. FIGURE 11a shows the ac inverter voltages (CH1 to CH3) and the Arduino DUE digital signals, including a mark for the beginning of the fundamental period (D6) and the gating signals (D0 to D5), while FIGURE 11b and FIGURE 11c show the difference between a no dead-time implementation and the dead-time implementation ( $2 \mu s$  in this case) and, including the digital signals for a power leg (D0 and D1) and the gate voltage at the SiC MOSFET (CH1 and CH2). Waveforms show the feasibility of the SPWM implementation, obtaining characteristic SPWM patterns at the line-to-line voltages and avoiding short-circuits when the state of the power valves changes.

**B. MODULATION FOR A CSC**

A CSC is modulated using Selective Harmonic Elimination, which allows getting a defined frequency spectrum with a low number of commutations. The modulator is defined by a modulation index and a phase angle, defining the gating angles of the switching pattern. For the above, two techniques are widely used are (i) to use look-up tables to get the gating angles for two points near the modulation index, and then to interpolate them or (ii) to use a polynomial function to describe the relationship between the modulation index and each gating angle. For this work, SHE used in [43] is implemented using 720 samples per cycle ( $0.5^\circ$  resolution), where the sampling is triggered by a timer using the DueTimer

**TABLE 5. Function for power converters applications.**

Modulation	Time ( $\mu s$ )	Transforms	Time ( $\mu s$ )	Algorithms	Time ( $\mu s$ )
Updating VSC-SPWM (using float format for the <i>abc</i> modulator values)	14	Clarke, <i>abc</i> to <i>a<math>\beta</math></i> (using float format)	11.6	Proportional-Resonant (PR) controller (using float format)	14.2
Updating CSC-SHE (using integer format for the gating angles, excluding overlap)	8.4	Clarke, <i>a<math>\beta</math></i> to <i>abc</i> (using float format)	11.4	Proportional-Integral (PI) controller (using float format)	11.5
Look-Up table reading for sine and cosine references (float format)	1	Park <i>abc</i> to <i>dq</i> (using float format)	16	Phase-Locked Loop (PLL) (using float format)	9.2
		Park <i>dq</i> to <i>abc</i> (using float format)	15.5		

\* without included the variables acquisition or transformations



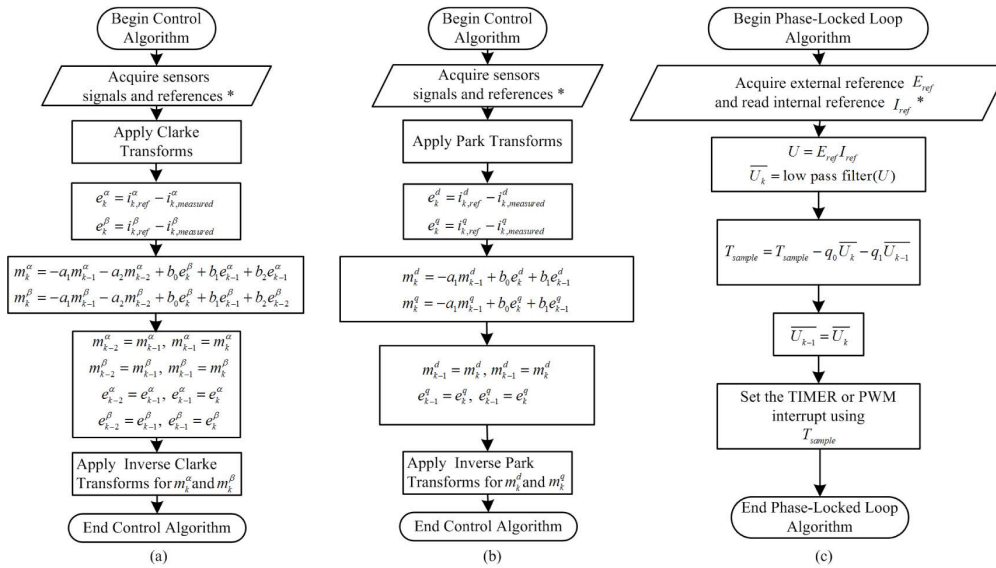


FIGURE 10. Algorithms Flowchart, (a) proportional-resonant controller, (b) proportional-integral controller, (c) phase-locked loop.

library. About the algorithm, (i) to read the phase references from look-up tables takes  $0.44 \mu s$ , (ii) to compare the gating angles with the phase references and define the six gating signals takes  $7.2 \mu s$ , and (iii) to write the digital port with the gating signals takes  $0.2 \mu s$  and increases the indexes for loop-up table takes  $0.5 \mu s$ . Overlap was implemented by software using a delay instruction, adding the overlap time to the modulation sample.

Experimental results are shown in FIGURE 12 for 50 Hz, 720 samples per period, and unitary modulation index. FIGURE 12a shows the ac inverter currents (CH1 to CH3) and the Arduino DUE digital signals, including a mark for the beginning of the fundamental period (D6) and the gating signals (D0 to D5), while FIGURE 12b and FIGURE 12c show the difference between no overlap implementation and overlap implementation ( $5 \mu s$  in this case), including the digital signals for a power leg (D0 and D5) and the gate voltage at the Si IGBT (CH1 and CH2). Waveforms show the feasibility of the SHE implementation, obtaining characteristics patterns at the phase current and avoiding open-circuit the dc inductor when the state of the power valves is changed.

