Laboratory Procedure for Real-Time Simulation Experiment of Renewable Energy Systems on OPAL-RT Digital Simulator

Elutunji Buraimoh
Smart Grids Research/Department of Electrical Power Engineering
Durban University of Technology
Durban, South Africa
https://orcid.org/0000-0002-0054-5778

Innocent Ewean Davidson
Smart Grids Research/Department of Electrical Power Engineering
Durban University of Technology
Durban, South Africa
https://orcid.org/0000-0002-2336-4136

***Abstract*— Simulation technologies have been extensively used to design and develop electrical systems since the mid-twentieth century. Simulation tools have grown in lockstep with the advancement of computational capabilities. Computing technology has greatly increased performance and has become widely available at a constantly falling cost in recent years. As a result, simulation tools have experienced tremendous performance advances and consistent cost reductions. Engineers and researchers can now access inexpensive, high-speed simulators hitherto too expensive, except for the major industries and utilities. This work utilized the RT-LAB tool of OPAL-RT to develop the 250-kW renewable energy system (single-stage grid-connected photovoltaic). Thus, the system design, analysis, and testing are done with an RT-LAB environment using the OPALRT-OP4510 Digital Simulator. The test results are obtained using UTD2025CL Digital Storage Oscilloscope. This work also presents the application RT-LAB software, SIMULINK-based model adjustment, and I/O system and therefore presented prospects of OPAL-RT system in Rapid Control Prototyping (RCP) and Hardware-in-the-Loop (HIL) testings.**

Keywords—modeling, simulation, real-time simulation, photovoltaic system, RT-LAB

I. INTRODUCTION

Using another process, a simulation represents a system's functioning or feature [1]. Each variable or system state is solved sequentially as a function of variables and states at the end of the previous time step to solve mathematical functions and equations at a specific time step. The actual time necessary to calculate all system functions during a discrete-time simulation time-step is quicker or greater than the simulation time-step duration. In contrast, the accuracy of calculations in real-time simulation is determined by an accurate dynamic representation of the system and the time it takes to produce results. The proper time-step duration must be selected to correctly describe system frequency response up to the quickest transient of interest. The simulation results may be validated when the simulator achieves real-time without overruns.

Figure 1 depicts a typical time step and computes power needs for various applications. The mechanical systems with weak dynamics on the left side of the chart need a fixed sampling between 1 and 10 milliseconds; the computational step will be less than 5% to 10% of the system's minimal time constant, the rule of thumb. A lower time step may be necessary to preserve numerical stability in stiff systems. Simulation time steps as little as 100 microseconds to 500 microseconds may be necessary with the presence of a friction process.

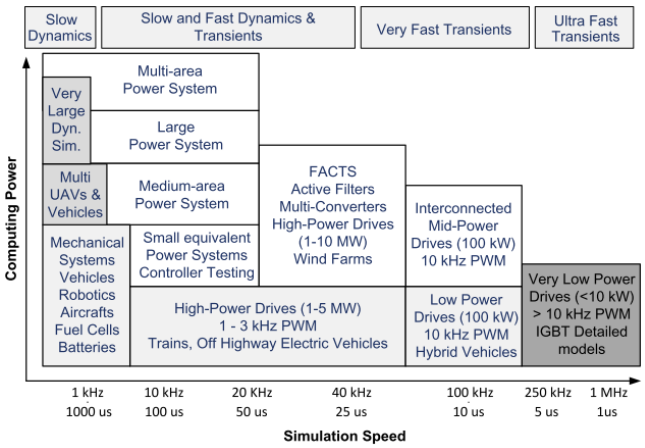

Fig. 1. Time-steps of various engineering simulations [1]

EMT simulators commonly utilize a simulation time step of 30 to 50 microseconds to get an accurate result for transients up to 2 kHz. EMT phenomena simulation with frequency content up to 10 kHz generally needs a simulation time step of around 10 microseconds since better precision can be attained with smaller time steps. System equations must be solved with very tiny time increments [citation]. Offline simulation is popular, but it takes a long time if no model precision is ignored (i.e., the use of average models). Low-power converters, for example, require time steps of less than 250 nanoseconds without interpolation or 10 microseconds with interpolation in power electronic converters with a higher PWM carrier frequency (10 kHz).

