From Simulation to Reality: IEC 61499 Compliant Control Applications for Solar Energy Systems

Marc Jakobi[1], Urs Stöckli[1], Luc Meier[1], Tjarko Tjaden[2] and Volker Quaschning[2]

[1] Vela Solaris AG, Winterthur (Switzerland)

[2] HTW Berlin (Germany)

Abstract

This contribution documents the development and validation of open source, model-based control applications for IEC 61499 compliant programmable logic controllers (PLCs) intended for the employment in solar energy systems, such as photovoltaic (PV) systems with a battery and/or heat pump (see Fig. 1). Communication interfaces were created between the simulation tools Polysun and Matlab and the PLC's runtime environment, 4diac RTE (FORTE). This allowed for the development and pre-validation of IEC 61499 control applications in co-simulations with established simulation models. The applications implement the PVprog algorithm, which uses locally sourced PV and load forecasts for charging optimized battery operation (Bergner, et al. 2016), combined with an optional PV curtailment and demand-side management (DSM) of SG Ready heat pumps. For validation, a control application was deployed to a Raspberry Pi 2 and employed in a field test.

Keywords: PV; IEC 61499; PLC; SG Ready; heat pump; battery; simulation; field test; forecasts; optimization

1. Introduction

Decreasing feed-in tariffs and the combination of declining PV energy costs and increasing grid electricity prices are driving the trend for PV systems further in the direction of self-consumption. More and more homeowners are incorporating batteries and heat pumps into their systems. However, political marginal conditions such as feed-in limitations have resulted in the need of more complex, intelligent operational strategies. Today, there is a wide range of "smart energy management" solutions available on the market. Their main tasks usually comprise of the shifting of the load to times of high PV production by means of forecast based battery operation and/or demand-side management (DSM).

The development of such control applications is often a long and tedious process. First, algorithms are prototyped and simulated using development environments such as Matlab/Simulink (Bundesverband Wärmepumpe e. V., 2014). Then, they must be ported to a programming language that can be compiled to the control hardware. Currently available proprietary applications are expensive and, in many cases, non-transparent for installers and end users, regarding the exact type of control that occurs. There exists an open source algorithm for the reduction of curtailment losses, which was devised by Bergner, et al. 2016 in Matlab (Bergner et al., 2016). To be usable in the field, however, it must be ported to other programming languages. The PV industry currently lacks generic, standards-based software for the management of multi-generator energy systems.

2. Objectives

This contribution has the objective to facilitate the development of intelligent control systems by means of open source libraries that enable co-simulations of IEC 61499 control applications and simulation tools. By minimizing the prototyping phase to a bare minimum, the need to port control software to other programming languages is eliminated. With these co-simulation tools, open source control applications that can readily be run on a large variety of programmable logic controller (PLC) hardware, are developed and validated. The applications are focused on grid-connected PV energy system configurations as illustrated in Figure 1.

doi:10.18086/eurosun2018.11.06 Available at http://proceedings.ises.org

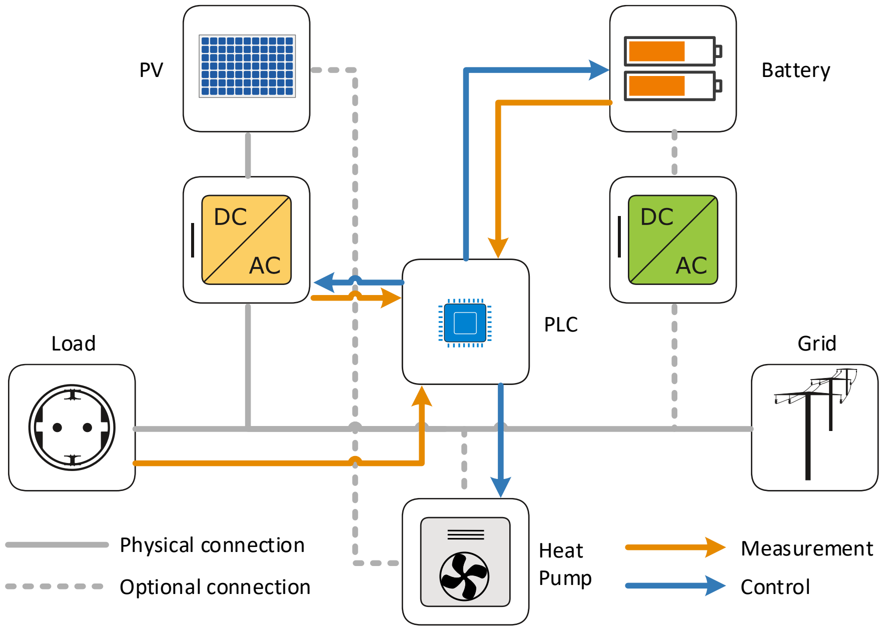

Fig. 1: System configurations and integration of the PLC.

Each system includes a PV generator, an electrical load, an optional battery and an optional heat pump. The controllers act upon the PV inverter, the battery, and the heat pump. Additionally, they take measurements from each of the components. Other household appliances that could potentially be used for DSM, such as washing machines or dishwashers, are not regarded within the scope of this contribution. However, a modular design is intended for the control applications, so that new components can be added later. The objectives can be summarized as follows:

- Facilitation of the development process of control applications by minimizing the prototyping phase.
- Elimination of the need to deploy prototypes into different programming languages.
- Design and validation of generic, standards-based, intelligent open-source control applications.