### C. PROPORTIONAL-RESONANT CONTROL FOR A VSC USING CLARKE TRANSFORMS

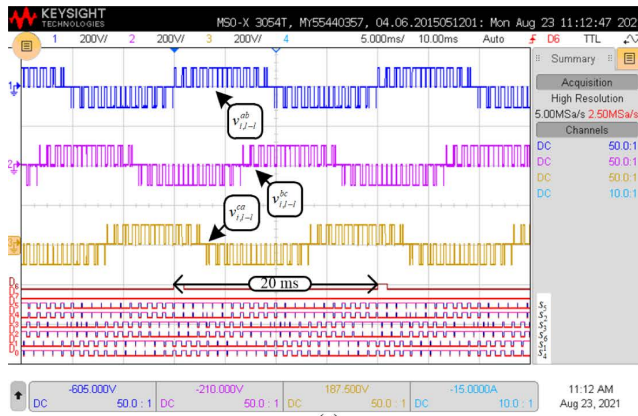
Using the Clarke transform, the three-phase signals are converted into two oscillatory signals –  $\alpha\beta$  frames- which can be controlled with two Proportional Resonant (PR) controllers [44], allowing power control using the p-q theory [45]. The PR controllers compute the modulators in the  $\alpha\beta$  frames, which must be converted to  $abc$  frames using the inverse Clarke transform. The controller steps -FIGURE 10a- are (i) to measure the ac currents and read the references from internal register, (ii) to convert them, and the references from  $abc$  frames to  $\alpha\beta$  frames, using Clarke transforms, (iii) to compute the error between the references and the measured currents, (iv) execute the PR algorithm to determine the modulator in  $\alpha\beta$  frames, (v) convert the modulator from

$\alpha\beta$  frames to  $abc$  frames and (vi) use the modulator in  $abc$  to drive the converter.

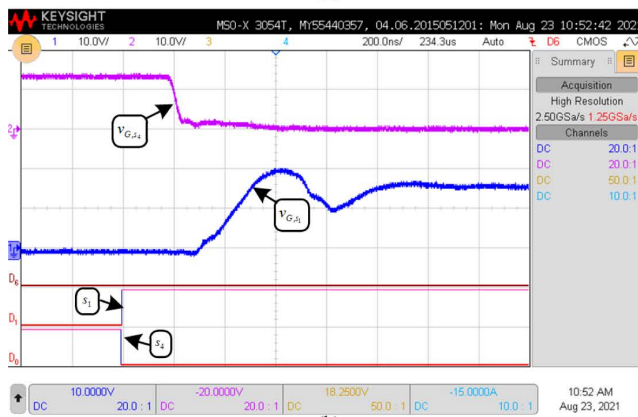
The algorithm was implemented using floating-point values, and the SPWM previously presented was used to drive the VSC for a fundamental frequency of 50 Hz and 1050 Hz (21 p.u.) for the carrier frequency. The PWM interrupt is used to trigger the controller algorithm, leading to 21 samples per fundamental period. To apply the Clark transform takes  $11.2 \mu s$ ; to compute each PR controller takes  $14.2 \mu s$  and to convert the modulator to the  $abc$  frame takes  $9 \mu s$ . Additionally, to acquire the three currents and the dc voltage and escalate them takes  $30.4 \mu s$ . FIGURE 13 shows the controller performance for a step reference change for the  $\alpha$  and  $\beta$  variables (FIGURE 13a and FIGURE 13b respectively); the internal variables and the references are shown using the DAC in CH1 (current) and CH2 (reference); CH3 and CH4 show the line-to-line voltage and the current for the  $a$  phase, D0 to D5 show the pulses for the power valves, D6 the samples (HIGH when the algorithm is executing and LOW when the Arduino DUE waits for the next interruption) and D7 is used to trigger the reference step change. From the waveform, it can be observed that the control implemented in the Arduino DUE accomplishes its objective, while the LOW period in D6 shows the available processing time on the processor between samples.

### D. PROPORTIONAL-INTEGRAL CONTROL FOR A VSC USING PARK TRANSFORMS

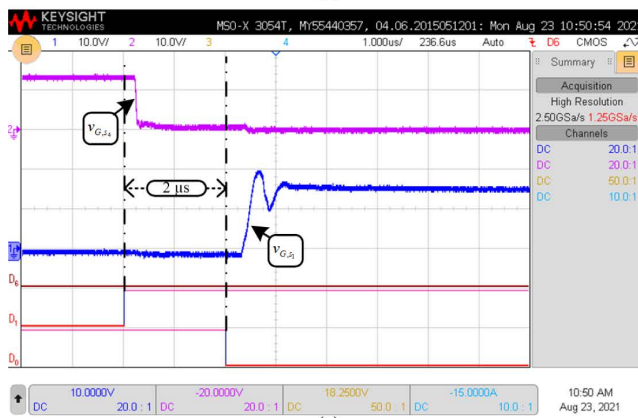
The Park transform allows the conversion of three oscillatory ac signals into two continuous signals; thus, a Proportional-Integral (PI) regulator can be used to control them [44], allowing power control using the p-q theory. The PI controller defines the modulator values in the  $dq$  frame, which could be used with a modulator strategy after applying inverse Park transform. The controller steps - FIGURE 10b- are (i) to acquire the ac currents, (ii) convert them from  $abc$  frames to  $dq$  frames using Park transforms, (iii) to compute the



(a)



(b)

