



Hardware-in-the-Loop Simulation of Controlled and Uncontrolled EV Charging in a Distribution Grid

Lode De Herdt

Hardware-in-the-Loop Simulation of Controlled and Uncontrolled EV Charging in a Distribution Grid

by

Lode De Herdt

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended privately on Friday October 23, 2020 at 15:00.

Student number:	4390601	
Project duration:	January 13, 2020 – October 23, 2020	
Supervisor:	Prof. dr. ir. P. Bauer,	TU Delft
Daily Supervisor:	Dr. ir. G. R. Chandra Mouli,	TU Delft
Advisors:	Ir. Y. Yu,	TU Delft
	Dr. ir. A. Shekhar,	TU Delft
Thesis committee:	Prof. dr. ir. P. Bauer,	TU Delft
	Dr. ir. G. R. Chandra Mouli,	TU Delft
	Dr. ir. M. Cvetković,	TU Delft

This thesis is confidential and cannot be made public until October 23, 2021.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

The charging of a rapidly increasing number of EVs places significant strain on electrical distribution grids. Controlled ('smart') charging of EVs offers a solution to this problem by shifting the charging of vehicles to the most optimal moments in time. Smart charging is not a completely new research topic; it has a high Technology Readiness Level and some commercial implementations already exist. There is a large amount of literature available that compares the effect of controlled and uncontrolled EV charging on the distribution grid. However, most of this literature is based on pure software simulations using power flow analysis and mathematical models of EV charging behaviour. Experimentally validating these results would require a large number of EVs and a controllable distribution grid, which is costly and unpractical. Therefore, this thesis presents a compromise; a hardware-in-the-loop (HIL) testbed in which the distribution grid is software-based but the EV and EV Supply Equipment (EVSE) is hardware-based. In this report, the design and implementation of the testbed is discussed, as well as results from HIL simulations that compare the effect of controlled and uncontrolled EV charging in a distribution grid.

The low-cost HIL testbed presented in this thesis is based upon an OPAL-RT OP5700 Digital Real-Time Simulator (DRTS) which runs a Newton-Raphson power flow analysis of a Dutch distribution grid. The DRTS also interfaces with all other hardware in the testbed. This includes a power amplifier which generates a three-phase grid at the voltage calculated by the power flow analysis. A commercially available EVSE is connected to this power amplifier and supplies power to an emulated EV which consists of communication hardware and two bidirectional back-to-back DC power supplies. These power supplies act as an AC load. Various communication protocols were implemented to exchange information between the different systems in a manner that closely represents commercial EVs.

A smart charging algorithm which determines the optimal charging current setpoints in real-time, based on external factors like local load power consumption, solar irradiance and energy prices, was implemented into the testbed. The testbed has subsequently been used to study the effect of uncontrolled and controlled charging on the simulated distribution grid for a total of eight different scenarios. Of these scenarios, one consists of traditional uncontrolled charging and the other seven are with the smart charging algorithm activated. These include firstly a base case, then two scenarios with an inaccurate solar irradiance and local load forecast, a further two scenarios with non-ideal EV charging behaviour, and lastly two scenarios with respectively centralized and decentralized curtailment implemented. In all scenarios, one EV was implemented as hardware and three others as software, so that a comparison could be made.

Even under non-ideal conditions, smart charging is found to reduce costs, grid overloads, and voltage deviations with respect to uncontrolled charging. The implemented centralized curtailment is shown to be more effective than decentralized and further reduces voltage deviations in the grid but at a slightly higher cost than without this curtailment. Lastly, based on observations made in the HIL simulation results, recommendations for future work and improvement of the smart charging algorithm are given.

Preface

This thesis project started 9 months ago with the personal goal of improving my practical skills and obtaining a deeper understanding of the effects of new sustainable technologies on our electrical infrastructure. During this time, I have had the unique opportunity to work with a variety of lab equipment and I was given significant freedom to design, construct, and test a large experimental setup by myself. Within this project, I have relied upon knowledge gained throughout my academic career, including my BSc Electrical Engineering, MSc Sustainable Energy Technology, and gap year at student-team Forze Hydrogen Electric Racing. Much of this thesis project took place during the unprecedented times of a pandemic, but I am very fortunate that the impact of this on my project remained limited.

The completion of this master thesis marks the end of my time as a student at the TU Delft. The past 6 years have been an amazing journey of personal and academic development and I would not be where I am now without the support of my friends and family. I would especially like to thank my parents, Myriam & Tim and Bart & Eva. Without their years of financial and emotional support, none of this would have been possible. I am forever grateful for the unique opportunities they have given me and realise that not everyone is so lucky.

I would also like to express my appreciation for the staff of DCE&S. My daily supervisors Yunhe Yu and Dr.ir. Aditya Shekhar provided guidance throughout the project and were always available for questions. Their continued enthusiasm whenever I showed my progress was a great source of motivation. My supervisor Dr.ir. Gautham Ram Chandra Mouli never failed to amaze me with his incredibly insightful feedback and suggestions during our monthly meetings. Despite supervising so many MSc students, he was always up-to-date on my progress and showed great expertise on the topic. The DCE&S lab technicians - Joris, Bart, Harrie, Vasileios and Geert Jan - helped me on countless occasions with ordering new supplies and providing advice for my experimental setup. I would especially like to thank them for setting up the remote access that allowed me to work from home during the COVID-19 lockdown.

Last but not least, I would like to thank Prof.dr. Pavol Bauer, Dr.ir. Gautham Ram Chandra Mouli and Dr. Miloš Cvetković for taking time out of their schedule to be part of my graduation committee. I look forward to your evaluation!

*Lode De Herdt
Delft, October 2020*

Contents

List of Figures	ix
List of Tables	xi
List of Abbreviations	xiii
1 Introduction	1
1.1 Research Motivation	1
1.2 Research Objectives	3
1.3 Thesis Outline	3
2 Literature Review	5
2.1 Smart Charging	5
2.1.1 Potential Benefits of Smart Charging	5
2.1.2 Main Algorithms in Literature	6
2.1.3 Relevant Protocols	7
2.2 Real-Time Simulations	8
2.2.1 Simulation vs Emulation	9
2.2.2 Types of Real-Time Simulations	9
2.3 Why PHIL?	10
2.4 Similar Experiments	11
2.4.1 AIT's FlexEVELab	12
2.4.2 PowerLabDK SYSLAB	12
2.4.3 University of Suceava's EV Simulator	13
2.4.4 Overview & Research Gap	14
3 Hardware	15
3.1 Overview	15
3.2 Real-Time Simulator	17
3.2.1 OPAL-RT OP5700	17
3.2.2 Modelling for Real-Time Simulation	18
3.3 Grid Emulation	18
3.3.1 AST1501 AC Source	18
3.3.2 Simulink Implementation	19
3.3.3 Validation	20
3.4 EVSE	20
3.4.1 Alfen EVE Single Pro-Line	20
3.4.2 PLC for Modbus TCP Communication	21
3.5 EV Emulator	22
3.5.1 Communication between EVSE and EV-Emulator	22
3.5.2 Walther-Werke Type 2 EV-Emulator	24
3.6 SM15K DC Source	25
4 Smart Charging Algorithm	27
4.1 Objective Function	27
4.2 Sub-Functions	28
4.3 Constraints	29

5	System Integration	31
5.1	Overview	31
5.2	Adaptation of Algorithm	33
5.3	Modbus Communication Using PLC.	34
5.4	Circulating Power.	36
5.4.1	Layout.	36
5.4.2	Measurements	36
5.5	Simulink Model	38
5.5.1	Top-Level	39
5.5.2	Master.	39
5.5.3	Slave	40
5.5.4	Console	41
6	Uncontrolled Charging	43
6.1	Simulation scenario	43
6.2	Simulation results.	46
7	Smart Charging	47
7.1	Case 1: Base case	47
7.2	Case 2: Inaccurate PV Forecast.	51
7.3	Case 3: Inaccurate Load Forecast	53
7.4	Case 4: EV Current Lower Than Setpoint.	55
7.5	Case 5: Battery Capacity Fading	56
7.6	Case 6: Centralized Curtailment.	58
7.7	Case 7: Decentralized Curtailment	60
7.8	Comparison.	63
8	Conclusions & Recommendations	65
8.1	Conclusion	65
8.2	Recommendations for Improvement of Smart Charging Algorithm	66
8.3	Recommendations for Future Work	67
A	Appendix A: EVSE Modbus Mapping Table	69
B	Appendix B: Simulink Model	75
	Bibliography	85

List of Figures

1.1	Predicted future global EV sales	1
1.2	Load, solar, and wind profiles for California on March 29 in a scenario with 11% annual wind and 11% annual solar	2
2.1	Overview of possible protocols involved in smart charging	7
2.2	Illustration of real-time (a) and non-real-time (b) simulation	9
2.3	Structure of HIL-based simulation concepts: CHIL (left) vs. PHIL (right)	10
2.4	A Top-Level Overview of AIT's FlexEVELab Setup	12
2.5	A Top-Level Overview of the PowerLabDK SYSLAB Setup	13
2.6	Overview of University of Suceava's EV Simulator and EVSE setup	13
3.1	Infographic of a PHIL setup proposed within the OSCD project	15
3.2	Overview of experimental setup	16
3.3	A typical PHIL setup	17
3.4	OPAL-RT OP5700 Real-Time Simulator	17
3.5	California Instruments AST1501	18
3.6	Measured Voltage Output of Power Amplifier	19
3.7	Single-line diagram of a Dutch rural grid used for grid simulation.	20
3.8	Comparison of Power Flow Results Calculated by DIgSILENT PowerFactory and the Adapted Newton-Raphson Simulation	20
3.9	Alfen EVE Single Pro-Line	21
3.10	Schneider Electric TM221ME16TG PLC	21
3.11	Labeled image of a type 2 plug	22
3.12	Graph of CP signal at different charging states	23
3.13	Walther-Werke EV-Emulator external (a) and internal (b) view	24
3.14	Delta-Elektronika SM500-CP-90 Bidirectional DC Power Supply	25
5.1	Overview of experimental setup	32
5.2	Flowchart showing how the Python algorithm runs alongside the real-time simulation for N nodes	33
5.3	Part of the Modbus register table of the Alfen EVE Single Pro-Line	34
5.4	Screenshots of the Channel Assistant (a) and input registers (b) of the PLC	34
5.5	Schematic overview of experimental setup	36
5.6	Annotated picture of experimental setup. The annotation colors used correspond to the colors of figure 3.2.	36
5.7	Circulating and input current on phase C with various circulating powers	37
5.8	Measured input power as a function of recirculating power	38
5.9	Graphical representation of a sine wave showing its digitized staircase form in greater detail	38
5.10	Constant current - constant voltage charging profile of a battery	40
6.1	Picture of complete experimental setup. The annotation colors correspond to figure 5.1.	43
6.2	Locations of the EVs, local PV generation and local loads	44
6.3	experimental results of uncontrolled charging	46
7.1	Day-Ahead market energy price	47
7.2	Experimental results of base case	48
7.3	Cause of spikes in grid import	49
7.4	Grid Import at Node 5, software simulation	49

7.5	Intra-minute exceeding of grid import limit, software simulation	50
7.6	Calculated and measured voltage	51
7.7	Setpoint and measured current	51
7.8	Forecast and actual PV power, used in case 2	52
7.9	Comparison of Grid Import for Case 2 and Base Case. A 5 minute moving average was used to more clearly show the difference.	52
7.10	Experimental results of case 2	53
7.11	Forecast and actual local load power, used in case 3	54
7.12	Experimental results of case 3	54
7.13	Experimental results of case 4	55
7.14	Setpoint and measured current of EV1	56
7.15	Experimental results of case 5	57
7.16	Perceived and actual SOC due to capacity fading	58
7.17	Experimental results of case 6	59
7.18	Current imported from grid at node 5 & 15, with variable limit	60
7.19	Curtailment of EV charging current as a function of bus voltage	61
7.20	Experimental results of case 7	61
7.21	Yo-yo effect of decentralized curtailment causes voltage oscillations	62
7.22	Probability density of bus 5 voltage	64
B.1	Top-Level overview of Simulink model	76
B.2	Master Subsystem	77
B.3	Subsystem within Master which converts the Control Pilot duty cycle to an analog setpoint for the AC load.	78
B.4	Subsystem within Master which converts a PWM duty cycle received from the PLC into the measured phase voltage	78
B.5	Subsystem within Master which performs sanity checks on the incoming duty cycle of the control pilot signal. For safety reasons, if the received signal is not within expected bounds, the AC load will not draw any power.	78
B.6	Subsystem within Master which implements EV charging behaviour; it stops the charging when $SOC \geq 1$ and implements constant-voltage charging.	79
B.7	Slave Subsystem	80
B.8	Subsystem within Slave which reads data from external files	81
B.9	Subsystem within Slave which calculates the EVs' SOC by integrating their charging power.	82
B.10	Subsystem within Slave which calculates power going to each EV based on current and bus voltage	83
B.11	User Interface (Console) Subsystem	84

List of Tables

2.1	Comparison of similar experimental setups found in literature	14
3.1	Most relevant features of TM221ME16TG PLC	22
3.2	Relationship between rated cable current and resistance between PP and PE	23
3.3	Resistances and voltages between CP and PE corresponding to each charging state	23
3.4	Relationship between CP duty cycle and maximum allowed current	24
4.1	Parameters of smart charging objective function	28
4.2	Parameters of smart charging sub-functions	29
4.3	Parameters of smart charging constraints	30
5.1	Demonstration of how float and word objects overlap in the PLC's memory	35
5.2	Input power provided by power amplifier	37
6.1	Parameters used for uncontrolled & controlled charging scenarios	45
7.1	Comparison of simulation results for different cases	63
7.2	Relative change w.r.t uncontrolled charging	63

List of Abbreviations

- AC** Alternating Current.
- AIT** Austrian Institute of Technology.
- API** Application Programming Interface.
- BNC** Bayonet-Neill-Concelman.
- CHIL** Controller Hardware-in-the-Loop.
- CP** Control Pilot.
- CPO** Charge Point Operator.
- CV** Constant-Voltage.
- DC** Direct Current.
- DCE&S** DC Systems, Energy Conversion & Storage.
- DRTS** Digital Real-Time Simulator.
- DSO** Distribution System Operator.
- EV** Electric Vehicle.
- EVSE** Electric Vehicle Supply Equipment.
- HIL** Hardware-in-the-Loop.
- HUT** Hardware Under Test.
- I/O** Input/Output.
- ICE** Internal Combustion Engine.
- IEC** International Electrotechnical Commission.
- IP** Internet Protocol.
- IRENA** International Renewable Energy Agency.
- ISO** International Organization for Standardization.
- KNMI** Royal Netherlands Meteorological Institute.
- NEDU** Association for Dutch Energy Data Exchange.
- OCPP** Open Charge Point Protocol.
- OSCD** Orchestrating Smart Charging in mass Deployment.
- PE** Protective Earth.

PHIL Power Hardware-in-the-Loop.

PK Peak.

PLC Programmable Logic Controller.

PP Proximity Pilot.

PV Photovoltaic.

PWM Pulse-Width Modulation.

RFID Radio-Frequency Identification.

RMS Root Mean Square.

SIL Software-in-the-Loop.

SOC State of Charge.

TCP Transmission Control Protocol.

V2G Vehicle To Grid.

Introduction

1.1. Research Motivation

With rapidly increasing global average temperatures, tackling climate change is one of the main challenges of this era. In order to stay beneath the 1.5°C limit agreed upon in the Paris Agreement, humanity must drastically reduce its greenhouse gas emissions in the coming years. One of the ways in which this is being achieved is by electrification of transportation, a sector currently responsible for 27% of greenhouse gas emissions in Europe [1]. Currently, most vehicles are powered by an Internal Combustion Engine (ICE) which runs on fossil fuels. These ICE vehicles are increasingly being replaced by Electric Vehicles (EVs) which can be powered by electricity produced by renewable sources, therefore significantly reducing the amount of emitted greenhouse gasses.

The amount of EV sales is expected to increase exponentially in the coming decades. To date, 17 countries (including The Netherlands) have announced a complete phase-out of new ICE vehicles by 2050 or sooner [2]. While these countries are particularly ambitious, almost all countries have goals to significantly increase their EV sales. According to the International Energy Agency's Sustainable Development Scenario, over 30% of all vehicles sold worldwide should be EVs by 2030, as shown in 1.1.

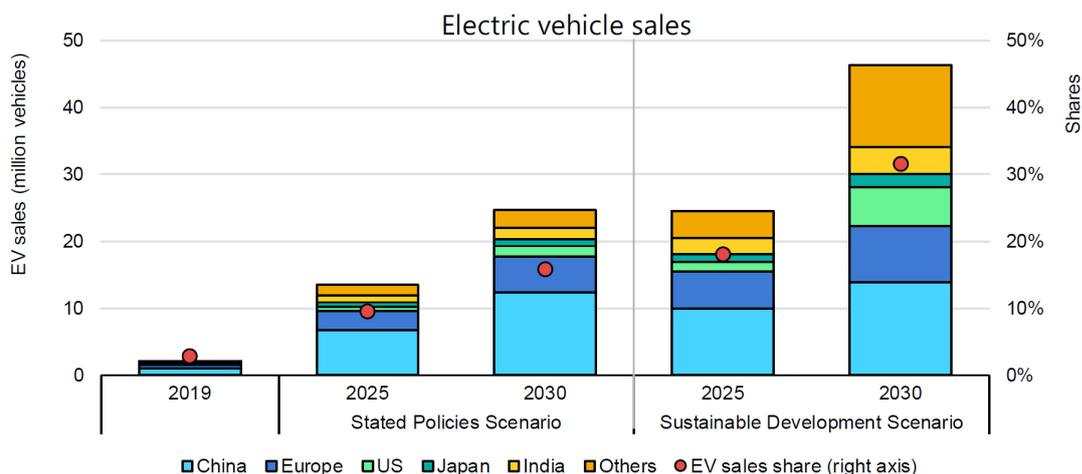


Figure 1.1: Predicted future global EV sales [2]

The increase in EV sales will have a significant impact on the electrical distribution grid. If EV charging power is not controlled, then peak demand is expected to increase at times when many people charge their EVs, for example at the end of the working day when people return home from work and plug in their EV. In many cases, the distribution grid is not designed for this increase in peak demand. While

it is possible to increase the grid's capacity, this is very costly as it would require replacing expensive components such as transformers and underground cables. A more cost-effective solution would be to automatically delay the charging to a later point in time whenever the distribution grid is near its limits. This is one application of controlled ('smart') charging.

Along with a growth in the share of electric vehicles, it is also expected that the share of electricity produced by renewable sources such as solar and wind will increase in the near future. This also presents some challenges to the electrical infrastructure as energy production from solar and wind is variable; the energy output varies depending on the weather and cannot be controlled. In the past, most of the electricity was generated by fossil-fueled thermal plants of which the energy output could be, to a large extent, controlled to match demand. However, this is not the case with most renewable sources. In some places, for example California, this is already leading to a phenomenon sometimes referred to as the 'duck curve', as shown in figure 1.2. The red and green lines represent wind and solar generation respectively. The dark blue and cyan lines represent the load, excluding and including variable generation respectively. During the daytime, the solar systems generate a large amount of the required energy which significantly reduces the net load that still needs to be supplied by thermal plants. However, as the sun sets, this solar energy generation decreases. Just as the load reaches its peak in the evening, the solar energy produced drops to almost zero. This is an issue because it means that the thermal plants need to ramp up/down very quickly, which is costly and not always possible. Besides this, it also means that thermal plants with high power production capacities still need to be kept operational but will rarely be running at full capacity, which is economically undesirable for the plant operators.

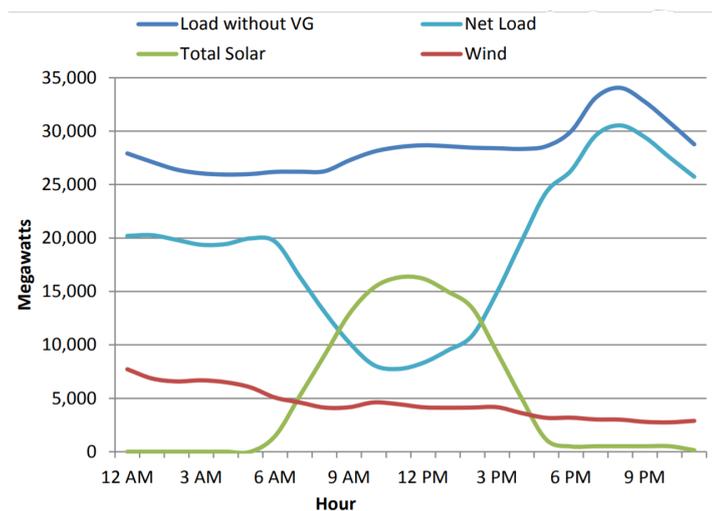


Figure 1.2: Load, solar, and wind profiles for California on March 29 in a scenario with 11% annual wind and 11% annual solar [3]

One solution for this might be to store energy at times of high generation from renewables and release it during times of low generation, but grid-scale energy storage is currently very costly and not yet widely implemented, so grid operators must find another way to match the grid's supply and demand. One of the ways of doing this is through demand response; the loads in a smart grid can adjust their behaviour depending on generation. This is another possible benefit of smart charging; at times when generation from solar/wind is high, EVs can be charged. Likewise, they can stop charging when generation is low.

The concept of smart charging is certainly not new as variations of it have been around for over a decade in academic literature. It has a high Technology Readiness Level and some small-scale commercial implementations can already be found. However, researching the effects of a distributed smart charging algorithm (i.e. with EVs in different grid locations) on the distribution grid is challenging. It can be done relatively easily with pure software-based simulations but this will always require real systems to be modeled and simplifications to be made. The alternative option is to use real EVs at different physical locations with a controllable distribution grid in between. Needless to say, making a distribu-

tion grid especially for an experiment is not very cost-effective. A compromise between these two is the use of a Hardware-in-the-Loop (HIL) setup.

A HIL simulation is a type of simulation which combines software models or control algorithms with real physical hardware. This type of simulation is run on a Digital Real-Time Simulator (DRTS) in which the model is executed in real-time, i.e. at the same rate as the actual wall clock time. This means that simulating a model for 1 minute in a DRTS would take exactly 1 minute in reality. This real-time behaviour is an important property in order for the software model to be able to interact with the physical hardware. The benefit of a HIL simulation is that it combines the flexibility of software-simulation with the actual response of real hardware components. This allows the researcher to interface real hardware components with complex systems which would otherwise be unpractical to use in an experiment. Specifically for this thesis project, the use of HIL means that one or more EVs could be implemented as hardware in the same lab, but through a (software-based) real-time grid simulation they can be made to behave as if they were in different cities. Without a HIL simulation, a real controllable distribution grid would be needed which would be very impractical. For that reason, this thesis describes the development of a HIL test-bench which can hopefully serve as a building block for further research into the topic.

1.2. Research Objectives

The main objective of this thesis is to develop a testbed to experimentally validate the functionality of EV smart charging algorithms in comparison to uncontrolled charging for different scenarios. The focus is to study a single EV charger connected to a given node in a larger network of EV chargers and therefore, a Hardware-in-the-Loop (HIL) based set-up is chosen. The main research question is *To what extent do controlled and uncontrolled EV charging differ in their impact on a distribution grid?*. This is divided into the following objectives:

1. Mimic EV charging behavior at the chosen node in the network by achieving high power capability in the testbed.
2. Validate the proper functionality of the developed EV smart charging algorithm in the real-time environment for different scenarios.
3. Compare the effect of uncontrolled and controlled charging on the distribution grid in terms of voltage deviation and grid connection overload using the HIL testbed.

This thesis will focus on AC charging as DC charging is mostly used for fast charging. In general, the whole purpose of fast charging is to charge the EV as quickly as possible which makes it an unideal candidate for smart charging, because this would interfere with the ability to limit/delay the EV charging. The parameters used for the controlled and uncontrolled charging experiments will be based on the scenario of EVs charging at a small business during office hours. Real historical data will be used for the energy prices, PV production, and local load power.

1.3. Thesis Outline

This report starts with a literature review that provides background information on smart charging, real-time simulations and similar experimental setups found in literature in chapter 2. Chapter 3 then introduces the layout of the proposed experimental setup and its various hardware components. Chapter 4 discusses the smart charging algorithm that was used in this project, and adaptations that were made to run it in real-time. The integration of the different hardware and the smart charging algorithm into one system is described in chapter 5. Chapter 6 shows the first experimental results, with the scenario of uncontrolled charging, meaning that the smart charging algorithm is not active and all EVs charge at their rated power. The parameters used for this first scenario are then reused for 7 additional scenarios in chapter 7. All of these scenarios include smart charging but each scenario has one parameter changed. The results of each scenario is analyzed and compared at the end of this chapter. Based on the observations made, chapter 8 presents recommendations for improving the smart charging algorithm, as well as a conclusion of the work and recommendations for future work.

2

Literature Review

2.1. Smart Charging

According to the International Renewable Energy Agency (IRENA), smart charging is defined as “*adapting the charging cycle of EVs to both the conditions of the power system and the needs of vehicle users.*” [4]. While the exact definition of the term varies depending on the source, the main idea of smart charging is to intelligently manage the charging power of EVs based on external factors such as energy prices, grid congestion, local generation, etc. This can potentially lead to a number of benefits for the involved parties.

2.1.1. Potential Benefits of Smart Charging

Reduce the need for grid infrastructure investments

Without smart charging, EVs will charge at full power regardless of the grid conditions. This can lead to high peaks in power consumption and this would require large investments in reinforcing grid infrastructure to be able to cope with these peaks without overloading the grid. Through smart charging, EV charging demand can be shifted and peak loads can be significantly reduced, at a cost of 10% of the total cost that would otherwise be needed for reinforcing the grid without smart charging [4]. A study by the Rocky Mountains Institute found that with an EV penetration of 23% (expected to be achieved in 2031), uncontrolled EV charging would increase peak loading by 11.1% in California, while smart charging of the same EVs would only lead to an increase of 1.3% [5].

