



Repositorio Institucional de la Universidad Autónoma de Madrid
<https://repositorio.uam.es>

Esta es la **versión de autor** del artículo publicado en:
This is an **author produced version** of a paper published in:

13th European Space Power Conference (ESPC), Elche, Spain, 2023

DOI: <https://doi.org/10.1109/ESPC59009.2023.10298161>

Copyright: © 2023 IEEE

El acceso a la versión del editor puede requerir la suscripción del recurso
Access to the published version may require subscription

Digital controllers design using the *ESA Control Toolbox* in MATLAB Simulink

1st Alberto Sanchez
HCTLab Research Group
Universidad Autónoma de Madrid
Madrid, Spain
alberto.sanchezgonzalez@uam.es

2nd Elyas Zamiri
HCTLab Research Group
Universidad Autónoma de Madrid
Madrid, Spain
elyas.zamiri@uam.es

3rd Angel de Castro
HCTLab Research Group
Universidad Autónoma de Madrid
Madrid, Spain
angel.decastro@uam.es

Abstract—While the demanded performance of power converter controllers keeps on increasing, the design time is also expected to decrease. This becomes an even more significant issue in space applications, where design quality is critical. In order to fulfill the desired performance, FPGA becomes a valuable option for implementing advanced digital controller algorithms for power converters due to its high component density, speed, and reconfigurability. However, implementing controller algorithms in FPGAs is not trivial, particularly for designers unfamiliar with VHDL. The main objective of this study is to provide a MATLAB Simulink Toolbox, i.e., library, to implement power converter controllers algorithms in FPGAs without requiring prior knowledge of VHDL or any other hardware description language. The proposed library consists of essential building blocks needed for designing these controllers, all of which are synthesizable in any FPGA. Furthermore, to facilitate the design and debugging processes, all the building blocks will be compatible with Simulink simulation. Thus, two twin descriptions for each block are provided: one in MATLAB Simulink (M-code along with basic and standard Simulink blocks) and another in VHDL language. This study compares the simulation results achieved by both descriptions for two different PID controllers connected to a synchronous buck converter to assure that the results obtained from the Simulink simulation are similar to those in the FPGA implementation.

Index Terms—FPGA, Digital controller, Power converter, MATLAB Simulink

I. INTRODUCTION

Recently, digital controllers have emerged as a critical component in the control of power converters, offering a range of advantages including cost reduction, programmability, and flexibility. Traditionally, classical digital controllers were implemented using Digital Signal Processors (DSPs) with relatively simple control algorithms [1], [2]. However, the use of DSPs, which are software-based solutions, poses challenges in the certification process. Ensuring the reliability and safety of software-based controllers can be more complex, requiring extensive testing and verification procedures.

Regarding the tendency to simplify the certifying process in digital controllers, especially in space applications, FPGAs become the common choice due to their higher level of reliability compared with software-based solutions [3]. They allow employing high-speed demanding algorithms because of their parallel nature for computing equations. In most FPGA-based applications, hardware description language, such as

VHDL/Verilog, is needed to describe the behavior of the system [4], [5]. In our case, VHDL is chosen because of the previous background of the group. However, it is not a trivial language among digital controller designers that are mainly acquainted with languages such as C or C++ for implementing controller algorithms [6].

Several High-Level Synthesis (HLS) tools are available to make FPGA implementation simpler, some of them are well-known among digital developers. They accept different formats as input and automatically translate them into VHDL/Verilog codes suitable for FPGA programming. For instance, the control algorithm written in C or C++ codes can be translated into VHDL/Verilog codes using *Xilinx Vivado HLS* tool. This eliminates the need for manual HDL coding, reducing design time and effort. However, its synthesis results (time/area) may not be as efficient as the hand-coded VHDL models [7].

