Article

Benchmarking Real-Time Control Platforms Using a Matlab/Simulink Coder with Applications in the Control of DC/AC Switched Power Converters

Jesus D. Vasquez-Plaza 1*, Andres F. Lopez-Chavarro 1, Enrique A. Sanabria-Torres 1, Juan F. Patarroyo-Montenegro 2 and Fabio Andrade 1

1 Sustainable Energy Center, University of Puerto Rico, Mayagüez Campus, Mayagüez, PR 00680, USA
2 Opal-RT Technologies, Montreal, QC H3K 1G6, Canada
* Correspondence: jesus.vasquez@upr.edu

Abstract: In the control of DC/AC switched power converters (SPC), one of the most important aspects to be considered is the selection of the real-time control platform. The real-time control platform must be able to meet the high performance efficiency and regulation requirements of the DC/AC SPC, as these typically operate at switching frequencies in the order of kHz to MHz. For this reason, the hardware characteristics of the ADC and PWM, and the processing capacity of the real-time control platform are of vital importance when implementing advanced digital controllers that maintain voltage and current levels within regulatory standards. In this context, we aimed to perform a comparative study of the computation times of different real-time control platforms when implementing different control strategies for DC/AC switched power converters. We also analyzed the impact of the real-time control platforms on the THD of the voltages generated by the DC/AC switched power converters. With the help of this paper, researchers and developers will have criteria to select which real-time control platform to use in real-time control for DC/AC SPC applications.

Keywords: real-time control platforms; DC/AC converters; computation time; total harmonic distortion; voltage and current controllers

1. Introduction

Due to the high demand for clean and sustainable electrical energy around the world, renewable energy sources’ (RES) applications are constantly growing. Switched power converters (SPCs) are an important means to transform RES into sustainable electrical energy and supply power to consumers. Mostly, SPCs are necessary for the integration of the RES into distributed generation systems, whether grid following or grid forming configurations [1]. A large number of these SPCs are controlled DC/AC converters incorporating a digitally programmable controller, implemented in a microprocessor or a dedicated device. Currently, the use of DC/AC SPCs is increasing due to the constant development of powerful transistors that can operate at ever faster switching frequencies, which has led to high requirements being placed on real-time control platforms (RTCPs) in terms of real-time control and simulation.

A large part of the research of the RTCP for DC/AC SPC revolves around the development and implementation of different control strategies that must maintain the values of current, voltage and frequency within the regulatory standards [2,3]. These control strategies change with the operation of the DC/AC SPC, which can operate connected to the main grid or in islanded mode connected to a load. Depending on the DC/AC SPC operating mode, different challenges are generated involving instability, waveform distortion, attenuation and major and minor disturbances. To overcome these challenges, appropriated controllers with fast responses, good reference tracking, short settling times, high reference tracking capacities and low total harmonic distortion (THD) must be implemented.
Many researchers have developed control strategies for the various DC/AC SPC modes of operation. Proportional-integral (PI) controllers in the \(dq\) frame and proportional-resonant (PR) controllers in the \(\alpha\beta\) frame are the most common. These controllers have a cascading architecture where the external control loop is the voltage control loop and the internal loop is the current control loop [4–6]. Another solution is to use optimal state feedback controllers such as LQR or LQI in the \(dq\) frame [7–9]. These control strategies achieve zero steady-state error at selected frequencies and provide a good combination of simplicity and high performance. Other widely used methods are \(H_\infty\), dead-beat, \(\mu\)-synthesis, sliding mode, fuzzy logic and hysteresis control [10,11]. In [12–16], exhaustive reviews of the different control strategies for DC/AC SPC were presented, taking into account the mode of operation.

Each of the aforementioned control strategies is implemented in RTCP. Figure 1 shows the typical architecture of an RTCP for a DC/AC SPC [17]. Typically, an RTCP for a DC/AC SPC consists of five components: (i) a sensing section, providing signal conditioning of electrical quantities, usually current and voltage measurements; (ii) a computational engine (CE), which is in charge of the implementation and execution of digital algorithms (control, filtering, modulation of output signals, among others); (iii) a power electronics section, which transforms and controls the voltages and currents to the desired levels; (iv) a communication section, which acts as the interface with other devices in order to incorporate the DC/AC SPC in a complex scheme; and (v) a human–machine interface (HMI), which allows users to configure and monitor the performances of the variables of interest of the DC/AC SPC.

![Typical architecture of an RTCP for DC/AC SPC control](image)

Nowadays, the CE of an RTCP consists of a micro-controller (\(\mu\)C) or digital signal processor (DSP), i.e., a very large-scale integration component incorporating a microprocessor core; digital inputs/outputs (DI-DO); analog–digital converters (ADCs); and a pulse-width modulator (PWM), which must be able to meet the strict regulatory requirements of the DC/AC SPC [18]. Therefore, the characteristics of the ADC and PWM, and the processing capacity of the CE, are of decisive importance when implementing advanced digital controllers which regulate the voltages and currents of the DC/AC SPC [19].
Unlike in other applications, the voltage and current signals of the DC/AC SPC are sinusoidal signals (time-varying), usually 50 or 60 Hz, depending on local regulations. This makes the design and practical implementation of digital controllers for DC/AC SPC have limitations at the ADC in terms of conversion speed and quantization error. The ADC resolution and sampling frequency determine the accuracy of the voltage and current signals being fed back, which affects the DC/AC SPC dynamics. On the other hand, DC/AC SPCs usually operate at switching frequencies in the order of tens of kHz. To achieve high switching frequencies, high-resolution, high-frequency digital PWM is required, which can allow high bandwidth, precise voltage regulation and reduced reactive element sizes in digitally controlled DC/AC SPCs [20,21].

Based on the requirements for DC/AC SPC described above, different RTCPs specialized for these applications have been developed and marketed. Among the main models, the dSPACE RTCPs represent a well-known example, offering a wide range of different solutions for SPC research [22]. Other popularly known solutions are the OPAL-RT [23], Typhoon [24] and RTDS [25] RTCPs, which allow real-time simulations, real-time control and hardware-in-the-loop (HIL) implementation of SPC. The RT-BOX [26], Imperix [27] and PED-Board [28] RTCPs were developed for real-time control and HIL tests with specialized SPC software and hardware. Despite the multiple advantages offered by the RTCPs mentioned above, in general, they are expensive, limiting their use for many researchers. To address this problem, various research groups have developed low-cost custom DSP/μC-based RTCPs for SPC applications [29–31]. They mostly use μCs from the C2000 branch of Texas Instruments (TI) which are particular for real-time control [32].

One of the greatest advantages that most of the RTCPs mentioned have is that they allow alternative programming in a conventional way, through a block diagram modeling environment developed in Matlab/Simulink. This facilitates the process that is usually done with C programming and a dedicated compiler. Matlab/Simulink has a tool called Embedded Coder™ that generates and executes C and C++ code through an automated process, including coding, compiling, linking and downloading to the RTCP [33–35]. In addition, it allows real-time control and changing and monitoring the variables and parameters of interest, making this ideal for the control of DC/AC SPCs [36,37].

Many researchers have published reviews comparing the different RTCPs in terms of their computational power, characteristics, advantages, disadvantages, etc. A general review where different RTCPs were compared in terms of prices, computing capabilities, hardware, software, communication and interface is presented in [31,38–41]. In [42], RTCPs are compared in SPC real-time simulation applications. This review focuses on modeling accuracy, bandwidth and system stability, showing the limitations of power and communication interfaces. On the other hand, some researchers have analyzed and compared the calculation power of RTCP in real time programmed with code generated directly from models in Simulink. Simulink offers several different ways to optimize the code that it generates and downloads in the RTCP. In [43], the computation times required by PID and LQI controllers are compared for the different optimization methods, in order to estimate the computational power of the RTCP. Similarly, A comparative study between RTCP of different characteristics was carried out in [44] to measure the computational power of each one. Several Simulink blocks of varying complexity were downloaded into the RTCP. For each downloaded block, the computation time that each block used in each RTCP was measured.