Reduction of energy costs

By shifting EV charging to off-peak times, the system can take advantage of time-of-use tariffs to charge the EVs at times when electricity prices are lower, thus reducing the total energy costs. Depending on the scenario, [6] found that cost savings of 10%-50% can be achieved.

Increased self-consumption

An increasing number of users are becoming ‘prosumers’, i.e. both consuming electricity from the grid and producing their own energy by e.g. an onsite PV system. The power provided by this PV system is variable and cannot be controlled. However, with smart charging, the EV charging profile can be shifted to attempt to match this PV generation and therefore increase self-consumption. Not only does this reduce stress on the electricity grid, it can also be desired by EV owners with environmental concerns who would like to maximize the amount of renewable energy used for charging their EV. Van der Meer et al considers the case of four EVs connected to one charging point at a workplace and finds an increase of 8.8% in self-consumption compared to uncontrolled charging [7].

Provision of ancillary services

The increasing amount of EVs form an opportunity for the power sector. These EVs can act as flexible loads and decentralized energy storage and can provide ancillary services to the grid, such as load regulation and real-time balancing of the grid [4]. Especially with increasing use of intermittent generation such as wind and solar, EVs with smart charging have the potential to improve grid stability.

EV owners would be remunerated for providing these grid services which could lead to negative total charging costs in some cases [8].

2.1.2. Main Algorithms in Literature

Smart charging has been a popular topic for several years, and some commercial implementations of smart charging are currently already available. Many earlier works that formulate an optimization problem to intelligently charge EVs based on external factors can be found.

Firstly, some existing smart charging algorithms take a centralized approach, where one central entity collects the charge parameters and requirements from all EVs and then runs an optimization to determine the optimal charging profile for all EVs. This generally produces the most optimal solution for the entire system as information about the global system is more readily available. However, privacy, cybersecurity and scalability concerns can be raised [9]. An alternative to this is a decentralized approach where each node in the system acts independently to calculate a solution which is optimal for its local conditions. This offers better scalability and is generally easier to implement because less communication is needed between the involved parties. However, this means that each node essentially operates blind to the other nodes so that makes coordination of the nodes challenging.

Within the smart charging algorithms available in literature, there are differences in the objective functions depending on the goals of the specific implementation. Common goals include [9]:

- Load regulation: reducing variation in the aggregate load (EV + non-EV loads) in order to reduce the need for up- and downramping of power plants and to avoid grid overloads.
- Provision of ancillary services: as discussed in the previous section, EVs can act as flexible loads and decentralized energy storage and can therefore provide ancillary grid services and contribute to grid stability.
- Reduce grid congestion: by time-shifting the charging of EVs, grid congestion resulting from peak loads can be reduced.
- Maximize self-consumption: by shifting EV charging to periods of high generation by onsite renewable sources such as PV or wind, the local self-consumption can be increased.
- Minimize cost of power system operations: from the grid operator's perspective, it can be beneficial to use smart charging to reduce total system costs, including fuel costs and startup/shutdown cost of generators.
- Minimize cost of EV charging: from the EV owners' perspective, the cost to charge their vehicle can be reduced with smart charging by time-shifting charging to take advantages of time-of-use tariffs, or by reducing electricity imports from the grid through an increase in self-consumption. The provision of ancillary services can also provide financial benefit to the EV owner.

A variety of different optimization techniques can be used for calculating the optimal EV strategy, including Game Theory [10, 11], (Mixed-Integer) Linear Programming [8, 12, 13], Fuzzy Logic [14] and others.

2.1.3. Relevant Protocols

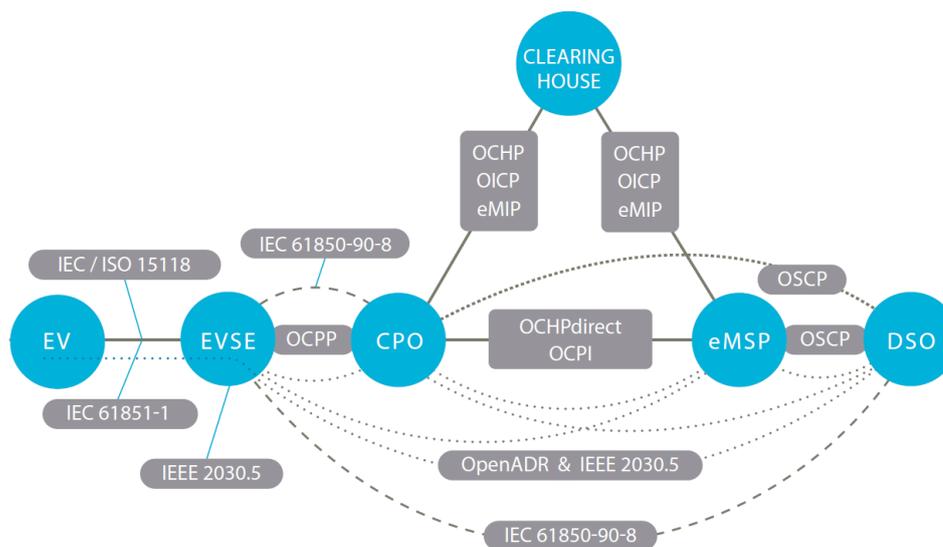


Figure 2.1: Overview of possible protocols involved in smart charging [15]

Many standardized protocols are applicable to facilitate communication between the different parties involved in smart charging. Figure 2.1 shows an overview of some of these protocols and their relation to the different roles within the EV market. For the purposes of this project, the focus will be on three of these protocols, namely Open Charge Point Protocol (OCPP) for communication between the Charge Point Operator (CPO) and Electric Vehicle Supply Equipment (EVSE), and IEC/ISO 15118 and IEC 61851-1 for communication between EVSE and EV.

OCPP

The Open Charge Point Protocol (OCPP) has been designed and developed by the Open Charge Alliance, a consortium of over 150 EV infrastructure leaders. Its purpose is to facilitate communication between Electric Vehicle Supply Equipment (EVSE) and a central entity which manages and operates the charging points, i.e. a Charge Point Operator (CPO). It is currently the de facto open standard for EVSE to CPO communications in Europe and therefore has high market adoption and maturity [15].

In April 2018, the newest version (2.0.1) was released. However, it is worth noting that not all hardware supports this new version yet. Its improvements over the previous version (1.6) include the addition of extra smart charging functionality and support for ISO 15118 [16]. The most important addition for smart charging purposes is that the amount of energy requested by the EV (user) can now be communicated over OCPP 2.0.1 [17]. The main use cases supported by OCPP are [15]:

- Authorization of charging session
- Collecting transaction information for billing purposes
- Limiting charging current
- Controlling the charge point, including locking/unlocking the connector, setting the charge point to available/unavailable and firmware updates
- Sending a reservation message
- Smart charging

Specifically for smart charging, the supported features of version 2.0.1 include [18]:

- Sending a fixed current limit, or a charging profile, for each individual EV. The charging profile consist of a list of current limits as a function of time.

- Receiving EV-initiated charging information (only when using ISO 15118).
- Reading data from the EVSE's energy meter.
- Receiving the amount of energy requested, as well as the maximum rated current for each EV.
- Reading the status of each EVSE (available, EV connected, error, etc.)

IEC 61851-1

IEC 61851-1 is a standard by the International Electrotechnical Commission which describes the communication between EVSE and EV. It has a high level of maturity and market adoption in Europe [15].

The communication is relatively simple and it is based on two signals; the Proximity Pilot (PP) and Control Pilot (CP). The PP indicates that a vehicle is connected and communicates the maximum rated current of the cable. The CP is a Pulse-Width Modulation (PWM) signal which communicates the charging state and the upper limit of current that can be drawn by the EV. When using communication based on IEC 61851-1 it is only possible for the EVSE to set/change the maximum charging current. It is not possible to receive any data such as the current State of Charge (SOC) or battery size directly from the EV. It is also not possible to use this communication for Vehicle To Grid (V2G) applications.

ISO 15118

ISO 15118 is a protocol that specifies a more advanced communication between EVSE and EV. The protocol is designed and developed by a Joint Working Group consisting of the IEC and the International Organization for Standardization (ISO). The maturity and market adoption of the 15118 protocol is relatively low compared to IEC 61851-1, but it is expected to increase in the coming years [15].

The benefit of ISO 15118 is that more information can be dynamically exchanged between EV and EVSE which is very useful for enabling smart charging. For example, unlike IEC 61851-1 communication, ISO 15118 allows the EVSE to read parameters of the EV. This includes the current SOC, requested energy, maximum/minimum voltage and current limits, and requested time of departure [19]. ISO 15118 also enables a bidirectional flow of power, i.e. the EV can also provide power back to the grid (V2G). An additional convenient feature is that the protocol allows for 'Plug & Charge', meaning that the EV can automatically identify itself and so there is no longer a need for the EV owner to scan his/her credit card or RFID card [20].

2.2. Real-Time Simulations

Software simulation tools are widespread within science and engineering. However, many of these simulation tools generally do not run in real-time; the speed of their computations are dependent on the complexity of the model and the processing power of the hardware it is running on. When an event occurs, the computation time to calculate the new system output can be significantly longer or shorter than what the reaction speed would be of the real system. Because of this, it is not possible to directly interface such a simulator with external hardware. For this to be possible, the simulation software would need to exchange data with the external hardware and so it has to be running synchronized and with the same time step as the external hardware, i.e. in real-time.

In a Digital Real-Time Simulator (DRTS), a model of a physical system is executed at the same rate as the actual wall clock time, meaning that simulating a model using a time period of 1 minute would take exactly 1 minute. This is important to be able to interface with external hardware because this ensures that the simulation is synchronized with its physical counterpart. In the case of a real-time simulation, the model equations need to be solved within fixed time steps. The model needs to be designed such that its execution time, T_e , does not exceed this fixed time-step. This way, overruns can be avoided and real-time behaviour can be maintained, as shown in figure 2.2. If T_e is larger than the fixed time step, an overrun occurs and the simulation is considered nonreal-time or 'offline'. In this case, the time-step can be increased or the model can be simplified to make it run in real-time [21]. If these conditions are satisfied, a real-time simulation can be run on a Personal Computer (PC). However, generally purpose-built real-time simulators are used. A good DRTS should have sufficient computing power to run complex models with small computation times. This is often achieved by using multiple

processors in parallel. Additionally, DRTS often have various input/output (I/O) terminals to interface with external hardware, and a communication network to allow very complex models to be split over several DRTS machines [21].

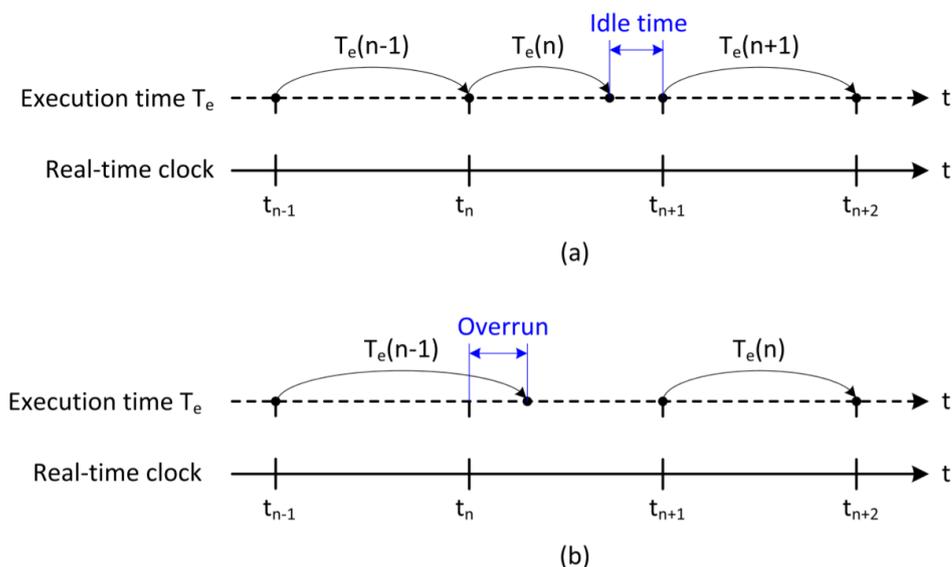


Figure 2.2: Illustration of real-time (a) and non-real-time (b) simulation [21]. If the model's execution time T_e is less than the simulation's time-step, it can run in real-time. Otherwise, there is an overrun and real-time operation is not achieved.

2.2.1. Simulation vs Emulation

Systems such as an EV can be included in a test setup in the form of physical hardware, an emulation, or a simulation. In the case of physical hardware, this would mean that the system in its entirety would be connected to the setup. In the case of experiments with EVs, this may well involve needing to bring a car into a laboratory. Besides the practical issues due to a car's size and weight, there are also other issues such as safety concerns regarding large batteries. Additionally, key parameters such as the charging profile or the battery's capacity cannot be changed without swapping out the EV for a different model.

An alternative to this is to either use a simulation or an emulation. Both are designed to mimic the behaviour of a real device. In the context of EV charging, the distinction is that a simulation is purely software-based, whereas an emulation is a physical device that has (some of) the same input and output connections and can mimic the behavior of the EV it is based upon [22]. Unlike a simulation, an emulator can process power and always runs in real-time. In both cases, the parameters are highly customizable and not limited to one specific model of EV.

2.2.2. Types of Real-Time Simulations

In the context of power systems, Digital Real-Time Simulations can be divided into a number of different categories [21, 23]:

- Software-in-the-Loop (SIL): both the power system and the controller are simulated on the same DRTS.
- Controller Hardware-in-the-Loop (CHIL): the power system is simulated in the DRTS while the control algorithms are executed on an external controller platform.
- Power Hardware-in-the-Loop (PHIL): the controller is simulated in the DRTS and interfaced with physical hardware through a power amplifier.
- Co-simulation: the power system and controller are simulated on different simulation environments. Synchronization of the two systems needs to be ensured.

Figure 2.3 shows the key differences between Controller Hardware-in-the-Loop (CHIL) and Power Hardware-in-the-Loop (PHIL). In the case of CHIL, the Hardware-in-the-Loop (HIL) is a controller but for PHIL this is power hardware. In both cases, signals are passed through analog-to-digital (A/D) and digital-to-analog (D/A) converters to and from the Hardware Under Test (HUT).

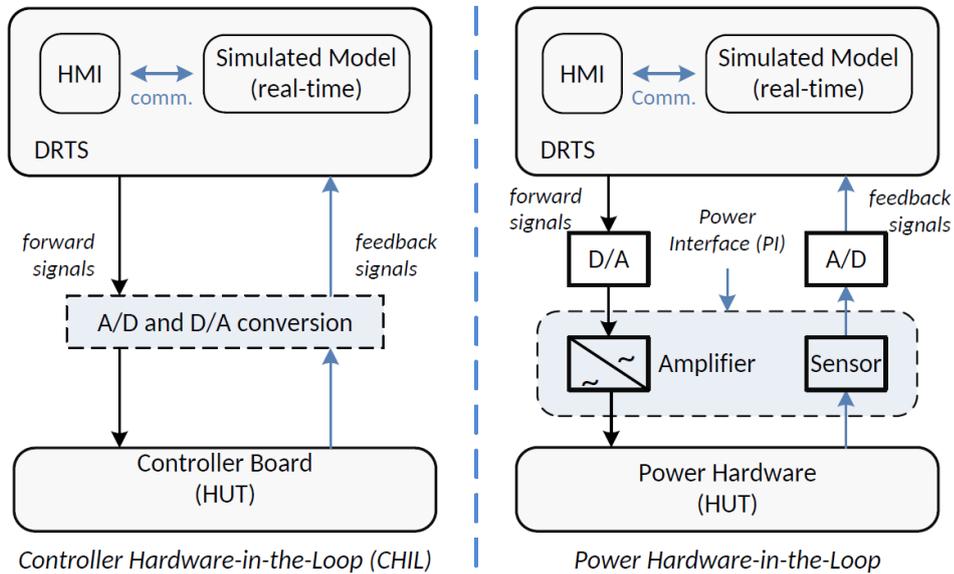


Figure 2.3: Structure of HIL-based simulation concepts: CHIL (left) vs. PHIL (right) [21, 23]. In the case of CHIL, the DRTS interfaces with an external controller. With PHIL, the DRTS interfaces with external power hardware through an amplifier.

Note that in the case of PHIL, a power amplifier is needed between the DRTS and HUT. Not only does this add size and cost to the system, care must also be taken to minimize the impact of the dynamics of this amplifier on the system. In extreme cases, the introduction of a power amplifier may cause the PHIL system to become unstable, even if the real system would be stable [24]. Additionally, the power amplification is not ideal so this introduces errors and delays into the system, which the interface algorithm between the DRTS and HUT must be able to handle. In literature, various solutions for minimizing the impact of these nonidealities can be found. These range from simple low-pass filtering of the feedback signals [25] to more complex solutions such as the 'damping impedance method' involving the use of a damping impedance equal to the impedance of the HUT [26, 27]. The conclusion found in literature is that there is no single optimal solution for all PHIL systems, so the most effective algorithm must be evaluated case by case by using Nyquist or Popov's circle criterion. For safety reasons, it is also common practice to run equipment at a power level well below its rating when testing the system's stability [27].

In this project, a PHIL test-bed is used. The distribution grid is modeled within the DRTS and the power hardware (consisting of an AC charger and EV emulator) is connected to the DRTS via a power amplifier.

2.3. Why PHIL?

This project uses a Power Hardware-in-the-Loop (PHIL) method for simulating the controlled and uncontrolled charging in a distribution grid. Because the use of PHIL is a key aspect in project, the justification for choosing PHIL and the benefits of PHIL in the context of this project are discussed below.

In the case of power systems, it is often much more practical to do pure software simulations than to experiment with a real system. Performing reproducible experiments on a large-scale physical distribution grid while controlling all relevant variables is very challenging. In the case of a software simulation, the user can have full control over all relevant parameters without needing access to any hardware other than a computer. This provides relatively high flexibility. For instance, different grid topologies,

PV penetration and load conditions can easily be programmed and their effect on the test object analyzed. Test scenarios such as catastrophic faults in the grid can be run without needing to worry about damage to physical systems. Nevertheless, experimental validation is still vital in any scientific discipline. Software simulations always involve modeling the behaviour of a physical system as a series of equations which inevitably requires some simplifications. These simplifications may well have an effect on the outcome of the system validation.

Hardware-in-the-loop is a hybrid solution; it combines the power and flexibility of real-time software simulation with the actual response of real hardware components. PHIL allows the validation of equipment within a larger virtual power system, under a wide range of conditions and in a repeatable, safe, and economical manner [24]. This method of validation is closer to reality than a pure software simulation, but less complicated than using a real distribution grid.

For the testing and validation of smart charging protocols, a PHIL setup offers the following benefits over a pure software simulation:

- Increasing the accuracy of the simulation with the use of real components, leading to less simplifications.
- Evaluating whether real components which have certain communication and power behavior are really able to interact with large systems. This behavior includes nonidealities such as but not limited to communication latency, amount of information flow, refresh rate and response speed of the physical system.
- Theoretical simulations make certain assumptions, including:
 - Accurate information about every part of the system is available
 - Information is available instantaneously to every relevant controller in the simulations
 - Every controllable element knows what every other controller in the system would do in the future

Real systems inevitably violate some of these assumptions and simplifications. The results of a PHIL simulation can be compared to a pure software simulation in order to experimentally demonstrate the influence these aspects have on the performance of the solution.

- Much of the hardware and infrastructure currently in use for EV charging is based upon commercial products. For a SIL setup, all devices must have validated models in order to achieve accurate results. The design and inner workings of these products is often not publicly available so it cannot be modeled in its entirety, and literature shows that in practice these products do not always show expected behaviour and conform to industry standards [28, 29]. Using real hardware in a PHIL setup could show the effect of this nonconformity on the performance of the solution.

Specifically for testing a distributed smart charging algorithm, a PHIL setup has the benefit that the effect of having chargers in different locations can be more easily investigated. For example, there could be two real EVSEs physically located in the same laboratory but through the use of a distribution grid simulation, their bus voltages can affect each other as if they were on opposite sides of the city.

2.4. Similar Experiments

Literature shows that extensive research has been done into the possible benefits of controlled charging on the distribution network [30–33]. However, much of this research is based on software simulation. Due to the inherent challenges, experimental validation is rarely touched upon. Nevertheless, some previous work involving experimental validation, using either PHIL setups or real distribution grids, can be found.

2.4.1. AIT's FlexEVELab

The most complete PHIL setup is the FlexEVELab at the Austrian Institute of Technology (AIT). It is a flexible topology, allowing the EV, EVSE, and grid to be set up as either a simulation, emulation, or real hardware component [22]. Both AC and DC charging can be used and the IEC 61851 and ISO 15118 are both available. To connect all devices together and allow them to be controlled from a single node, Lablink is used as shown in figure 2.4. Lablink is a co-simulation 'middleware' tool developed by AIT specifically for the purpose of evaluating the effect of large scale EV penetration. This tool enables the exchanging of data between the different actors in the system and ensures synchronization [34].

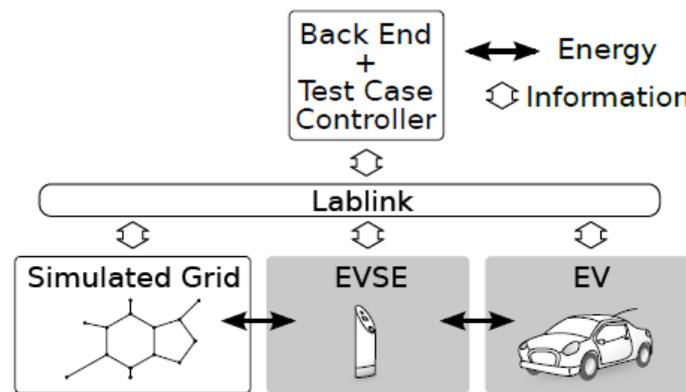


Figure 2.4: A Top-Level Overview of AIT's FlexEVELab Setup [22]. The different devices exchange information via a 'middleware' tool named Lablink.

In order to accurately simulate and emulate the charging process of an EV within FlexEVELab, comprehensive measurements of several real EVs were used to develop a model of the charging process. The measurements showed a number of special characteristics, including [35]:

- All measured cars showed a reaction delay up to 1.5 seconds in responding to a current commands
- The measured EVs did not start charging immediately but had an initialization phase of approximately 20 seconds
- EVs do not exactly apply the maximum charging current sent by the EVSE; it is always a certain percentage (dependent on the EV) lower
- Not all EVs show a constant-voltage phase during charging

This model is used as a control software for hardware components such as an RLC load or an electronic load, which can then mimic the charging behaviour of a real EV without needing to have a physical car in the laboratory. The test setup presented by AIT can be used for a variety of prospective testing applications, ranging from load balancing possibilities to future V2G applications [22]. So far, the setup has been used by researchers of the University of Passau to validate a smart charging algorithm for power quality control using a simulation of a Bavarian distribution grid [36].

2.4.2. PowerLabDK SYSLAB

For the experimental validation of a smart charging algorithm, the setup shown in figure 2.5 has been used. The algorithm is based on a droop controller and aims to use EVs to improve voltage quality in distribution grids, without relying on any V2G capability. The setup used does not rely on HIL methods, but instead relies completely on real hardware connected to the local distribution grid. The connected hardware includes an 11 kW wind turbine, 3 Nissan Leaf EVs, and a resistive load of 15 kW per phase. The wind turbine's purpose is mainly to provide stochastic active and reactive power variation to the system in order to make the test grid more realistic. Due to the test grid's relatively low X/R ratio, active power modulation is the most effective way to control the voltage, so the controllable loads are used for this purpose [29].

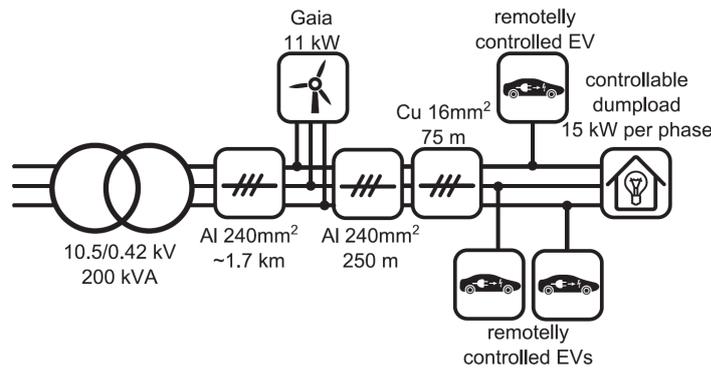


Figure 2.5: A Top-Level Overview of the PowerLabDK SYSLAB Setup [29]. A real distribution grid is used with 3 EVs, a small wind turbine for stochastic variation, and a variable resistive load for voltage control.

The paper shows that the results of this experimental validation show some unexpected issues such as unwanted controller oscillations, differences in reaction times to control signals between the different EVs, and one EV violates the set charging current limit by 1A. Nevertheless, the results also show that the smart charging controller works as it is able to improve the voltage quality when compared to uncontrolled charging.

The droop-based controller described by Martinenas et al is relatively simple, and the fact that they are using a real grid restricts their workable voltage range and the amount of test scenarios. Therefore, the paper suggests further research should focus on more sophisticated control algorithms and a more expansive list of test scenarios [29].

2.4.3. University of Suceava’s EV Simulator

At the University of Suceava, an EV simulator of 24 kW was developed with the purpose of verifying the correct working of a simple EVSE (figure 2.6a) that was developed in-house. The system is designed for the IEC 61851 standard and type 3 charging [37].

The EV simulator (shown in figure 2.6b) consists of three load resistors and several smaller resistors. There are several resistors between L3 and ground in order to measure the EVSE’s tripping time at different values of residual current. The paper shows that the developed EV simulator allows an operator to check if all charging stages are being correctly executed by the EVSE, as well as verify the correct functioning of the EVSE’s residual current monitoring device.