3. Technology Selection

The criteria for the control applications are discussed in the following subsections. Various technologies are compared and those that best fit the set criteria are selected.

3.1 Criteria

At the time of beginning this project, it was unclear whether deployment and field testing of a control application would be possible within the time constraints. Therefore, one of the main criteria is flexibility. The control applications need to be designed in such a way that their functionality can be validated using the tools at hand. Even with the possibility of field tests, one of the goals is to speed up the development process by reducing the prototyping phase. The control applications designed with simulation tools should be deployable to real systems with as few changes made to them as possible. As mentioned in section 2, the software should be generic and based on standards, in order to achieve a high portability and good comprehensibility for the end-user and developer. Finally, the project is intended to be fully open source. As such, the final products' usability should not be bound by any proprietary software or hardware limitations.

3.2 Standard Selection

Currently, two main standards are defined for PLC control applications: IEC 61131-3 (International Electrotechnical Commission (IEC), 1993) (published in 1993) and IEC 61499 (International Electrotechnical Commission (IEC), 2017) (defined in 2005, revised in 2012), whereby the former has been established in the automation industry (Bolton, 2015). IEC 61131-3 defines five programming languages: Function Block Diagram (FBD), Instruction List (IL), Ladder Diagram (LD), Sequential Function Chart (SFC) and Structured Text (ST). Compliant PLC applications can be programmed in integrated development environments (IDEs) such as CoDeSys, using combinations of these graphical and textual languages.

Tab. 1: Advantages and disadvantages of the PLC standards.

IEC 61131-3	IEC 61499
++ Experience	- Not yet adopted by the industry
+ Large community	- Small community
+ Wide range of software/hardware	o Growing range of software/hardware
- - Low flexibility	++ High flexibility
- - Platform dependent	++ Platform independent
- - Little to no interoperability	++ High interoperability
- - Little to no portability	++ High portability
- Designed for centralized systems	+ Distributed systems possible
- - Limited to simple control systems	++ Complex control systems possible
- Low reliability	+ High reliability

The fact that the standard has been established in the automation industry is the main argument for its use. There exists a large user community and a broad selection of software environments. In the long run, however, the procedural approach cannot live up to the rising requirements for PLCs (Zoitl and Strasser, 2016). Furthermore, IEC 61131-3 does not make any specifications about cross-compatibility between different platforms. As a result, a control application written for one piece of hardware may have to be rewritten from scratch for another controller.

To tackle these drawbacks, IEC 61499 was devised. The standard defines a graphical programming language that uses function blocks (FBs) similar to those defined in FBD. FBs and the control applications are saved in an Extensible Markup Language (XML) format and are not bound to any hardware, thanks to the "IEC 61499 compliance profile for feasibility demonstrations" (Holobloc Inc, 2017). Furthermore, rather than being procedural, the applications are object- and event-oriented. This results in a higher modularity and the possibility of complex, distributed control applications. The disadvantage of the standard lies in the fact that it has yet to be fully adopted by the automation industry (Zoitl and Strasser, 2016). Compared to IEC 61131-3, the standard has a small community and the range of IEC 61499 compliant software and hardware is still rather limited. It is, however, envisaged that IEC 61131-3 will be revised in the future and made compatible with IEC 61499 (Zoitl, 2014). The advantages and disadvantages of the two PLC standards are summarized in Table 1. Since the benefits of IEC 61499 far outweigh the drawbacks, it is chosen for the control applications developed within the scope of this contribution. As an IDE, the open source software 4diac is used.

4. Software Libraries for the Co-Simulation of IEC 61499 control applications with Matlab and Polysun

Because the chosen IEC 61499 compliant development- and runtime environment, 4diac, is still rather young, the available testing and validation utilities are very limited. To work around these limitations, and to enable the validation of IEC 61499 compliant control applications through simulations, two communication libraries were developed in Matlab and JAVA. Both libraries are based on (Holobloc Inc, 2017) and use the Transmission Control Protocol (TCP/IP) and/or the User Datagram Protocol (UDP/IP) to transfer data between the simulation programs and the control applications. The libraries convert IEC 61499 data types (e.g., LREAL, REAL, INT, …) into corresponding Matlab- (double, single, int16, …) or JAVA data types (double, float, int, …), and vice versa. For the use of either of the two libraries, skills in the respective programming languages are required. To extend its usability to engineers without programming skills, the JAVA library was integrated into a Polysun controller plugin. The plugin contains controllers that can be set as actors and sensors for various simulated components (i.e. the battery or heat pump) and configured via a graphical user interface (GUI) to connect and communicate with the control application using one of the aforementioned protocols. For the connection of the control application with the plugin, a corresponding IEC 61499 function block library was developed. The co-simulation between the control logic and a Polysun model is visualized qualitatively in Figure 2. Due to the modular design of the communication interface, the only logic that must be replaced to use the application in a real system is the IEC 61499 communication logic that was previously used for the connection with Polysun. The source code of the Matlab and JAVA libraries, as well as the Polysun plugin, are available on the internet portal GitHub.