Real-time control of SPC involves some hardware challenges, especially the restrictions on the CE capabilities of the RTCPs and the fact that the ADC and PWM frequencies must be synchronized. This last restriction indicates that the ADC and PWM tasks must be executed in the same sampling period, which is challenging. For this reason, it is of great interest to study how the computation times of the different RTCPs are affected when implementing different control strategies in SPC applications.

We aimed to perform benchmarking of RTCP using a Matlab/Simulink Coder with SPC control applications. A methodology is proposed to measure and compare computation
times in different RTCPs that are programmed from Matlab/Simulink. Moreover, we intended to carry out a comparative study of the computation times of different RTCPs when implementing different control strategies for DC/AC SPC and to analyze the impact of the RTCP on the THD of the voltages generated by the DC/AC SPC. With the use of the information provided by this paper, researchers and developers will be better able to choose which RTCP to employ when developing a new real-time control method for SPC applications.

The rest of this paper is organized as follows: In Sections 2 and 3, a brief description and justification of the RTCPs and control strategies for the DC/AC SPC used in this paper are presented. Then, in Section 4, the proposed computation time measurement methodology is presented. Section 5 presents how to measure computation times for each selected RTCP. Section 6 presents the experimental results of computation times and THD obtained with each RTCP. Finally, discussions and highlighting are presented in Section 7.

2. Real-Time Control Platforms for Rapid Control Prototyping

The selection of RTCP must consider all the hardware implementation challenges mentioned above. In order to cover the wide range of RTCPs for SPC applications on the market, four different RTCPs with different characteristics were selected that allow real-time control from Matlab/Simulink.

2.1. Technical Description of the RTCP

We considered the most powerful and expensive RTCPs and the cheapest and most simple RTCPs. The RTCPs considered for this research are the following: (i) High cost and performance: OPAL-RT OP5700. (ii) Medium cost, high performance: dSPACE Scalexio. (iii) Medium cost and performance: dSPACE 1006. (iv) A low-cost, custom RTCP based on a TI $\mu$C of the C2000 branch TMS320F28379D developed in [30]. Table 1 shows technical benchmarking of the mentioned RTCPs in terms of DI, DO, ADC, PWM, CE, software and price. The more $ signs the platform has, the higher the price is in the market.

Table 1. Technical benchmarking of the RTCPs used in this research.

| RTCP Brand/Model | Features | Computational Engine | Price | Software |
|------------------|----------|----------------------|-------|----------|
|                  | DI No. | DO No. | ADC No. | Resolution | Conv. Time | PWM No. | Max. Freq. |
| Based in TI $\mu$C C2000 F28379D | 169 Shared | 169 Shared | 16 | 16/12 bits | 0.91 $\mu$s | 24 Shared | 100 MHz |
| dSPACE 1006 | 96 Shared | 96 Shared | 48 | 16 bits | 0.80 $\mu$s | 16 | 2 MHz |
| dSPACE Scalexio | 96 Shared | 96 Shared | 48 | 16 bits | 0.25 $\mu$s | 96 Shared | 0.5 MHz |
| OPAL-RT OP5700 | 32 | 32 | 16 | 16 bits | 1 $\mu$s | 32 Shared | 0.5 MHz |

Most of the RTCPs used in this research can handle a minimum of 32 DI, 32 DO, 16 ADC and 16 PWM, which are sufficient for single-phase or three-phase DC/AC SPC applications. The OP5700 has independent dedicated boards for the DI and DO, unlike the other RTCPs that are shared. It should be noted that dSPACE 1006 has dedicated boards specifically to generate the PWM, unlike the other RTCPs that are shared with the DI and DO. This dedicated hardware allows one to generate PWM of up to 2 MHz. The $\mu$C-based RTCP has fixed and unalterable hardware, unlike the dSPACE and OPAL-RT ones, which allow adding or changing the ADC and DI-DO boards for other more powerful ones. This advantage allows one to enhance the RTCP, but leads to quite high additional
costs. Likewise, it is observed that the conversion times of the ADC and the maximum frequencies of the PWM vary, as does the speed of the CE according to the RTCP, which will be studied in the experimental results section.

2.2. Code Generation Process

As mentioned above, SPC real-time control is now widely used and is constantly expanding. The control methods must be implemented in RTCPs, since SPCs operate at kHz-order frequencies and are composed of switching elements. All tasks performed by the RTCPs must be completed within real-time constraints to establish real-time control of the SPCs [45]. This means that both inputs and outputs in the RTCP must be processed within the sampling time window \((T_s)\), as shown in Figure 2.

![Figure 2. Tasks at a sampling time in real-time control applications.](image)

The main tasks in a \(T_s\) are: (a) measurement, (b) program execution and (c) command. These tasks must be executed within the fixed \(T_s\) window considering the “idle” time. The first task acquires the input physical/logical variables from the system. The control program is executed in the second task, where all DI-DO, ADCs, PWM and modulation techniques are configured, along with the general control structures. The command task executes the system output control actions. If the sum of all tasks exceed the \(T_s\), the RTCP goes into overrunning, which generates inaccurate and inconsistent system output states.

Programming control strategies for different processing RTCPs require prior knowledge of the programming environment and the peripherals built into the RTCPs to be used. C or C++ code is commonly used to implement these control strategies. Simulink Coder\(^\text{®}\) provides the ability to generate source code used in real-time applications [46,47]. The main structure of code generation by Simulink\(^\text{®}\) is shown in Figure 3.

![Figure 3. Code generation process using Simulink Coder.](image)

The creation of the Simulink/Matlab file is carried out in step 1. Block programming is done using support packages for electrical devices, control systems blocks, mathematical blocks, PWM blocks, analog/digital input and output blocks, etc. [48]. Usually, each RTCP has its own analog/digital PWM, input and output blocks developed according to its hardware build. Then, in step 2, the conversion of the Simulink programmed blocks to a programming language is carried out. The input/output parameters, the initial states and the block configurations are represented in C/C++ by means of Simulink Coder\(^\text{®}\) [49–51]. Step 3 deals with the compilation of the C/C++ code in the native language that is executed within each RTCP. Finally, in step 4, the files compiled in step 3 are downloaded into the RTCP (e.g., TMS320F28379D, dSPACE 1006 or OP5700).
3. Controller Description

As mentioned above, there are many control architectures for DC/AC SPCs. In order to cover the different control architectures for DC/AC SPCs that can be found in the literature, a PI controller in the $dq$ framework, a PR controller in the $\alpha\beta$ framework and an LQI controller in the $dq$ framework were chosen. Each control strategy was implemented in each of the RTCPs mentioned in Section 2. Figure 4 shows the general structure of the control of a DC/AC SPC connected to a non-linear load.

Each of the selected control strategies is composed of different mathematical functions, such as sine, cosine, matrix operations and integrations. Each control strategy is presented in more detail in the following subsections.

Figure 4. General structure of the control of a DC/AC SPC connected to a non-linear load.

3.1. Proportional-Resonant (PR) Controller

The voltage controller follows a cascade control architecture, using proportional-resonant (PR) controllers in the $\alpha\beta$ reference frame to track the reference voltage signal [52]. The general PR controller structure for DC/AC SPC is shown in Figure 5.