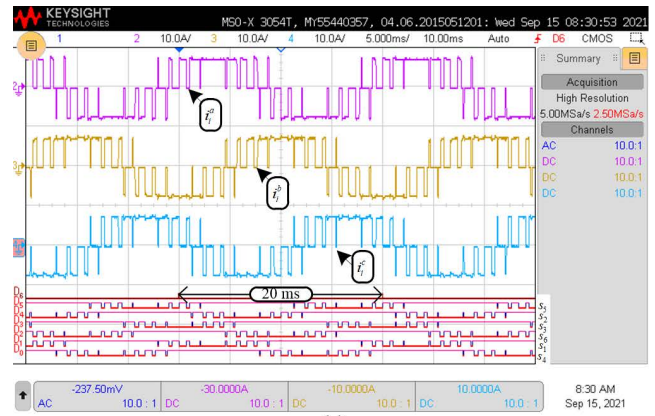


(c)

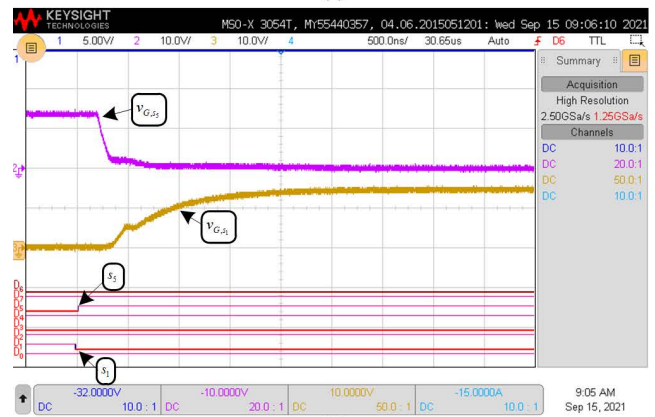
**FIGURE 11.** Key waveforms for VSC modulation using SPWM (a) line-to-line voltage and Arduino digital pulses (CH1 to CH3 line-to-line voltages, D6 begin of the internal period, D0 to D6 are the gating signals by the Arduino DUE), (b) gating pulses for a power leg without dead-time, (c) gating pulses for a power leg with 2 μs dead-time (please note the inverted relation between the Arduino signal and the gate voltage).

error between the references and the currents, (iv) execute the PI algorithm to determine the modulator in  $dq$  frames, (v) convert the modulator from  $dq$  to  $abc$  and (vi) use the modulator in  $abc$  to drive the converter using a modulation technique.

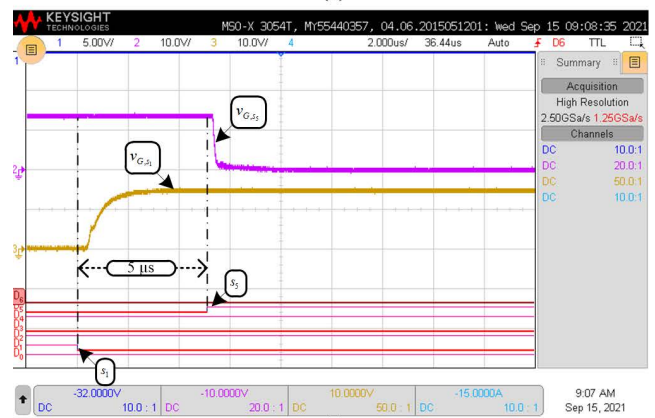
The algorithm was implemented using floating-point values. The SPWM previously presented was used to drive the VSC, using 50 Hz for the fundamental frequency and 1050 Hz (21 p.u.) for the carrier frequency. The PWM interrupt is used to trigger the control algorithm, obtaining



(a)



(b)

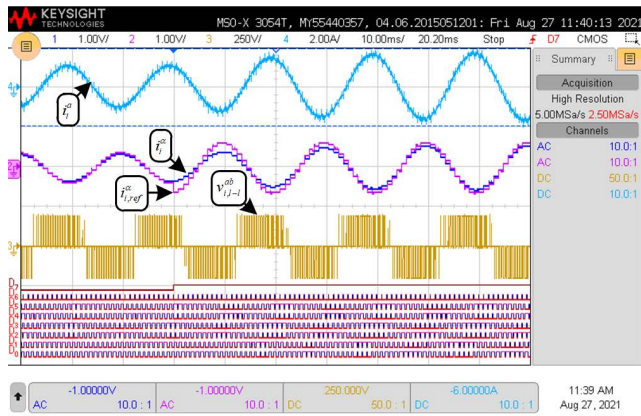


(c)

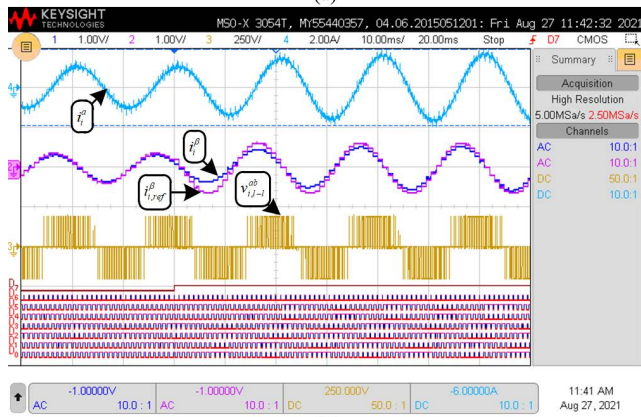
**FIGURE 12.** Key waveforms for CSC modulation using SHE (a) phase currents and Arduino digital pulses (CH2 to CH4 are the phase currents, D6 shows the begin of the internal period, D0 to D6 gating signals by the Arduino DUE), (b) gating pulses and gate voltages without overlap, (c) gating pulses and gate voltages with 5 μs overlap (please note the inverted relation between the Arduino signal and the gate voltage).

21 samples per fundamental period. The sine and cosine values for the Park Transform were read from a look-up table. To apply the Park transform takes 16 μs (8 μs each variable); to compute each PI controller takes 11.5 μs and to convert the modulator to the  $abc$  frame takes 15.5 μs. Additionally, to acquire the three currents and the dc voltage and escalate them takes 30.4 μs.