AC circuits having a higher frequency response and shorter lines, such as those used in low-voltage networks and electric rail power feeding systems, may require time steps of less than 20 microseconds. To identify the least time-step size and processing power required to accomplish the specified time-step, tests using real-world system set-ups and characteristics are required. Modern digital real-time simulators can have jitter and overhead of less than 1 microsecond, allowing time-step values as low as 10 microseconds while still providing enough computing resources for model calculation. If necessary, simulation time steps can be lowered to a minimal amount to enhance precision or avoid numerical instability.

II. SYSTEM DESCRIPTION

A. Simulink Model of 250 kW PV System

The sun's energy is captured in the form of DC by a photovoltaic (PV) array. As needed, the obtainable DC

voltage is transformed to AC for home or industrial usage. The obtained DC voltage is ramped up to a significant level of DC using a boost or buck-boost converter and then converted to AC using an inverter in some topologies [2]. However, due to the vast number of components used, this procedure is rather expensive. The Single-Stage Inverter is a cost-effective substitute for the two-stage technique (SSI). The name Single-Stage Boost Inverter is because it boosts DC and inverts it to AC using only one circuit. The advantages of SSBI include simplified and robust circuitry, dependability, and efficiency. This study examines the many (but not all) SSI topologies used in PV systems.

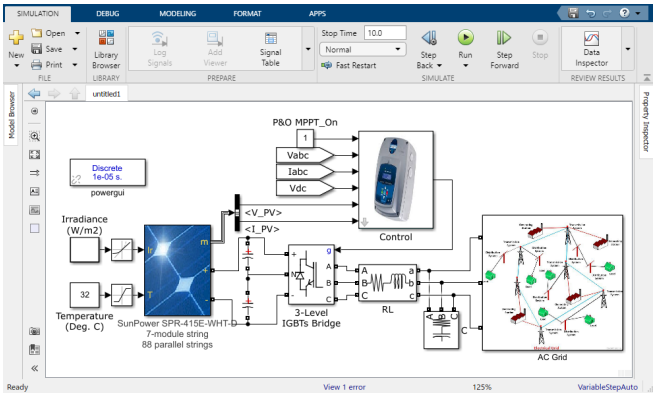

Fig. 2. Single Stage Grid-Connected PV System in MATLAB SIMULINK

There are 86 parallel strings in the PV array. Seven SunPower SPR-415E modules are linked in series in each string. Figure 3 shows the I-V and P-V descriptions of the chosen module or the entire array.

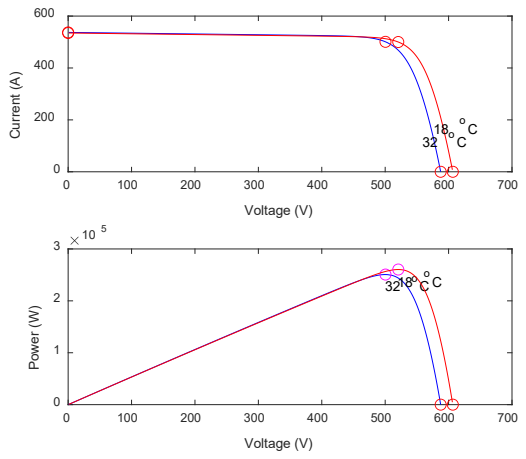

Fig. 3. I-V and P-V characteristics of the selected SunPower SPR-415E modules

A 3-level IGBT bridge with PWM control is used to represent the converter. The inverter choke RL and a tiny harmonics filter C filter the harmonics generated by the IGBT bridge. To link the inverter to the utility distribution system, a 250-kVA 250V/25kV three-phase transformer is utilized.