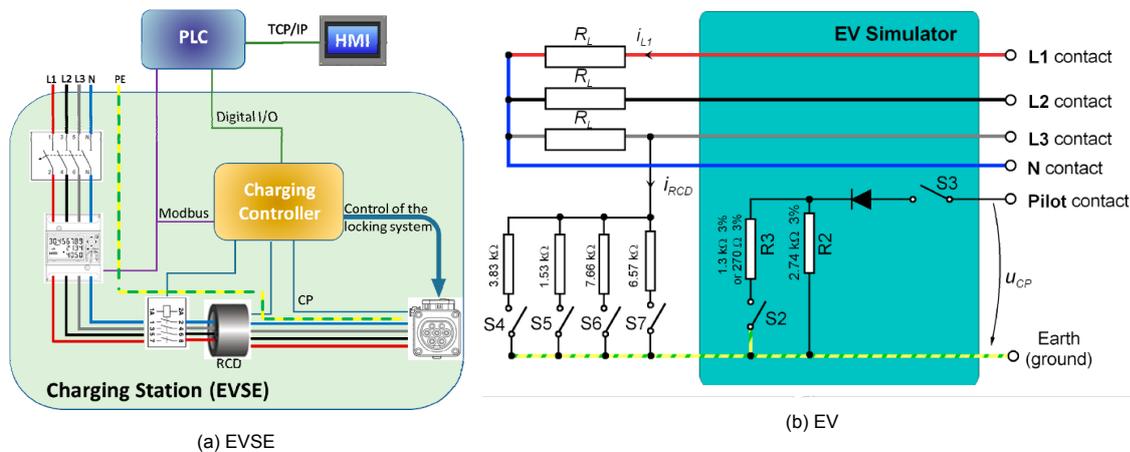


Figure 2.6: Overview of University of Suceava’s EV Simulator and EVSE setup [37]. The EV simulator is designed to verify the correct execution of all charging stages by an EVSE, and its residual current monitoring device.

2.4.4. Overview & Research Gap

Table 2.1 presents an overview of the previously discussed similar experiments found in literature.

	Smart Charging Algorithm	Power Flow Simulation	Controllable Grid Voltage	Protocols	Maximum Power
FlexEVELab [22, 34–36]	Yes	Yes	Yes	ISO 15118 IEC 61851	22 kW (AC) 200 kW (DC)
SYSLAB [29]	Yes	No	Partially	IEC 61851	200 kVA (AC)
University of Suceava [37]	No	No	No	IEC 61851	24 kW (AC)

Table 2.1: Comparison of similar experimental setups found in literature

The amount of similar PHIL setups found in literature is relatively limited, despite the popularity of electric mobility and smart charging as research topics. Yet, as discussed in sections 1.1 and 2.3, PHIL simulations offer significant benefits for researching the effect of EV charging on a distribution grid. The FlexEVELab developed at the Austrian Institute of Technology is the only one that could be found which has all the features of the setup proposed in this report. While AIT's FlexEVELab is a very complete setup, it is also relatively costly and complicated, as the communication between different setups relies on a sophisticated custom-made middleware named Lablink [34]. Therefore, this thesis project will focus on a low-cost PHIL system based on commercially available components and communication protocols.

3

Hardware

3.1. Overview

This project is part of a larger project named Orchestrating Smart Charging in mass Deployment (OSCD). Within the OSCD project, the infographic seen in figure 3.1 was previously created. This infographic serves as a starting point for the design of the PHIL setup of this thesis.

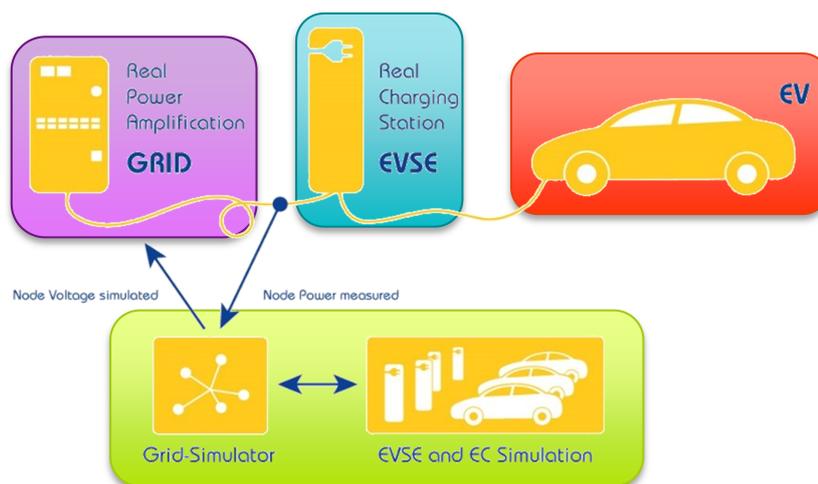


Figure 3.1: Infographic of a PHIL setup proposed within the OSCD project [38]

The experimental setup which was used in this thesis project is shown in figure 3.2, where the colored blocks correspond to those in figure 3.1. Solid lines represent power flows and dotted lines represent information flows. The green block contains the Digital Real-Time Simulator which runs a grid simulation and the smart charging algorithm. This is described in more detail in section 3.2. The purple block contains a power amplifier which emulates a three-phase grid with controllable voltage. The voltage produced by the power amplifier is determined by the grid simulation. The grid emulation is discussed in section 3.3. The power amplifier is connected to the Electric Vehicle Supply Equipment (EVSE) shown in the teal-colored block and described in section 3.4. Lastly, there is the EV-emulator in the red block. In the OSCD diagram (figure 3.1), this is depicted as a real EV. However, for the thesis project it was decided to use an EV emulator instead, for practical reasons and also because this allows more flexibility in terms of emulating EVs with different battery sizes and charging characteristics. This is described in more detail in section 3.5. The EV emulator used does not process any power by itself, so an external load is needed. For this purpose, two bidirectional DC power supplies are used back-to-back and act as a controllable AC load.

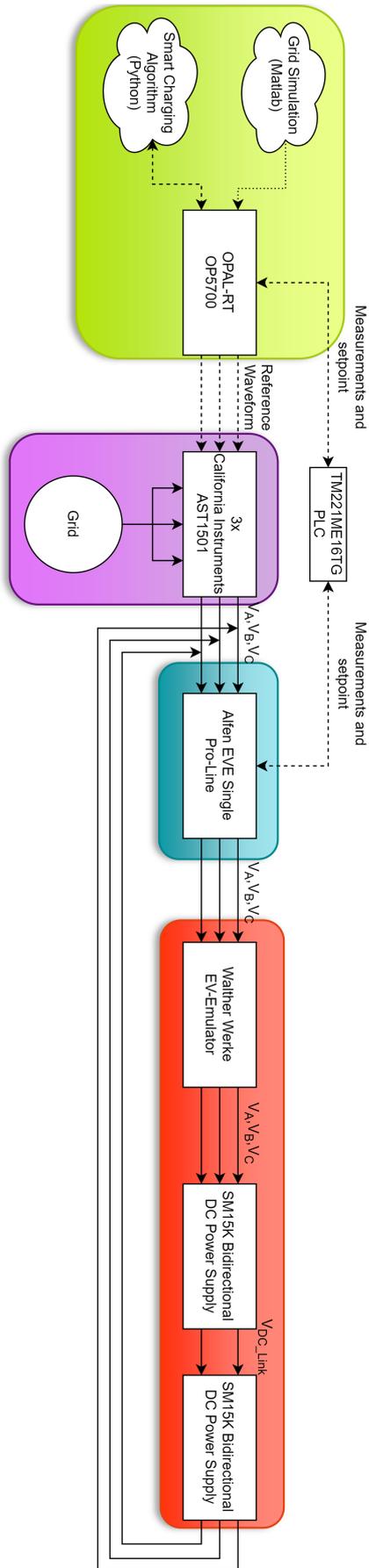


Figure 3.2: Overview of experimental setup

3.2. Real-Time Simulator

The Digital Real-Time Simulator (DRTS) is the heart of the experimental set-up. It communicates with all other components of the setup and controls the actions of the system in real-time. In this section, the DRTS used, an OPAL-RT OP5700, is described in more detail.

3.2.1. OPAL-RT OP5700

As discussed in section 2.2.2, there are several types of real-time simulations. In this project, a Power Hardware-in-the-Loop (PHIL) setup is used. Figure 3.3 shows what such a setup would look like when using an OPAL-RT DRTS. The system consists of two main hardware components; firstly a workstation running OPAL-RT's software package RT-LAB that acts as the 'host' computer. This host computer is used to create and edit a MATLAB Simulink model which is then converted to a real-time application in C code through a process called code generation. The host is also used to control the real-time model while it is running and visualize some of its outputs. Secondly, there is the real-time simulator itself, referred to as the 'target'. This machine runs the real-time model prepared by the host and takes care of the physical I/O signals to external hardware.

The host and target machines communicate through TCP/IP protocols in order to load/run/pause/stop the model and to exchange data related to the model.

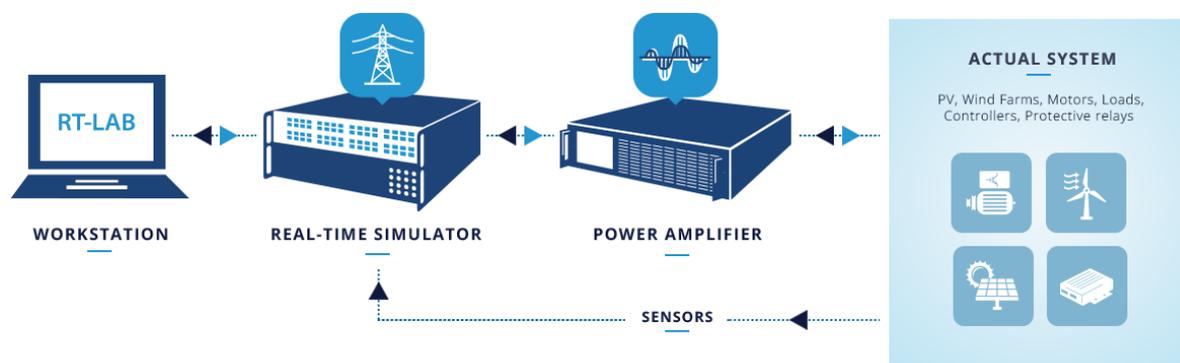


Figure 3.3: A typical PHIL setup [39]

The DRTS used in this project is the OP5700 model from OPAL-RT. At the time of writing, this model is the top-of-the-line flagship simulator offered by OPAL-RT. Some key features include:

- CPU: Intel® Xeon® E5, 8 cores, 3.2 GHz
- FPGA: Xilinx® Virtex®-7 FPGA, 485T
- 32 digital outputs, 5V to 30V
- 32 digital inputs, 4V to 30V
- 16 analog outputs, -16V to 16V, 16 bits
- 16 analog inputs, -20V to 20V (true differential), 16 bits



Figure 3.4: OPAL-RT OP5700 Real-Time Simulator [40]

3.2.2. Modelling for Real-Time Simulation

Once the desired model has been made in MATLAB Simulink, it can be imported into RT-LAB on the host computer where it will be converted to a real-time application in C code, through a process called code generation. Note that not all Simulink blocks are supported for code generation, so this must be kept in mind when making the model.

The Simulink model can be split into several different subsystems. There are three possible types of subsystems:

- **Console:** this allows the user to interact with the model while it is running. It can contain data inputs such as adjustable sliders and outputs connected to a scope that allow the user to visualize certain output signals. The console is an asynchronous subsystem that runs on the host computer instead of the target so no critical logic should be included in this subsystem; it acts purely as a user interface.
- **Master:** this subsystem runs in real-time on the target. It contains all the critical logic and calculations, as well as the I/Os that communicate with external hardware. There is always one master subsystem in the model.
- **Slave:** this is an optional type of subsystem that can be added by the user if the model is complex and needs to be distributed. The benefit of using slave subsystems is that each subsystem can run on its own CPU core or they can even be distributed among several targets. If desired, each subsystem can run with a different timestep and the calculations for each subsystem can be run concurrently.

3.3. Grid Emulation

One of the research objectives mentioned in section 1.2 is to obtain an amplified 3-phase voltage corresponding to a given distribution grid node. This voltage should be determined through real-time power flow simulations. This section describes how this is achieved, both in terms of hardware and software.

3.3.1. AST1501 AC Source

To emulate a three-phase distribution grid, three California Instruments AST1501 power sources are used. Each source can output DC or single-phase AC with a power of 1500 W or 1500 VA. The output voltage is controllable between 0-400 V(RMS).



Figure 3.5: California Instruments AST1501 [41]

The magnitude of the grid voltage is calculated by the DRTS so the DRTS needs to be able to communicate this voltage to the three AST1501 sources. The easiest way to do this would be to use the functionality named *External Analog Programming of Output Voltage Amplitude* which allows an analog signal to set the amplitude of the output signal, while the waveform of the output is generated internally [42]. However, to be able to make use of this functionality, the optional clock and lock modules are needed to synchronize the three sources and ensure 120 degrees phase difference between each source. Unfortunately the AST1501 sources available in the laboratory do not have these optional

modules. A workaround is to use the *External Analog Programming of Output Voltage Waveform* functionality in which not just the amplitude but the entire waveform is given by an external analog signal. For this purpose, three of the OP5700's analog outputs are used to generate three sine waves with 120 degrees phase difference. The amplitude of the sine wave can be between 0 - 10 V, where a 10 V(PK) analog input would correspond to the full-scale RMS output voltage of 400 V(RMS). Therefore, to emulate a three-phase grid connection, three sine waves with a nominal amplitude of $10 \cdot 230/400 = 5.75V$ should be generated by the OP5700. This value is multiplied by the per-unit bus voltage determined by the grid simulation, as described in the next section.

Figure 3.6 shows the measured output voltage of the three AST1501 sources when an input of three sine waves with an amplitude of 5.75 V is given. As can be seen, the three phases of 230 V (RMS) are generated correctly.

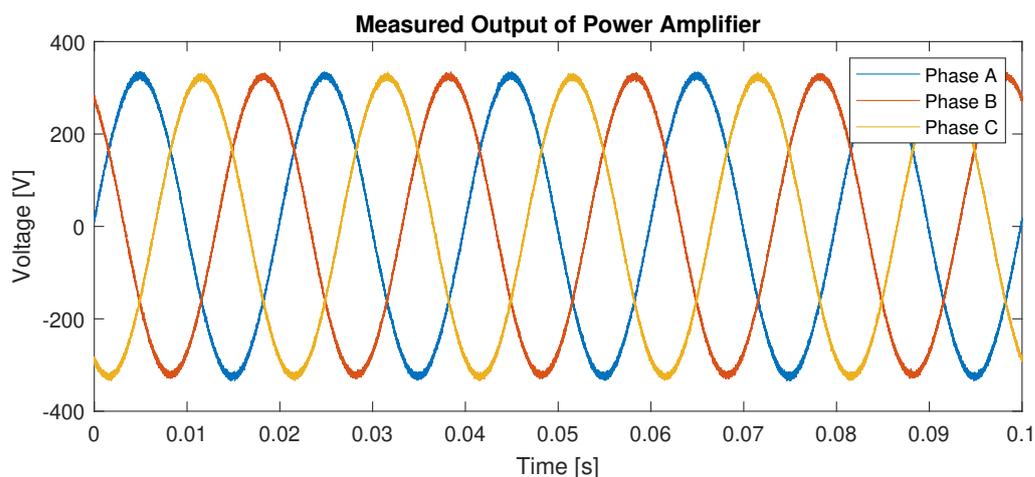


Figure 3.6: Measured Voltage Output of Power Amplifier

3.3.2. Simulink Implementation

In order to determine what the output voltage of the AST1501 sources should be, a power flow simulation should be implemented in the DRTS. This requires the use of a power flow method which can be implemented in Simulink. Simulink allows the use of MATLAB code but not all functions are supported. This is because, as discussed in section 3.2.1, the Simulink model (as well as any MATLAB code used) is converted to a real-time application in C code through a process called code generation. Not all MATLAB functions are supported for code generation so care must be taken to avoid code that uses these functions when selecting a power flow implementation.

The MATLAB code used for implementing a power flow simulation was adapted from code found in literature [43]. Some of the MATLAB functions used were not supported for code generation and were rewritten. This code is based on the Newton-Raphson method. Newton-Raphson was chosen as it is faster than Gauss-Seidel and more accurate than the Fast-Decoupled method [44]. If, in the future, a much more complex grid is used such that convergence cannot be achieved within an acceptable timestep, then the algorithm could be switched to the Fast-Decoupled method. In radial networks with a high R/X ratio, the Forward/Backwards Sweep method may be more appropriate [45] but for the given grid, Newton-Raphson is chosen as its implementation in MATLAB is readily available in literature and the results converge well within the set timestep.

The grid model used is based on data of a real Dutch distribution grid provided by a Distribution System Operator (DSO). As can be seen in figure 3.7, it has a total of 19 nodes, with loads connected at nodes 5, 15 and 19. Between each node is a branch of known resistance R and reactance X, but these have been removed from figure 3.7 for confidentiality purposes. With this data and the load powers for each point in time, the power flow simulation can be run and the voltage of each node can be calculated.

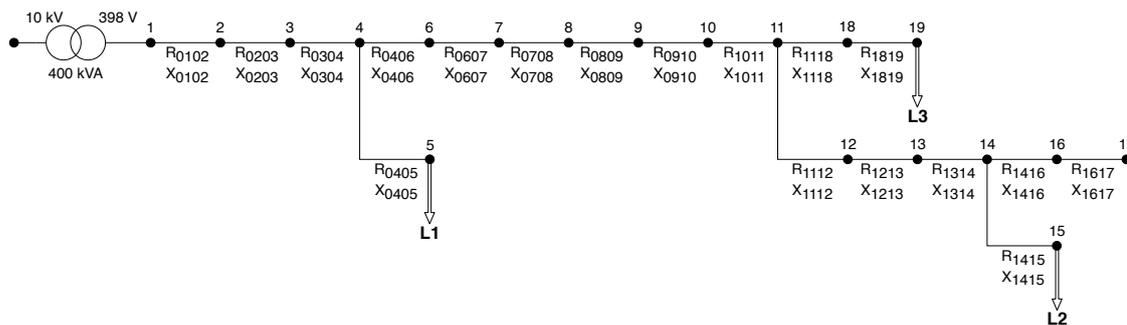


Figure 3.7: Single-line diagram of a Dutch rural grid used for grid simulation.

3.3.3. Validation

In order to validate the power flow method that was implemented in the DRTS, its results were compared to a simulation with exactly the same parameters in DlgSILENT PowerFactory. Figure 3.8 shows that the results of both simulations are exactly the same. This is to be expected as PowerFactory also uses the Newton-Raphson method for its load flow calculations. It can be concluded that the Newton-Raphson-based power flow analysis implemented in the DRTS works as expected.

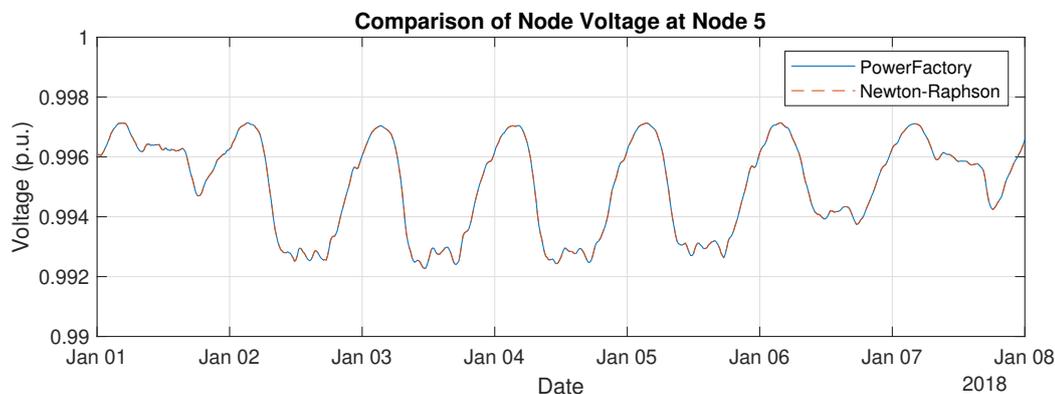


Figure 3.8: Comparison of Power Flow Results Calculated by DlgSILENT PowerFactory and the Adapted Newton-Raphson Simulation

3.4. EVSE

As part of the experimental setup, a commercially available Electric Vehicle Supply Equipment (EVSE) is used. This is integrated into the setup and communication between the EVSE and the DRTS is established.

3.4.1. Alfen EVE Single Pro-Line

The EVSE used is the EVE Single Pro-Line manufactured by Alfen N.V., shown in figure 3.9. It is an AC charger that supports both single- and three-phase charging up to 32A. It has one socket for a type 2 charging cable. The EVSE can be monitored and its parameters changed via a software named *ACE Service Installer* on any computer connected to the same Ethernet network.

One notable feature of the EVE Single Pro-Line is that it has been designed with smart-charging applications in mind and thus can communicate with 3rd party devices. It can communicate either via GPRS (using GSM network) or via Ethernet and uses Open Charge Point Protocol (OCPP) version 1.6 or Modbus TCP/IP. These protocols can be used to read measurements such as voltages and currents, or to send commands to the EVSE such as a maximum allowed charging current. Note that the newest version of OCPP, 2.0.1, is not yet supported by this hardware. As discussed in section 2.1.3, the smart charging functionality of version 1.6 is more limited than version 2.0.1. As is the case with most commercial EVSEs at this time, the ISO 15118 protocol is also not yet supported so the communication

between the EV and EVSE is very limited, as will be discussed in section 3.5.



Figure 3.9: Alfen EVE Single Pro-Line [46]

3.4.2. PLC for Modbus TCP Communication

In this setup, the maximum allowed current will be determined by the smart charging algorithm during the real-time simulation. The DRTS will need to communicate this value to the EVSE. Because these two devices are physically close to each other, connecting them via Ethernet and using the Modbus TCP protocol is most convenient. Modbus TCP is an open communication protocol that is widely used for communicating between (mostly industrial) electronic devices.

The DRTS used in this setup unfortunately does not natively support the Modbus TCP so a third-party device is needed to convert the Modbus TCP communication to an analog or digital signal compatible with the DRTS. It was decided to use a Programmable Logic Controller (PLC) from Schneider Electric, shown in figure 3.10 to do this conversion. A summary of the most relevant features of this PLC can be seen in table 3.1. Note that for PWM signals, a 'fast' input/output is needed.



Figure 3.10: Schneider Electric TM221ME16TG PLC [47]

Table 3.1: Most relevant features of TM221ME16TG PLC

Rated supply voltage	24 V DC
Digital Inputs	8 (4 fast inputs)
Digital Outputs	8 (2 fast outputs)
Digital Output Voltage	24 V DC
Digital Input Voltage	<= 5 V for logical low >= 15 V for logical high
Analog Inputs	2
Analog Outputs	0
Analog I/O Voltage Range	0 - 10 V
Supported Modbus Protocols	Modbus TCP Server/Client Modbus RTU Master/Slave Modbus ASCII Master/Slave

3.5. EV Emulator

3.5.1. Communication between EVSE and EV-Emulator

A type 2 AC charging cable (shown in figure 3.11) uses two pins for communication between the EV and EVSE; a Proximity Pilot (PP) to indicate that a vehicle is connected and to communicate the maximum current of the cable, and a Control Pilot (CP) to communicate the charging state and the maximum current the EV is allowed to draw.



Figure 3.11: Labeled image of a type 2 plug [48]

Proximity Pilot

Detachable charging cables have a resistor between the Proximity Pilot (PP) and Protective Earth (PE) which indicates the maximum current rating of the cable, as per table 3.2. The EVSE is continually measuring the resistance between the PP and PE. If an open circuit is measured, the EVSE can conclude that no charging cable is present and no further action is needed. However, if a resistance of $R \leq 2.7k\Omega$ is measured, this indicates a cable is connected with a current rating indicated in table 3.2.

Table 3.2: Relationship between rated cable current and resistance between PP and PE [49]

Rated current of cable	Nominal Resistance, PP-PE	Interpretation range
13 A	1.5 kΩ	1 kΩ - 2.7 kΩ
20 A	680 Ω	330 Ω - 1 kΩ
32 A	220 Ω	150 Ω - 330 Ω

Control Pilot

When the EVSE detects that a cable is connected, it puts a voltage of 12 V over the CP and PE. Once an EV is connected to the other side of the cable, the EV will activate a resistor between the CP and PE which will lower the voltage. The activated resistance and corresponding voltage is dependent on the charging status, as per table 3.3. The EV will only be able to charge in status C or D, the latter requiring the EV and EVSE to be in a well-ventilated area. In states B, C and D, the CP signal is pulse-width modulated (PWM) signal instead of a constant voltage. The voltages in table 3.3 are then the upper value of the PWM signal and the lower value is -12 V regardless of the state, as can be seen in figure 3.12. This is achieved by placing a diode in series with the resistors.

Table 3.3: Resistances and voltages between CP and PE corresponding to each charging state [49]

State	Description	Resistance, CP-PE	Voltage, CP-PE
A	EV not connected	$\infty \Omega$	+12 V
B	EV connected, not ready	2740 Ω	+9±1 V
C	Ready	882 Ω	+6±1 V
D	Ready, ventilation required	246 Ω	+3±1 V
E	Shut off		0 V
F	Error		-12 V

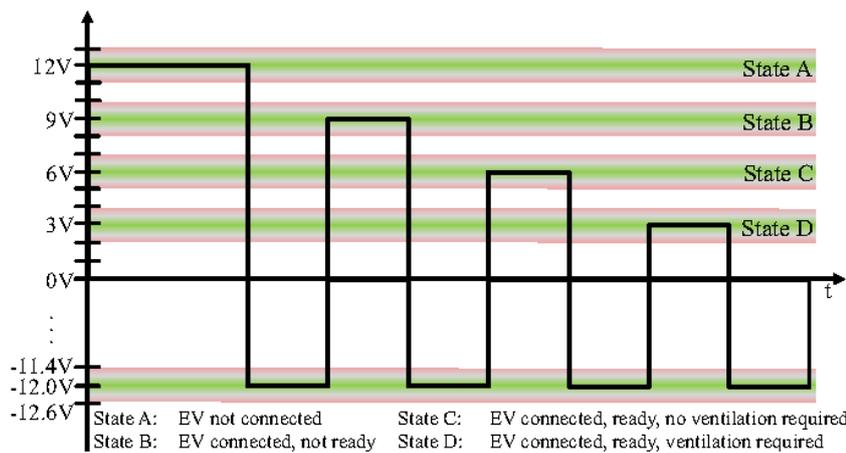


Figure 3.12: Graph of CP signal at different charging states [50]

The usage of a PWM signal instead of a constant voltage means that another variable can be communicated by changing the duty cycle; the maximum available current. This will signal to the EV's on-board charger the maximum AC current it is allowed to draw. The EV is allowed to draw less current than what is indicated by the PWM signal, but not more. Table 3.4 shows the relationship between the PWM's duty cycle and the maximum allowed current. A 10% duty cycle would indicate a maximum current of 6 A, 20% indicates 12 A, etc. Note that the minimum value that can be set by the EVSE is 6 A. If the duty cycle is any lower than 8%, the EV should stop charging.