FIGURE 14 shows the control performance for reference changes for the  $d$  and  $q$  variables (FIGURE 14a and FIGURE 14b, respectively). In each figure, the internal vari-



(a)



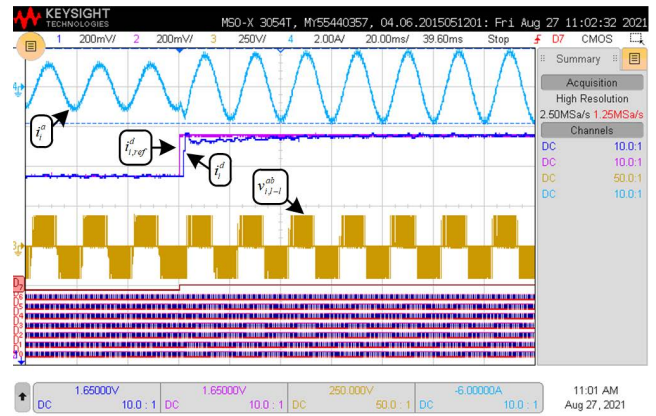
(b)

**FIGURE 13. Key waveforms for the PR controller performance (a) response for  $i^d$  response for step reference change, (b) response for the  $i^q$  response for  $i^d$  change (CH1: internal variable, CH2 internal reference, CH3 line to line voltage, CH4 ac current, D0 to D5 gating signal generated by the Arduino DUE, D6: executing of the algorithm, D7 trigger of the step reference).**

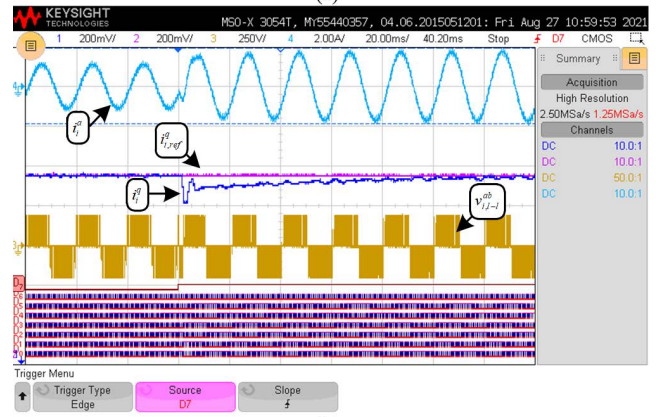
ables and the reference are shown using the DAC in CH1 (current) and CH2 (reference), while CH3 and CH4 show the line-to-line voltage and the current for the  $a$  phase. D1 to D5 show the pulses for the power valves, D6 shows the algorithm samples (HIGH when the algorithm is executed and LOW when the Arduino DUE is waiting for the next interruption), and D7 is used to trigger the reference step change. From the waveform, it can be observed that the control implemented in the Arduino DUE accomplishes its objective, while the LOW period in D6 shows the available processing time on the processor

**E. PHASE-LOCKED LOOP**

For grid-connected applications, a Phase-Locked Loop (PLL) is necessary to track the grid voltage frequency and phase angle and synchronize the power converter controller with it. As an example, the PLL presented in [46] was implemented in the Arduino DUE. This PLL requires measuring one voltage as an external reference, which is multiplied by an internal cosine reference. A PI regulator is used to follow a zero value as a reference by modifying the sampling time. (see FIGURE 10c). Because the multiplication of the two signals has an oscillatory component, a low-pass filter is required



(a)



(b)

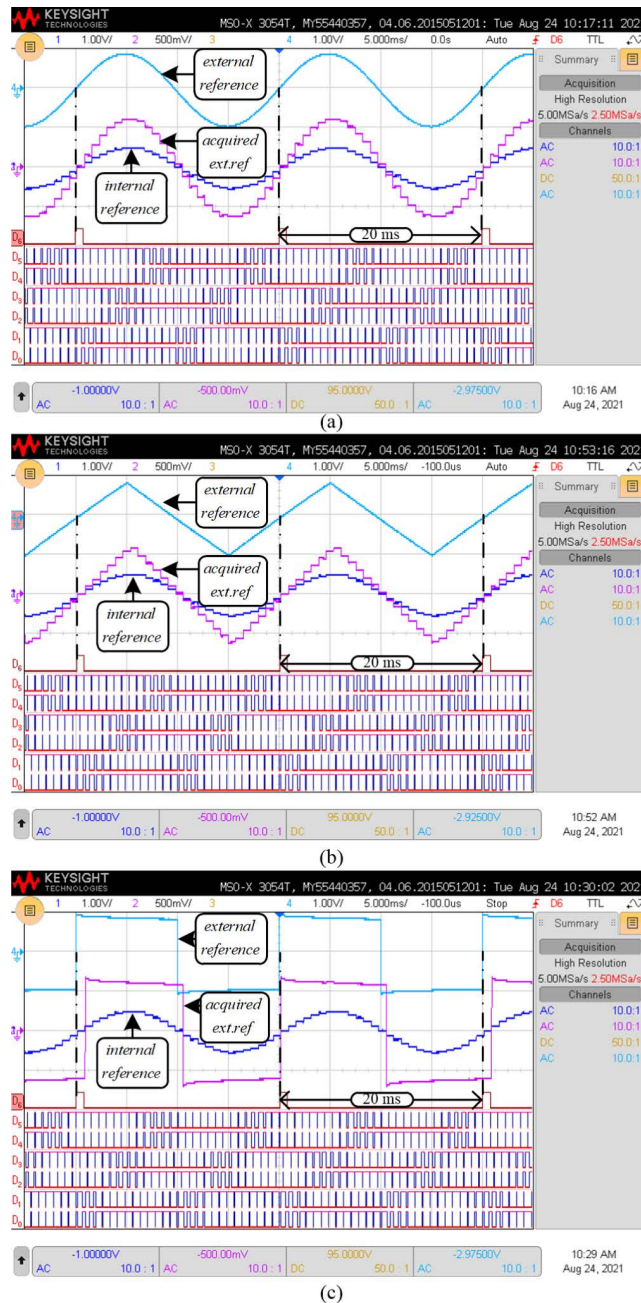
**FIGURE 14. Key waveforms for the PI controller performance (a) response for  $i^d$  response for step reference change, (b) response for the  $i^q$  response for  $i^d$  change (CH1: internal variable, CH2 internal reference, CH3 line to line voltage, CH4 ac current, D0 to D5 gating signal generated by the Arduino DUE, D6: executing of the algorithm, D7 trigger of the step reference).**