The descriptions of the five primary Simulink®-based subsystems that make up the control system are given in Table 1:

TABLE I. CONTROLLER'S DESCRIPTION

Subsystem	Description
Maximum Power Point Tracking (MPPT) Controller	In the 'Perturb and Observe,' the MPPT controller uses strategy. This MPPT system automatically adjusts the inverter VDC regulator's VDC reference signal to get a DC voltage that extracts the most power from the PV array.
DC link Voltage Regulator	Regulate the present regulator's needed Id (active current) reference
Current Regulator	It determines the appropriate reference voltages for the inverter based on the active and reactive current references. The Iq reference is set to zero in our case [3]
Pulse Width Modulation (PWM) Generator	Using the needed reference voltages, generate firing signals for the IGBTs. The carrier frequency, in this case, is 1980 Hz (33*60)
Phase-Locked Loop (PLL) & Measurements	For synchronization and voltage/current measurements, a PLL is required

These complete model parameters are given in Table 2.

TABLE II. OFFLINE MODEL PARAMETERS

Parameters	Values
3-phase power (VA)	250e3 VA
Primary line-to-line voltage	25e3 V
DC link voltage	480 V
Proportional gain of DC voltage regulator	4
Integral gain of DC voltage regulator	500
Proportional gain of Current regulator	0.4
Integral gain of Current regulator	50

B. OP4510 RT-LAB-RCP/HIL Systems

The OP4510 is a small device with 128 fast I/O channels, signal conditioning, additional RS422 streams (extra low-speed fibre-optic channels), high-speed communication ports (SFPs), and Simulink and Simscape integration [4]. The combination of cutting-edge INTEL multi-core processors with the strong Kintex 7 Field Programmable Gate Arrays (FPGA) delivers increased simulation time steps (microsecond) and power, allowing rapid power electronic systems to be more accurate. The OP4510 simulates up to 200 nodes in a power grid [2]. Figure 4 shows the OP4510 system architecture (standard configuration. This real-time power grid simulator offers HIL and RCP simulation with the significantly best performance and at a relatively affordable price.

OPAL-RT technology was used to create the simulator, which includes real-time and distributed simulation applications. It can work in real-time on a PC cluster with fixed-step solvers in the Simulink SimPower Systems platform [3]. RT-LAB is made up of non-proprietary commercial off-the-shelf (COTS) PC modules. It's a real-time simulation platform that lets you create and implement automatic system-level and block diagram models on regular PCs. It also comes with the MATLAB/Simulink/Simscape software as a front-end platform for modifying and displaying block diagram visual models.

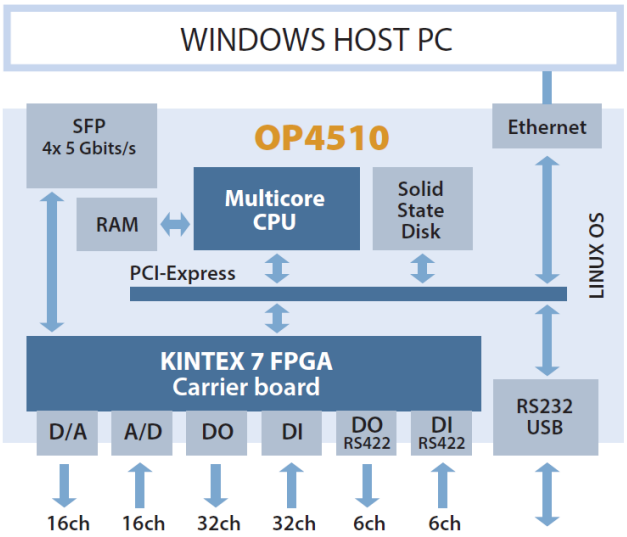

Fig. 4. Configuration and system architecture of OP4510 [2]