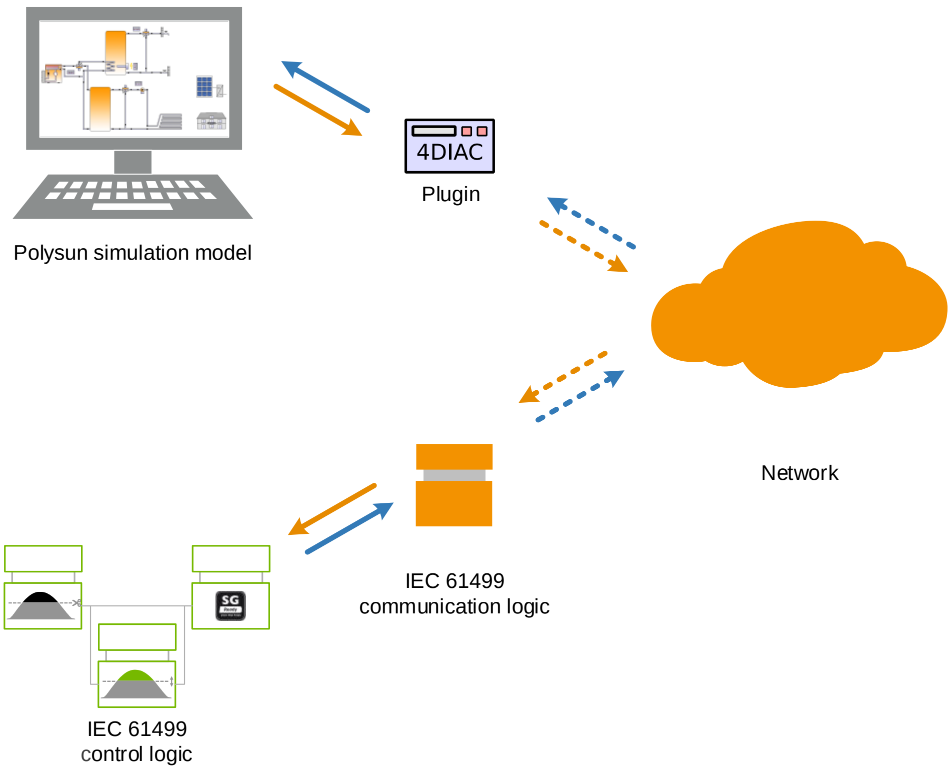

Fig. 2: Visualization of the communication interface between the simulation software Polysun and an IEC 61499 compliant control application (qualitative depiction).

5. Implementation and Co-Simulation of the Control Applications

Three IEC 61499 compliant control sub-applications were developed and pre-validated in co-simulations using the previously devised communication interfaces. The sub-applications can be used alone or connected with each other, respectively. They are described in the following subsections.

5.1 Self-Sufficiency Optimization of SG Ready Heat Pumps

With over 1000 heat pumps certified with the "SG Ready" label (Bundesverband Wärmepumpe e. V., 2014a), the specified control interface has established itself as a standard for heat pumps in Germany. The standard dictates that heat pumps carrying the label must make it possible to influence their operation using two switch contacts. Each contact can have the state "on" (1) or "off" (0). This results in four operation modes, which are listed and described in Table 2. Unfortunately, there is currently no fixed definition for the SG Ready modes 3 and 4. Normally, the heat pump's internal controller automatically switches back to mode 2 based on different set temperature limits for the storage tank and a hysteresis. From a user's perspective, it would be desirable to have clearly defined factory settings for the control modes, i.e. fixed set points for the storage tank temperature thresholds or increases thereof, in addition to clarifications on whether or not cartridge heaters are used. As part of this project, a control application was devised based on (Tjaden et al., 2017). It performs DSM on the heat pump to shift its use to times of high excess PV power, in order to optimize self-supply and minimize the amount of electricity purchased from the grid.

Tab. 2: The four states of SG Ready heat pumps. Source: (Bundesverband Wärmepumpe e. V., 2014b).

#	Contact States	Short Description	Long Description
1	1 : 0	Off	The heat pump is turned off for a maximum of 2 hours.
2	0 : 0	Normal	The heat pump runs in normal operation (determined by an internal controller)
3	0 : 1	Amplified I	A recommendation for the heat pump to turn on. Whether the heat pump is turned on or not, is determined by an internal controller.
4	1 : 1	Amplified II	A definitive instruction to turn the heat pump on as long as its internal controller deems it possible.