before the PI regulator. The output of this PI was used to set the control interrupt.

The implemented PLL algorithm uses floating-point values and takes  $9.2 \mu s$ , excluding the external signal acquisition and the set of the interruption period. In this case, an SPWM with 21 samples per period is implemented, and the PLL sets the PWM duty register. The experimental results for an external 50 Hz sine reference are shown in FIGURE 15a, where CH1 shows the internal reference for a sine, CH2 shows the external reference acquired by the Arduino DUE, and CH4 is the original reference. Also, D6 shows the synchrony of the control algorithm, and D0 to D5 show the gating pulses for the VSC. It can be noticed that the implemented PLL allows the synchronization of the internal reference phase with the external reference phase for a sinusoidal reference, but also for distorted references as triangular (FIGURE 15b) and square waveforms (FIGURE 15c).

**V. CONCLUSION**

This work studies the use of the Arduino DUE as a digital control platform for a three-phase power converter and the modulation for two-level converters. The study includes (i) the requirements to modulate and control two-level voltage-source and current-source topologies and (ii) the Arduino



**FIGURE 15.** Key waveforms for PLL performance (a) performance for a 50 Hz sinusoidal signal, (b) response for distorted signals (triangular and square waveform) (CH1 internal reference, CH2 acquire external reference, CH4 external reference, D0 to D5 gating signals, D6 begin of the internal period).

DUE board capabilities to perform the required task. Finally, the experimental results revised the modulation, control, and PLL implementation and measured the time required to execute each task in the Arduino DUE microcontroller.

This study demonstrates the feasibility of the Arduino DUE board to execute the required task in acceptable times for digital modulation and control of the power converters, allowing the computation of the algorithms using the floating-point emulation provided by Arduino IDE. Additionally, the low cost of the board and the software required to program

it made the Arduino DUE board an attractive device to implement the modulation and control strategies for power converter applications. Experimental results show the feasibility of the proposal and the available computing time between control interrupts to implement master control loops or communication algorithms.

## ACKNOWLEDGMENT

The authors would like to thank the Chilean Government projects ANID/FONDECYT/1201683, ANID/FONDECYT/1201308, and the technical support of the projects ANID/FONDECYT/11181203 and FONDAP/SERC/15110019; to the Universidad del Bío-Bío and the support of the projects DIUBB 182510 4/R and GI-160510EF and to Macrofacultad de Ingeniería UFRO-UBB-UTAL, Chile, 14ENI2-26866.