Figure 5. General structure of the PR controller for DC/AC SPC.
where \( V_{ref} \) is the amplitude voltage reference value; \( \omega_0 \) is the angular frequency; \( k_{pV}/k_{pI} \) and \( k_{iV}/k_{iI} \) are the proportional and resonant gains of the voltage and current controllers. The controllers are selected according to their zero steady-state tracking error characteristics and simple implementation [52,53]. A set of resonant controllers are used in the voltage and current control loops to suppress harmonic distortion in the voltage and ensure the sharing of harmonic currents.

3.2. Proportional-Integral (PI) Controller

The same analysis performed in the previous controller can be applied to a synchronous \( dq \) frame. In \( dq \) frame, the PR controller is replaced by a PI controller and harmonic frequencies are displaced to the left of the spectrum by \( \omega_0 \). Some of the applications of PI controllers for the V-I control level are presented in references [54,55]. The PI controller structure is shown in Figure 6.

![Figure 6. General structure of the PI controller for a DC/AC SPC.](image)

3.3. Linear-Quadratic-Integral (LQI) Controller

The optimal control aims to minimize a cost function and obtain the desired trajectories of the system with optimal performance [56–58]. LQI controllers are static feedback controllers based upon the augmented states (plant states \( (i_{dq}, i_{oq}, V_{Cdq}) \) plus integral of tracking error \( e_{dq} \)). The LQI controller is designed to minimize the tracking error by computing the integral of the error vector \( e_{dq} \).

As seen in Figure 7, unlike the PR (\( \alpha\beta \)) and PI (\( dq \)) control strategies, the LQI control strategy requires measuring all the states of the plant to be controlled. The reference signal \( V_{ref_{dq}} \) is generated in the same way as the previous controllers applying synchronous \( dq \) frame conversion. Finally \( K_{LQI} \) is the LQI gain matrix that multiplies each output of the multiplexer shown in Figure 7. More information on the design and implementation of this optimal control strategy is provided in [7,8,59].
4. Methodology for the Measurement of Computation Time Profiles

The measurements of the computation time profiles for each of the RTCPs mentioned above were evaluated. To observe the differences in the computation times in each RTCP used in this research in the DC/AC SPC control, the PI ($\alpha\beta$), PR ($dq$) and LQI ($dq$) controllers were used. On each of the RTCPs, each of the selected control strategies were implemented in separate Matlab/Simulink files. Each Matlab/Simulink file was divided into four subsystems called Analog-In, Controller, PWM and Monitoring, as shown in Figure 8.

Each of the subsystems contains the following blocks:

1. The Analog-In subsystem contains the ADC reading blocks and the conditioning of the voltage and current signals that will be used by the Controller. These change according to the RTCP under test in terms of quantity, resolution and conversion time.
2. The **Controller subsystem** contains the control strategy and generates the control signals that the **PWM subsystem** will use. The content of this subsystem varies according to the control strategy used and is the same for all RTCPs.

3. The **PWM subsystem** holds the PWM blocks that generate the switching signals that go to the power electronics stage (see Figure 1). The amount of blocks in this subsystem varies according to the PWM block configuration of each RTCP.

4. Finally, the **Monitoring subsystem** contains the parameters that can be changed in real-time, such as the gains of the controllers, the frequency and the reference voltage for the DC/AC SPC. In addition, it holds the blocks for calculating the active-reactive power, voltage–current RMS and THD values.

4.1. Matlab/Simulink Common Blocks in Real-Time Control of DC/AC SPC

The control strategies for the RTCP selected in this research were programmed in Matlab/Simulink. The most common function blocks used in the aforementioned controllers are listed in Table 2. ‘x’ indicates the presence of the component within the controller.

| Blocks                          | LQI ($dq$) | PR ($\alpha\beta$) | PI ($dq$) |
|---------------------------------|------------|--------------------|-----------|
| PWM                             | x          | x                  | x         |
| Duty Generator                  | x          | x                  | x         |
| DI                              | x          | x                  | x         |
| DO                              | x          | x                  | x         |
| ADC                             | x          | x                  | x         |
| Voltage Reference Generator     | x          | x                  | x         |
| Integrator                      | x          | x                  | x         |
| ABC/$dq$ Conversion             | x          | x                  |           |
| $dq$/ABC Conversion             | x          |                    |           |
| ABC/$\alpha\beta$ Conversion    | x          |                    |           |
| $\alpha\beta$/ABC Conversion    | x          |                    |           |
| RMS                             | x          | x                  | x         |
| THD                             | x          | x                  | x         |
| Matrix Product                  | x          |                    |           |

PWM is used by the DC/AC SPC to turn on and off the transistors within a certain period of time. Thus, the electrical effect at the output resembles a sinus wave. (DI) and (DO) are used as control flags between the inverter and the RTCP. The analog voltage and current measurements from the DC/AC SPC are converted to digital signals by the ADC. Typically, in DC/AC SPC applications, this conversion is done on the order of tens of kHz. The voltage reference generator block generates the voltage reference signal to be generated by the DC/AC SPC via the controllers. Usually, this reference signal has a magnitude/frequency of 120 VRMS/60 Hz or 220 VRMS/50 Hz, as required by local regulations.
The blocks described as converters have the task of transforming the reference frame of the signals captured by the ADC. The RMS and THD functions are considered for the validation of the performance of the controller. In summary, Table 2 shows the differences that exist between the control strategies used in this research in terms of blocks used in Matlab/Simulink. Their effect will be analyzed in the following sections.

### 4.2. Scenarios Designed to Compare RTCPs

In order to compare the computational power of the selected RTCPs, the computation time profiles of each of the most common blocks and functions in each control strategy were measured according to Table 2. This was done for each of the RTCPs. Additionally, to carry out a comparison between the time profiles of the RTCPs depending on the control strategy that is implemented, two scenarios were proposed, which are described below.

#### 4.2.1. Scenario 1: Voltage–Current Controllers

This scenario is used to compare the voltage–current control strategies’ computation times. Since the performance variables in a DC/AC SPC control change depending on the interests of the designer, the number of blocks vary according to the designer’s needs. For this reason, for each Matlab/Simulink file of each control strategy, two versions were created:

- The **full version** contained an instantaneous, three-phase active–reactive power calculation block with two first-order low pass filters and the RMS calculation blocks for the currents ($i_1$, $i_2$) and the voltages ($V_c$) of the LCL filter (see Figure 4). In addition, it contained an error calculation block for $V_c$.
- The **debugged version** only contained a block for calculating the RMS error of voltage $V_c$ and a block for calculating the RMS values of voltage $V_c$.

The creation of these two file versions was done to analyze the impact that each of these functions generates in the computation time profiles in each RTCP, which are of high mathematical complexity. In order to compare the computational power of the selected RTCPs, each subsystem’s computation time mentioned above was measured for the control strategies in all RTCPs.

#### 4.2.2. Scenario 2: Voltage Harmonic Controller

A DC/AC SPC’s high frequency switching makes THD management of voltage signals an important factor to take into account in the control. The higher the harmonic to be controlled, the more precision the ADC will require to capture the high frequencies and thus perform adequate control. Additionally, the PWM signals must have the ability to generate control signals that include those high frequencies. Since each RTCP has different ADC and PWM characteristics, it is necessary to compare the impacts they may have on the THD generated by the DC/AC SPC. For this reason, this scenario aims to compare the performance of each RTCP according to the impact it generates on the THD of $V_c$.