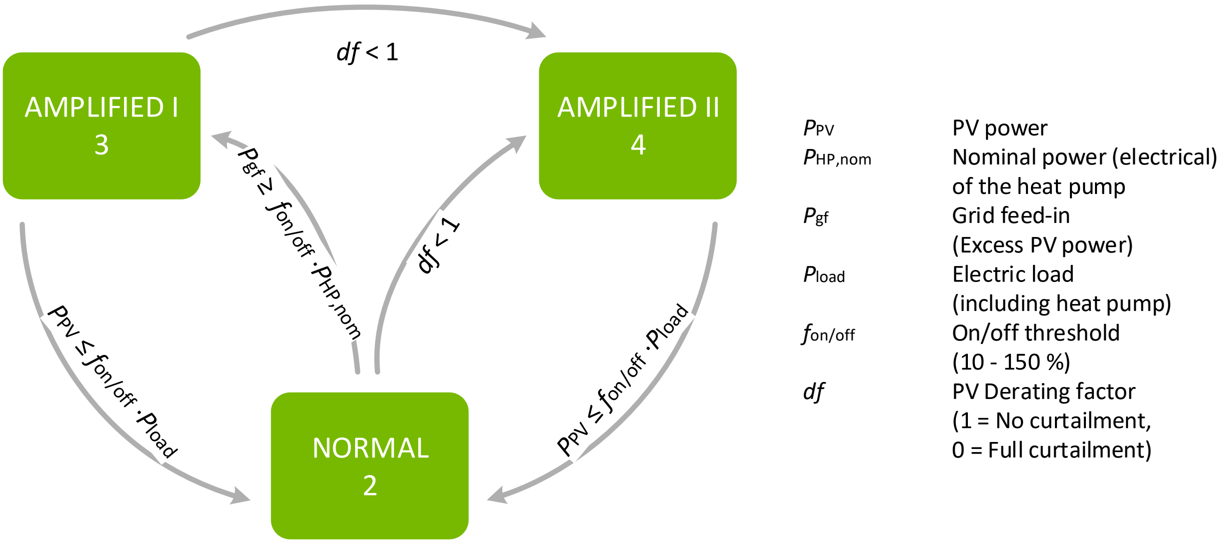

Fig. 3: Switching conditions of the SG Ready heat pump control logic.

The conditions for switching the heat pump from one operation mode to another is illustrated in Figure 3. Control mode 1 is not utilized by the control application. In the simulation model, it is assumed that the heat pump's internal control unit switches back to normal mode from mode 3 at a temperature threshold of 55 °C in the third layer of the storage tank, with a hysteresis of 5 K. For mode 4, the threshold is set to 70 °C. The IEC 61499 control application uses an on/off threshold of 100 %, as recommended by (Tjaden et al., 2017) for systems with a PV/heat pump nominal power ratio of 3 : 1. Figure 4 compares two PV systems with a heat pump used for heating and domestic hot water on a selected sunny day, simulated in Polysun.

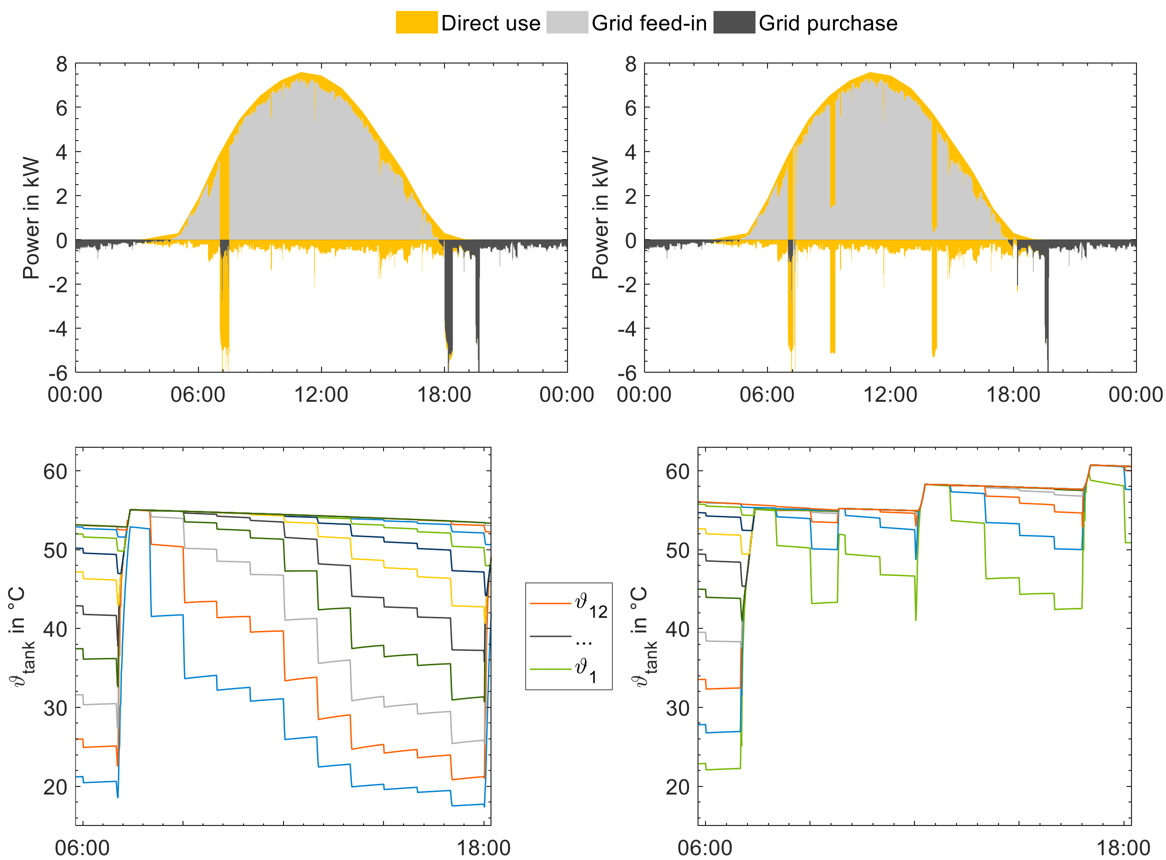

Fig. 4: Comparison of a PV system with a heat pump without (left) and with (right) an SG ready controller, simulated in Polysun. Above: Power flows on a sunny day. Below: Temperatures in the layers of the drinking water tank on the same day. Nominal PV power: 9.9 kWp; Nominal heat pump power: 3.3 kW (electrical), 10.1 kW (thermal); Household load profile: 5000 kWh/a; Heat pump consumption: Approx. 5400 kWh/a (depending on the operation strategy).

