TIME SYNCHRONOUS CONTROL OF GRID- AND PV-EMULATORS FOR LABORATORY TESTING WITHIN A CO-SIMULATION ENVIRONMENT

Christian SEITL
AIT Austrian Institute of Technology – Austria
christian.seitl@ait.ac.at

Thomas STRASSER
AIT Austrian Institute of Technology – Austria
thomas.strasser@ait.ac.at

Marios MANIATOPOULOS
National Technical University of Athens – Greece
mmaniato@mail.ntua.gr

Panos KOTSAMPOPOULOS
National Technical University of Athens – Greece
kotsa@power.ece.ntua.gr

ABSTRACT

The increasing share of renewables and electric vehicles leads to new challenges and new control strategies. They result in an additional need of complex simulations and testing. Investigating novel controllers often requires to simultaneously simulating multiple domains and controlling power electronics. The AIT Lablink enables coupling of different simulation tools and laboratory hardware. This work discusses the development of Lablink drivers for two grid and two PV emulators. Beside control possibilities also the latencies of the implementations are examined.

INTRODUCTION

The shift from a centralized grid with conventional bulk generation towards a decentralized and sustainable one with a significant share of renewable energies leads to new challenges like fluctuation in renewable energy production as well as to new control approaches. To address them and to be capable of operating a possible future power grid the EU funded ELECTRA IRP project has developed the Web-of-Cells (WoC) concept [1]. In the context of this concept, cells can be seen as embedded microgrid where each cell contributes to voltage and frequency control based on its own flexibility of reactive and active power.

New functionalities and control concepts result in an additional need of testing. Therefore, laboratory tests are essential as some scenarios cannot be tested in the real world. Though, a lab with emulators for grid, photovoltaic (PV), batteries, etc. can test extreme conditions like over-voltage situations, voltage dips, under-frequency, fast change of PV-power, and many more.

Testing control algorithms of a cell within the WoC-concept as well as controllers for isolated and embedded microgrids increases the complexity of laboratory tests significantly [2]. On the one hand, there is a demand for a time synchronous control of various emulators, for example when testing ride-through capabilities of a PV inverter during a simultaneous dip of PV power. On the other hand, often a co-simulation of different domains like the grid, vehicle movements, prosumer households, etc., is necessary, sometimes with a real home storage system whose behaviour depends on all of the above mentioned domains. To tackle these issues AIT developed the Lablink, a middleware which enables coupling of different software simulation tools as well as laboratory hardware.

This work discusses the external control of two grid and two PV emulators using self-developed Lablink drivers, with the focus of using the emulators within a co-simulation environment.

IMPLEMENTATION

AIT Lablink

The Lablink allows to interface arbitrary software and hardware systems by providing a data agnostic communication middleware [3, 4]. It is based on an MQTT broker (Message Queuing Telemetry Transport protocol) with overlaid communication schemes (cf. Figure 1).

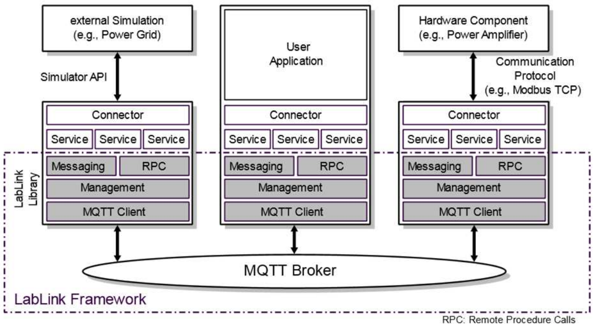

Figure 1: AIT Lablink System Architecture [3]

Lablink supports among others publish/subscribe messaging with complex data types, remote procedure calls and data points. By writing Lablink connectors for software simulation tools, power sources, measurement devices, etc. they can be easily coupled. A sync service facilitates synchronization and central control of simulators. A Python interface extends the setup by scripting possibilities. More information about Lablink can be found in [4].

Control of Lab Hardware

The manufacturers of laboratory hardware usually provide an interface to externally control their devices. Its type and the control possibilities vary from manufacturer to manufacturer and sometime among different devices of the same vendor.

Using those interfaces Lablink connectors for two grid emulators

- Spitzenberger & Spies PAS
- Regatron TopCon TC.ACS

and for two PV emulators

- AIT Photovoltaic Array Simulator
- Regatron TopCon Quadro TC.P

were developed. Both Regatron devices are switched mode devices whereas the two others are linear amplifiers. Subsequent, the interface types and implementations of those four drivers are discussed.