Another conventional alternative to acquiring VHDL code without any preknowledge of hardware language is using MATLAB Simulink HDL Coder that converts MATLAB code/Simulink model into portable and synthesizable VHDL code. This approach is well-known among digital controller designers and it achieves better synthesis results compared with Xilinx Vivado HLS tool [7], [8]. However, it is important to note that some modifications may be required in the model in order to successfully convert it into HDL code [9]. Furthermore, the certifying process which is very important in space applications will be more complex for two main reasons. First, with the release of new versions of MATLAB tools, the provided VHDL code will be updated so the final VHDL code must be tested and verified again. Second, a small change in the controller algorithm can result in a totally different VHDL code due to the simplification steps in HDL Coder.

This paper introduces the *ESA Control Toolbox*, a novel MATLAB toolbox that offers a convenient and reliable solution for generating static VHDL code that is significantly needed in sensitive applications. The toolbox addresses the challenges associated with verifying VHDL code performance when an algorithm changes or updates to the MATLAB tool version occur. The controller design using *ESA Control Toolbox* is mainly divided into three parts: connecting several blocks to implement the algorithm in a graphical way, resembling the HDL Coder interface; providing the

TABLE I
COMMON BLOCKS FOR CONTROLLER DESIGN IN THE *ESA Control Toolbox*

Category	Some defined blocks
ADC Controllers	AD7476_Controller, THS1030_Controller
Arithmetic operators	Adder(Un), Multiplier(Un), Subtractor(Un), Minus
Comparators	EqualComparator(1b), LowerThanComparator(Un)
Controllers	Controller, HysteresisController
Logic operators	AndBus(1b), NotBus(1b), OrBus(1b)
Protections	FaultDetectorHyst, Saturator(Un), Protection(1b)
PWM Generators	PWMTrailing, PWMLeading, ComplDeadtimeGen
Registers	Reg(1b,En,Bus,BusEn), SRRegister
Others	Resize, Counter, Timer, Mux, Accumulator(Un), ...

static VHDL code; and programming the FPGA, enabling the implementation of digital control algorithms in hardware.

II. ESA CONTROL TOOLBOX

The *ESA Control Toolbox* is based on the MATLAB Simulink environment. The library consists of several subcategories including some common blocks for digital control applications such as *PWM Generators*, *ADC Controllers*, *Registers*, etc. The available subcategories along with their main blocks are listed in Table I. It is notable that all synchronous blocks have clock and reset inputs and the reset input is active-high in all cases. The ESA toolbox can be implemented and used like any other Simulink library and it is integrated into MATLAB HDL Coder, enabling the user to incorporate HDL coder blocks.

In certain cases, there are blocks available in two versions: *Bus* and *1b*, such as *AndBus* and *And1b*. The *Bus* version uses *std_logic_vector*, which allows for vectors defined as *std_logic_vector(num_bits-1 downto 0)*, where *num_bits* is determined by a generic port. On the other hand, the *1b* version is designed for single-bit ports and uses *std_logic*. Simulink HDL Coder has limitations that prevent the use of vectors defined as (0 downto 0), as it automatically converts boolean/1-bit wires to *std_logic* instead of *std_logic_vector(0 downto 0)*. Therefore, if 1-bit ports are required, the *1b* version of the block should be used. As can be seen in Table I, blocks may have different versions that support different arithmetics, such as signed or unsigned. In such instances, the block without a suffix denotes the signed version, while the suffix "Un" is added to indicate the unsigned version. For example, the blocks *Multiplier* and *MultiplierUn* represent the signed and unsigned versions, respectively.