The system depicted on the left-hand side uses a regular heat pump and the system on the right is controlled using the SG Ready heat pump controller. The respective power flows are illustrated on top, with temperature levels in the various drinking water tank layers plotted below. Without the controller, the heat pump turns on once in the morning and heats up the tank. During the day, the temperatures in the individual tank layers decrease gradually, so that the stored energy is not enough to cover the demand in the evening. The controller adds two load peaks of the heat pump to the daytime, thus reducing its use at night and almost eliminating the large grid-supply peak at 6 PM. Instead of falling steadily during the day, the temperatures within the controlled system's buffer storage tank are kept at higher levels. By 6 PM, the second layer has a temperature above 50 °C, which is more than enough to satisfy the hot water demand. In the one-year simulation, the degree of self-sufficiency is increased from 21.2 % to 24.5 %, the own consumption ratio improves from 22.3 % to 26 % and the grid feed-in power and grid purchase are each reduced by 364 kWh and 298 kWh, respectively. This load shift is approximately equal to the monthly electricity consumption of the heat pump without an SG Ready controller outside of the heating period. By fine-tuning the on/off threshold, adding a larger drinking water tank and with a more energy efficient building (Tjaden, 2015), the benefits of using the controller would increase even further.

5.2 PV Curtailment

The simplest way to limit PV feed-in power to a set value is curtailment. This is usually implemented by PV systems e.g., from SolarLog or SMA. Control applications that require the theoretical PV production before curtailment as an input, however, operate more effectively if they can communicate with the control unit performing curtailment to estimate the derated power (Jakobi, 2017). For this purpose, a subapplication for the curtailment of PV systems was developed as part of this project. It implements a simple proportional-integral-derivative (PID) controller. The set-point value is the running 10 min average of the grid feed-in power (as specified by the German KfW incentive programme, "Erneuerbare Energien, Speicher" (Forum Netztechnik/Netzbetrieb im VDE (FNN), 2014)). In a real system, curtailment can be determined by communicating with inverters that implement the "SunSpec open protocol for interoperability between devices in renewable energy systems".

5.3 Forecast-Based Operation of PV Battery Systems

It is possible to eliminate curtailment losses by intelligently operating a PV battery system. The simplest way to achieve this is to set a fixed feed-in limit, above which the battery is charged. The downside of this method is that it results in a lower use of the battery, compared to a feed-in limitation only through curtailment (Weniger, 2013). By using PV and load forecasts, a dynamic feed-in limitation that solves this problem can be realized. The research group "PV storage systems" at HTW Berlin developed "PVprog", a model-based control algorithm, in Matlab. It utilizes measurement-based PV and load forecasts to dynamically limit the PV feed-in power using a battery (Bergner et al., 2016). For this contribution, the algorithm was ported to an IEC 61499 sub-application (see Figure 5) and validated against a Matlab simulation model provided by HTW Berlin.

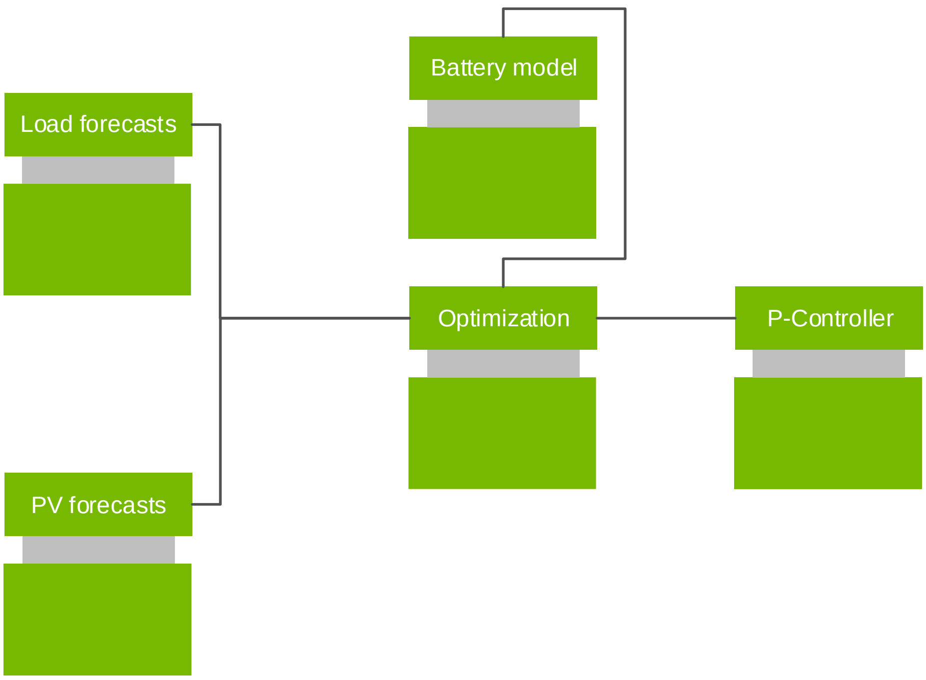

Fig. 5: Simplified depiction of the IEC 61499 implementation of the PVprog algorithm.