Figure 5 shows the OP4510 set-up at the Smart Grids Laboratory, Durban University of Technology. The set-up consists of the Host PC (monitor for graphical interface and view the power system/algorithms/control/physical models, CPU), Real-Time Target PC, and UTD2025CL Digital Storage Oscilloscope for observation. Between the Host and Target is the TCP/IP

Fig. 5. OP4510 set-up at the Smart Grids Laboratory, Durban University of Technology

The following describes how the RT-LAB simulator is set up in DUT. The OP4510 is a full simulation system operating with a Kintex 7 FPGA. It's intended for use as a workstation (or desktop) simulator with Intel Xeon 3.5 GHz central processing unit. A sophisticated Target Computer, a flexible, high-speed Front End Processor, and a signal conditioning stage are all part of it. Its design makes it simple to utilize with common connectors (DB37, DB9). In the entire system, there are four targets (total core number); the master target handles all communications with both the target and the host PC and any other targets. However, only one target core is activated out of the entire four used for the real-time experiment conducted. The target is running the Redhat v2.6.29.6-opalrt-6.2.1 operating system, which the target uses to function in real-time. There is only one host PC used in the experiment. However, there may be several host PCs, just like there could be multiple targets. Each host allows several end-users to ping the targets, with the master host having total control over the simulator. On the other hand, other hosts are maintained in read-only mode, allowing them to only receive and display signals from the simulator. Fixed processors deployed across several nodes can control I/Os.

The target computer can work as a standalone computer in its standard configuration. The target computer in chassis in the Smart Grids Lab at DUT includes the features listed in Table 2. The RT-LAB digital simulator supports C37.118 IEEE Standards and IEC 61850 standards to communicate between hosts and targets [5]. It also works with SPECTRACOM standards. Ethernet connection is constructed between the target and the hosts' PCs to achieve quick transmission and reception. It also includes a unique capability of simultaneous calculation of system models, which may reduce the step size from milliseconds to microseconds.

TABLE III. CONTROLLER'S DESCRIPTION

Items	Quantity	Description
Operating System	1	Redhat v2.6.29.6-opalrt-6.2.1
Chassis Type	1	OP4510-1, V2
CPU (Central Processing Unit)	1	Intel Xeon 3.5 GHz
Total Core #	4	8 GB
Memory Motherboard	1	X11SSM-F-O Supermicro Server Motherboard,
AC Input		100-240V, 60-50Hz
FPGA Board Index		00

III. RT-LAB SIMULATION PROCEDURE

A. Step 1; Model Preparation

A new project is created on the RT-LAB window interface of the host PC screen. A new model is also created under the model section of the newly created project. This new model is opened as an empty MATLAB Simulink file (.slx). The Simulink model blocks shown in Figure 2 can now be copied with all its parameters in Table 2 into the new model created on the RT-LAB platform. The new model can now be compiled and run offline. The configuration parameters must be changed under the model set. Thus, fixed time steps under the solver details must be reduced. The periodic sample time option must be unconstrained, and none of the option boxes must be ticked. It must be noted that the sample time specified in the configuration parameters must be the sample with the one specified in the powergui.

Similarly, all the control blocks' sample times must be multiple of the sample time specified in the powergui. Run the model and look at the results on the different scopes. The PV voltage is 481 V, and the power extracted from the array is 236 kW when the steady-state is reached (about t=0.15 sec).

B. Step 2; Block Grouping

Now that the model is running properly offline, remove all the scopes and replace them with a single scope that can accommodate a maximum of eight signals. It must be noted that the experiment can be conducted several times to observe other system dynamic parameters in batches. The model block is now grouped into three. The first is the group of all the power, control, mathematical, and signal blocked, and this group is turned into a subsystem named "SM_name." The second is a subsystem of only the single scope with a

subsystem name "SC_name". The last group is the powergui which is left alone without being enclosed into a subsystem. In RT-LAB platforms, the subsystems objective is to distinguish computation subsystems and GUI subsystems.