A. Design translation into VHDL

The *ESA Control Toolbox* consists of a comprehensive library where each block is represented as a Simulink subsystem. These subsystems offer two distinct implementations with identical functionality: a MATLAB Simulink implementation represented in the schematic, and a corresponding text block containing pre-coded VHDL implementation. When the user runs a regular simulation in Simulink, the schematic implementation of the block is simulated. It allows fast simulations as Simulink cannot simulate directly VHDL codes, and using

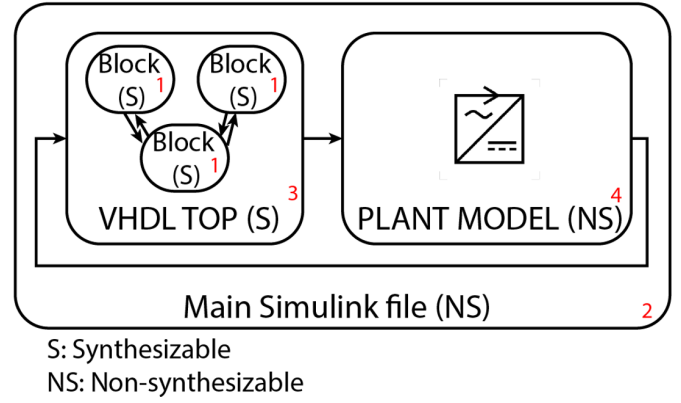


Fig. 1. The architecture of the system using *ESA Control Toolbox*.

an HDL simulator instance for every block in the design is not a scalable procedure. However, when the blocks are integrated for implementation on an FPGA, only the VHDL block is employed. This dual implementation approach offers the best of both worlds. During the design and simulation stages, the Simulink schematic provides a user-friendly and intuitive representation, allowing for rapid prototyping and testing. On the other hand, when it comes to the final FPGA implementation, the VHDL block delivers the necessary hardware-level functionality and performance.

The architecture of the system using the *ESA Control Toolbox* is demonstrated in Fig. 1. As can be seen, before inserting the blocks from the ESA library (1) into a Simulink file (2), a new Subsystem block should be inserted (3), which will be the top module of the future VHDL implementation, i.e. the FPGA block. In a typical control system, the main Simulink file (2) contains other non-synthesizable blocks (4) to model the plant to be controlled. Therefore, block (3) should contain only blocks from ESA library and any other standard blocks presented in the HDL coder library (e.g. constant block or 1-D look-up tables). It is notable that all the blocks inside the HDL coder library can be translated into VHDL, but this translation is automatic and it has not been checked during this study, as MATLAB does not provide their codes/algorithms. As long as the automatic translation can be trustworthy, it can be used. However, blocks (2) and (4) may contain any Simulink block, as they are only used for simulation purposes and will not be implemented inside the FPGA.

Scope blocks can be inserted anywhere, including the VHDL Top subsystem (3), as they will be ignored in the VHDL translation. Finally, after testing the design in the Simulink environment, the system designed in Subsystem block (3) will be translated into VHDL. As the ESA library already contains the translation of all the blocks into VHDL, the HDL Coder toolbox will only create the translation of constant values, and the connection between blocks, using only a small amount of the toolbox's features. The created files in VHDL using ESA library will be independent of the version of MATLAB and they can be imported into any EDA tool, like *Vivado*, *Quartus*, or any other. Naturally, the synthesis

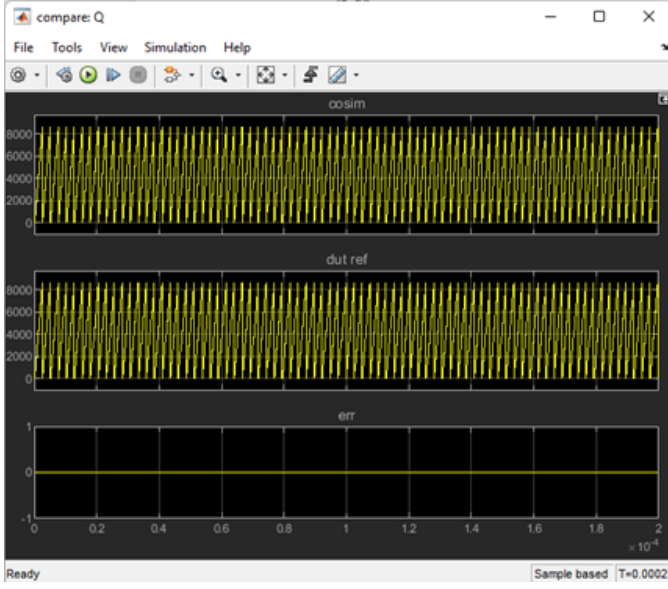


Fig. 2. Results comparison between the *ESA Control Toolbox* and MATLAB Simulink for a multiplier block, showing the perfect match.