Due to the flexibility of the PR ($\alpha\beta$) control strategy regarding the selection of the resonant frequency, this control strategy is used for this scenario. To mitigate each harmonic in $V_c$, it is necessary to implement a new PR ($\alpha\beta$) voltage controller in parallel with the PR ($\alpha\beta$) controller of the fundamental frequency ($\omega_0$), as shown in Figure 9. In other words, for each harmonic, it is necessary to include a new block in the Matlab/Simulink of the PR ($\alpha\beta$) controller that was presented in Figure 5.

In this scenario, each PR ($\alpha\beta$) controller is gradually added for each harmonic, and the THD in $V_c$ generated by each of the RTCPs is recorded by a power analyzer. This is done until the DC/AC SPC becomes unstable with each RTCP; in this way the best THD and the best waveform that each RTCP generates are recorded.
5. Computation Time Profile Measurement

Each of the RTCPs has different hardware structures in terms of the characteristics of each of their components, as mentioned in Section 2. Despite the fact that the control strategies of all the RTCPs selected in this research were performed in Matlab/Simulink, each one of them has its own toolbox and blocks that allow one to configure and to program the ADC, DI, DO, PWM, etc. In addition, each RTCP has its own procedure to capture the computation times profiles of the different blocks that are implemented in Matlab/Simulink. Next, the methodology that must be carried out to measure the computation time profiles of each of the RCTPs used in this research is briefly explained.

5.1. Low-Cost µC-Based RTCP

To analyze the computation time that the TMS320F28379D microcontroller requires according to the model developed in Simulink, the support package for Texas Instruments C2000 was used. This package generated a report from a real-time simulation; each computational time unit per Simulink block was calculated with recorded data [60,61]. This tool normally takes the number of samples specified by the user and obtains the computation time of each programmed block offline. Finally, the generated report describes maximum, minimum and average computation times, and the number of calls of the function or block. Additionally, a time window is generated where each task is separated for each time interval. In this way it is possible to know which blocks have priority over others.

5.2. dSPACE Scalexio RTCP

To measure the computation times of the blocks and the function in Matlab/Simulink, dSPACE Scalexio uses the subsystems generated within the model and new sources (lines within the profiler GUI) during the initialization of the system in the dSPACE Profiler. The dSPACE Profiler is used to analyze the time behavior (real-time tasks) of dSPACE applications running on a platform. The Profiler displays events in a line chart with event markers and a taskbar. The new fonts mentioned above are registered and displayed in the dSPACE Profiler with the event label to return the identifier of the font [62].
System initialization uses block code to create the profiler source. This code runs once during system initialization and creates the necessary source to generate events within subsystems. Likewise, the subsystem parameters to be measured must be selected as an atomic subsystem to take the subsystem as a unit and measure the start and end times without considering other blocks or subsystems. The out function parameters for the time measurement must be included in out function block to be executed at the end of the subsystem execution. Together, two events (System Initialization and System Exits) create a visible sample period within the dSPACE Profiler. The parameters of the output function for time measurement must be included in an output function block that will be executed at the end of the subsystem execution.

5.3. dSPACE 1006 RTCP

For this RTCP, for each task in the real-time program, there is a variable named turnaround time (RTI), which is available in the Task Information Variable description files group of dSPACE software, ControlDesk. This variable indicates how long it takes to run the entire task. However, to measure the time of the blocks in dSPACE 1006, the RTCP measures the computation time of the parts of the model through atomic subsystems, which means that the RTCP controls the execution parts in the model. Additionally, it forces Simulink to execute all the blocks in that subsystem as if they were their own separate “functions”. In addition, other blocks, such as Data Store Memory and a Data Store Read, are involved in the process of measurement time; likewise, for the dSPACE Scalexio to measure computation time, it is necessary to include the System Outputs Function block, and add code into the block. For more details on how to implement this process, see [22]. Finally, the time measurement is shown on ControlDesk as a event in a line chart with event markers and a taskbar.

5.4. OPAL-RT OP5700 RTCP

With the use of the Simulink library “Monitoring” by RT-LAB, the user can generate in real-time the computation time profile of a specific block from a model, and the full model if needed. The blocks to be monitored must be located inside a subsystem configured as an atomic unit [63]. On the other hand, by enabling the monitoring option in the RT-LAB software, the “MonitoringViewer” will show in real-time the name of the subsystem or block, with values of their respective minimum times, maximums and averages. It is worth mentioning that these computation times are expressed in $\mu$s with two decimals.

6. Experimental Results

For this section, the parameters provided in Table 3 were used. The experimental results were obtained using the microgrid test-bench of the University of Puerto Rico-Mayaguez Campus [64]; see Figure 10. This configuration uses four Danfoss 2.2 kVA DC/AC SPCs with LCL filters, four sensor boxes with voltage—current LEM sensors and one non-linear load. Due to the complexity of physical connections between the DC/AC SPC, the sensor box and the RTCP, each RTCP was connected to one of the DC/AC SPCs in the setup and independent tests were carried out. All compared RTCPs were programmed with code generated by Matlab/Simulink, both for Simulink Coder and for the C compiler, with their default optimization parameters. In order to make a fair comparison, the space vector—Triangular 1/4 PWM technique was used in all experiments [65].
Table 3. Setup parameters.

| Parameter                          | Value                |
|------------------------------------|----------------------|
| Switching frequency, \( f_s \)    | 10 kHz               |
| Sampling Period, \( T_s \)         | 100 µs               |
| Nominal frequency, \( f_o \)       | 60 Hz                |
| Rated voltage, \( V \)            | 120 V\text{RMS}     |
| Filter capacitance, \( C_f \)     | 8.8 µF               |
| Filter internal inductance, \( L_i \)| 1.8 mH               |
| Filter output inductance, \( L_o \)| 1.8 mH               |
| DC link voltage, \( V_{DC} \)     | 350 V                |

Control Strategy

|           | \( K_{IV}, K_{PV}, K_{II}, K_{Pl} \) | 1.5, 100, 0.1, 200 |
|-----------|--------------------------------------|-------------------|
| PI (\( dq \)) | \( K_{RV}, K_{PV}, K_{RI}, K_{PI} \) | 1.5, 100, 0.1, 200 |
| PR (\( \alpha \beta \)) | \( K_{LQI} \) | \[a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9 \ a_{10}\] |

Non-Linear Load

|                     |                     |
|---------------------|---------------------|
| Capacitance, \( C_1 \) | 100 µF               |
| Inductance, \( L_1 \)  | 2 mH                |
| Resistance, \( R_1 \)   | 250 Ω               |

where \( a_1 = 4.23, a_2 = 0.215, a_3 = 0.638, a_4 = 0.038, a_5 = -0.337, a_6 = -0.021, a_7 = 0.321, a_8 = 0.011,\)
\( a_9 = -179.31, a_{10} = 5.37, b_1 = -0.215, b_2 = 4.23, b_3 = -0.03, b_4 = 0.638, b_5 = 0.021, b_6 = -0.337, b_7 = -0.011,\)
\( b_8 = 0.321, b_9 = -5.37, b_{10} = -179.31.\)

Figure 10. Microgrid Experimental Platform. 1—dSPACE 1006, 2—TI µC C2000 F28379D, 3—Danfoss 2.2 kVA DC/AC SPCs, 4—LCL output filters, 5—linear load, 6—PA2203A KEYSIGHT Power Analyzer, 7—non-linear load, 8—sensor boxes, 9—dSPACE-SCALEXIO, 10—OPAL-RT OP5700.