Table 3.4: Relationship between CP duty cycle and maximum allowed current [49]

Duty cycle interpreted by EV	Maximum current
Duty cycle <8 %	Charging not allowed
$8 \% \leq \text{duty cycle} < 10\%$	6 A
$10 \% \leq \text{duty cycle} \leq 85 \%$	Available current = (% duty cycle) x 0.6 A
$85 \% < \text{duty cycle} \leq 96 \%$	Available current = (% duty cycle - 64) x 2.5 A
$96 \% < \text{duty cycle} \leq 97 \%$	80 A
Duty cycle >97 %	Charging not allowed

3.5.2. Walther-Werke Type 2 EV-Emulator

As part of the experimental setup, a device needs to be connected between the EVSE and AC load which has a type 2 cable and can mimic the communication behaviour of a real EV. The device used for this was a commercially available EV-emulator from Walther-Werke. As can be seen in figure 3.13a, it can be plugged into an EVSE using a type 2 cable. It has switches that allow it to communicate status A, B, C or D and a rated cable current of 13 A, 20 A, 32 A or 63 A. Note that while it allows the user to select a rated cable current up to 63 A, the actual wires it uses are only 1.5 mm² so a maximum of approximately 16 A can be drawn to prevent overheating. Furthermore, the EV-emulator has lights to show when the phases are live and it has a Bayonet-Neill-Concelman (BNC) coaxial connector that allows monitoring of the CP signal.

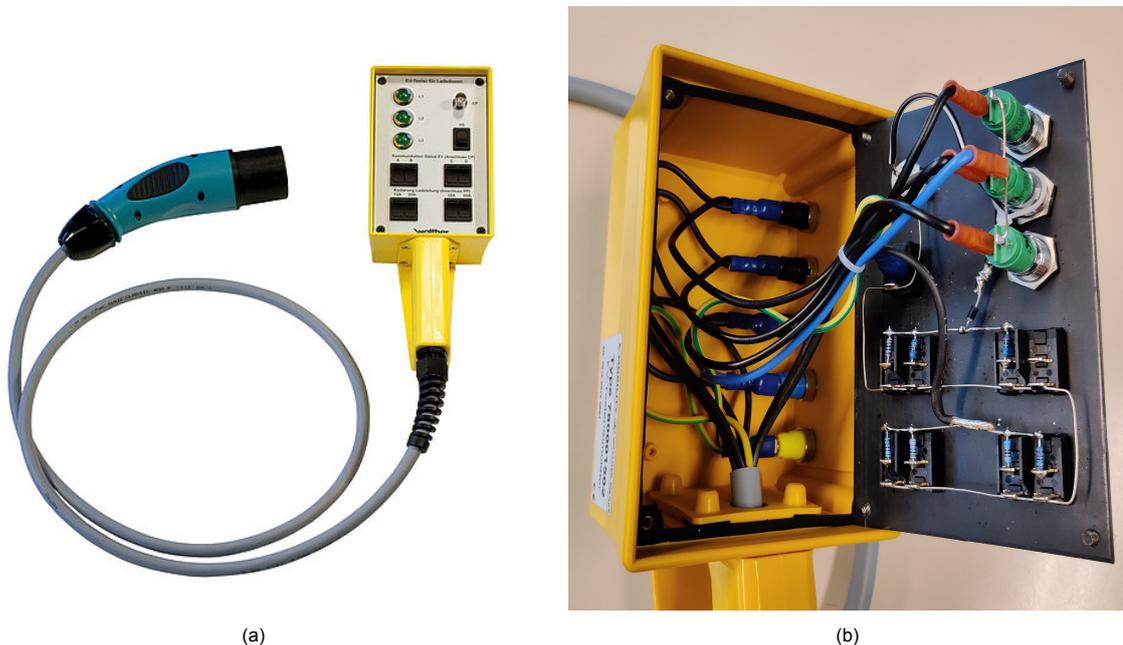


Figure 3.13: Walther-Werke EV-Emulator external [51] (a) and internal (b) view

Figure 3.13b shows the relatively simple internal workings of the EV-emulator. The three phases, neutral and PE are routed to safety sockets on the side of the device which can be used to connect an external load. They are also routed to the lights so that the user can see which phases are live. The CP, PP and PE are routed to the switches to allow the status and rated cable current to be set by the user. Each switch can activate a resistor between CP/PP and PE, as per table 3.2 and 3.3.

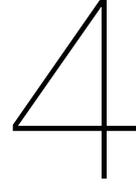
3.6. SM15K DC Source

Lastly, some hardware is needed that can act as a load. The EV emulator described above does not do anything with the power coming from the EVSE, but simply feeds the three phases (plus neutral and PE) through to 5 safety sockets. In a real EV, the power supplied by the EVSE eventually makes its way into the EV's battery where it is stored. Therefore, in this setup, there must be a way to draw power in a manner similar to a real EV. The easiest solution to this would be to use an AC load. However, no appropriately sized AC load was available for this project, so two DC power supplies are used instead. The power supplies used are two Delta Elektronika SM15K series, specifically model SM500-CP-90. These are bidirectional DC supplies which can both sink and supply 15kW at a maximum of 500V or 90A. For the purposes of this project, these two power supplies are connected back-to-back to form an AC load that can circulate power back to the (emulated) grid, as described in section 5.4. While the SM15K power supplies are not advertised for this purpose, their bidirectionality makes it possible. These devices by themselves are AC/DC converters so connecting them back-to-back realizes an AC-DC-AC topology, which is essentially what is used by a conventional regenerative AC load.

The benefit of using such a regenerative load, which is capable of feeding power back to the grid as opposed to using a resistive load, is that it significantly reduces the wasted energy. Given the real-time nature of this project, test scenarios will be running for several hours so if a resistive load were used, this would result in a lot of energy being wasted as heat and a rather toasty lab.



Figure 3.14: Delta-Elektronika SM500-CP-90 Bidirectional DC Power Supply [52]



Smart Charging Algorithm

The development of a smart charging algorithm is not an explicit goal of this thesis; the main focus is on the development of the PHIL test-bed. Nevertheless, an existing algorithm is used and slightly adapted, so knowledge of how it works is needed. The main concepts are described in this section. A more detailed explanation of the algorithm can be found in [53].

The algorithm is written in Python. At its heart, it uses a mathematical optimization solver named Gurobi to calculate the optimal setpoint for each EV connected to a node, based on a set of input parameters and constraints.

4.1. Objective Function

The main goal of the algorithm is to determine the optimal charging setpoint for each EV connected to a node. This optimization consists of minimizing the objective function show in equation 4.1. This objective function represents the total cost of smart charging for each node C_n^{opt} . It is made up of 3 parts [53];

- Penalty that needs to be paid to the EV user if the demanded energy d_j is not delivered by the departure time.
- In some cases, energy generated by an onsite PV installation is not free. For example, if the installation is owned by a third-party, there may be a pre-determined cost per kWh of energy generated C^{PV} .
- Cost of buying/selling energy from/to the grid.

Note that the total cost C_n^{opt} can be either positive or negative. In the case where there is relatively high local generation, the energy exported to the grid can be higher than the imported energy and thus the total cost can become negative.

$$\begin{aligned} \text{Min. } C_n^{\text{opt}} = & \sum_{j=1}^J \left(B_{n,j,T_j^a} + d_{n,j} - B_{n,j,T_j^d} \right) C_{n,j}^p + \Delta T \sum_{t=1}^T p_{n,t}^{PV} C^{PV} \\ & + \Delta T \sum_{t=1}^T \left(p_{n,t}^{g(\text{imp})} C_t^{e(\text{buy})} - p_{n,t}^{g(\text{exp})} C_t^{e(\text{sell})} \right) \end{aligned} \quad (4.1)$$

Table 4.1: Parameters of smart charging objective function, based on [53]

Parameter	Note	Unit	Type	Source
C_n^{opt}	Total costs of EV charging at node n	€	Output parameter	Optimization
B_{n,j,T_j^a}	Energy in battery of j^{th} EV at node n at arrival	kWh	Input parameter	User/EV (depending on protocol)
$d_{n,j}$	Demanded energy of j^{th} EV at node n	kWh	Input parameter	User
B_{n,j,T_j^d}	Energy in battery of j^{th} EV at departure time T_j^d at node n	kWh	Variable	Optimization
$C_{n,j}^p$	Penalty for not meeting energy demand by departure time of j^{th} EV at node n	€/kWh	Input parameter	CPO
$p_{n,t}^{\text{PV}}$	PV energy generated at time t at node n	kW	Input Parameter	Forecast
C_{PV}	Cost of obtaining PV energy	€/kWh	Input Parameter	CPO
$p_{n,t}^{g(\text{imp})}$	Power imported from the grid at time t at node n	kW	Variable	Optimization
$C_t^{e(\text{buy})}$	Market clearing price for buying electricity from grid at time t	€/kWh	Input Parameter	TSO
$p_{n,t}^{g(\text{exp})}$	Power exported to the grid at time t at node n	kW	Variable	Optimization
$C_t^{e(\text{sell})}$	Market clearing price for selling electricity to grid at time t	€/kWh	Input Parameter	TSO

4.2. Sub-Functions

To accompany the main objective function, some sub-functions are defined that show how variables relate to each other:

- The imported/exported power at the node is a function of voltage and current

$$\begin{aligned} p_{n,t}^{g(\text{imp})} &= V_{n,t} \times i_{n,t}^{g(\text{imp})} \\ p_{n,t}^{g(\text{exp})} &= V_{n,t} \times i_{n,t}^{g(\text{exp})} \end{aligned} \quad (4.2)$$

- The energy in an EV's battery at time t is equal to its battery size multiplied by the SOC. This is also equal to the energy in its battery at arrival plus the sum of power between arrival time and time t:

$$B_{n,j,t} = B_{n,j}^{\text{max}} \times S_{n,j,T_j^a} = B_{n,j,T_j^a} + \Delta T \sum_{T_j^a}^t (p_{n,j,t}^{e+} \eta_{n,j}^{ev}) \quad (4.3)$$

- The demanded energy is the arrival SOC minus the requested departure SOC, multiplied by battery size:

$$d_{n,j} = (S_{n,j,T_j^a} - S_{n,j,T_j^d}^{\text{req}}) \times B_{n,j}^{\text{max}} \quad (4.4)$$

- The imported power minus exported power at the node is the local load power minus the PV

power, plus the sum of EV powers

$$p_{n,t}^{g(imp)} - p_{n,t}^{g(exp)} = p_{n,t}^{LL} - p_{n,t}^{PV} + \sum_{j=1}^J p_{n,j,t}^{e+} \quad (4.5)$$

- The local load power at time t is the normalized load profile multiplied by the rate yearly energy consumption.

$$p_{n,t}^{LL} = p_{n,t}^{LL,norm} \times p_n^{LL,r} \quad (4.6)$$

- The PV power at time t is the normalized PV profile multiplied by the rated PV size.

$$p_{n,t}^{PV} = p_{n,t}^{PV,norm} \times p_n^{PV,r} \quad (4.7)$$

Table 4.2: Parameters of smart charging sub-functions

Parameter	Note	Unit
$V_{n,t}$	Voltage at node n at time t	V
$i_{n,t}^{g(imp)}$	Current imported from the grid at time t at node n	A
$i_{n,t}^{g(exp)}$	Current exported to the grid at time t at node n	A
$B_{n,j,t}$	Energy in battery of j th EV at node n at time t	kWh
$B_{n,j}^{max}$	Battery size of j th EV at node n	kWh
S_{n,j,T_j^a}	SOC of j th EV at node n at time t	
$p_{n,j,t}^{e+}$	Charging power of j th EV at node n at time t	kW
$\eta_{n,j}^{ev}$	Charging efficiency of j th EV at node n	
$S_{n,j,T_j^d}^{req}$	Requested SOC at departure time T_j^d of j th EV at node n	
$p_{n,t}^{LL}$	Power of local load at node n at time t	kW
$p_{n,t}^{PV}$	Power of PV at node n at time t	kW
$p_{n,t}^{LL,norm}$	Normalized power of PV at node n at time t	kW
$p_n^{LL,r}$	Rated size of local load	kWh/yr
$p_{n,t}^{PV,norm}$	Normalized power of local load at node n at time t	kW
$p_n^{PV,r}$	Rated size of PV	kWp

4.3. Constraints

The optimization is subject to certain constraints:

- Current AC charging standards (IEC 61851) dictate that the lowest allowed charging current is 6A [49]. Under this standard, it is not possible to set a maximum allowed current lower than 6A. If less than 6A is available then the charging must be stopped.

$$\left(i_{n,j,t}^{e+} = 0 \right) OR \left(i_{n,j,t}^{e+} \geq 6 \right) \quad (4.8)$$

- The imported and exported current $i_{n,t}^{g(imp)}$ $i_{n,t}^{g(exp)}$ of the node are subject to limitations imposed by the distribution network capacity so there is a maximum current that can be imported and exported, $i_{n,t}^{G+}$ and $i_{n,t}^{G-}$ respectively:

$$\begin{aligned} i_{n,t}^{g(imp)} &\leq i_{n,t}^{G+} \\ i_{n,t}^{g(exp)} &\leq i_{n,t}^{G-} \end{aligned} \quad (4.9)$$

- Each EV has a maximum charging current and cannot charge at a current above this value:

$$i_{n,j,t}^{e+} \leq i_{n,j}^{EVr} \quad (4.10)$$

Note that in this implementation, it will be assumed that all EVs use three-phase charging and the currents are balanced.

Table 4.3: Parameters of smart charging constraints

Parameter	Note	Unit
$i_{n,j,t}^{e+}$	Charging current of j^{th} EV at time t and node n	A
$i_{n,t}^{G+}$	Distribution network capacity for importing current from grid at node n	kW
$i_{n,t}^{G-}$	Distribution network capacity for exporting current to grid at node n	kW
$i_{n,j}^{EVr}$	Rated current of j^{th} EV at node n	A

5

System Integration

5.1. Overview

Figure 5.1 shows the layout of the complete experimental setup with all its components. Solid lines represent power flows and dotted lines represent information flows. This figure is similar to figure 3.2 except that all necessary communication lines have been added. This chapter will describe the workings of the communication between different hardware, as well as the general integration of the different components into one system.

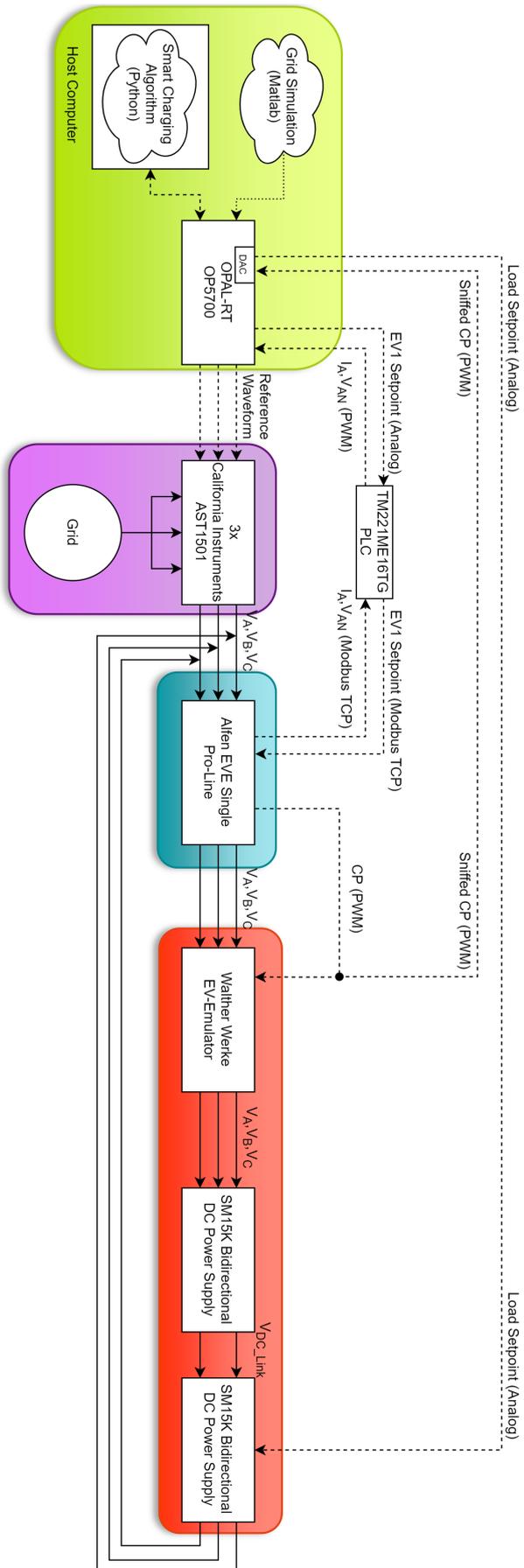


Figure 5.1: Overview of experimental setup. The colored blocks correspond to those of figure 3.1.

5.2. Adaptation of Algorithm

The smart charging algorithm described in section 4 was originally designed to run locally on a PC and to calculate the optimal charging behavior of several EVs over the course of several days. It was not designed to run in real-time nor as part of a PHIL system. To be able to incorporate the algorithm into the proposed PHIL system, some adaptations are needed.

As mentioned in section 3.2, the DRTS used runs a model made in MATLAB/Simulink. The optimization algorithm is written in Python, so this cannot be directly incorporated into the real-time simulation model. Several methods of incorporating the Python algorithm into the model were explored and in the end it was chosen to make use of RT-Lab's Application Programming Interface (API) functionality. This API can be used to allow a Python algorithm on the target computer to read and write variables within the real-time model running on the target. Strictly speaking, because this exchange of variables relies on the TCP/IP network that is in between the host and target, real-time behaviour cannot be guaranteed as there are many external (e.g. network congestion) that can influence the communication delay. However, the optimization will only be performed once per minute, so this relatively large time-step makes any communication delay insignificant.

In practice, this means that the Python algorithm and the real-time model can run in parallel on the host and target computer respectively. The RT-Lab API is used to exchange variables between the two systems and to keep them synchronized. Figure 5.2 shows how this works in practice from the perspective of the Python algorithm running on the host computer. Once all variables have been initialized in the Python algorithm, the RT-Lab API is used to read the current simulation time of the model. The algorithm then waits until the next full minute, at which point it will read the bus voltage of *node n* and each EV's SOC at this node, from the model. Because the optimization was designed to run only on one node, it must be run separately for each node. Therefore, after completing the optimization for *node n*, the bus voltage and SOC of *node n+1* can be read and the optimization can be run again with the new inputs. This is repeated for all nodes with EVs. Once all optimizations are completed, the algorithm sends the calculated setpoint for each EV to the DRTS. This process is repeated every minute.

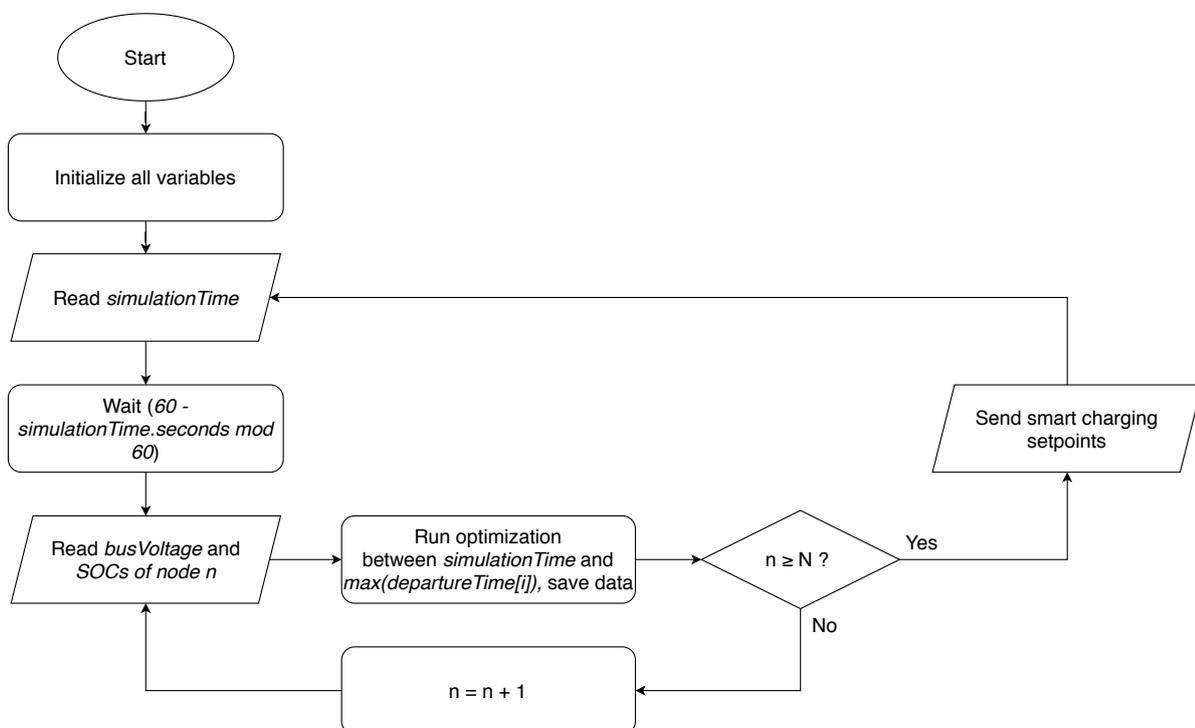


Figure 5.2: Flowchart showing how the Python algorithm runs alongside the real-time simulation for N nodes

5.3. Modbus Communication Using PLC

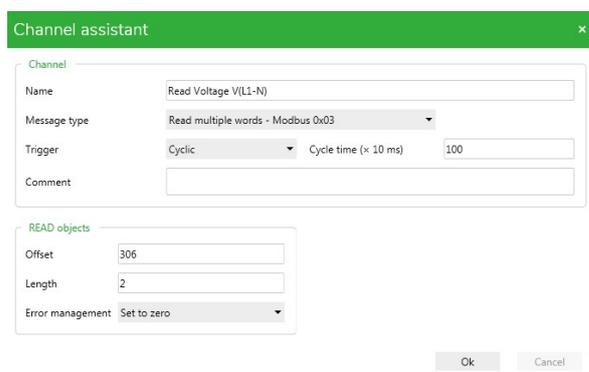
In section 3.4.2, details of the Schneider Electric Programmable Logic Controller (PLC) are shown, which is used to interface between the DRTS and EVSE. The PLC has two functions; the first is to read the analog signal coming from the DRTS, convert this to Modbus TCP and write it to the EVSE as the maximum allowed charging current. The second function is to read the current and phase voltage of phase 1 that is measured by the EVSE, and convert this information from a Modbus TCP format to a Pulse-Width Modulation (PWM) signal which can be interpreted by the DRTS.

The PLC can be programmed using a programming language known as Ladder Logic, in Schneider Electric's EcoStruxure Machine Expert software. Here the PLC needs to be programmed to read/write to the correct Modbus registers of the EVSE. The address of the needed registers can be found in a Modbus Register table provided by Alfen, part of which has been reproduced in figure 5.3 [54]. The entire table can be found in appendix A.

Description	Start address	End address	Number of 16 bit registers	Read or Write	Data Type	Step size & Units	Additional info
Energy measurements							
Meter state	300	300	1	R	UNSIGNED16	n.a.	Bitmask with state: Initialised: 0x01 Updated: 0x02 Warning: 0x04 Error: 0x08
Meter last value timestamp	301	304	4	R	UNSIGNED64	0.001s	Milliseconds since last received measurement
Meter type	305	305	1	R	UNSIGNED16	n.a.	0:RTU, 1:TCP/IP, 2:UDP, 3:P1, 4:other
Voltage Phase V(L1-N)	306	307	2	R	FLOAT32	1V	
Voltage Phase V(L2-N)	308	309	2	R	FLOAT32	1V	
Voltage Phase V(L3-N)	310	311	2	R	FLOAT32	1V	

Figure 5.3: Part of the Modbus register table of the Alfen EVE Single Pro-Line [54]

All registers that will need to be written to or read can be programmed in the Channel Assistant within EcoStruxure Machine Expert, as shown in figure 5.4a. The message type can be specified (in this case, 0x03 denotes reading multiple words from holding registers), as well as its offset and length. Each 'channel' will be written/read at a set time interval, in this case every second. Each configured channel is assigned an address in the PLC's input/output registers, as shown in figure 5.4b. Because the PLC works with 16-bit words and the values read from the EVSE are 32-bit floats, each float is split into two words, each with its own address.



(a)

Input registers (IOScanner) properties

Used	Address	Device name	Channel
<input checked="" type="checkbox"/>	%IWN300.1.0	EVSE	Read Voltage V(L1-N)
<input checked="" type="checkbox"/>	%IWN300.1.1	EVSE	Read Voltage V(L1-N)
<input checked="" type="checkbox"/>	%IWN300.2.0	EVSE	Read Current L1
<input checked="" type="checkbox"/>	%IWN300.2.1	EVSE	Read Current L1

(b)

Figure 5.4: Screenshots of the Channel Assistant (a) and input registers (b) of the PLC

Once the channel assistant is configured and the desired variables are being read into the input/output

registers of the PLC, they are saved into the PLC's memory so that the variables can be manipulated and used in calculations. To understand how the variables are saved in the memory, it is important to understand that there are several types of memory objects:

- Memory bits (%Mi): Stores a single bit
- Memory words (%MWi): Stores a 16-bit word (integer)
- Memory double words (%MDi): Stores a 32-bit integer word (integer)
- Memory floating point (%MFi): Stores a 32-bit floating point according to IEEE 754 standard: the most significant bit signifies the sign, the next 8 bits signify the exponent and the 23 least significant bits signify the fraction.