outcomes will differ based on the chosen FPGA technology, resulting in variations in the final area utilization (hardware resources) and the achievable minimum clock period, both being specific to the targeted FPGA device. To modify the sample time for all blocks in the design, users can utilize the provided script called *SetSampleTime* which must be executed after adding the library blocks and before any simulation or translation to VHDL.

B. Results verification

A testbench file can be generated automatically by the user including the inputs for the FPGA simulation and expected outputs. All this information is created using the simulation designed in Simulink. Therefore, the testbench generates the same stimuli in the input ports, and will check if the FPGA outputs are the expected ones, comparing them with the Simulink outputs. A simple way of checking the translation of the system to VHDL code is to perform an HDL cosimulation available in MATLAB. MATLAB HDL cosimulation is compatible with Cadence (*Incisive* and *Xcelium*) and *Mentor Graphics* (*ModelSim* and *QuestaSim*) simulators.

The provided toolbox has been designed to get a cycle-by-cycle match between the Simulink and VHDL implementations. The comparison is done by copying automatically the inputs of the Simulink implementation to the inputs of the VHDL one. That assures that the behavior of the MATLAB/Simulink simulation is equivalent to the VHDL simulation. There are as many scopes as outputs the FPGA has. Every scope shows a comparison between the Simulink and VHDL implementations. If there is any mismatch, the scope will show it, as can be seen in Fig. 2 that is obtained from the test of a *Multiplier* block. The first wave is the HDL result, while the second one is the result of the Simulink simulation.

TABLE II
THE SIMULATION PARAMETERS OF THE BUCK CONVERTER

V_{in}	V_{out}	L	C	R_{out}	f_{sw}
28 V	14 V	22 μH	103 μF	10 Ω	100 kHz

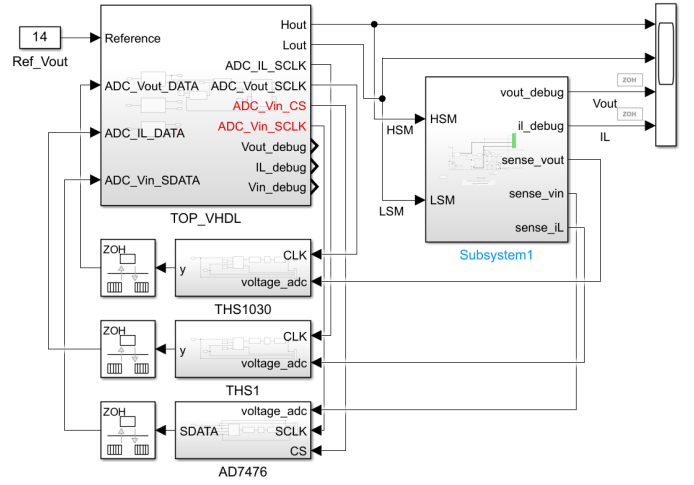


Fig. 3. The schematic of the tested system designed using *ESA* library blocks.

Finally, the third one is the comparison, in which 0 means no mismatch, and 1 means discrepancy. The user is not supposed to check this expected behavior, since the accuracy of both simulations has been previously checked. In fact, all the blocks in the proposed toolbox have passed unitary tests to check that the behavior of the Simulink and VHDL implementations is 100% accurate cycle by cycle and bit by bit.