6.1. Computation Time Profiles per Block

The computation time profiles of the block functions described in Section 3 are shown in Figure 11. The computation times were taken with a sampling time of \( T_s = 100 \) µs for each block, from the different RTCPs. The results show differences in the computation times of the selected blocks depending on the RTCP; however, the average time consumption of the block of the RTCPs is due to the capabilities and characteristics of each one. The highest average time consumption was for the low-cost µC-based RTCP. In contrast, the OP5700
evidently had the shortest average block computation time, followed by dSPACE Scalexio and dSPACE 1006.

Figure 11. Computation time profile of each block in each RTCP.

Some computation times present similarities among blocks. For example, PWM had high consumption of resources in all RTCPs due to its high average computation time, and this was also true for ADC blocks.

Isolated cases were also observed. The low-cost µC-based RTCP’s computation time not only showed high consumption of resources in the PWM and ADC, but the voltage reference block (yellow bar) was the block with the highest computation time, due to a mathematical process that involves the creation of a sinusoidal signal, as described in Section 4.1.

On the contrary, the conversion frame blocks DI and DO do not consume a lot of resources in the RTCPs; therefore, they have the lowest computation times.
6.2. Scenario I: Computation Time Profiles for Voltage–Current Controllers

Each of the voltage–current control strategies presented in Section 3 was implemented in each of the RTCPs described in Section 2 using Matlab/Simulink. For each voltage–current control strategy in each RTCP, the computation time profiles were measured for each of the Analog-In, Controller, PWM and Monitoring subsystems, as described in Section 4.2.1. Measurement of the computation time profiles was done according to the RTCP, as described in Section 5. For each file version of each control strategy, time computation profiles are presented in bar graphs for each subsystem of each RTCP with respect to $T_s$. Comparative bar graphs of the total computation times taken by each voltage–current control strategy in each RTCP are also presented.

6.2.1. PR ($\alpha\beta$) Controller

The results of the computation time profiles for the full and debugged versions of the PR ($\alpha\beta$) controller (see Section 4.2) are shown in Figure 12 and Figure 13, respectively. It can be observed that the low-cost $\mu$C-based presents the highest computation time profiles in all the subsystems while the OP-5700 presents the lowest. Additionally, it is observed that for the two versions the subsystem with the highest computation time in all RTCPs is the Monitoring subsystem, which indicates that the RMS blocks generate high computation times due to their mathematical complexity. In both versions the times of the controller and PWM subsystems remain the same since the Debugged version only affects the Analog-In and Monitoring subsystems. In addition, in the Debugged version it is observed that the times of the Monitoring subsystem are considerably reduced in the low-cost $\mu$C-based RTCP compared to the other RTCPs.

![Figure 12. Computation time profiles for the PR ($\alpha\beta$) controller of the Full version file.](image1)

![Figure 13. Computation time profiles for the PR ($\alpha\beta$) controller of the Debugged version file.](image2)
On the other hand, it would be expected that the time profiles for the Analog-In subsystem will decrease in all platforms since it does not include power calculation block, in other words the ADC blocks are removed for the $I_\text{c}$ currents. On the contrary, it is observed that the Analog-In subsystem times for the OPAL-RT OP-5700 RTCP remain the same, this is because there is a single ADC configuration block in Matlab/Simulink for the OP-5700 RTCP that includes all the ADC ports. This means that all the ADC ports of the OPAL-RT OP-5700 RTCP will be working at the same time even if they are not necessary. This may be a disadvantage, but the computation time is still significantly lower than the other RTCPs.

### 6.2.2. PI ($dq$) Controller

The results of the computation time profiles for the **full** and **debugged** versions of the PI ($dq$) controller (see Section 4.2) are shown in Figure 14 and Figure 15, respectively. It can be observed that the time profiles of the Analog-In, PWM and Monitoring subsystems are approximately the same in both versions with respect to the PR ($\alpha\beta$) controller, since the only change is the control strategy within the Controller subsystem—in this case, the PI controller ($dq$). Additionally, it can be observed that the low-cost µC-based RTCP presents the highest computation time profiles in all the subsystems, and the OPAL-RT OP-5700 RTCP presents the lowest. On the other hand, the Controller subsystem has a lower computation time profile compared to the PR ($\alpha\beta$) controller, which indicates that the PI ($dq$) controller consumes less resources.
6.2.3. LQI (dq) Controller

The results of the computation time profiles for the full and debugged versions of the LQI (dq) controller (see Section 4.2) are shown in Figure 16 and Figure 17, respectively. As for the other two controllers, it is noted that the computation times of the Analog-In, PWM and Monitoring subsystems are the same in both versions, since the only change is the control strategy within the Controller subsystem, in this case the LQI (dq) controller. The time profile of the Controller subsystem was too small in the low-cost µC-based RTCP, so it was impossible to measure it; hence, it is represented in the bar graph as N/M. On the other hand, the computation time profiles of the other RTCPs have values around the profiles of the other control strategies. Unlike the previous controllers, the LQI (dq) does not present changes in the computation times of the Analog In subsystem; this is because, being a state feedback controller, it requires all the voltage and current measurements from the DC/AC SPC, which means it uses more ADC blocks than the PI (dq) and PR(αβ) controllers.

Figure 16. Time Profiling for LQI (dq) controller, full.

Figure 17. Time Profiling for LQI (dq) controller, debugged.

6.2.4. Total Computation Time Profiles

The results of the total computation time profiles for full and debugged versions of each control strategy are shown in Figure 18 and Figure 19, respectively. It can be observed that for both the full and debugged versions, the low-cost µC-based RTCP presents the highest computation time profiles and the OPAL-RT OP-5700 RTCP the lowest for each of the control strategies. The profiles of the RTCPs dSPACE 1006, dSPACE Scalexio and
OP-5700 present computation times profiles with around the same values for each of the control strategies in both versions.

Similarly, it can be observed that for the **debugged** version of the low-cost μC-based RTCP that the computation time profiles drop considerably for all control strategies. This is directly related to the RMS function blocks within the Monitoring subsystem. Thus, if a designer wants to use a low-cost μC-based RTCP of the C2000 branch, it is advisable not to obtain the RMS values in real-time and rather to obtain them after performing the experiment due to the high computation times, which is due to its mathematical complexity.

On the other hand, in the low-cost μC-based RTCP, the LQI controller’s (dq) computation times are considerably lower compared to the other two control strategies’ times. This means that the impact on the computation time profiles of the blocks of matrix operations, such as those contained in the LQI controller (dq), is less than that of the integrators that contain the PR (αβ) and PI (dq). For this reason, if the designer wants or needs to obtain several RMS values at the same time in an RTCP such as the dSPACE 1006 or a low-cost μC-based RTCP, it is recommended that he implement control strategies with the minimum of integrators, such as optimal control strategies involving only matrix products.

![Figure 18. Time profiling for full block controllers.](image1)

![Figure 19. Time profiling for debugged block controllers.](image2)
Finally, it is observed that the RTCPs such as the dSPACE Scalexio and OP-5700 have very small computation time profiles in both versions compared to maximum computation time, $T_s = 100$ μs. This indicates that in these RTCPs we can calculate multiple RMS values at the same time without any problem, and use more ADC and PWM ports if the hardware allows it, since the computation time they have remaining compared to $T_s$ is approximately 98%. This means that with these platforms it is possible to control several three-phase DC/AC SPCs at the same time. On the other hand, with the low-cost μC-based RTCP, it is only possible to control a single three-phase DC/AC SPC because it is limited in computational time and in the number of ADC ports. In turn, according to the observed computation time profiles, it can be said that it is possible to control a maximum of three three-phase DC/AC SPC with dSPACE 1006 in terms of processing, given that the debugged version takes about 30% of its total computing capacity. In addition, this RTCP allows one to connect more ADC, PWM, DI and DO modules, thereby increasing their capacities but generating a higher cost.