The "SM_name" is the computation subsystem, while the "SC_name" is the GUI subsystem. If the four cores for the OP4510 are activated, creating subsystems ensures the assignment of computation subsystems to different CPU cores. The GUI subsystem can also contain displays, scopes, manual switches, constants and other user interface blocks. Thus, "SC_name" runs on the host PC asynchronously from the "SM_name". "SC_name" is not linked to a target CPU core. Therefore no mathematical operations, signal generation, and physical model must be found in "SC_name". On the other hand, "SM_name" must contain all the computational elements of the model, signal generators, I/O blocks, mathematical operations, physical models, etc.

The computation subsystem is implemented in real-time (or enhanced simulation form) on one CPU core of the target. The GUI subsystem is displayed on the Host PC. The data between the computation subsystem and the GUI is traded asynchronously through the TCP/IP link [6]. The computation blocks can be split into different computation subsystems. Each computation subsystem is executed on a CPU core of the real-time target. Data is exchanged simultaneously between two computing subsystems via a common system memory. Each additional computation subsystem is named "SS_name". Thus, each additional SS subsystem uses an additional CPU core.

C. Step 3; Adding Communication Block

Communication between computation subsystems is synchronous. However, communication between computation subsystems and the GUI subsystem is asynchronous. The block responsible for the communication in the RT-LAB is the OpComm block. OpComm ensures communication between computation subsystems and GUI subsystems. In case there is more than one computation subsystem, OpComm also ensures communication between 2 computation subsystems. OpComm block can be found in the RT-LAB library in the Simulink® library browser once RT-LAB has been installed.

Before any actions on the related signals, all subsystem inputs (SM, SS, SC) must pass through an OpComm block. After the subsystems have been created and renamed (SM name, SS name, SC name), the OpComm block must be included. One OpComm block can accept multiple inputs in one subsystem—open the block to choose the needed inputs number. Thus, the OpComm is connected between the input points of the SC_name subsystem and the single scope used. Each input signal can be a scalar or a vector. In the console or GUI subsystem, One OpComm is enough in most cases, and one is used in the 250 kW PV system model case study. However, more OpComm blocks (up to 25) may be inserted to receive signals from the real-time subsystems. The model must be run offline and subsequently saved to ensure it remains intact.

D. Step 4; Building

The system is set to confirm that the target platform is still "OPAL-RT Linux (x36-based)" on the RT-LAB platform. After this confirmation, the model is built. The building is done in six steps by the simulator in the following order; Select development platform, Model Separation, Generating C code. The generated C code is transferred, the generated C code is compiled, and the created model is transferred. The results of the building process can be checked under the compilation view. C code is generated by calling a code generator from MATLAB and applying it to each model according to each platform's specific template. Transferring the generated C code is done through an internal RT-LAB process. Next, the target compiler builds and links the files to generate a real-time executable to build the generated C code. Lastly, Executables are transferred back to the host computer.

E. Step 5; Assigning subsystems

After successfully building the system, the next step is to assign the subsystems in case there are two computational subsystems. Subsystems can be run on the same target or on different ones, limiting the number of cores. In the case of the 250 PV system simulated, the "SM_name" was run on one target activated by OPAL-RT. Most importantly, the OFF option under the XHP is checked and changed to ON.

F. Step 6; Loading and Executing

The real-time simulation mode is changed to software synchronized mode under the execution properties. Other options available are "Simulation" and "Hardware synchronized". These will be explained later. Thus, the model is loaded and subsequently run or executed. During this real-time simulation, a Simulink model is generated with the inscription "automatically generated by RT-LAB during compilation" is opened automatically by the RT-LAB. Double click on the scope to see the simulation results.

The model can be reset to stop the simulation. This would allow editing the model and incorporating blocks to ensure external data acquisition from the experiment.