## REFERENCES

- [1] B. Wu, *High-Power Converters and AC Drives*, 2nd ed. Hoboken, NJ, USA: Wiley, Jan. 2017.
- [2] M. Rashid, *Power Electronics Handbook*. Oxford, U.K.: Butterworth-Heinemann, Sep. 2017.
- [3] B. Wu, Y. Lang, N. Zargari, and S. Kouro, *Power Conversion and Control of Wind Energy Systems*, 2nd ed. Hoboken, NJ, USA: Wiley, Jul. 2011.
- [4] E. F. Fuchs and M. A. Masoum, *Power Conversion of Renewable Energy Systems*. Boston, MA, USA: Springer, 2011. [Online]. Available: <https://link.springer.com/book/10.1007/978-1-4419-7979-7#about>, doi: 10.1007/978-1-4419-7979-7.
- [5] A. Ahmadi, A. Tavakoli, P. Jamborsalamati, N. Rezaei, M. R. Miveh, F. H. Gandoman, A. Heidari, and A. E. Nezhad, "Power quality improvement in smart grids using electric vehicles: A review," *IET Electr. Syst. Transp.*, vol. 9, no. 2, pp. 53–64, Jun. 2019, doi: 10.1049/iet-est.2018.5023.
- [6] J. Y. Yong, V. K. Ramachandaramurthy, K. M. Tan, and N. Mithulananthan, "A review on the state-of-the-art technologies of electric vehicle, its impacts and prospects," *Renew. Sustain. Energy Rev.*, vol. 49, pp. 365–385, Sep. 2015, doi: 10.1016/j.rser.2015.04.130.
- [7] H. L. Ginn, N. Hingorani, J. R. Sullivan, and R. Wachal, "Control architecture for high power electronics converters," *Proc. IEEE*, vol. 103, no. 12, pp. 2312–2319, Dec. 2015, doi: 10.1109/JPROC.2015.2484344.
- [8] A. Bendre, R. Cuzner, and S. Krstic, "Three-level converter system," *IEEE Ind. Appl. Mag.*, vol. 15, no. 2, pp. 12–23, Mar. 2009, doi: 10.1109/MIAS.2009.931810.
- [9] C. Buccella, C. Cecati, and H. Latafat, "Digital control of power converters—A survey," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 437–447, Aug. 2012, doi: 10.1109/TII.2012.2192280.
- [10] M. M. Morcos and A. Lakshminathan, "DSP-based solutions for AC motor drives," *IEEE Power Eng. Rev.*, vol. 19, no. 9, pp. 57–59, Sep. 1999, doi: 10.1109/MPER.1999.1236744.
- [11] Y. Tian, X. Yang, and W. Chen, "A novel copper layer-based field-to-trace coupling model and EMS evaluation method for DSP-based control circuit in MMC-HVDC system," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 1, pp. 1133–1146, Feb. 2021, doi: 10.1109/JESTPE.2019.2958142.
- [12] J. Ye, S. Huang, L. Liu, L. Li, J. Xu, and A. Shen, "Accurate harmonic calculation for digital SPWM of VSI with dead-time effect," *IEEE Trans. Power Electron.*, vol. 36, no. 7, pp. 7892–7902, Jul. 2021, doi: 10.1109/TPEL.2020.3041632.
- [13] X. Guo, Y. Yang, R. He, B. Wang, and F. Blaabjerg, "Transformerless Z-source four-leg PV inverter with leakage current reduction," *IEEE Trans. Power Electron.*, vol. 34, no. 5, pp. 4343–4352, May 2019, doi: 10.1109/TPEL.2018.2861896.
- [14] Y. Yang, X. Guo, H. Ding, Z. Lu, C. Hua, M. Malinowski, and F. Blaabjerg, "Zero dynamic DC-link voltage control for current source converter under grid disturbances," *IEEE Trans. Power Electron.*, vol. 37, no. 1, pp. 855–864, Jan. 2022, doi: 10.1109/TPEL.2021.3099622.
- [15] B. Bossoufi, M. Karim, M. Taoussi, H. A. Aroussi, M. Bouderbala, O. Deblecker, S. Motahhir, A. Nayyar, and M. A. Alzain, "Rooted tree optimization for the backstepping power control of a doubly fed induction generator wind turbine: dSPACE implementation," *IEEE Access*, vol. 9, pp. 26512–26522, 2021, doi: 10.1109/ACCESS.2021.3057123.