### 6.3. Scenario II: Harmonics Compensator with Voltage Harmonics Controller

For this scenario, the parameters of the PR($\alpha\beta$) controllers to compensate for harmonics are shown in Table 4. The maximum harmonic to compensate for is the 17th, which is equal to 1020 Hz. This frequency is the maximum recommended frequency to control, since it is a decade before the switching frequency ($f_s = 10$ kHz). It is at the limit of instability. When trying to compensate for harmonics higher than the 17th, the DC/AC SPC becomes unstable.

**Table 4. Setup parameters for harmonic compensators.**

| Harmonics Compensator ($\alpha\beta$) | Parameter | Value |
|-------------------------------------|-----------|-------|
| Fundamental ($\omega_0$)            | $K_{RV}$, $K_{PV}$, $K_{RI}$, $K_{PI}$ | 1.5, 100, 0.1, 200 |
| Fifth Harmonic ($5\omega_0$)        | $K_5$     | 80    |
| Seventh Harmonic ($7\omega_0$)      | $K_7$     | 95    |
| Eleventh Harmonic ($11\omega_0$)    | $K_{11}$  | 65    |
| Thirteenth Harmonic ($13\omega_0$) | $K_{13}$  | 105   |
| Seventeenth Harmonic ($17\omega_0$) | $K_{17}$  | 100   |

The measurement of the harmonic content of the $V_c$ signal was carried out with a PA2203A KEYSIGHT power analyzer, as shown in Figure 10, highlighted with the number 6. Following the methodology described in Section 4.2.2, harmonic compensators were added for each RTCP. Harmonic voltage content was measured each time a harmonic compensator was added to the Controller subsystem for each RTCP. For each RTCP, bar graphs are presented comparing the effects on the harmonic content of $V_c$ as compensators are added. A bar graph is also presented comparing the THD in each RTCP each time harmonic compensators are added. In the bar graphs of the harmonic contents for each RTCP, the term noHC refers to the harmonic content of the $V_c$ signal only with the PR ($\alpha\beta$) controller at the fundamental frequency ($\omega_0$). Similarly, HC$_5$ represents the addition of the compensators of the 5th harmonic, HC$_{(5,7)}$, of the 5th and 7th harmonics, and so on.

#### 6.3.1. Low-Cost μC-Based RTCP

The results of the harmonic contents of the $V_c$ signal for each compensator added in the low-cost μC-based RTCP are shown in Figure 20.

It can be observed that the maximum compensator that could be implemented was the HC$_{(5,7)}$ due to the fact that with higher compensators, the system presented unwanted behaviors. When adding the compensator HC$_{(5,7,11)}$, the DC/AC SPC remained stable, and the harmonic content increased, but the fundamental frequency ($\omega_0$) of the system was reduced from 60 to 45.44 Hz. In the same way, by adding the compensator HC$_{(5,7,11,13)}$, $\omega_0$, it was reduced to 30.02 Hz and after several seconds became unstable. This indicates
that the low-cost \( \mu \)C-based RTCP does not have the ability to compensate for harmonics higher than 7th. In other words, the ADC and PWM of this RTCP do not have the ability to measure/generate these high frequency signals, which causes unwanted behaviors in the voltage and current signals of the DC/AC SPC.

On the other hand, it was observed that by not adding harmonic compensators (noHC), the 5th harmonic (orange bar) presented a higher percentage with respect to the others, which is reflected in the voltage \( V_c \) THD percentage. This was substantially reduced from 2.6% to 0.2% by adding the HC$_5$ compensator. Similarly, by adding compensator HC$_{(5,7)}$, the percentage of the 7th harmonic was reduced from 1.4% to 0.1%. This shows that the low-cost \( \mu \)C-based RTCP presents desirable results in harmonic compensation with frequencies from 300 to 420 Hz in DC/AC SPC applications.

6.3.2. dSPACE 1006 RTCP

The results of the harmonic contents of the \( V_c \) signal for each compensator added in the dSPACE 1006 RTCP are shown in Figure 21. Unlike the low-cost \( \mu \)C-based RTCP, it was observed that in the dSPACE 1006 RTCP, the maximum compensator that could be implemented was HC$_{(5,7,11,13)}$, but for noHC, it presented a content harmonic higher than low-cost \( \mu \)C-based RTCP. In turn, by adding the 17th harmonic compensator, the DC/AC SPC became unstable, indicating that the dSPACE 1006 RTCP has the ability to compensate for harmonics up to 780 Hz when it has a sampling frequency of 10 kHz. On the other hand, it is shown that by gradually adding harmonic compensators from noHC to HC$_{(5,7,11,13)}$, the percentage of harmonic content decreased according to the compensator added, thereby fulfilling its objective.

Figure 20. Harmonic \( V_c \) content controlled by low-cost \( \mu \)C-based RTCP.

On the other hand, it was observed that by not adding harmonic compensators (noHC), the 5th harmonic (orange bar) presented a higher percentage with respect to the others, which is reflected in the voltage \( V_c \) THD percentage. This was substantially reduced from 2.6% to 0.2% by adding the HC$_5$ compensator. Similarly, by adding compensator HC$_{(5,7)}$, the percentage of the 7th harmonic was reduced from 1.4% to 0.1%. This shows that the low-cost \( \mu \)C-based RTCP presents desirable results in harmonic compensation with frequencies from 300 to 420 Hz in DC/AC SPC applications.

6.3.2. dSPACE 1006 RTCP

The results of the harmonic contents of the \( V_c \) signal for each compensator added in the dSPACE 1006 RTCP are shown in Figure 21. Unlike the low-cost \( \mu \)C-based RTCP, it was observed that in the dSPACE 1006 RTCP, the maximum compensator that could be implemented was HC$_{(5,7,11,13)}$, but for noHC, it presented a content harmonic higher than low-cost \( \mu \)C-based RTCP. In turn, by adding the 17th harmonic compensator, the DC/AC SPC became unstable, indicating that the dSPACE 1006 RTCP has the ability to compensate for harmonics up to 780 Hz when it has a sampling frequency of 10 kHz. On the other hand, it is shown that by gradually adding harmonic compensators from noHC to HC$_{(5,7,11,13)}$, the percentage of harmonic content decreased according to the compensator added, thereby fulfilling its objective.
6.3.3. dSPACE Scalexio RTCP

The results of the harmonic contents of the $V_c$ signal for each compensator added in the dSPACE Scalexio RTCP are shown in Figure 22. Like the dSPACE 1006 RTCP, it is shown that in the dSPACE Scalexio RTCP, the maximum harmonic compensator that could be added was $HC_{(5,7,11,13)}$, since when adding the compensator of the 17th harmonic, the DC/AC SPC became unstable. It is also shown that for $noHC$, the harmonic content is higher compared to the two RTCPs already mentioned, but when adding the $HC_5$ compensator, the harmonic content decreases to the same values as for the other RTCPs. In general, by gradually adding compensators, the value of the harmonic contents decreases, which indicates that this RTCP can be used for harmonic control in DC/AC SPC applications.
6.3.4. OPAL-RT OP5700 RTCP

The results of the harmonic contents of the $V_c$ signal for each compensator added in the OPAL-RT OP5700 RTCP are shown in Figure 23. Unlike the RTCP already mentioned, this RTCP allowed us to add compensators up to compensator $HC_{(5,7,11)}$, since when adding the compensator of the 13th harmonic, the system becomes unstable. It was also observed that for noHC, the harmonic content was the highest of all RTCPs, but in the same way, when adding compensator $HC_{5}$, the harmonic content decreased to the same values as for the other RTCPs. In general, like the other RTCPs, by gradually adding each voltage compensator, the value of the harmonic contents decreases, indicating that this RTCP can be used for harmonic control in DC/AC SPC applications.
The THD results of the $V_c$ signal for each RTCP when adding each compensator are shown in Figure 24. It can be observed that for each added compensator, the low-cost µC-based RTCP has the lowest THD till $HC_{(5,7)}$, which is up to the harmonic for which this RTCP provides desirable results. This is due to the characteristics of ADC as already mentioned, but it is also related to the fact that this RTCP, compared to the others, has the computational engine with the lowest processing capacity, as shown in Table 1. On the other hand, the THD for the $HC_{(5,7,11)}$ compensator is shown just to show that by adding this compensator, the THD increases from 2.1% to 3.4% as the fundamental frequency of the DC/AC SPC decreases.