G. Step 7; Adding Controller Block

The OpCtrl block facilitates the programming of one OPAL-RT card and its initialization and hardware synchronization mode selection. It also allows Send/Recv and I/O blocks to be bound to that specific card [7]. Only one OpCtrl block must be found for each card used in the model. OpCtrl block can be found in the RT-LAB I/O library (COMMON>CONTROLLER BLOCK) in the Simulink® library browser once RT-LAB has been installed. This block is added to the "SM_name" subsystem. The board type must be changed from "VC707" to "TE0741". This can make the board switching process easier for the user.

H. Step 8; Adding AnalogOut Block

This block is used to transmit to a physical I/O card the voltage to be supplied to Analog Output channels. In the RT-LAB I/O library, locate AnalogOut Block under the library COMMON>SUPPORTED FUNCTIONALITIES. The block is to be posted within the "SM_name" subsystem.

The Analog Out block is used to provide analogue output voltage values to the analogue output channels of an OP5330 module equipped with various types of carriers that are compatible with the FPGA board. OP5330 is the Digital to Analog Converter Module of the target PC. The data values are transmitted to the FPGA from the RT-LAB model via the PCIe bus of the target computer via one DataIn port of the bitstream. Each bitstream with Analog Output channels includes a configuration file that lists the Data port number as well as the position of the OP5330 modules in the system. The configuration file has the same name as the bitstream file name supplied in the OpCtrl block, but instead of ".bin," it has

the extension ".conf.". These two files are located on the computer via Local Disk (C:)>OPAL-RT>eFPGASIM>v2.3.3.56>Common>fpgalib>firmwares. They are both copied and pasted on the current folder path in the main MATLAB command window. Therefore, the Primary Bitstream FileName parameter in the OpCtrl block is changed to copied ".bin" file. Depending on the FPGA related to the controller they are connected to, this configuration file utilizes various parameters to specify the position of the OP5330 channels.

Lastly, the Simulink "Mux" block, which combines its different input signals into a single vector output, is used to connect the output points of the "SM_name" subsystem as its own input. Thus, the output of the Mux is connected to the input of the AnalogOut Block.

İ. Step 9; Datalogging

The block responsible for the data logging in the RT-LAB is OpWriteFile blocks. This block saves the input signal(s) to a file. It works like the Simulink ToFile icon, but if Simulation Mode is not checked, It records all signals using the RT-LAB data collecting architecture without interrupting the real-time simulation.

OpWriteFile block can be found in the RT-LAB library (data logging) in the Simulink® library browser; once RT-LAB has been installed, Data is stored in Matlab file format (.mat). Thus, the "variable name" parameter of the OpWriteFile is changed from the default "opvar" to "usernamespecified" and the file size limit (in Bytes) is adjusted to "[10000000]".

IV. Data Acquisition and Results

Now, the model is saved on the Simulink file. On the RT-LAB platform, the system is now built and assigned. Most importantly, the real-time simulation model is now changed from "Software synchronized" to "Hardware synchronized". The system is now executed. The plotting of various graphs can now be done from the data logged. The peak values of the measured parameters are given in Table IV. The simulation can be stopped by resetting the system.

TABLE IV. Peak Values Obtained from Datalogging

Parameters	Values
inverter primary line-to-line voltage	2.030e4 V
inverter primary line-to-line current	1.028e1 A
AC Power	2.636e2 W
DC Power	2.415e2 W
DC voltage reference	4.829e2 V
DC voltage measured	5.036e2 V
direct-axis current	1.274e0
direct-axis current reference	1.192e0
quadrature-axis current	2.363e-1
PV voltage	5.036e2
PV current	5.391e2
PV diode current	6.040e1

The captured peak values are used as inverse gains for their respective parameter signal line. This ensures they are input in a format readable by the UTD2025CL Digital Storage Oscilloscope. Lastly, the system is built, assigned, loaded, and executed after saving the model properly.