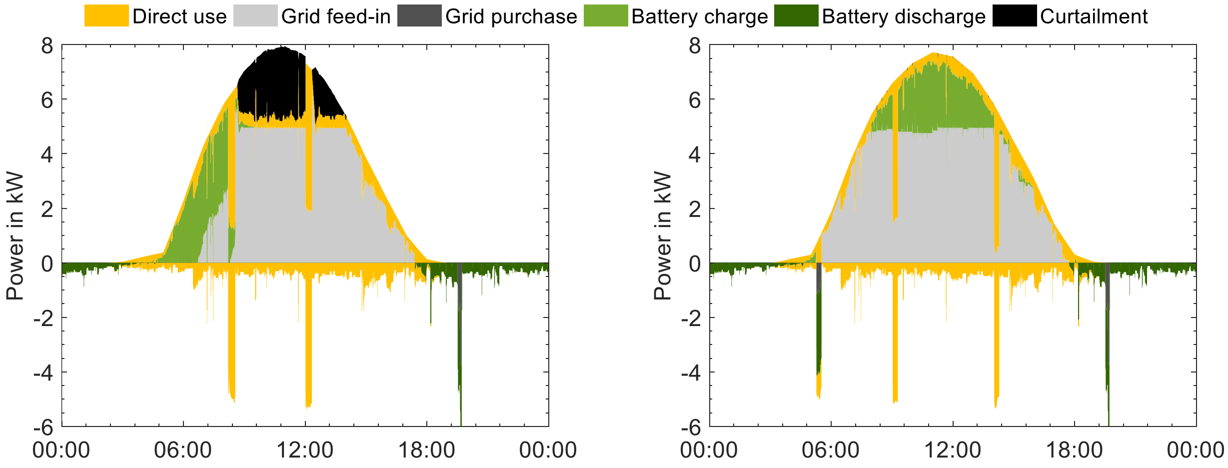

Fig. 6: Comparison of a PV system with an SG Ready heat pump and a battery (power flows on a sunny day). Left: The battery is charged as soon as a PV surplus occurs. Right: Forecast-based battery charging. Nominal PV power: 9.9 kWp; Usable battery capacity: 10 kWh; Nominal heat pump power: 3.3 kW (electrical), 10.1 kW (thermal); Household load profile: 5000 kWh/a; Heat pump consumption: Approx. 5400 kWh/a (depending on the operation strategy); Feed-in limitation: 50 %.

For a higher modularity, the sub-application decouples the battery model from the optimization, which resulted in slight changes to the optimization algorithm. The original Matlab implementation iterates through feed-in limitations between 0 % and 100 % of the nominal PV power in equal intervals and pre-simulates the battery over the forecast horizon of 15 h. An optimum is found when the limitation at which the used battery capacity can be kept as close as possible to the forecasted PV surplus energy while adhering to the maximum feed-in power. The IEC 61499 implementation, on the other hand, increases the dynamic feed-in limitation step-by-step, until the battery model cannot be charged any further or until the maximum allowed feed-in limit is reached. This results in a slightly higher resolution of the iteration steps, which in turn results in a slight change in the simulation results. The degree of self-sufficiency is increased by 0.1 % and the curtailment losses are increased by 0.5 %. These deviations are most likely purely stochastic. After the first validation, the control application of the system simulated in Polysun in section 5.1 was extended with the PVprog sub-application and the curtailment sub-application described in section 5.2. Additionally, a battery with a capacity of 10 kWh was added to the system and the feed-in limit was set to 50 %. Figure 6 compares the simulation results of a system with (right) and without (left) the PVprog sub-application. The use of the battery alone does not disturb the heat pump operation. However, it is not enough to prevent PV curtailment at noon. As long as the PVprog and SG Ready subapplications communicate with each other (Jakobi, 2017), the forecast-based operation of the battery can prevent almost all curtailment losses. In the co-simulated system, the degree of self-sufficiency is increased to 39.5 % by adding the battery and again to 42 % by using the PVprog sub-application (cf. section 5.1). The own consumption ratio increases to 44.7 % and then to 45.1 %. Adding the battery reduces the grid feed-in power by another 2622 kWh and incorporating the PVprog sub-application increases it again, by 281 kWh. The electricity purchased from the grid is reduced further by 1606 kWh and by another 229 kWh, due to the forecast-based operation.

6. Field Test

Among other goals, objectives of this project include a high flexibility and the minimization of the prototyping phase by making it possible to directly use the developed control applications in real systems with as few as possible changes made to the control logic. To achieve this, the interfaces between the simulation software and the control logic was implemented in a modular manner, so that only the communication logic must be swapped out (see section 4). Today, commercial IEC 61499 IDEs like ISaGRAF and nxtSTUDIO can already automatically configure the communication logic. This is also planned for 4diac (Zoitl and Strasser, 2016). To validate that this project's goal has been achieved, the PVprog sub-application from section 5.3 was deployed to a Raspberry Pi 2 and tested in a real PV battery system.

6.1 Set-Up

The control application was integrated into the "Living Equia" house at HTW Berlin (HTW Berlin, 2017). The building has a PV system with a nominal capacity of 4.6 kWp and a "Sonnenbatterie" battery system with a usable capacity of 5.3 kWp, as stated by the manufacturer. To be able to control the battery, a module for the 4diac runtime environment that implements the Hypertext Transfer Protocol (HTTP) had to be developed.