III. CASE STUDY

In order to evaluate the capabilities of the *ESA Control Toolbox*, a specific plant (synchronous buck converter) is chosen for testing purposes. The converter is characterized by the parameters listed in Table II. The objective of the integration tests is to assess the performance of the proposed toolbox in regulating the output voltage of the converter. In this example, the desired output voltage is set to $v_{out} = 14$ V, and the controller operates at a switching frequency of $f_{sw} = 100$ kHz.

The integration tests are conducted using a synchronous buck converter topology, as depicted in Fig 3. The controller employed in the tests is a simple PID controller implemented within the *TOP_VHDL* block. The output voltage and inductor current are measured using the *THS1030* ADC, while the input voltage is monitored using the *ADC7476* module. The following section presents a comprehensive analysis of the results obtained from these integration tests, providing insights into the performance and effectiveness of the *ESA Control Toolbox* in controlling power converters.

In the first test, the performance of the single-loop voltage controller demonstrated in Fig. 4 (a) is thoroughly evaluated. The main objective of this test is to investigate the transient behavior of the buck converter during the transition from the

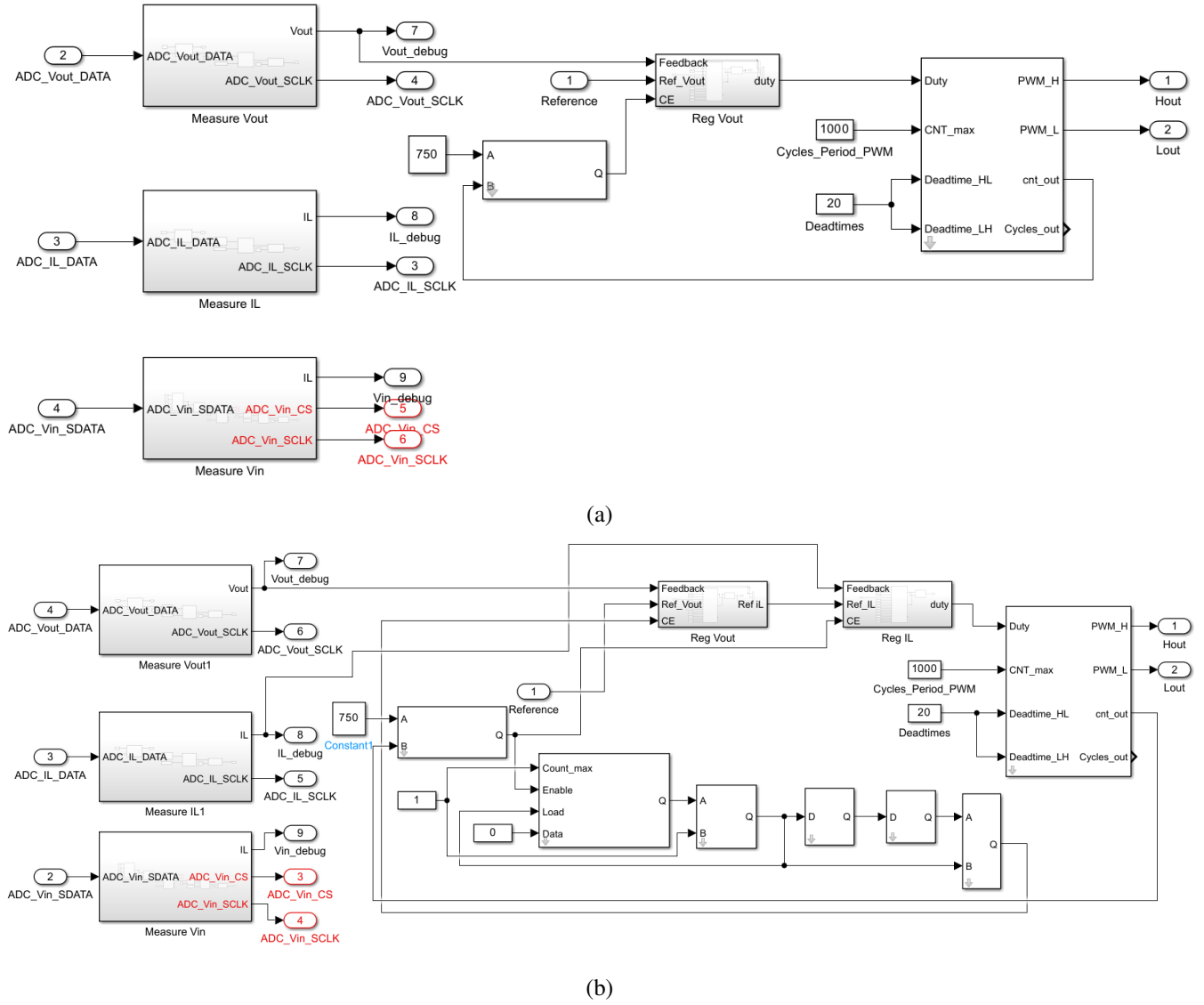
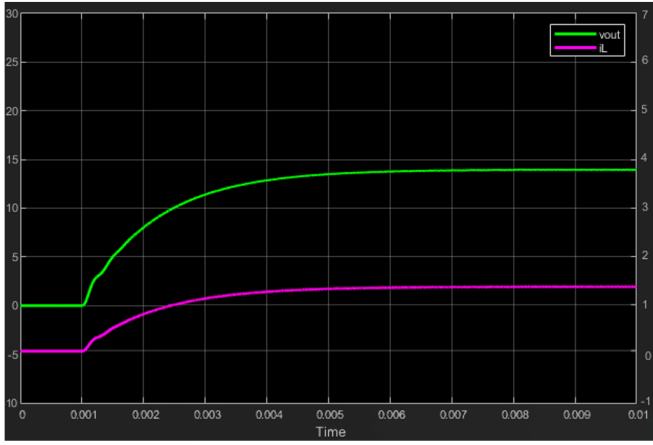