Spitzenberger & Spies PAS

Spitzenberger & Spies (SPS) uses SCPI commands ("Standard Commands for Programmable Instruments") via an IEEE 488.2 (also known as GPIB, "General Purpose Interface Bus") or an RS 232 interface (depending on the particular device) to control the amplifier. All functions and features that are available within the official software can also be implemented in your own driver or software.

The interface is fully documented and good examples how to use it exists. For the RS 232 interface any programming language that can read and write to the serial interface can be used. For GPIB various vendors for controllers or cards exist, each with libraries to control them. We used a GPIB-USB controller from National Instruments that utilizes VISA ("Virtual Instrument Software Architecture") and `PyVISA` to control the amplifier via Python.

Regatron TopCon TC.ACS

In contrast to SPS, Regatron provides an own .NET API for their TC.ACS grid simulator. This API provides less functionalities than the official software, e.g. use the device as bidirectional DC-source. However, the desired functions, like changing amplitude and frequency within a specific time, are feasible.

For our solution we implemented a .NET DLL in C# which accomplishes all the communication with the devices and `pythonnet` to directly call the methods within Python. The intermediate step of an own .NET DLL was necessary because Regatron uses an obfuscator to protect its intellectual property. As an unintentional side effect loading the DLL from Python fails. As side benefit of our two-step solution, debugging the communication to the device within a .NET IDE is easier than in a setup where the methods are directly called in Python.

AIT Photovoltaic Array Simulator

The AIT Photovoltaic Array Simulator (PVAS) is a special case of the four described devices, because both the source code of the power electronics and the of the software to control it would be available. Thus, every desired function could be implemented. However, the focus of this work is comparing the given possibilities and not extending them. The PVAS holds up to three PV curves in its memory. Its control software can load curves to the memory and fade the curves in a memory to create arbitrary working points with a desired MPP ("maximum power point") power, voltage and a specific panel curve curvature. This process can be automated using a script generator.

For the PVAS, the only possibility of an external control is to fade the curves stored in the memory via Modbus. Our Lablink implementation writes values into those registers to set the maximum PV power of the PV emulator.

Regatron TopCon Quadro TC.P

The control of the PV emulator from Regatron is similar to their grid emulator. There exists a .NET API where one can set real PV curves, e.g. for crystalline silicon (cSi)or thin film PV panel according to EN 50530, or synthetic curves with constant MPP voltage.

The .NET API is intended for high level use. In contrast to the grid emulator, there exists also a classical ANSI C DLL which provides access to low level functions. The low-level DLL provides additional features like setting the protection limits. Those protection limits can be used to turn off the output immediately when the voltage or current exceeds the limit for a specified time – an important feature to protect the device under test.

To call the low-level DLL functions via Python the package `ctype` was used. The rest was implemented the same way as for the Regatron grid simulator (wrapper in C# and `pythonnet` Python package).

ANALYSIS

In this section the control possibilities of the suppliers' software and the Lablink implementation are compared. Afterwards, timing issues are discussed.

Controllability

Grid Emulators

As shown Table 1, the two grid emulators support all basic functions like setting amplitude and frequency, both direct and via a ramp. It is also possible to setup voltage dips for LVRT tests and start them. Here are no differences between the possibilities of the vendor software and the Lablink implementation.

Looking at harmonics the situation is different. For the Spitzenberger and Spies, one has to create curve forms with the desired harmonics, transfer them to the device and execute it, which is overall quite cumbersome. The Regatron ACS API in the current version 1.4 does not support harmonics.

The vendor software of both amplifiers is not made for running long profiles which last for days and have a new set value every few seconds or even minutes. The Regatron software has no possibility. For the SPS it is possible to run scripts that include wait-commands. So one could write a that generates a script that retraces the set values from a CSV file and this script will then be executed via the SPS software – an unfeasible solution. Such long profiles are the perfect use case for the APIs to externally control the devices, in our case via Lablink.

Table 2: Comparison of different grid emulator functions of vendor software (VS) and Lablink implementation (LL)

	SPS PAS		Reg. ACS	
Function	**VS**	**LL**	**VS**	**LL**
Set Amplitude	✓	✓	✓	✓
Set Frequency	✓	✓	✓	✓
Unbal. grid voltage	✓	✓	✓	✓
Harmonics	✓	~	✓	✗
Ramps	✓	✓	✓	✓
Fast/short profiles	✓	✓	✓	✓
Slow/long profiles	~	✓	✗	✓
Start LVRT pulse	✓	✓	✓	✓
DC Mode	✓	✓	✓	✗
Set values of/via other software	✗	✓	✗	✓

Both grid emulators can run as a DC source/sink. This feature can be utilized to emulate a battery with a controllable battery voltage. The Regatron API does not support to activate the DC mode, which is not an issue because this scenario is not a typical use case and the company has other products for such use cases.