The dSPACE 1006 RTCP presents the best performance of all the RTCPs in terms of voltage harmonic compensation, since it allows one to add compensator up to $HC_{(5,7,11,13)}$ and also generates the lowest THD of the RTCPs. This is because this RTCP has dedicated
hardware for the PWM generation and the second shortest ADC conversion time, as shown in Table 1.

On the other hand, the dSPACE Scalexio RTCP, despite allowing compensator to be added up to $HC_{(5,7,11,13)}$ and having a short conversion time for the ADC, as shown in Table 1, it presented the highest THD for all the added compensators. This is attributed to the fact that it does not have dedicated hardware for the PWM, since the ports are shared with DI-DO, and the documentation is quite limited, which can lead to an erroneous configuration.

The OPAL-RT OP5700 RTCP, despite being the most expensive, has the same limitations as the other RTCPs and does not generate the lowest THD, since it also has shared PWM blocks. It is expected that better results could be obtained in terms of harmonic compensation; they could be improved if the ADC hardware is exchanged for other hardware that has a shorter conversion time.

For the harmonic compensation groups $noHC$, $HC_5$ and $HC_{(5,7)}$, it is shown that the low-cost µC-based RTCP and dSPACE 1006 RTCP present the best results in terms of harmonic compensation. These results are due to the fact that these platforms have dedicated hardware for the PWM.

![Figure 24](image-url) THD for each compensator added in each of the RTCPs.

Finally, to experimentally demonstrate the results of the voltage harmonic controllers, Figure 25 shows the best waveforms achieved by each of the RTCPs. It can be observed that the best $V_c$ waveform when implementing harmonic compensators was obtained with the low-cost µC-based RTCP with a THD of 2.14% and the highest with the dSPACE Scalexio RTCP with a THD of 2.73%. It should be noted that in the low-cost µC-based RTCP, only compensators up to $HC_{(5,7)}$ were added, whereas for the dSPACE Scalexio RTCP, compensators up to $HC_{(5,7,11,13)}$ were used.
7. Discussion

This paper presented benchmarks to illustrate the computing power of four different RTCPs with different characteristics programmed with code automatically generated from Matlab/Simulink in DC/AC SPC control applications. The aim of this paper is to provide information that allows researchers to select the most appropriate RTCP according to their control interests in DC/AC SPCs. In each of the RTCPs, three different strategies of voltage–current control and harmonic controllers were implemented. The computation times generated in each RTCP and the THD in the capacitor voltage signal of the DC/AC SPC were compared.

In general terms, the RTCPs compared fulfill their purpose, which is to control the voltage and current in the DC/AC SPC, although each one has different computation times and generates different THD in the voltage signal. The low-cost µC-based RTCP takes more time for each of the different Matlab/Simulink blocks than the other RTCPs. Excessive inclusion of RMS and THD calculation blocks in this RTCP can cause time overruns and instability of the DC/AC SPC. For this reason, for voltage and current control applications for three-phase DC/AC SPC, it is advisable to use the least THD and RMS blocks possible, and if possible, perform these operations after running the experiment. Despite not allowing the compensation of harmonics higher than the 7th, this RTCP presents better results in terms of THD generated in the DC/AC SPC compared to the other RTCPs, making it attractive due to its cost–performance ratio. The main limitation of this RTCP is that it is
only possible to control a three-phase DC/AC SPC at a time due to its limited number of ADCs and that a three-phase DC/AC SPC takes about 70% of its computing capacity.

The computation times of the other RTCPs are much shorter compared to the low-cost μC-based RTCP, which demonstrates their high computational power. The RTCP dSPACE 1006 showed good results in the implementation of voltage–current controllers and harmonic control similar to the low-cost μC-based RTCP. This RTCP allows the integration of higher harmonic compensators, and their total computation times are shorter. On the other hand, the RTCPs dSPACE Scalexio and OPAL-RT 5700 present much lower computation times, but their performances in the control of harmonics are poorer.

Unlike the low-cost μC-based RTCP, the dSPACE 1006, dSPACE Scalexio and OPAL-RT OP5700 RTCPs allow the control of more than one three-phase DC/AC SPC at the same time, due to their high computing power and greater numbers of ADCs. However, their costs are too high, and each increase in hardware generates more costs. The dSPACE 1006 and dSPACE Scalexio RTCPs have dedicated hardware and display software that make them superior to the other RTCPs in terms of operability and flexibility. On the other hand, the structure of the program in Matlab/Simulink that the OPAL-RT 5700 RTCP must have is more complex and makes the implementation of controllers difficult compared to the other RTCPs. In addition, to date this RTCP does not have specialized visualization software, and visualization must be performed directly in Matlab/Simulink scopes, which generates delays with what is really happening in the DC/AC SPC.

**Author Contributions:** Conceptualization, J.D.V.-P.; methodology, J.D.V.-P., A.F.L.-C., E.A.S.-T. and J.F.P.-M.; software, J.D.V.-P., A.F.L.-C. and E.A.S.-T.; validation, A.F.L.-C., E.A.S.-T. and J.D.V.-P.; formal analysis, J.D.V.-P.; resources, F.A.; writing—original draft preparation, J.D.V.-P., A.F.L.-C. and E.A.S.-T.; writing—review and editing, J.F.P.-M. and F.A.; supervision, J.F.P.-M. and F.A.; funding acquisition, F.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper is supported by the U.S. Department of Energy Office of Science, Office of Basic Energy Sciences, EPSCoR Program; Office of Electricity, Microgrid R&D Program; and Office of Energy Efficiency and Renewable Energy, Solar Energy Technology Office under EPSCoR grant number DE-SC0020281.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

**References**

1. Rashid, M.H. *Alternative Energy in Power Electronics*, 1st ed.; Elsevier Inc.: Amsterdam, The Netherlands, 2014; pp. 1–363. [CrossRef]
2. Corradini, L.; Maksimovic, D.; Mattavelli, P.; Zane, R. *Digital Control of High-Frequency Switched-Mode Power Converters*; Wiley-IEEE Press: Piscataway, NJ, USA, 2015; p. 357.
3. *IEEE Std. 1547-2018*; Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces. IEEE Standard Association: Piscataway, NJ, USA, 2018; pp. 1–138. [CrossRef]
4. Vasquez-Plaza, J.D.; Patarroyo-Montenegro, J.F.; Campo-Ossa, D.D.; Sanabria-Torres, E.A.; Lopez-Chavarro, A.F.; Andrade, F. Formal Design Methodology for Discrete Proportional-Resonant (PR) Controllers Based on Sisotool/Matlab Tool. In Proceedings of the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 8–21 October 2020; Volume 2020, pp. 3679–3684. [CrossRef]
5. Curi Busarello, T.D.; Zeb, K.; Simões, M.G. Highly Accurate Digital Current Controllers for Single-Phase LCL-Filtered Grid-Connected Inverters. *Electricity 2020*, 1, 12–36. [CrossRef]
6. Han, Y.; Shen, P.; Zhao, X.; Guerrero, J.M. Control Strategies for Islanded Microgrid Using Enhanced Hierarchical Control Structure with Multiple Current-Loop Damping Schemes. *IEEE Trans. Smart Grid 2017*, 8, 1139–1153. [CrossRef]
7. Patarroyo-montenegro, J.F.; Vasquez-plaza, J.D.; Andrade, F. A State-Space Model of an Inverter-Based Microgrid for Multivariable Feedback Control Analysis and Design. *Energies 2020*, 13, 3279. [CrossRef]
8. Lopez-Chavarro, A.F.; Patarroyo-Montenegro, J.F.; Sanabria-Torres, E.A.; Campo-Ossa, D.D.; Vasquez-Plaza, J.D.; Andrade-Rengifo, F. A Design Algorithm for Multivariable Linear Quadratic Integral Controllers in Voltage-Source Converters. In Proceedings of the IECNON Proceedings (Industrial Electronics Conference), Singapore, 8–21 October 2020, pp. 4025–4030. [CrossRef]