Fig. 4. The schematic of the tested system designed using ESA library blocks; (a) Closed loop voltage control, and (b) Double loop (voltage/current) control.

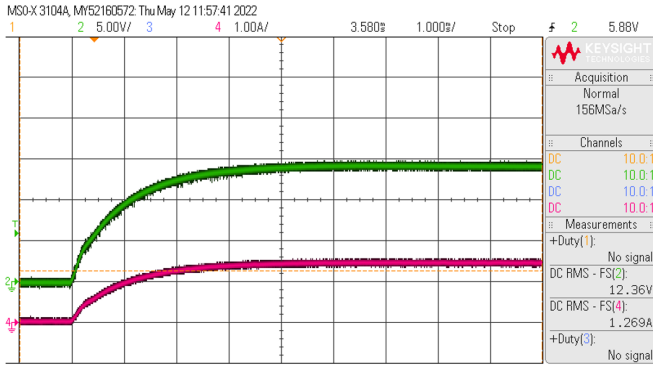
switched-off state to its nominal operating conditions, with an output reference voltage set at 14 V. Fig 5 illustrates the results obtained from the experimental implementation of the controller on an FPGA, and it compares these results with the simulation outcomes obtained using MATLAB Simulink. The figure demonstrates a close correspondence between the experimental results achieved through the *ESA Control Toolbox* and the simulation results from MATLAB Simulink. This alignment between the two sets of results serves as evidence of the effectiveness and accuracy of the proposed toolbox for implementing the digital voltage controller. Furthermore, the steady-state behavior of the controller is examined and presented in Fig 6. This analysis offers insight into the performance of the toolbox in steady-state conditions, highlighting its ability to maintain desired output voltage levels consistently. The observed steady-state behavior further reinforces

the reliability and functionality of the proposed toolbox in real-world applications.