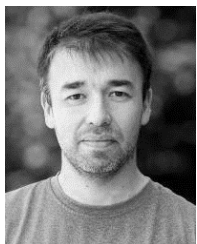
- [16] Y. Zuo, J. Mei, C. Jiang, and C. H. T. Lee, "Digital implementation of deadbeat-direct torque and flux control for permanent magnet synchronous machines in the M-T reference frame," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4610–4621, Apr. 2021, doi: [10.1109/TPEL.2020.3025332](https://doi.org/10.1109/TPEL.2020.3025332).
- [17] A. Dekka, B. Wu, N. R. Zargari, and R. L. Fuentes, "Dynamic voltage balancing algorithm for modular multilevel converter: A unique solution," *IEEE Trans. Power Electron.*, vol. 31, no. 2, pp. 952–963, Feb. 2016, doi: [10.1109/TPEL.2015.2419881](https://doi.org/10.1109/TPEL.2015.2419881).
- [18] C. R. Baier, R. O. Ramirez, E. I. Marciel, J. C. Hernandez, P. E. Melin, and E. E. Espinosa, "FCS-MPC without steady-state error applied to a grid-connected cascaded H-Bridge multilevel inverter," *IEEE Trans. Power Electron.*, vol. 36, no. 10, pp. 11785–11799, Oct. 2021, doi: [10.1109/TPEL.2021.3065003](https://doi.org/10.1109/TPEL.2021.3065003).
- [19] M. Badoni, A. Singh, A. K. Singh, H. Saxena, and R. Kumar, "Grid tied solar PV system with power quality enhancement using adaptive generalized maximum Versoria criterion," *CSEE J. Power Energy Syst.*, doi: [10.17775/CSEEJPES.2020.04820](https://doi.org/10.17775/CSEEJPES.2020.04820).
- [20] C. A. Sepúlveda, J. A. Muñoz, J. R. Espinoza, M. E. Figueroa, and C. R. Baier, "FPGA v/s DSP performance comparison for a VSC-based STATCOM control application," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1351–1360, Aug. 2013, doi: [10.1109/TII.2012.2222419](https://doi.org/10.1109/TII.2012.2222419).
- [21] V. K. Singh, R. N. Tripathi, and T. Hanamoto, "Implementation strategy for resource optimization of FPGA-based adaptive finite control set-MPC using XSG for a VSI system," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 2, pp. 2066–2078, Apr. 2021, doi: [10.1109/JESTPE.2020.2999267](https://doi.org/10.1109/JESTPE.2020.2999267).
- [22] L. He, F. Wang, and D. Ke, "FPGA-based sliding-mode predictive control for PMSM speed regulation system using an adaptive ultralocal model," *IEEE Trans. Power Electron.*, vol. 36, no. 5, pp. 5784–5793, May 2021, doi: [10.1109/TPEL.2020.3028545](https://doi.org/10.1109/TPEL.2020.3028545).
- [23] J. M. Blanes, R. Gutiérrez, A. Garrigós, J. L. Lizán, and J. M. Cuadrado, "Electric vehicle battery life extension using ultracapacitors and an FPGA controlled interleaved buck-boost converter," *IEEE Trans. Power Electron.*, vol. 28, no. 12, pp. 5940–5948, Dec. 2013, doi: [10.1109/TPEL.2013.2255316](https://doi.org/10.1109/TPEL.2013.2255316).
- [24] W. Li, L.-A. Gregoire, and J. Belanger, "A modular multilevel converter pulse generation and capacitor voltage balance method optimized for FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2859–2867, May 2015, doi: [10.1109/TIE.2014.2362879](https://doi.org/10.1109/TIE.2014.2362879).
- [25] C. A. Sepulveda, J. A. Muñoz, J. R. Espinoza, M. E. Figueroa, and P. E. Melin, "All-on-chip dq-frame based D-STATCOM control implementation in a low-cost FPGA," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 659–669, Feb. 2013, doi: [10.1109/TIE.2012.2206353](https://doi.org/10.1109/TIE.2012.2206353).
- [26] (Feb. 5, 2018). *What is Arduino*. Accessed: Apr. 20, 2021. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>
- [27] *Portenta H7*. Accessed: Apr. 20, 2021. [Online]. Available: <https://store.arduino.cc/usa/portenta-h7>
- [28] *Arduino Uno Rev3*. Accessed: Apr. 20, 2021. [Online]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>
- [29] *Arduino Mega2560 Rev3*. Accessed: Apr. 20, 2021. [Online]. Available: <https://store.arduino.cc/usa/arduino-mega-2560-r3>
- [30] *Arduino Due*. Accessed: Apr. 20, 2021. [Online]. Available: <https://store.arduino.cc/usa/duo>
- [31] (Mar. 23, 2015). *Smart ARM-Based MCU Datasheet*. Accessed: Apr. 20, 2021. [Online]. Available: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf)
- [32] S. Jayalath and M. Hanif, "An LCL-filter design with optimum total inductance and capacitance," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 6687–6698, Aug. 2018, doi: [10.1109/TPEL.2017.2754100](https://doi.org/10.1109/TPEL.2017.2754100).
- [33] R. A. Fantino, C. A. Busada, and J. A. Solsona, "Grid impedance estimation by measuring only the current injected to the grid by a VSI with LCL filter," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 1841–1850, Feb. 2021, doi: [10.1109/TIE.2020.2973910](https://doi.org/10.1109/TIE.2020.2973910).
- [34] P. Mishra and R. Maheshwari, "Design, analysis, and impacts of sinusoidal LC filter on pulsewidth modulated inverter fed-induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 2678–2688, Apr. 2020, doi: [10.1109/TIE.2019.2913824](https://doi.org/10.1109/TIE.2019.2913824).
- [35] R. N. Beres, X. Wang, M. Liserre, F. Blaabjerg, and C. L. Bak, "A review of passive power filters for three-phase grid-connected voltage-source converters," *IEEE Trans. J. Emerg. Sel. Topics Power Electron.*, vol. 4, no. 1, pp. 54–69, Mar. 2016, doi: [10.1109/JESTPE.2015.2507203](https://doi.org/10.1109/JESTPE.2015.2507203).
- [36] J. R. Lopez, P. P. Cruz, and A. M. Gutierrez, "VSI LC filter optimized by a genetic algorithm from connected to island microgrid operation," *Energy Syst.*, 2021, doi: [10.1007/s12667-021-00432-0](https://doi.org/10.1007/s12667-021-00432-0).
- [37] N. Zhu, D. Xu, B. Wu, N. R. Zargari, M. Kazerani, and F. Liu, "Common-mode voltage reduction methods for current-source converters in medium-voltage drives," *IEEE Trans. Power Electron.*, vol. 28, no. 2, pp. 995–1006, Feb. 2013, doi: [10.1109/TPEL.2012.2201174](https://doi.org/10.1109/TPEL.2012.2201174).
- [38] R. A. Torres, H. Dai, T. M. Jahns, and B. Sarlioglu, "Design of high-performance toroidal DC-link inductor for current-source inverters," in *Proc. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, Mar. 2019, pp. 2694–2701, doi: [10.1109/APEC.2019.8722096](https://doi.org/10.1109/APEC.2019.8722096).
- [39] P. E. Melin, J. I. Guzman, J. A. Rohten, C. R. Baier, J. A. Muñoz, E. E. Espinosa, J. A. Riedemann, and J. R. Espinoza, "Study of reactive power compensation capabilities and LC filter design for a three-phase current-source STATCOM," in *Proc. IEEE Int. Conf. Automat./23th Congr. Chilean Assoc. Autom. Control (ICA-ACCA)*, Oct. 2018, pp. 1–5, doi: [10.1109/ICA-ACCA.2018.8609717](https://doi.org/10.1109/ICA-ACCA.2018.8609717).
- [40] P. E. Melin, J. I. Guzman, F. A. Hernandez, C. R. Baier, J. A. Muñoz, J. R. Espinoza, M. I. Gonzalez, and E. E. Espinosa, "Study of reactive power compensation capabilities and LC filter design for a multi-level three-phase current-source D-STATCOM," in *Proc. IECON 44th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2018, pp. 3640–3645, doi: [10.1109/IECON.2018.8591466](https://doi.org/10.1109/IECON.2018.8591466).
- [41] O. E. Amestica, P. E. Melin, C. R. Duran-Faundez, and G. R. Lagos, "An experimental comparison of Arduino IDE compatible platforms for digital control and data acquisition applications," in *Proc. IEEE CHILEAN Conf. Electr., Electron. Eng., Inf. Commun. Technol. (CHILECON)*, Nov. 2019, pp. 1–6, doi: [10.1109/CHILECON47746.2019.8986865](https://doi.org/10.1109/CHILECON47746.2019.8986865).
- [42] A. Boulmane, Y. Zidani, and D. Belkhatay, "Implementation of modulation techniques on TMS320F28379D launchpad using the BOOSTXL-DRV8305EVM," in *Proc. Int. Symp. Adv. Electr. Commun. Technol. (ISAECT)*, Nov. 2018, pp. 1–6, doi: [10.1109/ISAECT.2018.8618756](https://doi.org/10.1109/ISAECT.2018.8618756).
- [43] P. E. Melin, J. I. Guzman, F. A. Hernandez, C. R. Baier, J. A. Muñoz, J. R. Espinoza, and E. E. Espinosa, "Analysis and control strategy for a current-source based D-STATCOM towards minimum losses," *Int. J. Electr. Power Energy Syst.*, vol. 116, Mar. 2020, Art. no. 105532.
- [44] R. Carnieletto, D. B. Ramos, M. G. Simoes, and F. A. Farret, "Simulation and analysis of DQ frame and P+Resonant controls for voltage source inverter to distributed generation," in *Proc. Brazilian Power Electron. Conf.*, Sep. 2009, pp. 104–109, doi: [10.1109/COBEP.2009.5347677](https://doi.org/10.1109/COBEP.2009.5347677).
- [45] H. Akagi, E. H. Watanabe, and A. Mauricio, *Instantaneous Power Theory and Applications to Power Conditioning*. Chichester, U.K.: Wiley, 2017.
- [46] M. A. Perez, J. R. Espinoza, L. A. Moran, M. A. Torres, and E. A. Araya, "A robust phase-locked loop algorithm to synchronize static-power converters with polluted AC systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 5, pp. 2185–2192, May 2008, doi: [10.1109/TIE.2008.918638](https://doi.org/10.1109/TIE.2008.918638).