For the purposes of this project, only memory words and memory floats are relevant. Note that only a float can store decimals and that two (16-bit) words are needed to store one (32-bit) float. Floats and words are stored in the same memory location of the PLC, so if a float is stored in %MFi, then its data can also be accessed by reading %MWi and %MWi+1. This is visually represented in table 5.1. This feature is used in the programming of the PLC. For example, for reading the measured voltage and current from the EVSE, the two input registers corresponding to one of these variables are saved to two memory words; %MWx and %MWx+1. However, these two words then each contain half a float so they cannot be interpreted by themselves. Fortunately, the corresponding memory float %MFx contains the bits of both these words and it can easily be changed from 32-bit floating point format to a regular binary integer using the $\%MWy = REAL_TO_INT(10 \times \%MFx)$ function. Note that the float %MFx is multiplied by 10 so that a resolution of 0.1V can be achieved. Memory words are always integers so the conversion rounds the float to the nearest whole number. Lastly, %MWy is then converted into a duty cycle of the digital output signal of the PLC, ready to be read by the DRTS.

Table 5.1: Demonstration of how float and word objects overlap in the PLC's memory

Memory Floats (even)	Memory Floats (odd)	Memory Words
%MF0		%MW0
	%MF1	%MW1
%MF2		%MW2
	%MF3	%MW3
%MF4		%MW4

...	%MFi	%MWi
%MFi+1		%MWi+1

The duty cycle for the voltage measurement is set as $D_v = 70 + (\%MW_y - 2300)/4$ such that 0% duty cycle represents a bus voltage of 0.88 p.u. and 100% represents 1.04 p.u. Similarly, the duty cycle representing the measured current is $D_i = 5 \times \%MW_z$, allowing for a range of 0A to 20A. The reason for these extra calculations is that it allows measuring of the full expected range of voltage and current with maximum resolution, as the PLC can only produce a duty cycle between 0 and 100% in steps of 1%, so only 101 steps are available.

The setpoint which is received from the DRTS and sent to the EVSE is converted in a very similar way as above, the only difference being that it is read from an analog input with a value from 0-1000 representing 0.00V to 10.00V. This value is divided by 50 before it is written to the EVSE, such that 0.00V would represent a 0A setpoint and 10.00V would represent a 20A setpoint.

5.4. Circulating Power

As an intermediate step to the full setup, a partial setup was built to test the proper operation of the DRTS, power amplifier and SM15K power supplies. The amount of power being circulated in the system was gradually increased in order to check if the system remained stable, which it did.

5.4.1. Layout

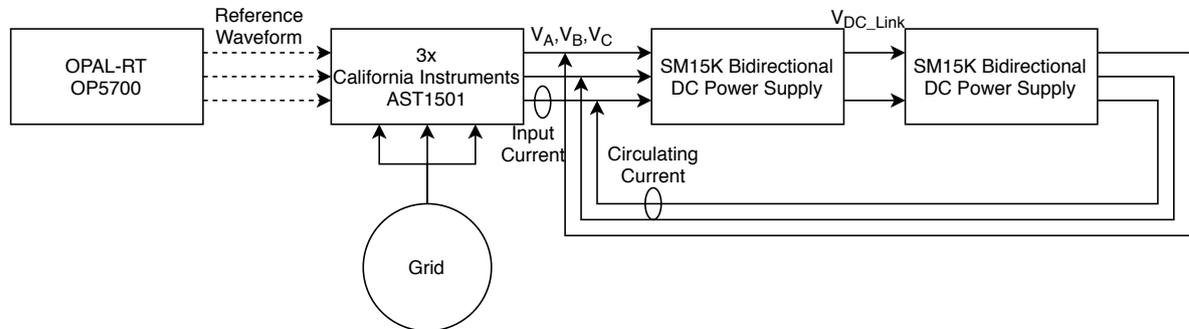


Figure 5.5: Schematic overview of experimental setup

Figure 5.5 shows an overview of the experimental setup. The DRTS generates three low-voltage sine waves of adjustable amplitude and with 120 degree phase difference. This is amplified by the power amplifier to generate a controllable grid voltage. Two back-to-back DC power supplies are plugged in to this, with a DC link in between. One power supply acts as a AC/DC converter and the other as a DC/AC converter. The voltage of the DC link is set to a fixed value so by manipulating the current drawn by the second power supply, the amount of recirculated power can be controlled. Two current clamps attached to a high-speed oscilloscope provide measurements of the input and circulating currents, which can be seen in figure 5.7.



Figure 5.6: Annotated picture of experimental setup. The annotation colors used correspond to the colors of figure 3.2.

5.4.2. Measurements

Figure 5.7 shows the measurements taken by the current clamps shown in figure 5.5. The top plot shows that, as is to be expected, that circulating current increases as the circulating power is increased. The bottom plot shows that the input current increases when the circulating power is increased, but its change is relatively limited. It is noticeable that even with no current circulating (i.e. the power supplies are set to 0A) there is still a significant current flowing. At $P_{\text{Circ}} = 2.5 \text{ kW}$, this current is approximately

90 degrees out of phase with the voltages, suggesting that this is purely reactive current, likely due to large input filters on the SM15K power supplies. This is confirmed by the results in table 5.2, which shows the apparent, active and reactive power along with the power factor as measured by the amplifier itself. Even with the power supplies off, there is a significant reactive power drawn so this suggest that it must be caused by passive components inside the power supplies. Do note that the power factor mentioned in this table is only the power factor of the input power. The circulating power has a much higher power factor of 0.99 at 2.5 kW and 1.00 at 10 kW.

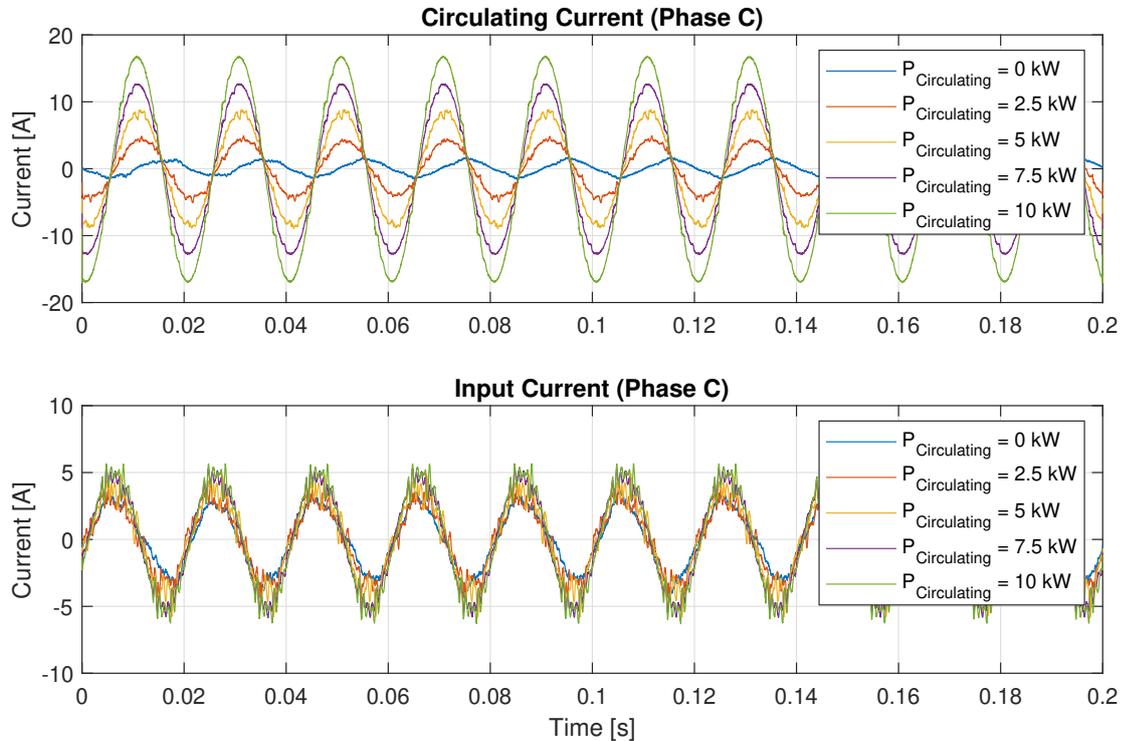


Figure 5.7: Circulating and input current on phase C with various circulating powers

Table 5.2: Input power provided by power amplifier

	S [kVA]	P[kW]	Q [kVAR]	PF
Supplies Off	1.49	0.00	1.49	0.00
Supplies On	1.47	0.19	1.46	0.13
DC Link on	1.45	0.36	1.40	0.25
P_{Circ} = 2.5 kW	1.72	0.42	1.67	0.24
P_{Circ} = 5 kW	2.01	0.51	1.95	0.26
P_{Circ} = 7.5 kW	2.34	0.64	2.26	0.27
P_{Circ} = 10 kW	2.55	0.79	2.43	0.31

Figure 5.8 is a graphical representation of some of the information in table 5.2, showing the active input power as a function of recirculating power. This active power is being supplied by the power amplifier and essentially represents the losses of the system. As the circulating power increases, so does the input power. This makes sense as the increased circulating current through the system means there are increased losses in the various components such as cables and power electronics.

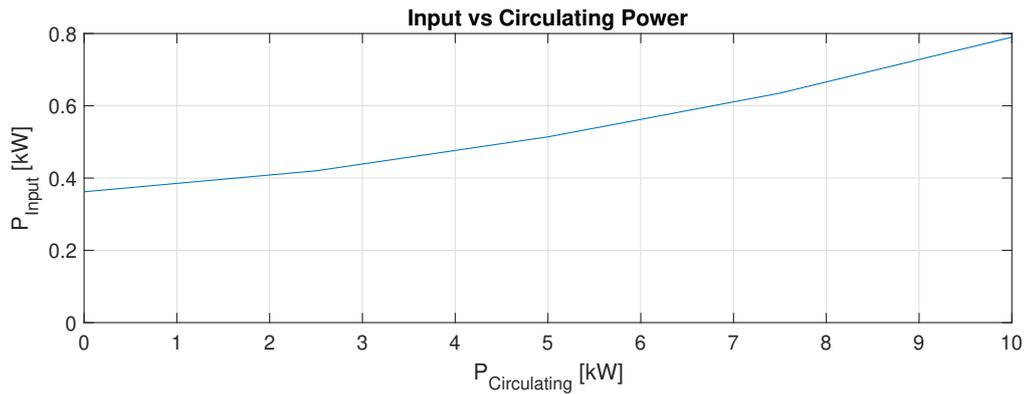


Figure 5.8: Measured input power as a function of recirculating power

5.5. Simulink Model

The Simulink model runs on the DRTS. It controls all of the simulation parameters, stores important measurement data and integrates the different processes within the setup into one coherent system. Screenshots of the model can be found in appendix B.

As discussed in section 3.2, a Simulink model on the DRTS can be split up into subsystems of three different categories;

- The console (or user interface) which the user can use to interact with the model while it is running.
- The master which contains critical logic as well as all I/Os that communicate with external hardware.
- (Optional) One or more slaves which can also contain critical logic. Each slave runs on its own CPU core and can perform calculations concurrently to the master's calculations. If desired, slaves can run at a larger timestep than the master.

In the Simulink model designed for this project, one subsystem of each type is used. The reason for using a slave subsystem is that the DRTS available in the lab is licensed to use a maximum of 2 cores, so one core can be used by the master and the other by a slave so that calculations can be executed concurrently and with different timesteps. This is important because the system needs to be able to generate three 50 Hz sine waves, as described in section 3.3. To accurately generate these sine waves with acceptable resolution, the master must run at a timestep T_{master} that is small enough, such that $\frac{1}{T_{master}} \gg 50\text{Hz}$. This is in order to reduce the 'staircase' effect of a digitized signal, as seen in figure 5.9. However, the DRTS is also running the Newton Raphson power flow analysis which is a relatively time-consuming process. To be able to complete the power flow analysis within the timestep and without overruns, even a simple grid such as the one in section 3.3 requires a timestep T_{slave} in the order of at least 1 ms. For more complex grids, a larger timestep would be needed. By separating the model into a master and a slave, it enables the possibility to make a 'multi-rate' model and run the master at a much smaller timestep than the slave. The T_{master} was set to 100 μs and T_{slave} to 10 ms. This way, the power flow analysis can run with a timestep large enough as to not cause overruns, while concurrently the sine waves can be generated with high resolution.

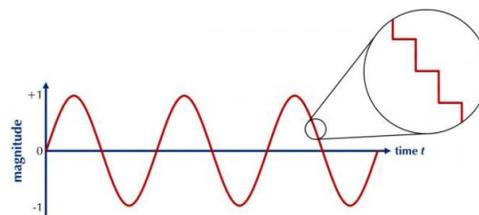


Figure 5.9: Graphical representation of a sine wave showing its digitized staircase form in greater detail [55]

5.5.1. Top-Level

The top-level of the Simulink model has three subsystems; master, slave and console (user interface). Variables are exchanged between the subsystems using signals. It is important to note that, as discussed in section 3.2, the master and slave are running on the DRTS target in real-time, so their communication is internal within the DRTS and therefore synchronous. The console is running on the host PC and the communication between the console and master/slave happens over TCP/IP so it is not completely synchronous and real-time behaviour cannot be ensured. The outputs of the console can be adjusted by the user while the simulation is running; other parameters cannot. The inputs to the console can be displayed on a scope and viewed on the host PC (with a few seconds delay) while the simulation is running.

5.5.2. Master

The master subsystem handles all analog and digital I/Os and user inputs coming from the console. Its key functions include:

- Converting CP duty cycle to an analog setpoint for the AC load
- Generating three sine waves based on the bus voltage calculated by the power flow analysis, to be used as reference waveforms for the power amplifier
- Emulating the charging behaviour of an EV by stopping the charge when $SOC \geq 1$ and a simple implementation of Constant-Voltage (CV) charging

Conversion of CP Duty Cycle to Analog Setpoint

The control pilot PWM signal generated by the EVSE needs to be converted to an analog setpoint for the AC load. This is so that the load can draw the power that would normally be drawn by the EV.

The *PWM In* is used to read the duty cycle of the DRTS' digital input pins. Firstly, this input is sanitized; if the duty cycle is below 9% or above 30%, this is assumed to be an unrealistic value and the load power is set to 0 for safety reasons.

Next, the duty cycle is multiplied by 60 (the inverse of the calculation described in section 5.3) to change the duty cycle into the corresponding current setpoint set by the EVSE. This current is multiplied by the nodal voltage at the node where the EV is located at that point in time to get a power. From this power, the expected losses are subtracted. The reason for this is that the load used is not a conventional AC load but two back-to-back bidirectional DC power supplies. Because of this, only the power on the DC bus can be controlled. Due to the losses of the AC/DC converter inside the power supply, the DC power is not exactly equal to the AC power. In section 5.4.2, these losses were measured so this data can be put into a Simulink look-up table so that an estimation of the losses can be made.

Once the expected converter losses have been subtracted, this DC-link power setpoint needs to be sent to the power supplies. The power supplies' DC current can be controlled via an analog signal between 0 V and 5 V, where 0 V represents 0 A and 5 V represents the maximum current (90 A). Therefore, the desired DC power setpoint is divided by the DC link voltage to get a DC current, and then multiplied by 5/90 to get the voltage of the analog signal between the DRTS and the power supply.

Generation of Reference Waveforms for Power Amplifier

Another key function of the master subsystem is the generation of three sine waves which are amplified by the power amplifier in order to form a local grid with controllable voltage. The master subsystem generates three sine waves with a frequency of 50 Hz and phase of 0, $2\pi/3$ and $4\pi/3$. The amplitude of these sine waves is determined by the results of the grid simulation. The calculated p.u. voltage is multiplied by 5.75 (as explained in section 3.3.1) and this becomes the amplitude.

EV Charging Behaviour

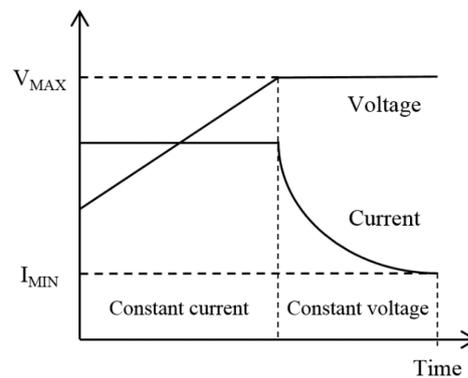


Figure 5.10: Constant current - constant voltage charging profile of a battery [56]

In order to realistically emulate an EV, it cannot be assumed that an EV will draw exactly the current that was set by the user or by the smart charging algorithm at any time. Because of this, two features are implemented. Firstly, if the battery is full ($SOC \geq 1$), the EV will stop charging, regardless of what setpoint it receives. Also, there is a simplified implementation of a battery's Constant-Voltage charging stage. A typical battery charging profile can be seen in figure 5.10. Near the end of the charge, when the battery is almost full, the EV no longer accepts full current and the current reduces as the SOC increases. A simplified implementation of this effect was realized by linearly reducing the current as a function of SOC, when the $SOC \geq 0.9$. These two features are summarized in equation 5.1.

$$I = \begin{cases} I_{setpoint} & \text{if } SOC < 0.9 \\ \min(I_{setpoint}, 10 \times I_{Rated} \times (1 - SOC)) & \text{if } 0.9 \leq SOC < 1 \\ 0 & \text{if } SOC \geq 1 \end{cases} \quad (5.1)$$

5.5.3. Slave

The slave subsystem runs at a larger timestep and can therefore do more time-consuming operations such as reading and writing files, and running the grid simulation. The slave subsystem has the following key functions:

- Reading the load and PV profiles from a file
- Calculating SOC
- Running the grid simulation
- Recording all relevant data to a file

Reading Load & PV Profiles

At the nodes within the simulated grid, there can be more than just EVs. Therefore, the possibility to add a local load or local (PV) generation is added. These powers vary throughout the day so real historical data is used. This data needs to be read from a file during the real-time simulation.

Before loading the model onto the DRTS, a MATLAB file containing the necessary data with timestamps is loaded onto the DRTS. During the simulation, this is read using RTLab's *OpFromFile* block in Simulink.

Calculating SOC

The SOC is calculated as the integral of power, divided by the battery size B_{sz} multiplied by 3600 (to convert from hours to seconds).

$$SOC = \frac{1}{3600 \times B_{sz}} \int P dt \quad (5.2)$$

The initial condition of the integrator block in Simulink is set to the arrival SOC of the EV and the upper saturation limit is set to the maximum allowed SOC of the EV.

Running Grid Simulation

The grid simulation is run using the *MATLAB Function* block, which has as input the load power for each load within the grid and outputs the p.u. voltage at every node in the grid. Note that the load power can be either positive (node imports power from the grid) or negative (node exports power to the grid). More information on the implementation of the grid simulation can be found in section 3.3.2.

Recording Data to a File

During the real-time simulation, all relevant data needs to be recorded and saved to a file so that it can be used for data analysis afterwards. The signals that need to be recorded are sent to an *OpWriteFile* block where it is written to a file during the simulation. Once the simulation is finished, the target automatically transfers the file to the host computer using TCP/IP.

5.5.4. Console

The console allows the user to interact with the simulation while it is running. It has the following key functions:

- Monitoring of signals on a scope, including:
 - All calculated and measured bus voltages
 - All measured currents and setpoints
 - SOC of each EV
 - Imported power from grid
 - Power consumed by each EV
 - Local load & PV power
- Possibility to manually set the output voltage of the power amplifier
- Possibility to manually set the charging current of all EVs
- Possibility to change the yearly consumption of the local load at node 5 & 15, and the PV size at node 5.

6

Uncontrolled Charging

In order to be able to evaluate the effect of controlled charging, it must be compared to uncontrolled charging. In this case, the smart charging algorithm is not active and the EVs charge at full power as soon as they arrive.

Figure 6.1 shows the complete physical setup that was used for all uncontrolled and controlled test cases.

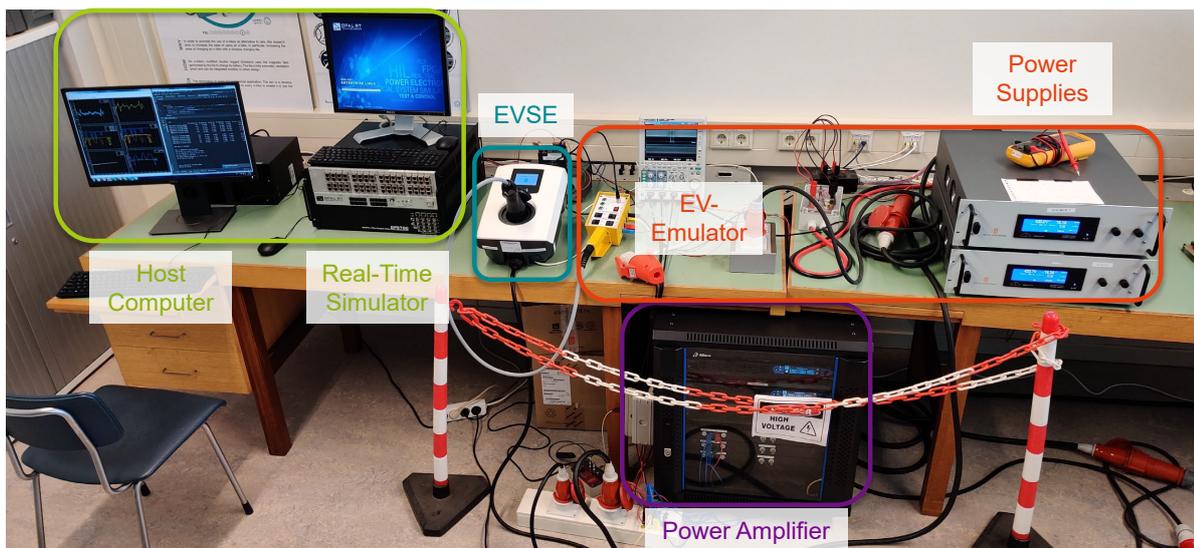


Figure 6.1: Picture of complete experimental setup. The annotation colors correspond to figure 5.1.

6.1. Simulation scenario

Table 6.1 shows the parameters used for this scenario. The exact same parameters will also be used for the smart charging base case in section 7.1.

The scenario is based on the case of two small businesses where employees park their EVs when they arrive at work and then depart at the end of the working day. Because of this, the simulation time is set to be from 9:00 to 17:00; a standard working day. As can be seen in figure 6.2, There are 4 EVs in total; 3 at node 5 and 1 at node 15. Only EV1 is represented by the PHIL system; the others are virtual. EV1, EV2, and EV3 have exactly the same parameters so that the difference between a hardware EV (EV1) and purely software EV (EV2) can be seen, as well as the difference between 2 EVs at different nodes (EV2 at node 5 and EV4 at node 15). EV3 has different parameters (larger battery, higher charging current, latter arrival and departure time) in order to see the effect of these different parameters.

The two types of EV parameters used are based on the two most common EVs currently registered in The Netherlands, a Tesla Model 3 and Tesla Model S respectively [57]. These have a battery size of 50 kWh and 100 kWh and a rated charging current of three-phase 16 A and 24 A respectively [58, 59].

Grid nodes 5, 15 and 19 (in figure 6.2) all have a local load. Node 5 also has a local PV installation. The profiles for this local load and solar irradiation were based on historical data taken from the Royal Netherlands Meteorological Institute (KNMI) and the Association for Dutch Energy Data Exchange (NEDU). Data from 01/06/2018 was arbitrarily chosen as it was a standard working day and there were significant fluctuations in solar irradiation, so the effect of that on the controlled charging can be studied. Each node has a current limitation of 50 A, which is a standard connection for small businesses provided by Dutch DSOs [60]. This limit need not necessarily be a physical component limit of the grid, it can also be a contractual limit imposed by the DSO and exceeding it may result in a penalty. In the case of a contractual limit, the DSO may also place a fuse which will blow when the limit is exceeded.

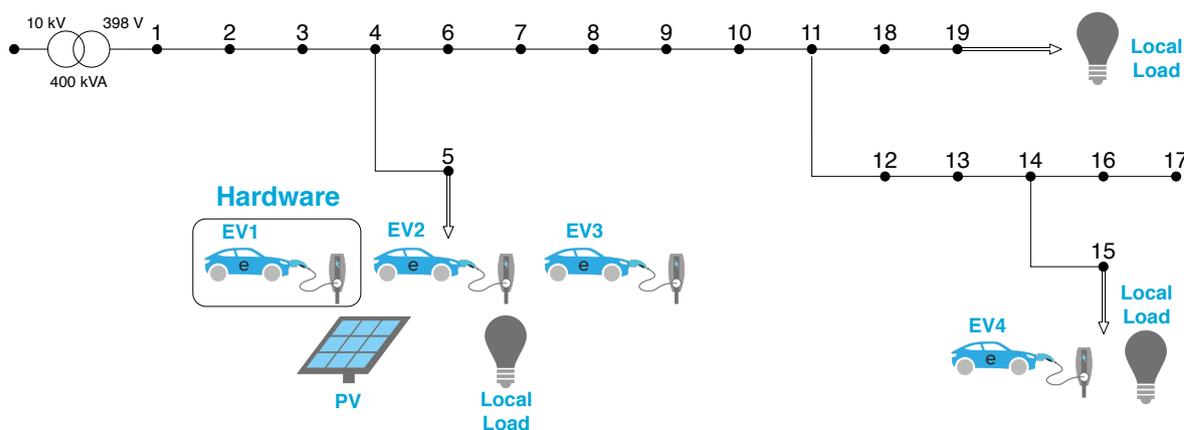


Figure 6.2: Locations of the EVs, local PV generation and local loads

As discussed in section 4.1, the objective function of the smart charging algorithm is based on minimizing cost. Therefore energy prices must be taken into account. The assumption is made that the EV charging station operates under a variable energy tariff so that the smart charging algorithm can benefit from these variable prices. The price of buying electricity from the grid is based on historical data from the Dutch Day-Ahead market on 01/06/2018. The price of selling electricity to the grid is set to a constant 0.02 €/kWh, which is always below the buying price. The cost of producing PV energy is set to 0 €/kWh as it is assumed that the PV installation is not owned by a third party so there are no variable costs associated with PV production. The penalty for not delivering the requested energy is set to a value that is always higher than the energy buying price, 0.10 €/kWh.