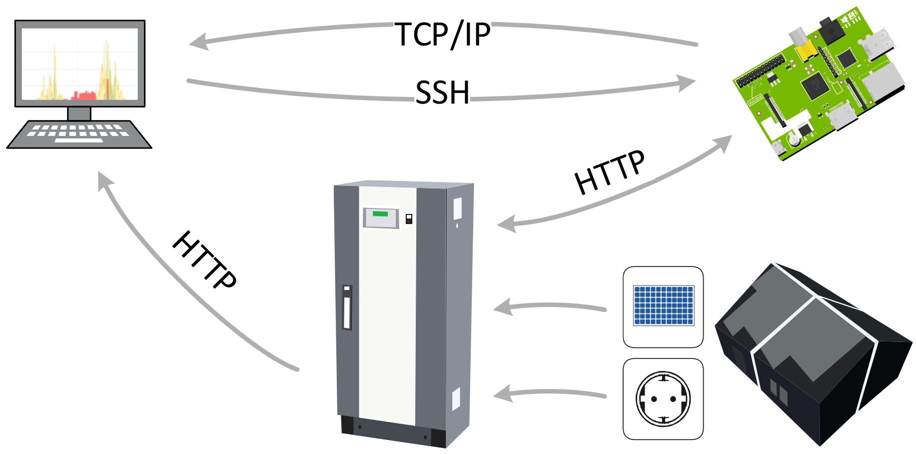

Fig. 7: Visualization of the field test set-up and the communication paths.

With this module, an IEC 61499 function block library for the Sonnenbatterie application programming interface (API) (Sonnen GmbH, 2013) was devised. The set-up of the field test is visualized in Figure 7. The control application on the Raspberry Pi obtains system measurements from a RESTful server, which is hosted on the Sonnenbatterie. On the one hand, the status and power flow measurements can be accessed by a web interface provided by the battery manufacturer. On the other hand, they can be obtained directly from the control application; together with information about its status. Access to the Raspberry Pi's operating system, on which the 4diac runtime environment is installed, can be achieved via Secure Shell (SSH).

6.2 Results

The power flows of the field test are depicted in Figure 8 for a selected sunny day. They prove that the IEC 61499 PVprog sub-application performs as envisaged in a real system. Early in the morning, the dynamic feed-in limit is set to a low value and adjusted for the high PV production during the day. The clouds in the late afternoon cause the limit to be lowered again. Reasons for the initially low feed-in limitation can include:

- High shadowing in the morning of the current day or evening of the previous day.
- The control application is still in an initializing state, which can take up to 10 days if every day is cloudy.
- The last 10 days were very cloudy, which can lead to an incomplete initialization of the PV forecasts.
- The useable battery capacity is lower than stated by the manufacturer (e.g., due to degradation).

In this case, the day before was completely sunny, which eliminates the second and third points as possible reasons. Apart from the last point, the first one could be a reason, because the test was performed in September.

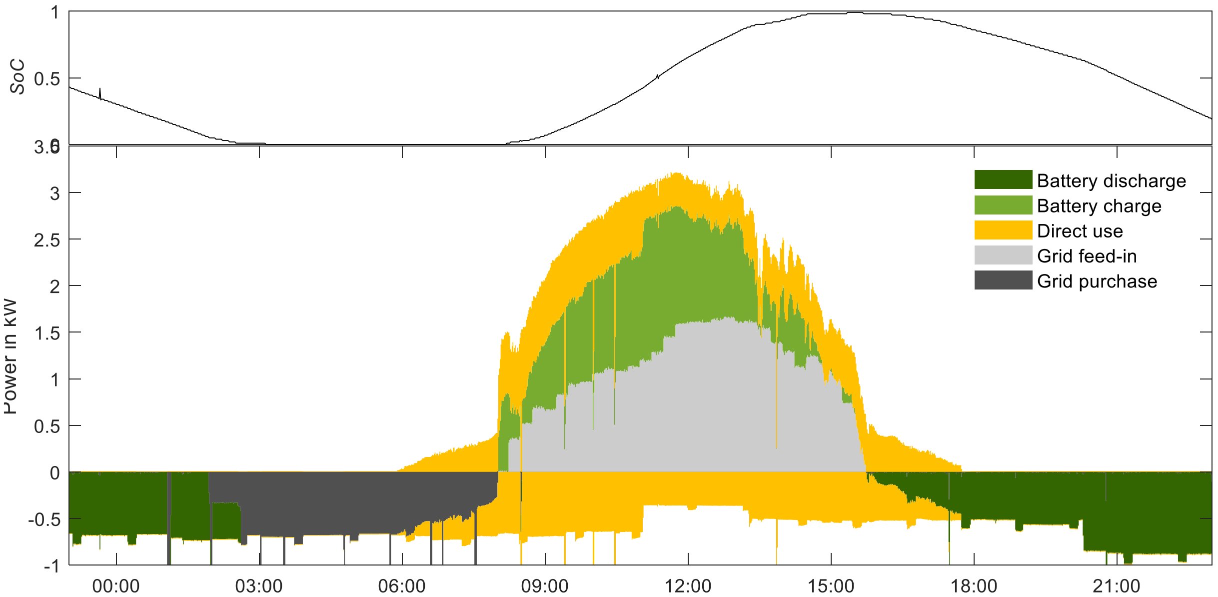

Fig. 8: Measured power flows (below) and battery state of charge (SoC) (above) on a selected sunny day during a field test of the IEC 61499 PVprog sub-application running on a Raspberry Pi 2. Nominal PV power: 4.6 kWp; useable battery capacity: 5.3 kWh.