9. Valencia-Rivera, G.H.; Merchán-Villalba, L.R.; Tapia-Tinoco, G.; Lozano-García, J.M.; Ibarra-Manzano, M.A.; Avina-Cervantes, J.G. Hybrid LQR-Pi control for microgrids under unbalanced linear and nonlinear loads. *Mathematics* 2020, 8, 1096. [CrossRef]

10. Petkova, M.; Antchev, M.; Gourgouliatsof, V. Investigation of Single-Phase Inverter and Single-Phase Series Active Power Filter with Sliding Mode Control. *Sliding Mode Control 2011*. [CrossRef]

11. Viswadev, R.; Mudlapur, A.; Ramana, V.V.; Venkatesaperumal, B.; Mishra, S. A Novel AC Current Sensorless Hysteresis Control for Grid-Tie Inverters. *IEEE Trans. Circuits Syst. II Express Briefs* 2020, 67, 2577–2581. [CrossRef]

12. Sahoo, S.K.; Sinha, A.K.; Kishore, N.K. Control Techniques in AC, DC, and Hybrid AC-DC Microgrid: A Review. *IEEE J. Emerg. Sel. Top. Power Electron.* 2018, 6, 738–759. [CrossRef]

13. Mehta, S. A comprehensive review on control techniques for stability improvement in microgrids. *Int. Trans. Electr. Energy Syst.* 2021, 31, e12822. [CrossRef]

14. Rosso, R.; Member, S.; Wang, X.; Member, S. Grid-Forming Converters: Control Approaches, Grid-Synchronization, and Future Trends—A Review. *Ind. Appl. 2021*, 2, 93–109. [CrossRef]

15. Ansari, S.; Chandel, A.; Tariq, M. A Comprehensive Review on Power Converters Control and Control Strategies of AC/DC Microgrid. *IEEE Access* 2021, 9, 17998–18015. [CrossRef]

16. Gao, F.; Kang, R.; Cao, J.; Yang, T. Primary and secondary control in DC microgrids: A review. *J. Mod. Power Syst. Clean Energy* 2019, 7, 227–242. [CrossRef]

17. Luo, F.L.; Ye, H.; Rashid, M.H.M. *Digital Power Electronics and Applications*; Elsevier Academic: Amsterdam, The Netherlands, 2005; p. 408.

18. Buccella, C.; Cecati, C.; Latafat, H. Digital control of power converters—A survey. *IEEE Trans. Ind. Inform.* 2012, 8, 437–447. [CrossRef]

19. Liu, Y.F.; Sen, P. Digital control of switching power converters. In Proceedings of 2005 IEEE Conference on Control Applications,CCA 2005, Toronto, ON, Canada, 29–31 August 2005; pp. 635–640. [CrossRef]

20. Maksimovic, D.; Zane, R.; Erickson, R. Impact of digital control in power electronics. In Proceedings of the International Symposium on Power Semiconductor Devices & ICs, Orlando, FL, USA, 18–22 May 2008; pp. 13–22. [CrossRef]

21. Yousefzadeh, V. Advances in digital power control. In Proceedings of the INTELEC, International Telecommunications Energy Conference, Orlando, FL, USA, 6–10 June 2010. [CrossRef]

22. dSPACE Systems. *Products—dSPACE*; dSPACE Systems: Paderborn, Germany, 2021.

23. OPAL-RT Technologies. *HIL Testing | FPGA Technology | RCP Technologies*; OPAL-RT Technologies: Montréal, QC, Canada, 2021.

24. Typhoon HIL. *HIL Hardware—Typhoon HIL; Typhoon HIL: Somerville, MA, USA, 2021.*

25. RTDS Technologies. *Power Electronics HIL—RTDS Technologies*; RTDS Technologies: Winnipeg, MB, Canada, 2021.

26. *Plexim: Electrical Engineering Software; RT Box *| Plexim: Zürich, Switzerland, 2021.

27. Imperix—Rapid Control Prototyping Solutions for Power Electronics. Power Electronic Controllers—Programmable Digital Controllers, 2021. Available online: https://imperix.com/products/power-electronic-controllers/ (accessed on 8 September 2022).

28. PED-Board. *PED-Board|Just Add Power*. Available online: https://www.ped-board.com/ (accessed on 8 September 2022).

29. Bastos, R.F.; Fuzato, G.H.; Neves, R.V.; Machado, R.Q. Model, design and implementation of a low-cost HIL for power converter and microgrid emulation using DSP. *IET Power Electron.* 2019, 12, 3833–3841. [CrossRef]

30. Vasquez Plaza, J.D.; Patarroyo-Montenegro, J.F.; Andrade, F. Development and Implementation of a Low-Cost Research Platform for Control Applications for Inverter-Based Generators. In Proceedings of the 2020 22nd European Conference on Power Electronics and Applications, EPE 2020 ECCE Europe, Lyon, France, 7–11 September 2020; pp. 1–9. [CrossRef]

31. Aravena, J.; Carrasco, D.; Diaz, M.; Uriarte, M.; Rojas, F.; Cardenas, R.; Travieso, J.C. Design and implementation of a low-cost real-time control platform for power electronics applications. *Energies* 2020, 13, 1527. [CrossRef]

32. Texas Instruments Inc. *C2000™ Real-Time Microcontrollers*; Texas Instruments Inc.: Dallas, TX, USA, 2021.

33. Yuan, S.; Shen, Z. The design of Matlab-DSP development environment for control system. In Proceedings of the 2012 3rd International Conference on Digital Manufacturing and Automation, ICDMA 2012, Guilin, China, 31 July–2 August 2012; pp. 903–906. [CrossRef]

34. Patil, P.H.; Kapse, K. Transition from Simulink to MATLAB in real-time digital signal processing. In Proceedings of the 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (IEESCO), Visakhapatnam, India, 24–25 January 2015; Volume 21. [CrossRef]

35. Hýl, R.; Wagnerová, R. Fast development of controllers with Simulink Coder. In Proceedings of the 2017 18th International Carpathian Control Conference, ICCC 2017, Sinaia, Romania, 28–31 May 2017; pp. 406–411. [CrossRef]

36. Dos Santos, B.; Araujo, R.E.; Varajao, D.; Pinto, C. Rapid Prototyping Framework for real-time control of power electronic converters using simulink. In Proceedings of the IECNON Proceedings (Industrial Electronics Conference), Vienna, Austria, 10–13 November 2013, pp. 2303–2308. [CrossRef]