The grid model used for all simulations is the same Dutch rural grid as shown in figure 3.7, based on data from a Dutch DSO. However, in order to more clearly demonstrate the effect of uncontrolled and controlled charging on voltage variations in the grid, the line lengths were multiplied by a factor of 4. This allows for more significant variations in the voltage of each bus so that the effects become more apparent.

Table 6.1: Parameters used for uncontrolled & controlled charging scenarios

General				
	Time of simulation	9:00 to 17:00		
	Number of EVs (J)	4		
	Number of nodes (N)	2		
	Electricity Buying Price ($C_t^{e(buy)}$)	Day-ahead prices [61], 01/06/2018		
	Electricity Selling Price ($C_t^{e(sell)}$)	0.02 €/kWh		
	Variable PV Cost (C_{PV})	0 €/kWh		
	Penalty ($C_{n,j}^p$)	0.10 €/kWh		
	Normalized PV Profile ($p_{n,t}^{PV, norm}$)	KNMI data [62], 01/06/2018		
	Normalized Load Profile ($p_{n,t}^{LL, norm}$)	E2A NEDU Profile[63], 01/06/2018		
Node 5				
	EVs at this node (j)	EV1	EV2	EV3
	Arrival time (T_j^a)	9:00	9:00	9:30
	Arrival SOC (S_{n,j,T_j^a})	40%	40%	40%
	Departure time (T_j^d)	16:30	16:30	17:00
	Requested departure SOC ($S_{n,j,T_j^d}^r$)	100%	100%	100%
	Battery size ($B_{n,j}^{max}$)	50 kWh	50 kWh	100 kWh
	Maximum charging current ($i_{n,j}^{EVr}$)	3x16 A	3x16 A	3x24 A
	Maximum node current ($i_{n,t}^{g(imp)}, i_{n,t}^{g(exp)}$)	3x50 A		
	PV Size ($p_n^{PV,r}$)	25 kWp		
	Local Load Yearly Consumption ($p_n^{LL,r}$)	88.779 MWh		
Node 15				
	EVs at this node (j)	EV4		
	Arrival time (T_j^a)	9:00		
	Arrival SOC (S_{n,j,T_j^a})	40%		
	Departure time (T_j^d)	16:30		
	Requested departure SOC ($S_{n,j,T_j^d}^r$)	100%		
	Battery size ($B_{n,j}^{max}$)	50 kWh		
	Maximum charging current ($i_{n,j}^{EVr}$)	3x16 A		
	Maximum node current ($i_{n,t}^{g(imp)}, i_{n,t}^{g(exp)}$)	3x50 A		
	PV Size ($p_n^{PV,r}$)	0 kWp		
	Local Load Yearly Consumption ($p_n^{LL,r}$)	3.556 MWh		
Node 19				
	EVs at this node (j)	None		
	Maximum node current ($i_{n,t}^{g(imp)}, i_{n,t}^{g(exp)}$)	3x50 A		
	PV Size ($p_n^{PV,r}$)	0 kWp		
	Local Load Yearly Consumption ($p_n^{LL,r}$)	5.618 MWh		

6.2. Simulation results

Figure 6.3 shows the results of a PHIL simulation with uncontrolled charging. As stated in table 6.1, the simulated time period is 8 hours. Since this is a real-time HIL simulation, this means that each simulation also took 8 hours to complete in reality. During this simulation, the smart charging algorithm was not activated. Instead, each EV was allowed to charge at its maximum power between its time of arrival and time of departure, irrespective of energy prices, grid limits, and local load and PV. As a consequence, all EVs draw most of their energy in the morning, when there is limited PV generation. The combination of very little power generated by PV and all EVs charging at full power in the morning means that a lot of energy has to be imported from the grid and the maximum grid import current is exceeded by 28 A. This also results in a very low bus voltage in the morning, with a minimum of 0.91 p.u. at node 5.

The SOC of EV1, EV2 and EV4 increase at a very similar rate during the charge because their battery size, arrival time and maximum current are all the same. The only difference is that EV1 and EV2 are at node 5 while EV4 is at node 4. Because the bus voltage at node 15 is higher than at node 5 during the charge, EV4 charges slightly faster than the other two. All three EVs are finished charging by 12:45. EV3 has a larger battery and arrives 30 minutes later, so it is finished charging later, at 14:38. Note that, near the end of the charge, the EVs reduce their charging power due to the constant-voltage charging region explained in section 5.5.2. The total energy cost for this scenario is €16.08, or 0.16 €/kWh of energy delivered to the EVs.

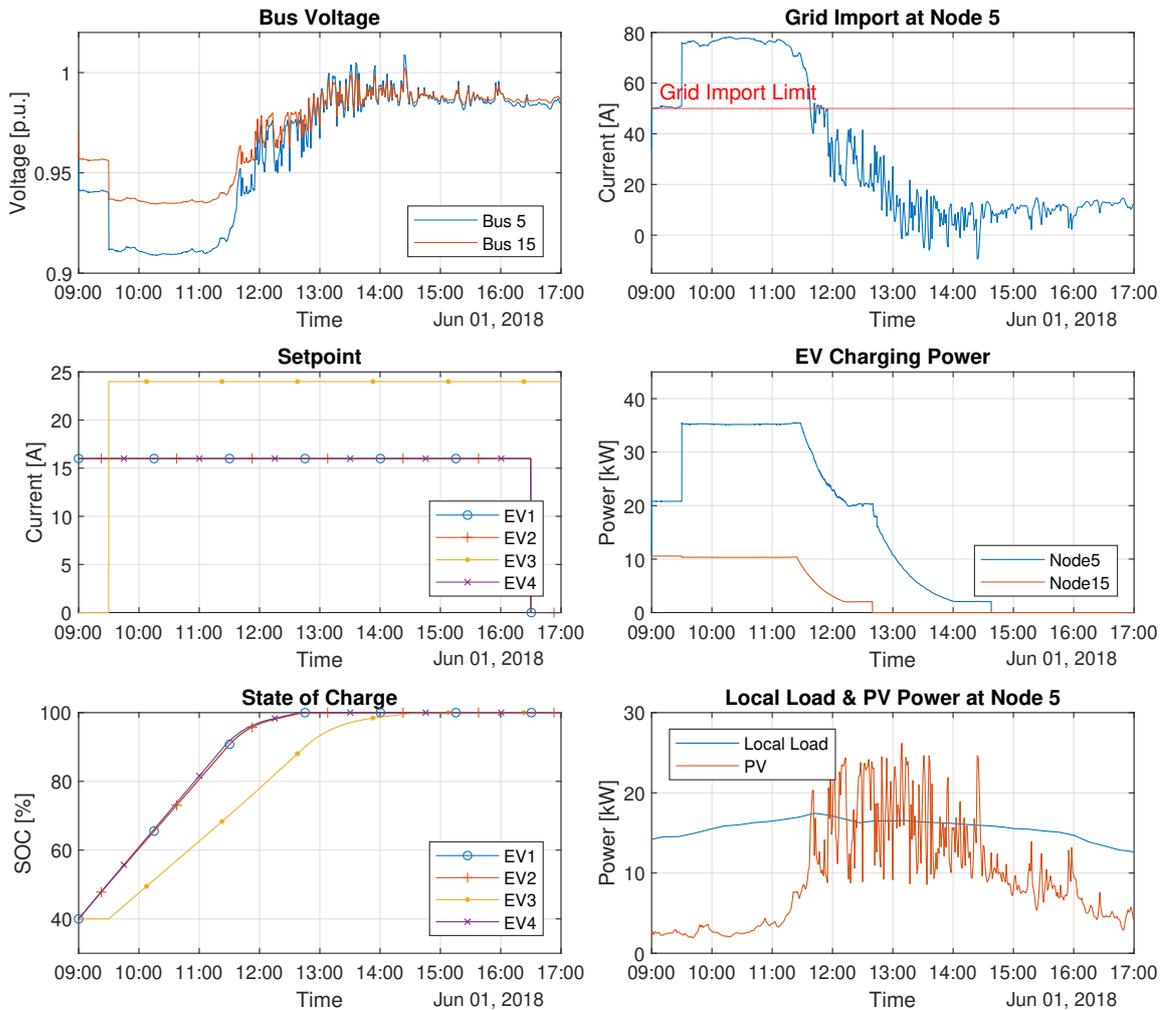


Figure 6.3: experimental results of uncontrolled charging

7

Smart Charging

7.1. Case 1: Base case

The same parameters as shown in table 6.1 are used for the smart charging base case. This base case is to serve as a benchmark for all other cases. Other cases will have only one parameter changed with respect to the base case, so that cross correlation issues can be avoided and the effect of this change can clearly be seen by comparing the results to the base case. In all smart charging cases, the smart charging algorithm described in chapter 4 was implemented.

Figure 7.2 shows an overview of the results of the base case simulation. Firstly, it can be noticed that, with the exception of a brief two-minute charge at 11:43, all EVs start charging only as of 12:00, despite the fact that they have already been connected to the EVSE for several hours. There are two reasons for this; the first can be seen in figure 7.1: the price of buying electricity from the grid is cheaper in the afternoon. Because of this, the optimizer within the algorithm tries to import power from the grid (and thus charge the EVs) as late in the day as possible (taking all constraints into account) in order to minimize cost. The second reason is that before 11:43, the PV production is not higher than the local load. If the PV had been higher than the local load earlier in the morning, then the EVs would have been charged with this excess energy because the cost of buying electricity from the grid $C_t^{e(buy)}$ is always greater than the money earned from selling this same power to the grid $C_t^{e(sell)}$. The combination of a high electricity buying price and no excess PV energy there means that no EVs are charged until 11:43.

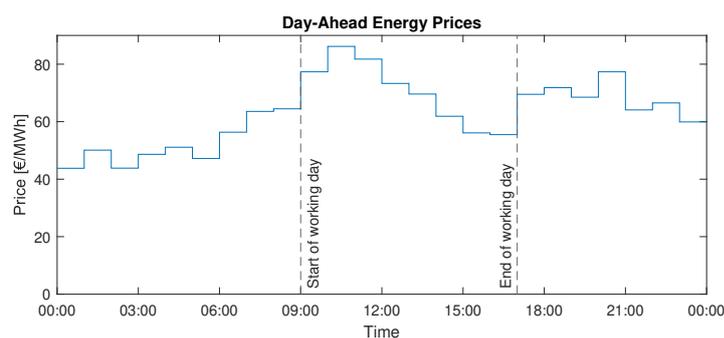


Figure 7.1: Day-Ahead market energy price

Another observation is that none of the EVs are fully charged at their time of departure; only to an SOC of 97.3%, 95.8%, 96.0% and 96.5% respectively. The reason for this is that the smart charging algorithm does not take into account the EVs' constant-voltage charging behaviour described in section 5.5.2. Instead, the algorithm assumes that all EVs always charge at their setpoint but this need not be the case. When the SOC > 90%, the EVs will charge with a current lower than their rated current and thus their SOC will increase at a rate slower than predicted by the algorithm. Additionally, when

the smart charging algorithm calculates its optimal power profile, it uses only the current bus voltage and does not take into account that the bus voltage may change in the future. Once the EVs start charging, the bus voltage drops and thus the power going to the EVs is reduced so their SOC will increase slower than predicted. This, in combination with the fact that the system waited as long as possible to commence the charge in order to minimize cost, means that it is not possible to fully charge the EVs in before their departure time. Since the optimization is rerun every minute, its prediction of the future SOC is continually updated so at some point, the algorithm would realise it is no longer possible to fully charge the EVs but it cannot compensate for it since it is already charging at the highest possible current within the given constraints.

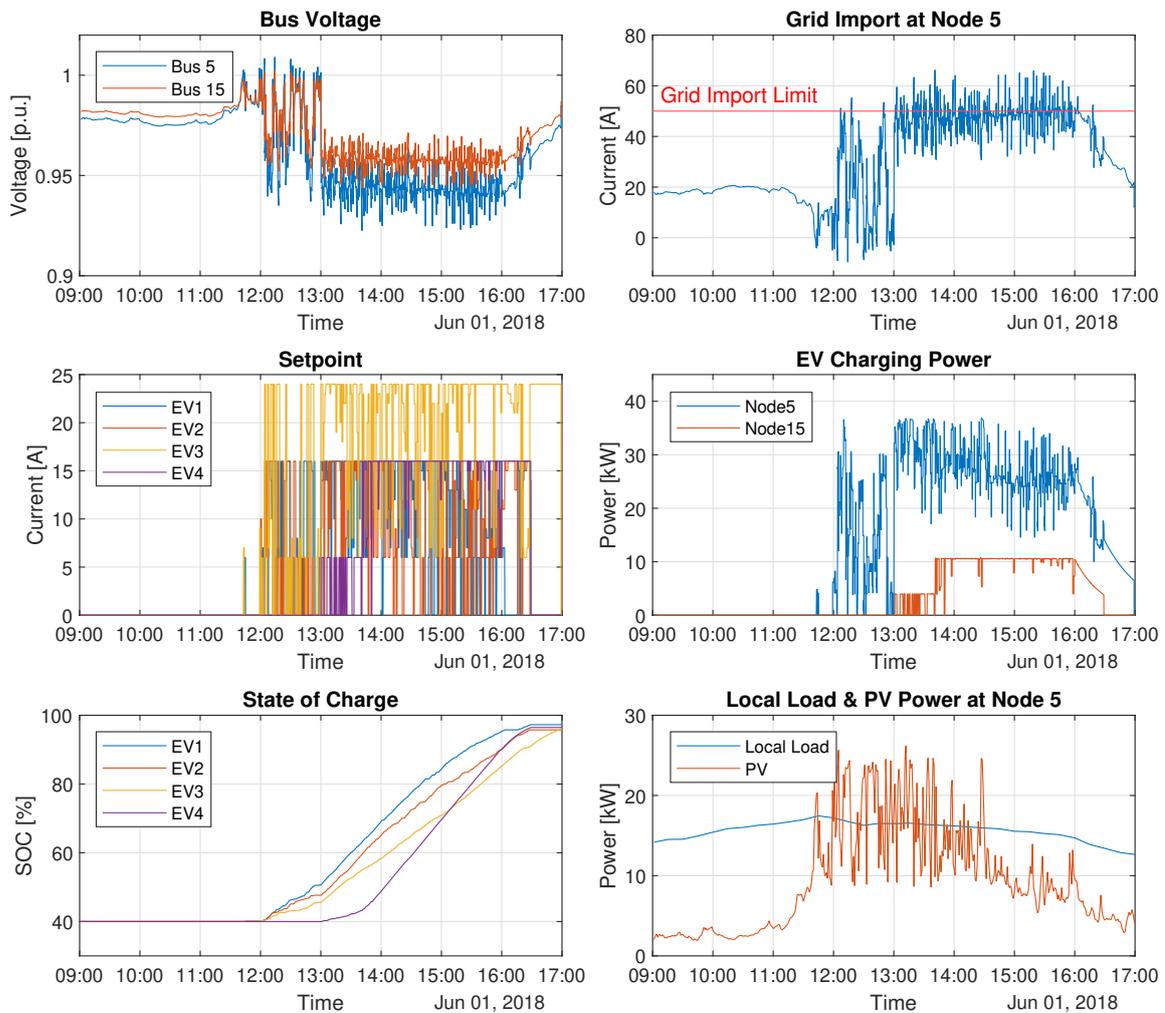


Figure 7.2: Experimental results of base case

The EV charging power shown in figure 7.2 shows a high amount of fluctuations. These are also seen in the grid import of node 5 and as a consequence, they also have an effect on the bus voltages. Part of these fluctuations can be explained by the plots in figure 7.3. The smart charging algorithm's optimization is completed at 15:11:03 and the new setpoints are sent at this time. The setpoint of EV1 is changed from 13 A to 0 A, EV2 from 0 A to 16 A and EV3 from 24 A to 22 A, all changed at exactly the same time. EV4's setpoint is unchanged. The net result of this change is only a 1 A increase in the total current of node 5. Yet, there is a brief 14 A increase in imported power. The reason for this is that EV1 is implemented as hardware and therefore has delays in responding to the new setpoint. This can be seen as EV1's measured current (purple line) does not instantly respond to its setpoint (blue line) and it takes about 9 seconds for it to go to 0 A. EV2 and EV3 on the other hand, are virtual and can respond instantly to changes in their setpoint. Because of this, there is an overlap of 9 seconds where EV2 has

already increased its charging current but EV1 has not yet fully reduced its current to 0 A. During this period of overlap, there is a spike in current imported from the grid by node 5 and consequently there is a dip in bus voltage. It is important to note that the root cause of this phenomenon is the difference in response times to a change in setpoint of each EV. In this case, it is perhaps not completely realistic that the virtual EVs have no communication delay, but it *is* realistic that not all EVs would have the same amount of delay. This is dependent on the internal workings of the EV's control system which will vary by EV model and manufacturer.

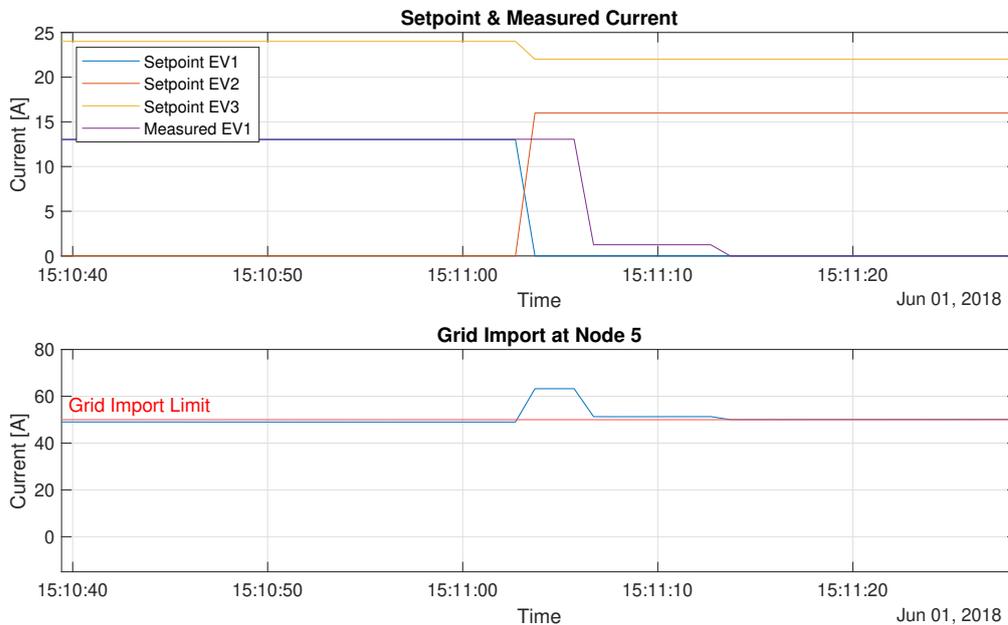


Figure 7.3: Cause of spikes in grid import

As explained, the effect seen in figure 7.3 is due to the fact that EV1 has a delay in responding to changes in setpoint while the other EVs do not. So what if this delay is removed, will the voltage fluctuations reduce? To test this hypothesis, the base case simulation was repeated with the same parameters, but with all EVs implemented as virtual EVs. This means that the simulation is still real-time, but it is purely software-based and the PHIL part of the setup was not used for this experiment. EV1's SOC was calculated based on its setpoint, and not using the measured current. The result of this can be seen in 7.4.

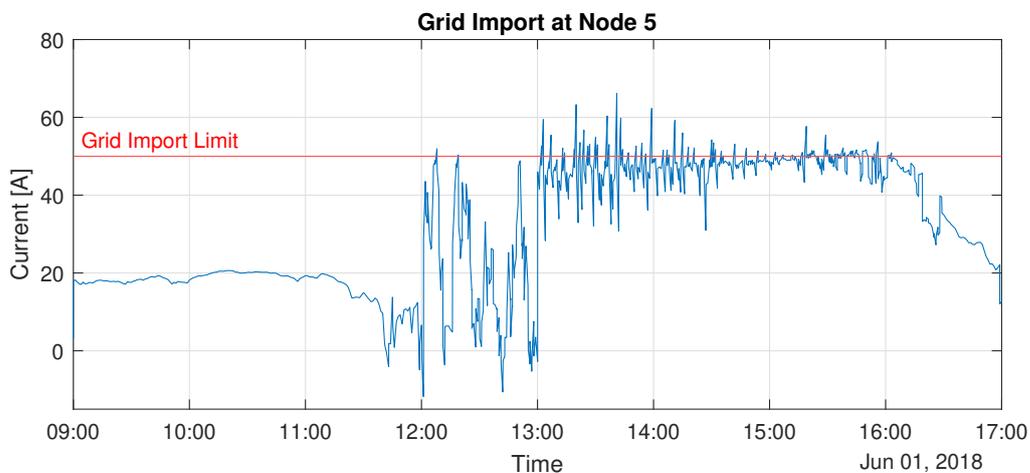


Figure 7.4: Grid Import at Node 5, software simulation

With respect to the hardware-in-the-loop results from figure 7.2, the oscillations have visibly reduced as the effect shown in figure 7.3 is no longer present due to EV1 no longer having a delay. However, there are still significant oscillations present and the grid import limit is still exceeded many times, but only for periods of less than a minute. This is because the smart charging algorithm is responsible for adjusting the EV power such that the limit is not exceeded. As described in 4, the algorithm has the local load and PV forecast and its optimization has a constraint that limits that total grid import power to below the grid import limit of 50 A. However, this optimization is only run once per minute and with timesteps of one minute. Therefore, the solution of this optimization will only need to satisfy its constraints at each timestep, and not in between timesteps. However, the PV power has continuous fluctuations so also between two optimization timesteps it is changing. The effect of this can be seen in figure 7.5. At 14:30:03, an optimization is completed and the EV setpoints are adjusted accordingly. At this time, the grid import current is below the limit and thus the constraint is satisfied. However, the PV power is decreasing and therefore the grid import is increasing. At 14:30:30, the 50 A limit is exceeded. Only at 14:31:03, the next optimization is completed and the EV power is reduced such that the grid import current is reduced and the constraint is once again met. This phenomenon continues whenever the PV power is decreasing and it leads to the grid import being exceeded for short periods of time very often. Practically speaking, whether or not this is a problem depends on how the 50 A limit is enforced. If, for example, the DSO has placed a 50 A fuse on the node's grid connection, then this fuse will allow the 50 A to be exceeded for a short amount of time (depending on its I^2t rating) without blowing.

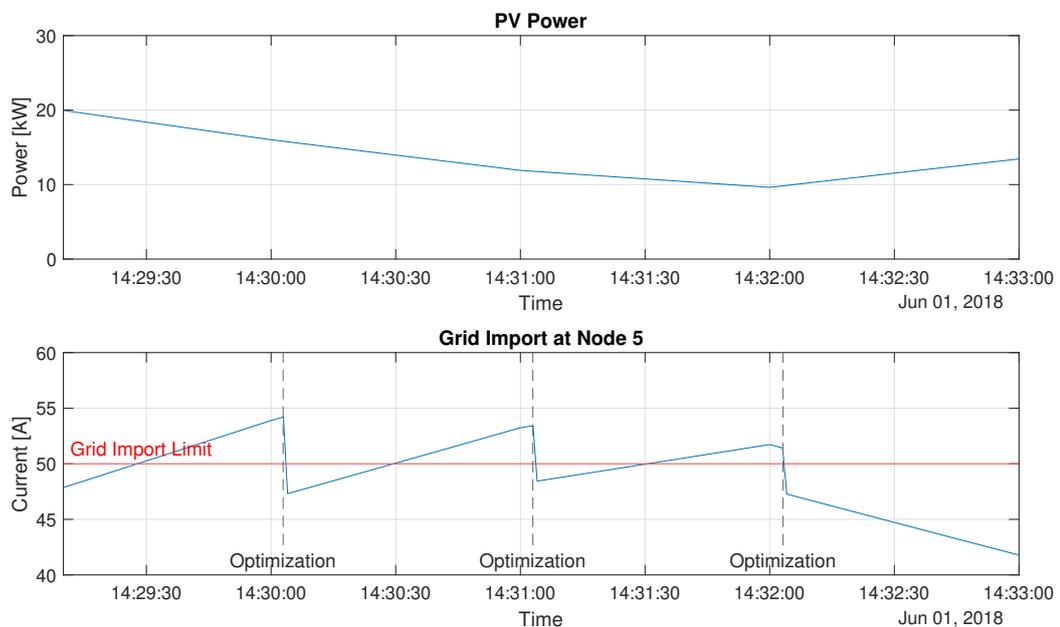


Figure 7.5: Intra-minute exceeding of grid import limit, software simulation

Besides the fluctuations in grid import power and bus voltage, figure 7.2 also shows that the EV charging current setpoints are changed often. The smart charging algorithm runs a new optimization every minute so the setpoints can potentially be modified every minute. While running the PHIL experiments, it was noticed that each time EV1's setpoint is changed from zero to nonzero or vice-versa, the EVSE's relays are opened/closed. As these relays are mechanical components, they can only be cycled a limited number of times before breaking. Therefore, if the smart charging algorithm changes from a zero to non-zero setpoint too often throughout the day, this will in the long term decrease the operational lifetime of the EVSE.