PV Emulators

The two DLLs to control the Regatron PV emulator give full access to the device. Thus, everything that can be done via the vendors software can also be done with the Lablink implementation (cf. Table 3).

Via the Modbus interface of the AIT PVAS one can only set the MPP power. It is not possible to set irradiation and panel temperature via Modbus. This would be necessary for specific, but varying operating points of a cSi module curve.

For typical co-simulation scenarios (e.g. with low voltage grid simulations or prosumers with PV, a home storage system and an electric vehicles) setting the MPP power is sufficient, since parameters like the exact MPP voltage or the exact u/i-curve curvature are not in the focus of examination.

Table 3: Comparison of PV emulator functions of vendor software (VS) and Lablink implementation (LL)

	AIT PVAS		Reg. TC.P	
Function	**VS**	**LL**	**VS**	**LL**
Set P_{MPP}	✓	✓	✓	✓
Set irradiation and panel temperature	~	✗	✓	✓
Set fade time	✓	✓	✓	✓
Create new curves	✓	✗	✓	✓
Load existing curves	✓	✓	✓	✓
Set protection limits	✗	✗	✓	✓
Run scripts	✓	✓	✓	✓
Read measurements and MPP match	✓	✓	✓	✓
Set values of/via other software	✗	✓	✗	✓

Timing considerations

When real lab hardware is used, real time capabilities are always an important issue. Since Lablink is based on MQTT with TCP communication and implemented in Java it is not *hard* real time capable. Though, this *hard* real time criteria was not a design constraint since most software and hardware coupled to the Lablink are not hard real time capable too. For example, because they run on Windows, or use programming languages like Java or C# with traditional garbage collection algorithms.

However, for a typical use case like a co-simulation of a large low-voltage grid, e.g. in DIgSILENT Power-Factory, a weather simulation as well as PV and grid emulators and controllable loads as lab hardware, the given setup is soft real time capable. Those setups have typical time cycle times of a few hundred milliseconds up to a few seconds.

Lablink

Lablink provides a *Sync-Host* which synchronizes and controls the simulation flow of all connected *Sync-Clients*, e.g. a Power-Factory simulation [4]. Using this sync service, the co-simulation participants are synchronous within few milliseconds.

To investigate the latency added by the Lablink framework, payload data has been packed into messages, sent to a broker, received by another process, and unpacked there. Figure 2 shows figure time measurements for a local broker running on the same notebook (*Eclipse Mosquitto*, a simple and lightweight standalone broker), and for *EMQ*, a remote broker running at a server in the company LAN. The tests were performed each *with a pause* of 50 ms between every message (denominated "w. p." in the figure) and *without a pause* (w.o. p.) where messages are sent immediately after each other.

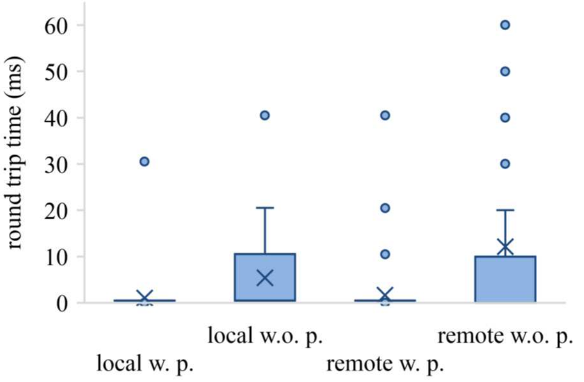

Figure 2: Round trip times of Lablinks Python implementation on a standard consumer notebook (Windows 7, Intel CORE i5 from 2015) with EMQ as local and remote broker, with and without pause between messages

When using a local MQTT broker and a pause between the messages, the time between sending and receiving a Message is below 1 ms. During the analysis, there was just one sample with a delay of 30 ms when the operating system had a high load due to external processes. When

Table 4: Latency between calling the set value function and measuring a new amplitude of AC voltage

Device	Latency
Spitzenberger Spies	
PyVISA, USB2GPIB, etc.	~ 350 ms
thereof SyCore	20 ms
Regatron ACS	
USB interface	~ 950 ms
Ethernet interface	~ 350 ms

the messages are sent immediately after each other, approx. 50% of the messages have a time delay of 10 ms, with an average time delay of 4.87 ms. Again, there are some rare outliers. The 10 ms delay is probably due to full cues in an event handler. Since the delay that small, the issue wasn't further investigated.