This is a time in which the PV power is reduced slightly every day, due to the Sun's declination; especially in the mornings and evenings, in which horizon shading is most prominent. The degraded useable battery capacity can be accounted for within the control application by implementing a sub-application that estimates the capacity according to how much energy is charged into and discharged from the battery.

7. Conclusion

The power flows of the field test in Figure 8 clearly show a shift of the battery charging to noon, which results in an elimination of PV curtailment losses. It can hence be concluded that the utilized simulation tools Polysun and Matlab have proven themselves to be reliable development aids for IEC 61499 control applications that can be implemented on a large variety of hardware. The need for standard-based control software has been met with a completely open source solution. Future work will include the implementation and development of additional communication protocols and robust standard control algorithms for multi-generator and multi-storage renewable energy systems.

8. Source Code

The source code of the projects described in this paper can be obtained via the following URLs:

- PVprog algorithm in Matlab:
www.pvspeicher.htw-berlin.de/veroeffentlichungen/daten/pvprog
- tcpip4diac: Matlab library for communication with IEC 61499 PLC applications:
www.github.com/MrcJkb/tcpip4diac
- Polysun4diac: Polysun plugin and JAVA library for communication mit IEC 61499 PLC applications:
www.github.com/MrcJkb/Polysun-4diac-ControllerPlugin
- IEC 61499 function block library and IEC 61499 PLC applications developed within the scope of this project:
www.github.com/MrcJkb/PVTControllerLib
- HTTP module for 4diac-RTE (discontinued, to be integrated into the official release of 4diac-RTE):
www.github.com/MrcJkb/forte_http_com

9. References

Bergner, J., Weniger, J., Tjaden, T., 2016. PVprog-Algorithmus - Algorithmus zur Umsetzung der prognosebasierten Batterieladung für PV-Speichersysteme mit messwertbasierten PV- und Lastprognosen. Hochschule für Technik und Wirtschaft HTW Berlin, Berlin.

Bolton, W., 2015. Programmable logic controllers. Newnes, Kidlington, Oxford, UK; Waltham, MA.

Bundesverband Wärmepumpe e. V., 2014a. Modell-Liste "SG Ready"-Label für Smart-Grid-fähige Wärmepumpen [WWW Document]. SG Reagy-Label - Für Smartgrid Geeignete Wärmepumpen. URL http://www.waermepumpe.de/waermepumpe/qualitaetssicherung/sg-ready-label.html (accessed 12.16.14).

Bundesverband Wärmepumpe e. V., 2014b. Regularien des SG Ready-Labels [WWW Document]. SG Reagy-Label - Für Smartgrid Geeignete Wärmepumpen. URL http://www.waermepumpe.de/fileadmin/redakteurdaten/Waermepumpe/Qualitaet/SG_READY/2012-11-01_SG_Ready_Regularien_Version1.1.pdf (accessed 12.16.14).

Forum Netztechnik/Netzbetrieb im VDE (FNN), 2014. Anschluss und Betrieb von Speichern am Niederspannungsnetz. Forum Netztechnik/Netzbetrieb im VDE (FNN), Berlin.

Holobloc Inc, 2017. IEC 61499 Compliance Profile for Feasibility Demonstrations [WWW Document]. URL http://www.holobloc.com/doc/ita/index.htm (accessed 4.26.17).

International Electrotechnical Commission (IEC), 2017. IEC 61499 - International Standard for Distributed Systems [WWW Document]. URL http://www.iec61499.com/ (accessed 4.27.17).

International Electrotechnical Commission (IEC) (Ed.), 1993. IEC 61131 Programmable controllers. International

Electrotechnical Commission (IEC), Geneva.

Jakobi, M., 2017. Development of model-based control applications compliant with IEC 61499 with a focus on PV (Master’s thesis). HTW Berlin, Berlin.

HTW Berlin, 2017. Solar Decathlon Europe 2010 - Living Equia - Teilprojekt HTW Berlin (ENOB) (SD Europe (BMWi-Energieoptimiertes Bauen Bundesförderung)) - Hochschule für Technik und Wirtschaft Berlin University of Applied Sciences - HTW Berlin [WWW Document]. URL http://www.htw-berlin.de/forschung/online-forschungskatalog/projekte/projekt/?eid=1161 (accessed 9.21.17).

Sonnen GmbH, 2013. Sonnenbatterie API for battery systems operated in slave mode.

Tjaden, T., 2015. PV-Systeme mit Wärmepumpe ideal betreiben. Pv Mag. 106–108.

Tjaden, T., Weniger, J., Quaschning, V., 2017. Richtige Dimensionierung von Photovoltaik, Wärmepumpen und Pufferspeichern.

Weniger, J., 2013. Dimensionierung und Netzintegration von PV-Speichersystemen (Masterthesis). Hochschule für Technik und Wirtschaft Berlin, Berlin.

Zoitl, A., 2014. Modelling control systems using iec 61499, 2nd edition. ed, IET control engineering series 95. The Institution of Engineering and Technology, Stevenage, Herts, United Kingdom.

Zoitl, A., Strasser, T., 2016. Distributed control applications: guidelines, design patterns, and application examples with the IEC 61499.