Figure 7.6 shows the calculated and measured bus voltage at node 5. As described in section 3.3, the bus voltage is calculated by the Newton-Raphson power flow analysis and this value is used to set the voltage of the power amplifier's output. The phase voltage of phase 1 is then measured by the EVSE and sent back to the DRTS to be recorded, as shown in the blue line. Figure 7.6 shows that

the measured and calculated voltage are very close to each other, showing that the power amplifier is able to accurately produce waveforms of the desired amplitude. It is noticeable that when EV1 is charging, the measured voltage drops with respect to the calculated voltage. The reason for this is that this voltage is measured inside the EVSE. There is approximately 4 meters of cables between the power amplifier and the EVSE. When EV1 is charging, there is current flowing through these cables. As the cables are not ideal and have some resistance, the current will cause a voltage drop over the cables, so the voltage measured by the EVSE will be lower than the voltage produced by the power amplifier. At a charging current of 16A, the voltage drop was found to be approximately 0.7 V.

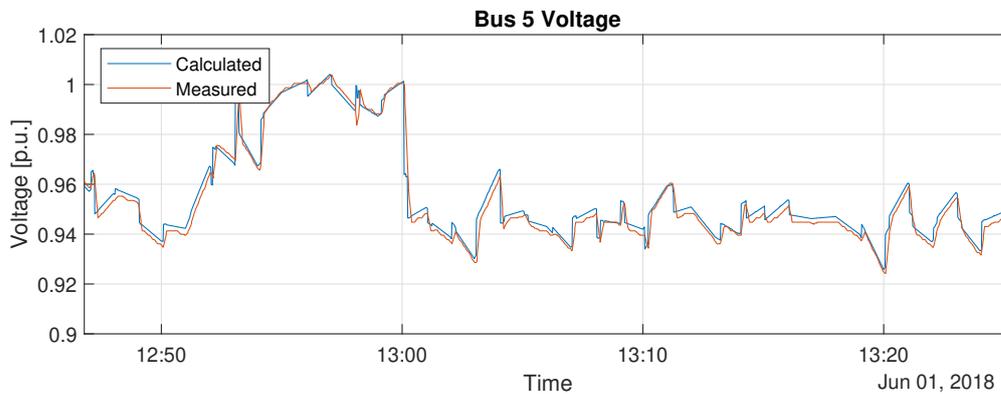


Figure 7.6: Calculated and measured voltage

Figure 7.7 shows the setpoint given by the smart charging algorithm and the current measured by the EVSE. It can be seen that the current drawn is very close (within 0.15 A) to the setpoint. The setpoint needs to be sent from the DRTS to the PLC, then to the EVSE where it is converted to a PWM signal, then sent back to the DRTS to be converted into an analog signal for the load which will then adjust the current. This current is measured by the EVSE and the measurement is sent via the PLC to the DRTS to be recorded. All of this results in a delay of approximately 9 seconds between the setpoint and measured current. Figure 7.7 also shows that, at approximately 15:25, EV1 enters its constant-voltage charging stage and its maximum current is limited is reduced as its SOC continues to further increase.

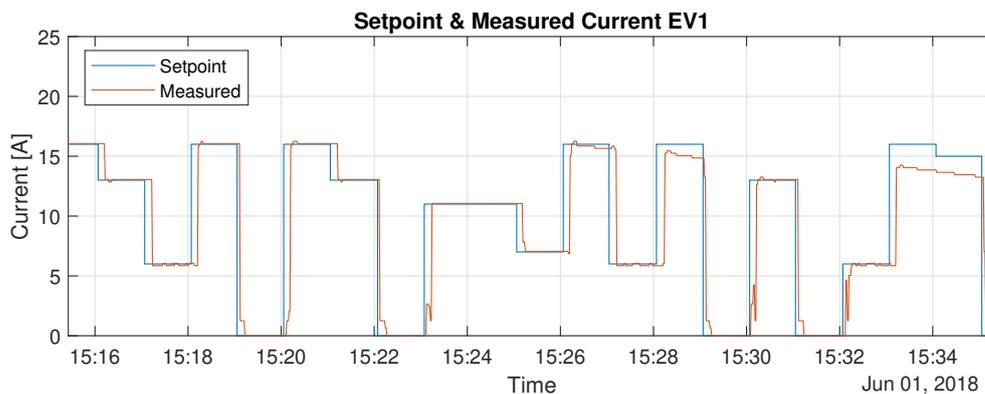


Figure 7.7: Setpoint and measured current

7.2. Case 2: Inaccurate PV Forecast

The scenario used for this case is the same as shown in table 6.1, the only difference being that the PV profile $p_{n,t}^{PV}$ used by the smart charging algorithm is different to the one used by the Simulink model. This simulates the scenario where the smart charging algorithm bases its optimization on a forecast of solar irradiance which is not completely accurate. This is a realistic scenario because weather phenomena can never be predicted with 100% accuracy. Even intra-hour forecasts can have an average root-mean-squared error of up to 30% [64].

To simulate these forecasting inaccuracies, random variations were added to the forecasted PV profile. This was done by multiplying the forecasted PV power by a different random multiplier every hour. The random multipliers were generated using a normal distribution with mean $\mu = 1$ and standard deviation $\sigma = 0.15$. To ensure a fair cost comparison with other cases, the total energy provided by the onsite PV system during the simulated time frame should be the same as in other cases. Since a mean of $\mu = 1$ is used, the Law of Large Numbers indicates that, with an infinitely high number of normally-distributed random multipliers, the total energy should remain unchanged. However, since a finite amount of multipliers (one per hour, so 8) is used, this does not necessarily hold true. This is corrected for with a correction factor, as shown in equation 7.1. The forecasted power is then used by the smart charging algorithm and the actual power is used by the Simulink model. The resulting equation can be seen in equation 7.2. Both powers can be seen in figure 7.8

$$C = \frac{\int p_t^{PV,forecast} dt}{\int (p_t^{PV,forecast} \times N_t(\mu, \sigma^2)) dt} \quad (7.1)$$

$$p_t^{PV,actual} = p_t^{PV,forecast} \times N_t(\mu, \sigma^2) \times C \quad (7.2)$$

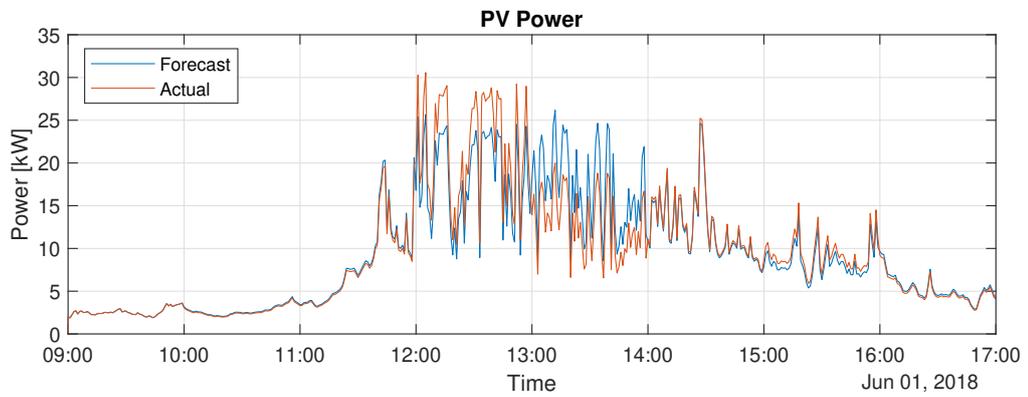


Figure 7.8: Forecast and actual PV power, used in case 2

The results of this scenario can be seen in figure 7.10. The main difference that can be seen with respect to the base case is that between 13:00 and 14:00, the grid import limit is exceeded significantly more in case 2 than in the base case, as can be seen in figure 7.9. This is because, as shown in figure 7.8, the actual PV production is significantly lower than the forecast. The smart charging algorithm does not use measured data and only takes the forecast into account so it cannot correct for this error.

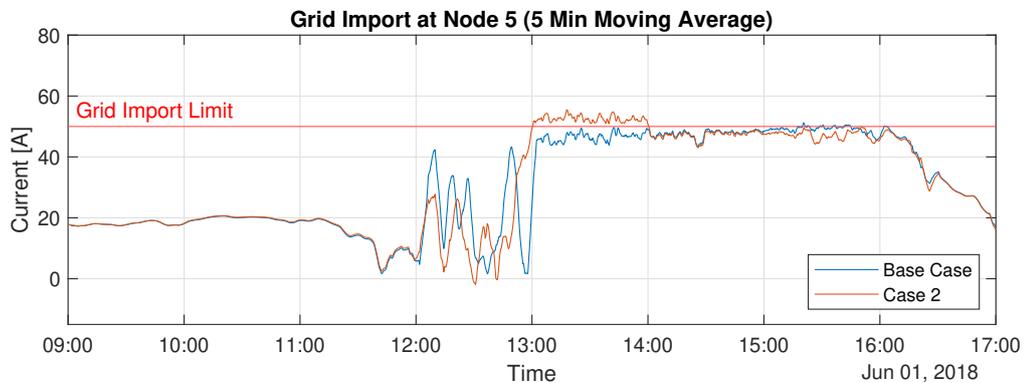


Figure 7.9: Comparison of Grid Import for Case 2 and Base Case. A 5 minute moving average was used to more clearly show the difference.

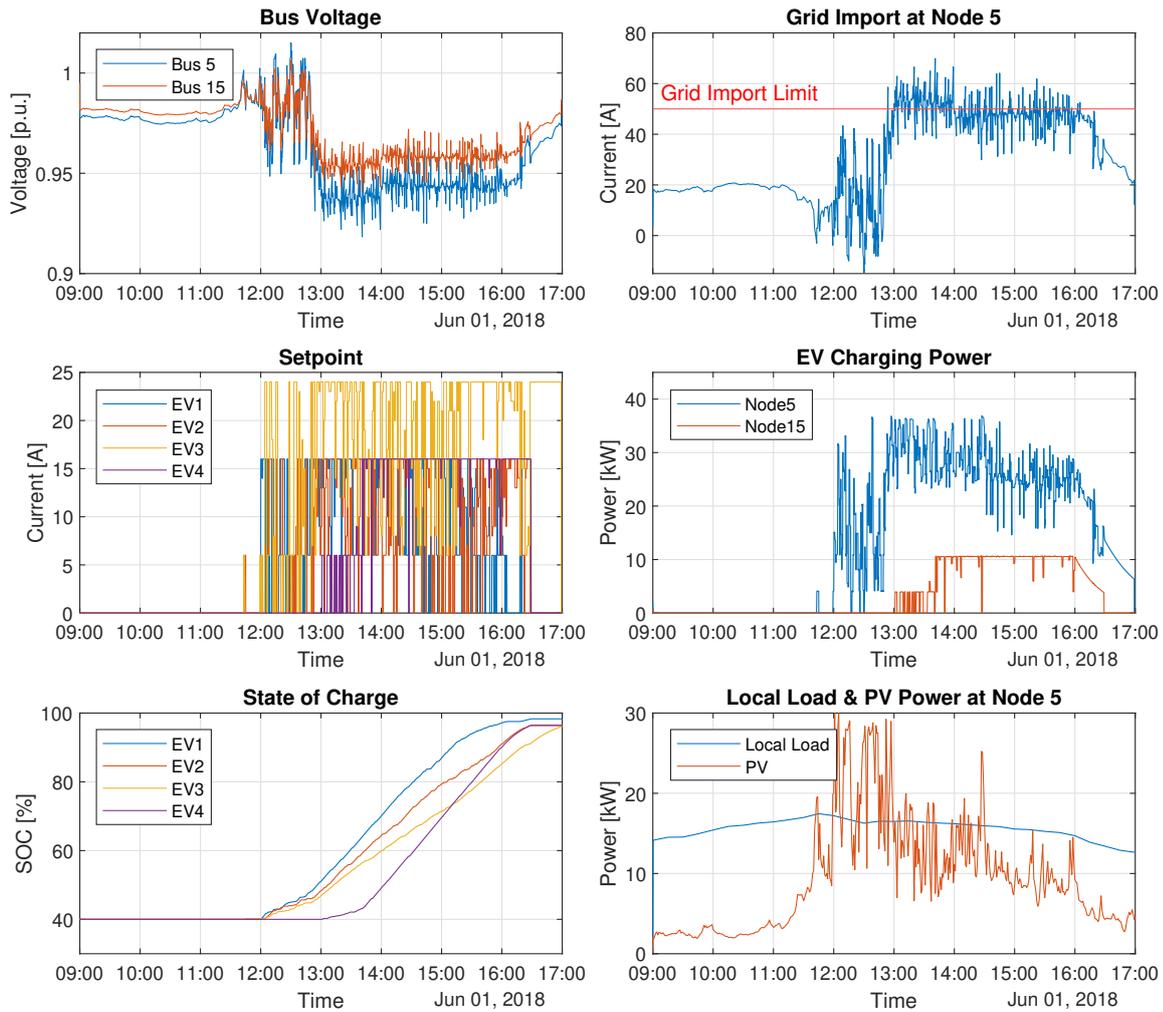


Figure 7.10: Experimental results of case 2

7.3. Case 3: Inaccurate Load Forecast

Similarly to the previous case, the parameters used for this case are the same as shown in table 6.1, with the exception that the local load profile $p_{n,t}^{LL}$ used by the smart charging algorithm is different to the one used by the Simulink model. Like in case 2, this simulates the scenario where the local load forecast used by the smart charging algorithm is not 100% accurate. This is realistic because the local load of a building can not be predicted perfectly as it is influenced by many factors within that building, such as the switching on/off of electric heating or large machinery. To simulate this, the forecasted power is, just like in section 7.2, multiplied by random factors with a normal distribution with mean $\mu = 1$ and standard deviation $\sigma = 0.15$. A correction factor is then applied such that the total load energy in the simulated time frame is equal to the other cases. The forecast and actual power can be seen in figure 7.11.

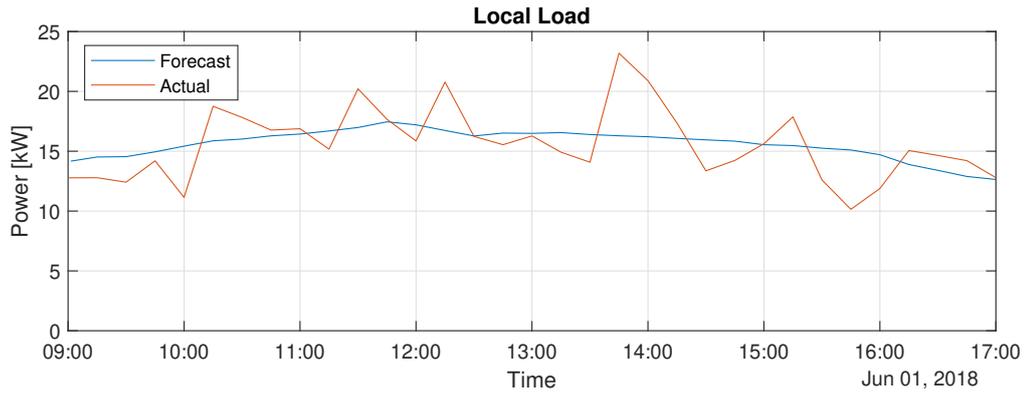


Figure 7.11: Forecast and actual local load power, used in case 3

The results of this case can be seen in figure 7.12. Similar effects are observed as in case 2; when the local load is significantly higher than the forecast (e.g. at 13:45), then the grid import is higher than in the base case and it goes above the limit. This is directly correlated; a 5 kW deviation from the forecast results in a 5 kW deviation in the grid import. The current version of the smart charging algorithm does not compensate for such a deviation at all.

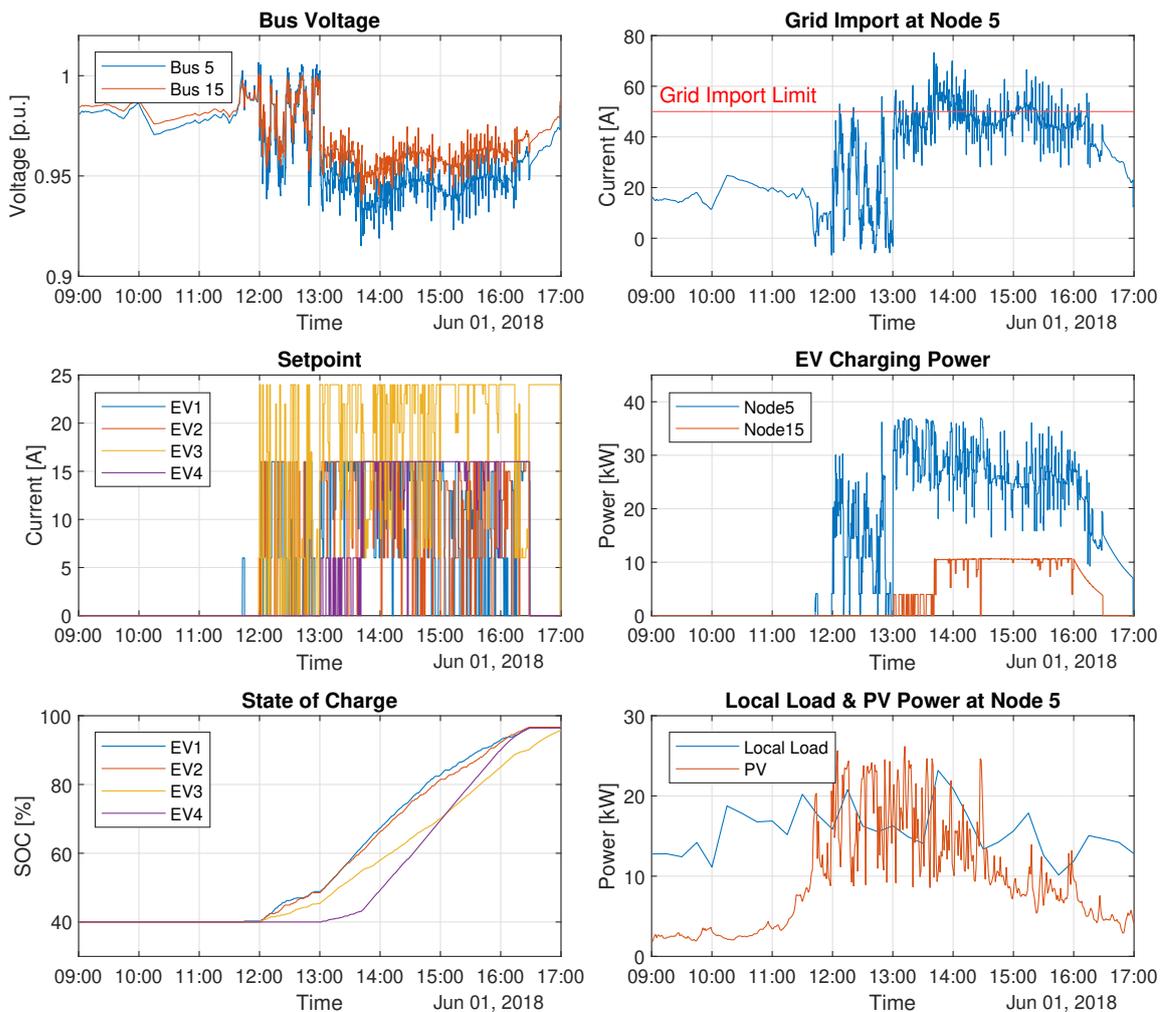


Figure 7.12: Experimental results of case 3

7.4. Case 4: EV Current Lower Than Setpoint

As explained in section 2.1.3, the currently most used standard for communication between EV and EVSE is IEC 61851. This standard allows the EVSE to communicate a current setpoint to the EV's on-board charger with a PWM signal. However, this setpoint is actually a maximum current and the EV is allowed to draw less (but not more) than this current. Therefore, it is not unusual for an EV to draw less current than the given setpoint. Measurements found in literature show that this deviation is usually around 10% [35]. This can be a problem for smart charging if the smart charging algorithm makes the assumption that the EV will always charge at the given setpoint.

To test the effect of this phenomenon on the implemented smart charging algorithm, case 4 has the same parameters as the base case but all EVs charge at a current 10% less than the smart charging setpoint. The results of this case are shown in figure 7.13 and 7.14. The setpoints given by the smart charging algorithm in this case are similar to those of the base case. However, the fact that the EVs charge only at 90% of the setpoint has some negative effects. The most noticeable effect is that the EVs are not charged fully; even less so than in previous cases. EV3 is only charged to 90% SOC instead of the requested 100%. This is because the EVs charge more slowly than anticipated by the algorithm, and since the algorithm starts charging as late as possible to profit from the cheaper energy prices in the afternoon, there is not enough time left to fully charge the EVs.

Another noticeable consequence is that the grid import is slightly less than in the base case. This is again due to the fact that the EVs draw less current than what is expected by the algorithm. This shows a possible area of improvement for the algorithm: if an EV draws less current than expected, the algorithm can increase the setpoint of other EVs without exceeding the grid import limit.

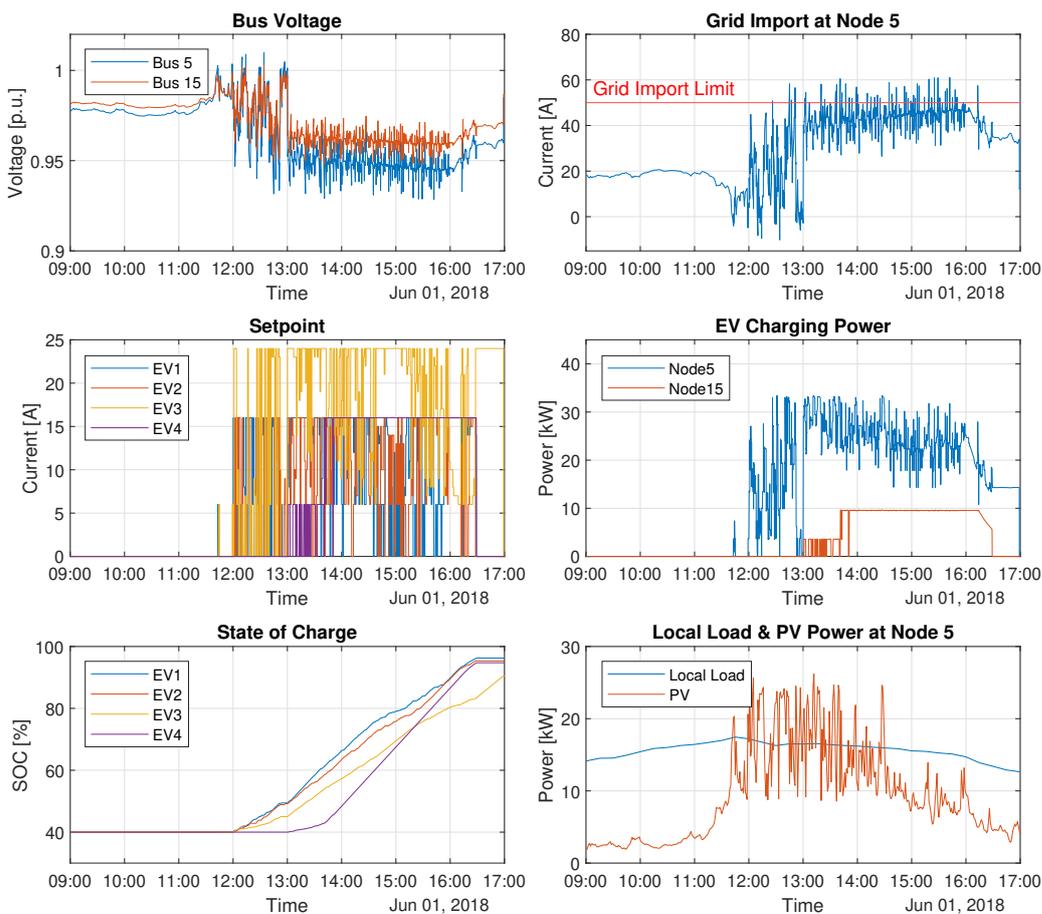


Figure 7.13: Experimental results of case 4

Figure 7.14 shows the setpoint given by the smart charging algorithm and the resulting measured current of EV1. As can be seen, the measured current is 10% less than the setpoint. Note that the reduction in current near the end of the charge is due to the EV entering its constant-voltage charging region, as explained in section 5.5.2.

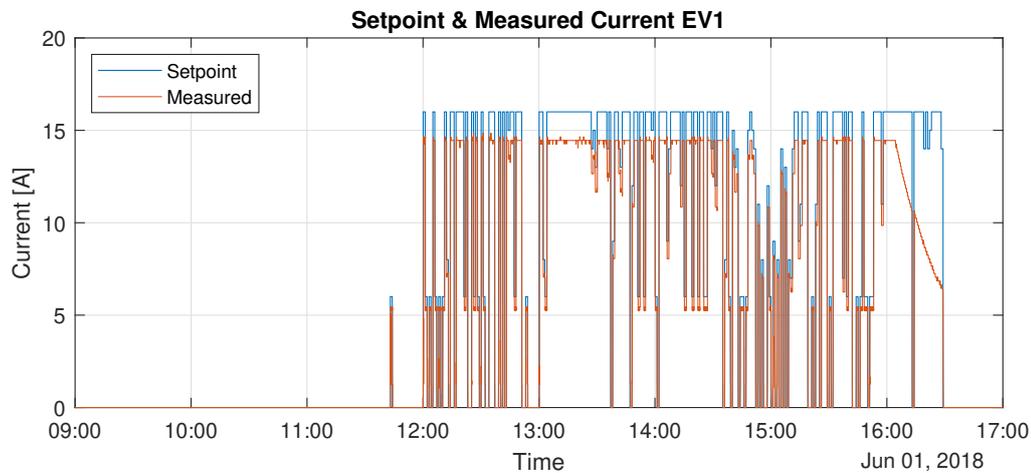


Figure 7.14: Setpoint and measured current of EV1

7.5. Case 5: Battery Capacity Fading

As EVs age, the performance of their batteries degrades, both due to the passing of time ('calendar aging') and battery cycling. A study of 100 EVs estimates that after 10 years, a typical EV battery will lose around 31% of its capacity [65]. This 'capacity fading' phenomenon, in combination with the limited communication between EV and EVSE provided by the IEC 61851 standard, presents another possible challenge for smart charging. As the EV's SOC is not communicated to the EVSE under this standard, it must be estimated based on measurements and the user's input of initial SOC and battery capacity. However, the user may not always know the EV's exact battery capacity as it will be reduced from the factory value due to aging. Therefore, the energy demand $d_{n,j}$ parameter in the objective function (equation 4.1) will be too high.