A second test is accomplished in this section to further investigate the performance of the proposed library, involving a more complex controller for the same switched-power converter. This controller consisted of separate current and voltage loops, allowing for more precise control of both the output current and voltage in the buck converter. In this test, the generation of complementary gate firing signals is achieved through the utilization of a *PWMTrailing2b* block. Fig. 7 highlights the effectiveness of *ESA Toolbox* for designing a controller with an even higher level of complexity for regulating both the output current and voltage, specifically in response to a step change in the reference voltage from zero to 14 volts. The results demonstrate the capability of the proposed toolbox to successfully handle such dynamic changes



(a)



(b)

Fig. 5. Output voltage (in green color) and output current (in pink color) during transient with voltage control; (a) MATLAB Simulink and (b) Experimental test using *ESA Control Toolbox*.

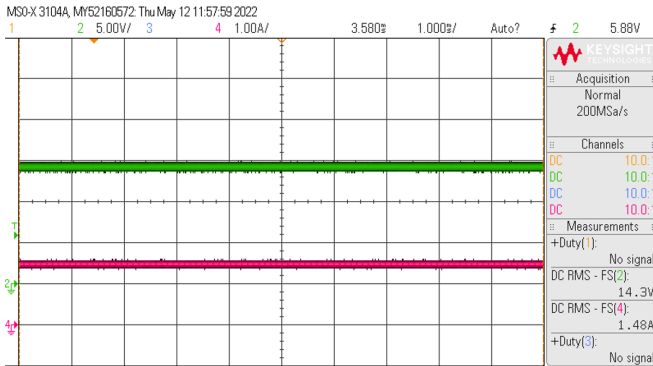
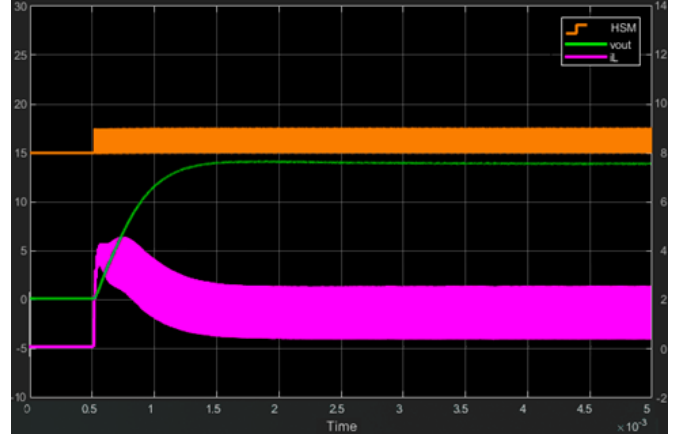


Fig. 6. Steady-state output voltage (in green color) and output current (in pink color) obtained by experimental test using *ESA Control Toolbox*.

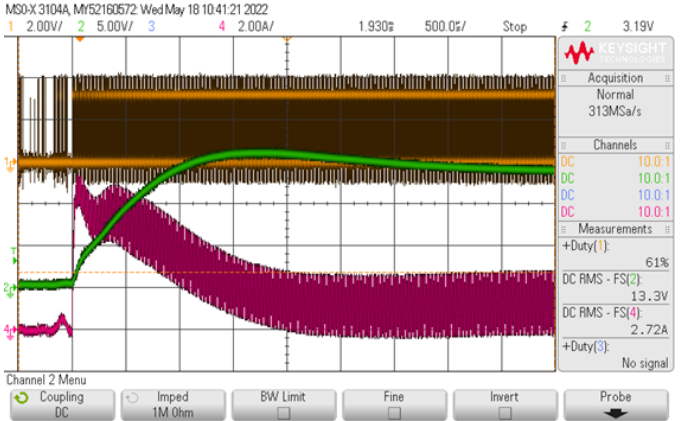
and maintain accurate control over the converter's output.