The results observed from the digital storage oscilloscope are shown below. The inverter primary line-to-line voltage (Channel 1) and inverter primary line-to-line current (Channel 2) are in Figure 6. The AC Power (Channel 1) and DC Power (Channel 2) are shown in Figure 7. Figure 8 inverter direct-axis current (Channel 1) and direct-axis current reference (Channel 2). Figure 9 shows the inverter quadrature-axis current.

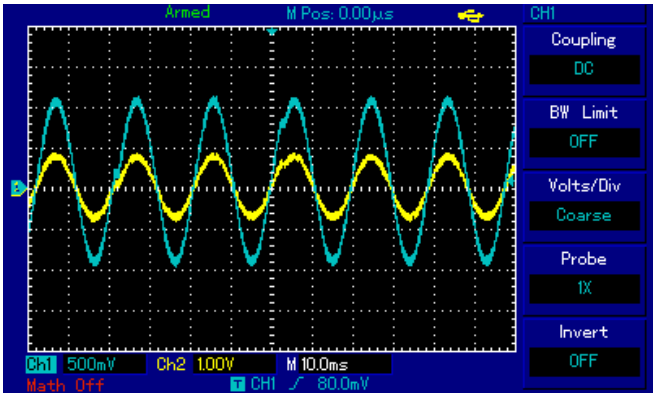

Fig. 6. inverter primary line-to-line voltage (Channel 1) and inverter primary line-to-line current (Channel 2).

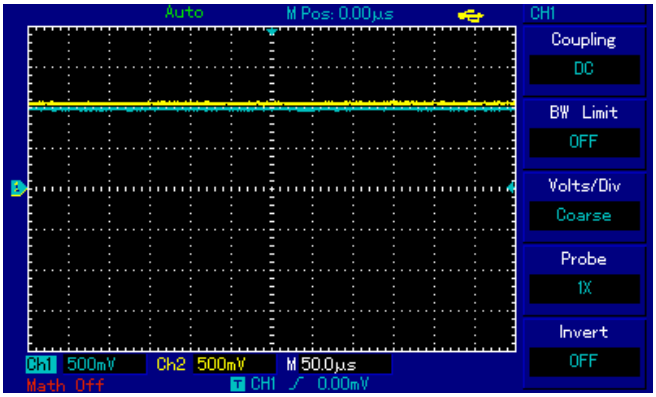

Fig. 7. The AC Power (Channel 1) and DC Power (Channel 2).

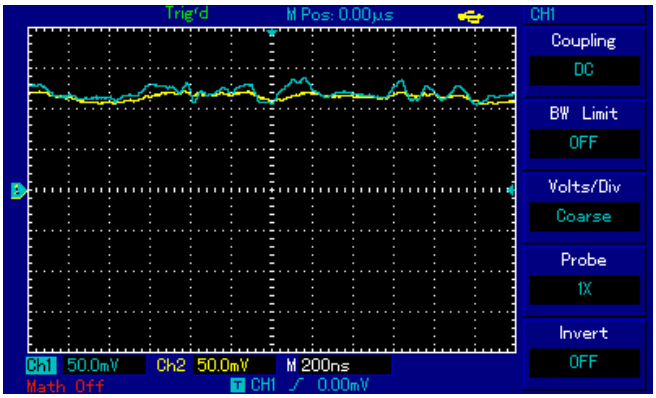

Fig. 8. inverter direct-axis current (Channel 1) and direct-axis current reference (Channel 2) .

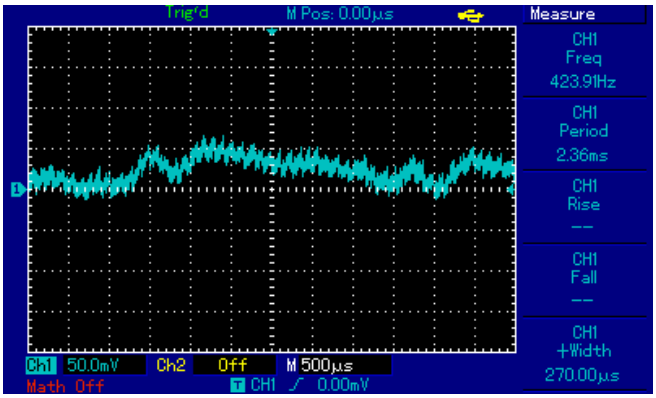

Fig. 9. Inverter quadrature-axis current.