**PEDRO MELÍN** (Member, IEEE) was born in Chillán, Chile, in 1982. He received the B.E. degree in electronic engineering and the M.Sc. and D.Sc. degrees in electrical engineering from the University of Concepcion, Concepcion, Chile, in 2006, 2010, and 2014, respectively. Since 2013, he has been with the Department of Electrical and Electronic Engineering, University of Bío-Bío, Concepcion, where he is currently an Associate Professor, teaching in the areas of electronic applications and digital systems. His research interests include electronics power converters and their application, including their design, digital control, and the application of this kind of topologies to AC drives, active power filters, and energy conversion.



**CARLOS BAIER** (Senior Member, IEEE) was born in Temuco, Chile, in 1979. He received the B.S., M.Sc., and D.Sc. degrees in electrical engineering from the University of Concepcion, Concepcion, Chile, in 2004, 2006, and 2010, respectively. Since 2009, he has been a Professor with the Faculty of Engineering, Universidad de Talca, Curicó, Chile, where he is teaching in the areas of industrial electronics. His research interests include improved control techniques for multicell converters, emerging converters, and strategies to inject power into the grid and high energy efficient improvements for medium-voltage converters.



**EDUARDO ESPINOSA** (Member, IEEE) was born in Concepción, Chile, in 1983. He received the B.E. degree in electronic engineering, in 2009, and the D.Sc. degree in electrical engineering from the University of Concepción, Concepción, in 2015, with a scholarship from the Chilean Research Fund ANID. Since August 2014, he has been with the Electrical Engineering Department, Universidad Católica de la Santísima Concepción, Concepción, where he is an Assistant Professor,

teaching and researching the power electronics and control systems. His research interests include model predictive control in power converters, multilevel converters, efficiency in power converters, and minimization of THD in multicell power converters.



**JOSE ESPINOZA** (Senior Member, IEEE) received the B.E. degree in electronic engineering and the M.Sc. degree in electrical engineering from the University of Concepción, Concepción, Chile, in 1989 and 1992, respectively, and the Ph.D. degree in electrical engineering from Concordia University, Montreal, QC, Canada, in 1997. Since 2006, he has been a Professor with the Department of Electrical Engineering, University of Concepción, where he is engaged in teaching

and research in the areas of automatic control and power electronics. He has authored and coauthored more than 150 refereed journal articles and conference papers and contributed to one chapter in the Power Electronics Handbook (Academic Press, 2011). He is currently an Associate Editor of the IEEE TRANSACTIONS ON POWER ELECTRONICS and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.



**JAVIER RIEDEMANN** (Member, IEEE) was born in Punta Arenas, Chile, in 1986. He received the degree in electrical engineering from the University of Magallanes, Punta Arenas, in 2010, and the D.Sc. degree from the University of Concepción, Concepción, Chile, in 2015. He was a Lecturer with the Department of Electrical and Electronics Engineering, University of Bío-Bío, Concepción, from 2014 to 2017, and a Lecturer with the Department of Electrical Engineering, Pontificia

Universidad Católica de Valparaíso, Valparaíso, Chile, from 2018 to 2019. He is currently a Research Associate with the Department of Electronic and Electrical Engineering, University of Sheffield, U.K. His research interests include control of power electronics converters, electric vehicles, and renewable energy systems.



**RUBEN PEÑA** (Senior Member, IEEE) was born in Coronel, Chile. He received the degree in electrical engineering from the Universidad de Concepcion, Concepcion, Chile, in 1984, and the M.Sc. and Ph.D. degrees from the University of Nottingham, Nottingham, U.K., in 1992 and 1996, respectively. From 1985 to 2008, he was a Lecturer with the Universidad de Magallanes, Chile. He has been with the Electrical Engineering Department, Universidad de Concepcion, since 2008, where he

is currently a Full Professor. His research interests include control of power electronics converters, AC drives, and renewable energy systems.

• • •