IV. CONCLUSIONS

In this paper, a new MATLAB Simulink toolbox for implementing digital controllers in FPGAs was presented. The toolbox offers a comprehensive set of blocks that integrates both MATLAB Simulink and VHDL language descriptions,



(a)



(b)

Fig. 7. Output voltage (shown in green) and output current (shown in pink) during transient with voltage/current control; (a) Simulation results in MATLAB Simulink and (b) Experimental test using *ESA Control Toolbox*.

providing a flexible and efficient platform for controller design. The experimental tests focused on implementing two different controllers, namely a single voltage loop controller and a double voltage/current controller, using the ESA library on an FPGA. The performance of these controllers was analyzed and compared with MATLAB Simulink simulations. Both steady-state and transient-state analyses were conducted, providing a comprehensive assessment of the controllers' functionality under various operating conditions. The FPGA implementation demonstrated a remarkable agreement with the MATLAB Simulink simulations, validating the successful implementation and functionality of the controllers facilitated by the proposed library. These findings highlight the viability and accuracy of utilizing the ESA Toolbox for designing controllers in real-time applications. The results provide valuable insights for advancing the field of design reliability in space applications and serve as a foundation for further research and development in FPGA-based controller design.

REFERENCES

- [1] A. R. Oliva, S. S. Ang and G. E. Bortolotto, "Digital control of a voltage-mode synchronous buck converter," in *IEEE Transactions on Power Electronics*, vol. 21, no. 1, pp. 157-163, Jan. 2006, doi: 10.1109/TPEL.2005.861193.
- [2] Bosheng Sun and Zhiqiang Gao, "A DSP-based active disturbance rejection control design for a 1-kW H-bridge DC-DC power converter," in *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1271-1277, Oct. 2005, doi: 10.1109/TIE.2005.855679.
- [3] P. K. Joseph, Bijeev N V, N. B. Sharma and V. H. Jani, "Analysis of digital PWM design for Zero Voltage Transition Phase Shifted Full bridge converters for space applications," 2016 International Conference on Emerging Trends in Communication Technologies (ETCT), Dehradun, India, 2016, pp. 1-5, doi: 10.1109/ETCT.2016.7882923.
- [4] Zamiri, E.; Sanchez, A.; de Castro, A.; Martínez-García, M.S., "Comparison of Power Converter Models with Losses for Hardware-in-the-Loop Using Different Numerical Formats," *Electronics* 2019, 8, 1255. <https://doi.org/10.3390/electronics8111255>.
- [5] R. Wiśniewski, G. Bazydło and P. Szcześniak, "Low-Cost FPGA Hardware Implementation of Matrix Converter Switch Control," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1177-1181, July 2019, doi: 10.1109/TCSII.2018.2875589.
- [6] P. K. Joseph, Bijeev N V, N. B. Sharma and V. H. Jani, "Analysis of digital PWM design for Zero Voltage Transition Phase Shifted Full bridge converters for space applications," 2016 International Conference on Emerging Trends in Communication Technologies (ETCT), Dehradun, India, 2016, pp. 1-5, doi: 10.1109/ETCT.2016.7882923.
- [7] Zamiri, E.; Sanchez, A.; Yushkova, M.; Martínez-García, M.S.; de Castro, A. Comparison of Different Design Alternatives for Hardware-in-the-Loop of Power Converters. *Electronics* 2021, 10, 926. <https://doi.org/10.3390/electronics10080926>.
- [8] F. M. Sánchez, R. Mateos, E. J. Bueno, J. Mingo and I. Sanz, "Comparative of HLS and HDL implementations of a grid synchronization algorithm," *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Vienna, Austria, 2013, pp. 2232-2237, doi: 10.1109/IECON.2013.6699478.
- [9] M. M. Ashtaputre, M. S. Sutaone and M. Rajne, "A Low Cost Solution for Hardware-in-Loop Software Testing," 2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA), Goa, India, 2021, pp. 449-452, doi: 10.1109/IRIA53009.2021.9588753.