Routing the data through the gigabit company network adds no significant delay. Only the amount and value of the outlier increase. Nevertheless, the delay always remained below 60 ms which was considered to be sufficiently low.

Laboratory Hardware

To examine the latency added by the device drivers and the lab hardware themselves, the latency between calling a set function and measuring an output change was determined. Only the overall time could be measured. It was not possible to split the overall time into the times needed for the library (e.g. `PyVISA`), the hardware driver (e.g. USB2GPIB-converter) and the processing time of the amplifier itself.

Also synchronizing the analog measurements with the computer executing the Python scripts is tricky and less accurate than measuring just execution times on a computer or the timespan between to sampled values.

Hence, the presented latencies are only good estimations to give an indication for minimum time steps of a co-simulation environment.

Table 4 shows the latencies for the two grid emulators. Both are in the range of 350 ms. The SPS' data sheet specifies the processing time of its control unit with 20 ms, so the library and the USB2GPIB converter sums up the about 330 ms. There exist also PCI-GPIB cards which probably have a much lower latency.

The Regatron ACS has two possible control interfaces: a serial-over-USB and an ethernet interface. The latency of the USB interface is almost three times the one of the ethernet. Thus, ethernet is preferred over USB.

For the PV emulators it is even more difficult to determine the latency. Setting a new MPP power or irradiance results in a change of the u/i-curve characteristics of the PV emulator. The PV inverter must react to this change before one can see changes.

Furthermore, the rate of set value changes is comparatively low due to cloud and shadowing speed as well as the spatial spread of PV panels. Hence, time resolutions of PV profiles below one second makes hardly any sense. The analysis showed that both emulators easily meet the requirement of a second cycle time.

Summing up, the Lablink framework itself has a latency of few milliseconds for the data exchange between two Lablink connectors. The drivers for controlling the examined lab hardware as well as the processing of the set values in the hardware adds a few hundred milliseconds.

CONCLUSION

In this work Lablink drivers for two grid and two PV emulators were developed. This enables them to be used within a co-simulation environment of pure software simulation and laboratory hardware. An analysis showed that all typical features of the emulators can be utilized using those drivers. An investigation of latencies identified overall delays of a few hundred milliseconds. Hence, the setup is suitable for time synchronous control of the emulators within the Lablink co-simulation environment.

REMARKS

This contribution at no point intends to devalue any implementation of any manufacturer as it discusses a very specific scenario. It is worth noting that all the suppliers compared in this publication provide means for external software and control applications to interact with their product. The suppliers also have a dedicated use-case portfolio in mind when they produce and sell their hardware.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme ([FP7/2007-2013] under grant agreement n°609687 (ELECTRA project), as part of the ELECTRA REX Researcher Exchange Programme).

REFERENCES

[1] L. Martini, H. Brunner, E. Rodriguez, C. Caerts, T. Strasser, G. Burt, 2017, *"Grid of the future and the need for a decentralised control architecture: the web-of-cells concept"*, CIRED – Open Access Proceedings Journal, vol. 2017, no. 1, pp. 1162-1166

[2] T. Strasser, F. Pröstl Andren, G. Lauss, R. Bründlinger, H. Brunner, C. Moyo, C. Seitl, S. Rohjans, S Lehnhoff, P. Palensky, P. Kotsampopoulos, N. Hatziargyriou, G. Arnold, W. Heckmann, E. Jong, M. Verga, G. Franchioni, L. Martini, A. Kosek, O. Gehrke, H. Bindner, F. Coffele, G. Burt, M. Calin, J. Rodríguez-Seco, 2017, *"Towards holistic power distribution system validation and testing-an overview and discussion of different possibilities"*, e & i Elektrotechnik und Informationstechnik, Springer, vol. 134, no. 1, pp. 71-77

[3] M. Nöhrer, M. Faschang, D. Reihs, 2017, *"AIT LabLink – The Link between Smart Grids and the Laboratory"*, Smart Energy Systems Week Austria 2017

[4] D. Stahleder, D. Reihs, M. Nöhrer, F. Lehfuß, 2018, *"Lablink – A novel co-simulation tool for the evaluation of large scale EV penetration focusing on local energy communities"*, CIRED Workshop 2018