V. Conclusion

Simulink and Simscape models may interact with the actual world in real-time using RT-LAB, which is fully integrated with MATLAB/Simulink®. The OP4510 system architecture has been shown in this work, and the laboratory set-up physical structure in Smart Grids Laboratory at DUT is shown. Engineers may use RT-LAB to swiftly design and verify their applications in real-time, regardless of their complexity. The set-up consists of the Host PC (monitor for graphical interface and view the power system/algorithms/control/physical models, CPU), Real-Time Target PC, and UTD2025CL Digital Storage Oscilloscope for observation. Between the Host and Target is the TCP/IP. The power network and control system of a 250 kW single-stage grid-connected photovoltaic (PV) system is simulated in real-time using OP4510. The waveforms of inverter primary line-to-line voltage, inverter primary line-to-line current, AC Power, DC Power, DC voltage reference, DC voltage measured, direct-axis current, direct-axis current reference, quadrature-axis current, PV voltage, PV current, and PV diode current were observed from both the oscilloscope and console computer connected with the OP4510. The real-time simulation data is also stored in .mat files from the console for further analysis. The next step is to implement the inverter control on Texas Instruments Digital Signal Controllers and conduct a further experiment using the OP4510 RT- LAB-RCP/HIL for both HIL and RCP testings. RT-LAB delivers flexible and scalable solutions for the power systems, power electronics, aerospace, and automotive sectors as a multi-domain platform.

References

[1] J. Bélanger, P. Venne, and J. N. Paquin, “The What, Where and Why of Real-Time Simulation,” *Planet RT*, vol. 1, no. 1, pp. 25–29, 2010, doi: 10.4028/www.scientific.net/MSF.800-801.489.

[2] E. Buraimoh and I. E. Davidson, “Overview of Fault Ride-Through Requirements for Photovoltaic Grid Integration, Design and Grid Code Compliance,” *9th Int. Conf. Renew. Energy Res. Appl. ICRERA 2020*, pp. 332–336, Sep. 2020, doi: 10.1109/ICRERA49962.2020.9242914.

[3] A. Delavari, P. Brunelle, and I. Kamwa, “Real-time Closed-loop PQ Control of NPC Multi-level Converter Using OPAL-RT and Speedgoat Simulators,” *2018 IEEE Electr. Power Energy Conf. EPEC 2018*, Dec. 2018, doi: 10.1109/EPEC.2018.8598323.

[4] “OP4510 RT-LAB-RCP / HIL SYSTEMS User Guide,” Quebec, Canada, 2016.

[5] S. S. Noureen, V. Roy, and S. B. Bayne, “An overall study of a real-time simulator and application of RT-LAB using MATLAB simpowersystems,” *2017 IEEE Green Energy Smart Syst. Conf. IGESSC 2017*, vol. 2017-Novem, pp. 1–5, 2018, doi: 10.1109/IGESC.2017.8283453.

[6] E. Naderi and A. Asrari, “Hardware-in-the-loop experimental validation for a lab-scale microgrid targeted by cyberattacks,” *9th Int. Conf. Smart Grid, icSmartGrid 2021*, pp. 57–62, Jun. 2021, doi: 10.1109/ICSMARTGRID52357.2021.9551023.

[7] F. Baccino, A. Brissette, D. Ishchenko, A. Kondabathini, and P. Serra, “Real-time hardware-in-the-loop modeling for microgrid applications,” *2017 6th Int. Conf. Clean Electr. Power Renew. Energy Resour. Impact, ICCEP 2017*, pp. 152–157, Aug. 2017, doi: 10.1109/ICCEP.2017.8004807.