Case 5 tests the effect of battery capacity fading on the smart charging algorithm's performance. The battery capacity is not actually reduced with respect to the base case because this would result in an unfair comparison with the other cases. Instead, the battery size given to the algorithm is increased to $\frac{50}{0.7} = 71$ kWh and $\frac{100}{0.7} = 143$ kWh for EV 1,2,4 and EV 3 respectively. It might seem counter-intuitive to increase the energy instead of decreasing it, but this essentially achieves the same thing; i.e. a situation where an EV's battery can store 30% less energy due to aging, but the smart charging algorithm still operates under the assumption that the energy stated in the datasheet is correct. All EVs still charge from 40% to 100% SOC but the actual energy needed to do this is less in reality than what the algorithm predicts. The energy that is really needed is still the same as in previous cases, such that a fair comparison can be made.

The results of this simulation are shown in figure 7.15 and 7.16. Noticeable is that, with respect to the base case, the EVs start charging much earlier. This is because the algorithm thinks that more energy needs to be delivered than in reality. As a result, the EVs are also finished charging well before their departure time, unlike in the previous cases. The disadvantage of this is that the charging occurs at sub-optimal times; i.e. the total charging cost is higher because charging happens at a time when energy prices (figure 7.1) are higher than later in the day.

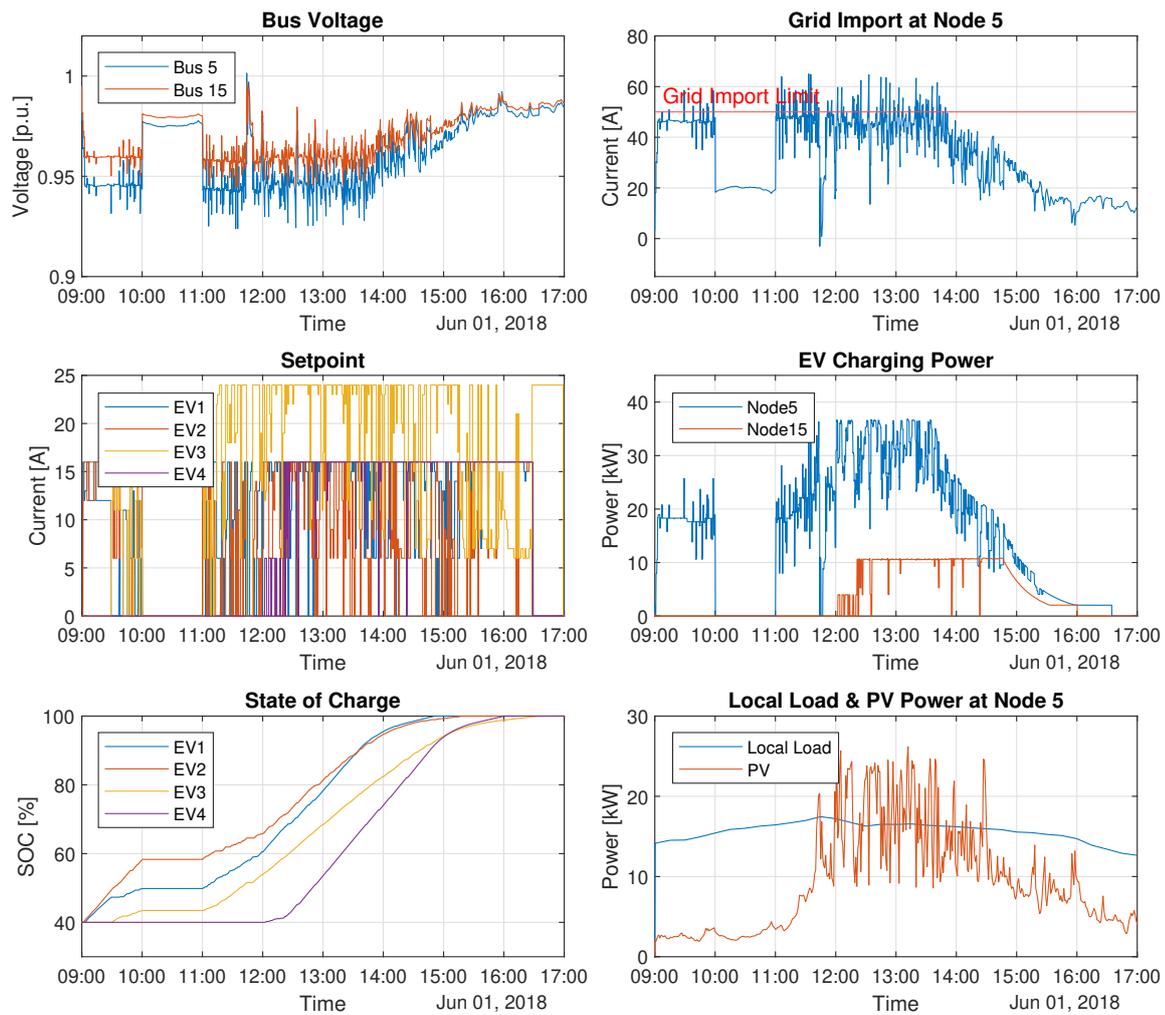


Figure 7.15: Experimental results of case 5

Figure 7.16 provides more insight into the effect of the simulated 30% capacity fade. As the smart charging algorithm thinks that the battery has a higher capacity than it does in reality, this causes a significant error in the estimated SOC. Without the EV being able to communicate its real SOC, the algorithm cannot correct for this effect.

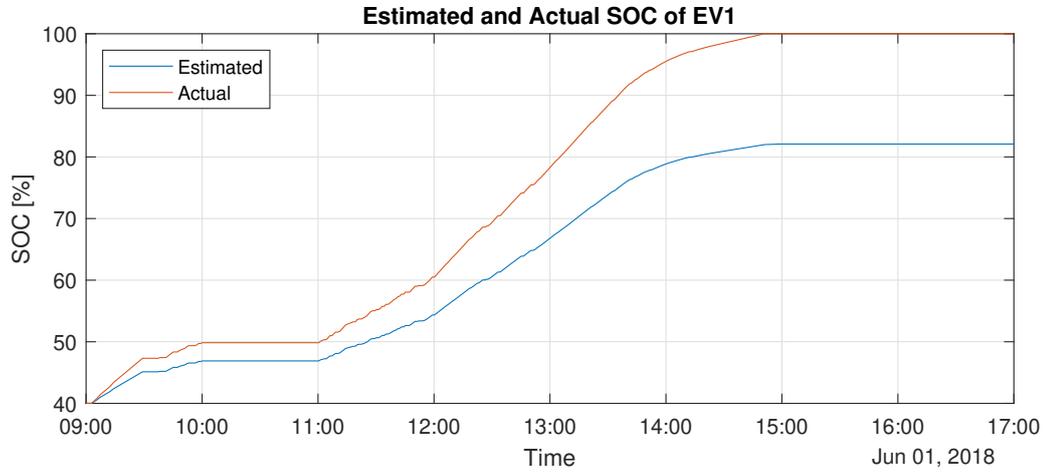


Figure 7.16: Perceived and actual SOC due to capacity fading

7.6. Case 6: Centralized Curtailment

This case has the same parameters as in table 6.1, except that the grid import limit of 50 A has to be shared among node 5 and node 15, i.e. $i_{5,t}^{g(impr)} + i_{15,t}^{g(impr)} \leq 50$ A. To be able to do this, the smart charging algorithm used for this case has some added functionality; it determines the grid import limit of each node based on how much energy its EVs still need. Additionally, there is a lower limit so that EVs can always still charge at their minimum current of 6 A.

$$i_{n,t}^{G+} = \max \left(6 + \frac{p_{n,t}^{LL}}{V_{n,t}}, 50 \times \frac{\sum_{j=1}^J (B_{n,j,T_j^a} + d_{n,j} - B_{n,j,t})}{\sum_{n=1}^N \sum_{j=1}^J (B_{n,j,T_j^a} + d_{n,j} - B_{n,j,t})} \right) \quad (7.3)$$

This simulates the scenario where there is a central entity, for example a DSO, that sets a current limit for each node within the network in order to e.g. reduce upstream grid congestion. Within each node, the charging of EVs can still be optimized, as long as the node's current limit is not exceeded.

The results of this case can be seen in figure 7.17. Different to the previous cases is that the grid import limit has become variable and it is now also lower because the 50 A limit is now not per node but needs to be shared among both nodes. The lower grid import limit leads to a higher minimum bus voltage of 0.93 p.u. as opposed to 0.92 p.u. in the base case. The EVs in this case also start charging significantly earlier than in the other cases. This is because more time will be needed to deliver the required energy to the EVs due to the lower grid import limit. Looking at figure 7.1 it can be seen that buying electricity from the grid is cheaper between 09:00 and 10:00 than between 10:00 and 12:00 so to minimize costs, the EVs are already partially charged from 09:00 to 10:00.

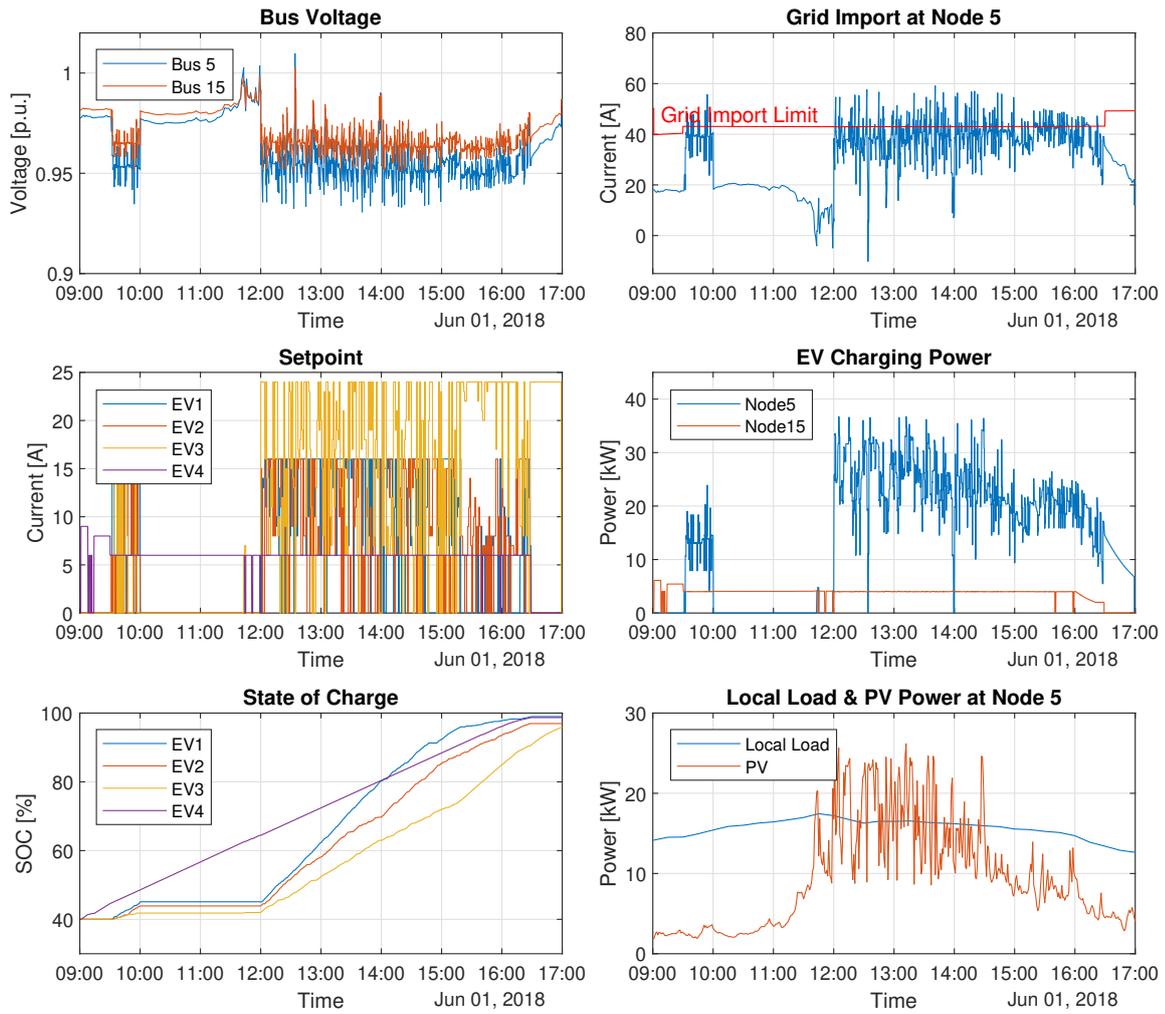


Figure 7.17: Experimental results of case 6

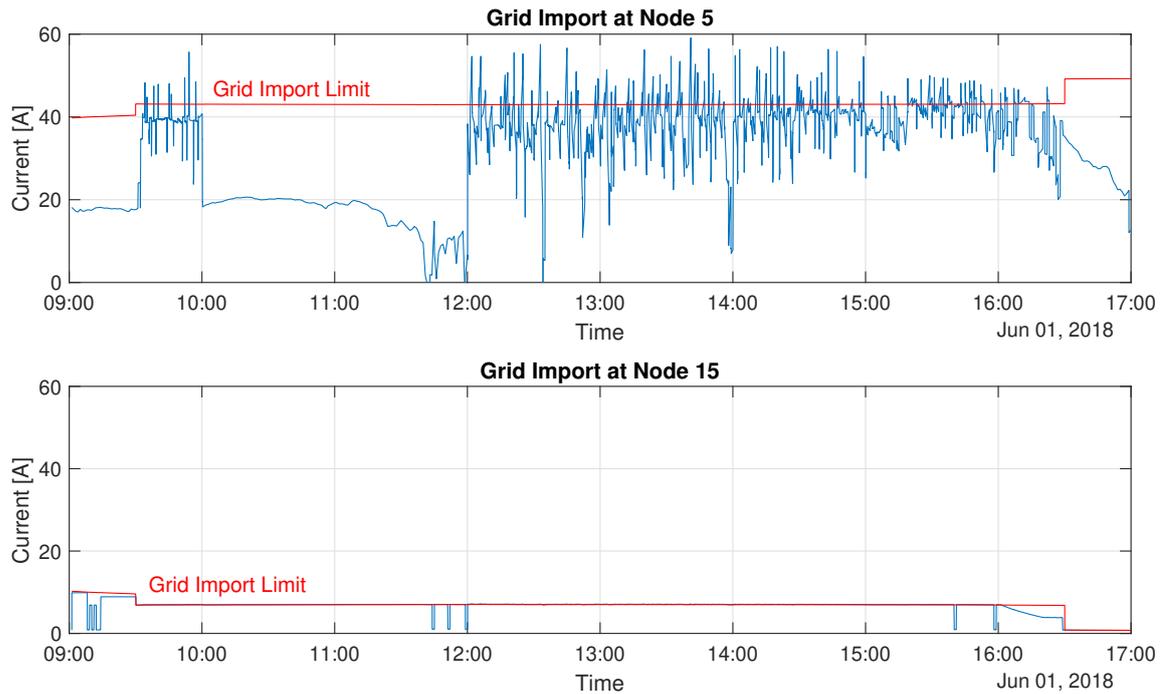


Figure 7.18: Current imported from grid at node 5 & 15, with variable limit

Figure 7.18 shows the current imported from the grid for both node 5 and node 15, as well as their dynamic limits. As can be seen, the limits do not vary much because the energy needed at node 15 is much lower than at node 5 so the grid import limit is at its minimum value of $6 + \frac{p_{n,t}^{LL}}{V_{n,t}}$ (see equation 7.3) most of the time. Node 15's limit is only above this minimum value between 09:00 and 09:30, because at that time EV3 has not arrived yet at node 5 so the energy needed at that node is relatively lower. Because node 15's limit is so low, the EV at that node is charging almost the entire day. There is very little flexibility at this node to minimize costs by time-shifting the charging to a time when energy prices are cheaper. At node 5, the limit is a lot higher so there is more flexibility to time-shift the charging. Node 15 is essentially forced to charge at times when energy prices are high, while node 5 is not. This leads to an unfair division of costs between the nodes. If these nodes are owned by different entities, this type of centralized curtailment could therefore lead to customer dissatisfaction. Since the 50 A limit is shared between the two nodes, the costs of curtailment should also be shared, so node 15 would need to be compensated.

7.7. Case 7: Decentralized Curtailment

Instead of centralized curtailment where a central entity such as a DSO would restrict the maximum current that can be imported from the grid, it is also possible for each node to independently reduce its current, for example as a function of bus voltage. For this case, curtailment based on the graph in figure 7.19 is used. If the bus voltage drops below 0.95 p.u., then the EV charging current will be reduced linearly until it is at its minimum setpoint of 6 A at 0.92 p.u..

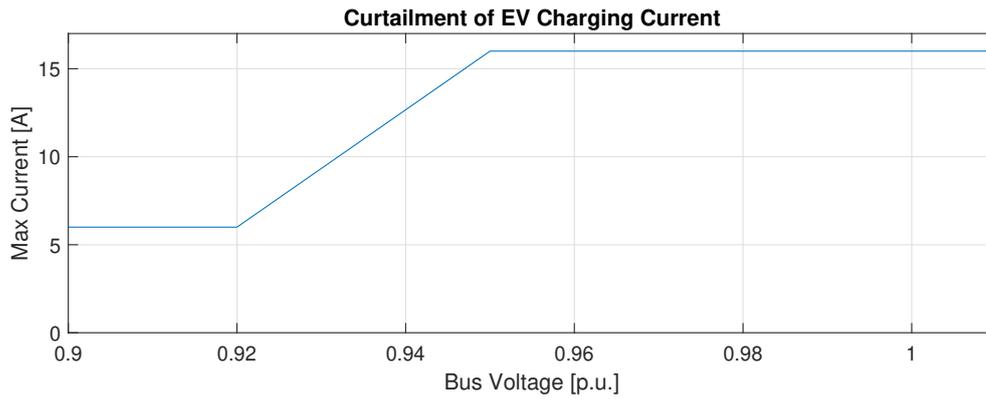


Figure 7.19: Curtailment of EV charging current as a function of bus voltage

The results of this case are shown in figure 7.20. The voltage of bus 15 barely reaches below 0.95 p.u. so EV4 is not curtailed significantly. However, bus 5 does reach lower voltages so this does lead to curtailment of EV1, EV2 and EV3. The decentralized curtailment has a positive effect on the bus voltage of node 5 as the minimum bus voltage is 0.93 p.u. instead of 0.92 p.u. in the base case.

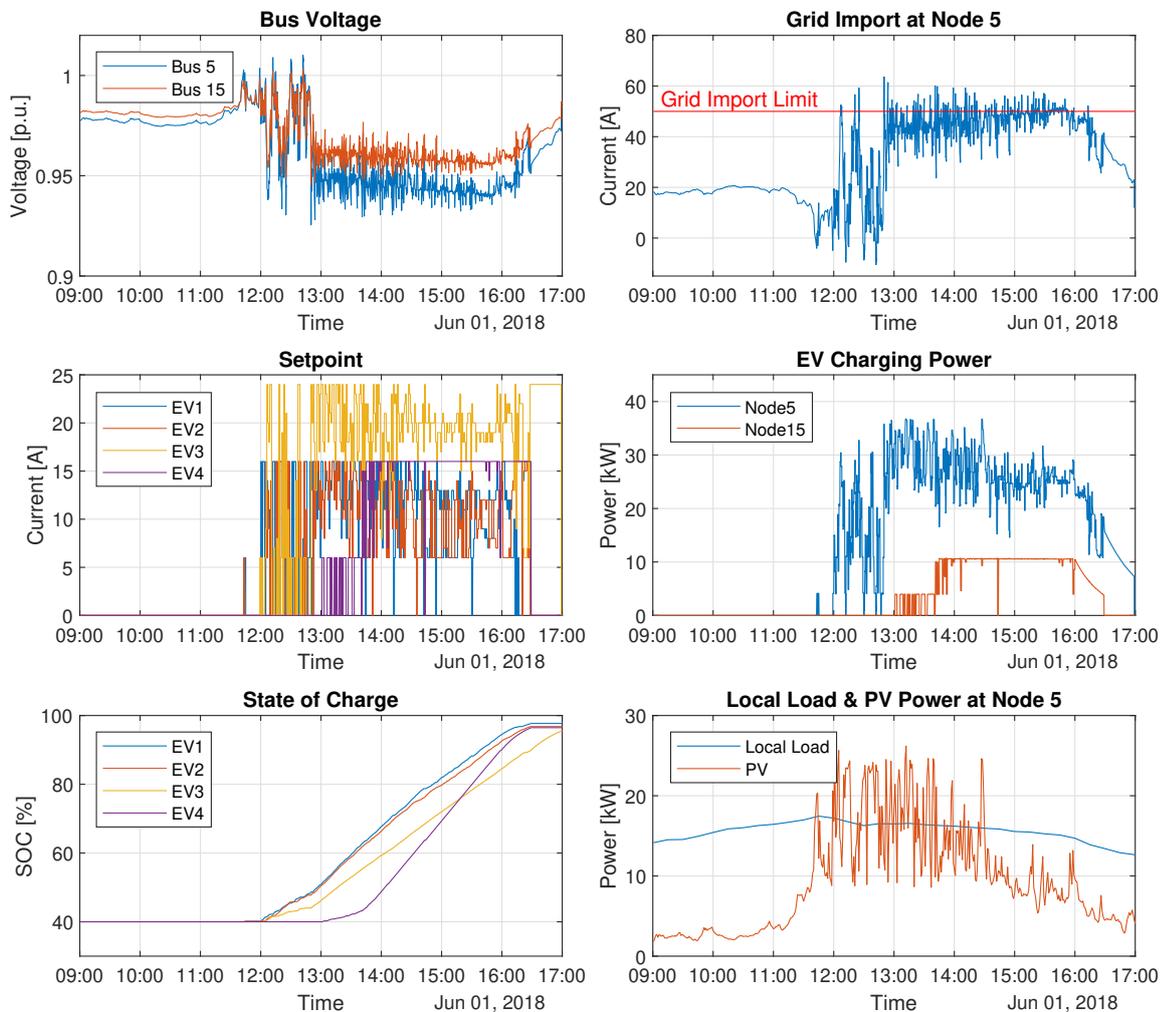


Figure 7.20: Experimental results of case 7

Unfortunately the effect of the decentralized curtailment is limited, and it even has some negative effects on the system as it introduces low frequency voltage oscillations. This can be seen in figure 7.21. At 13:20, an optimization is run and the EV charging power is reduced as the bus voltage is below 0.95 p.u.. As a result of this reduction in EV power, the bus voltage rises again and when the next optimization is run at 13:21, the bus voltage is above 0.95 p.u. so the EV setpoints are increased to their maximum again, which in turn leads to a reduction in bus voltage. This yo-yo effect causes oscillations in the voltage of the system, with a period approximately equal to the optimization timestep of 1 minute.

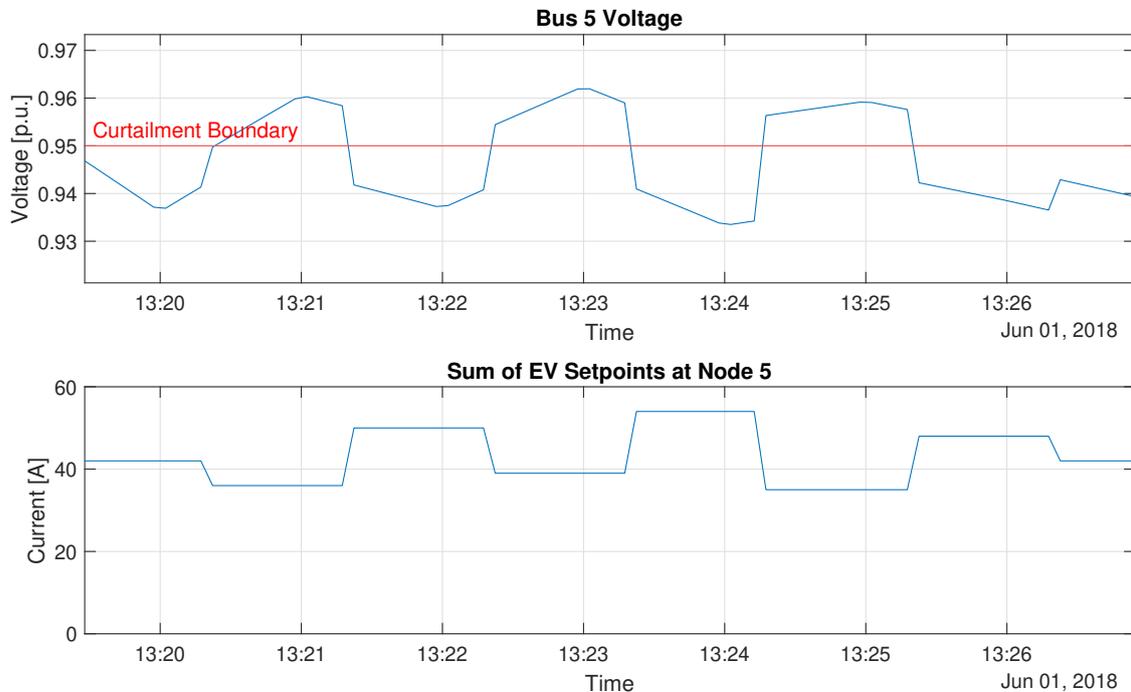


Figure 7.21: Yo-yo effect of decentralized curtailment causes voltage oscillations

Two potential solutions for this issue can be considered. Firstly, the curtailment shown in figure 7.19 could be based on a moving average of the voltage, instead of an instantaneous measurement. Alternatively, hysteresis could be added to the curtailment curve.

7.8. Comparison

Table 7.1 shows a comparison of some key results of the different simulations described in chapters 6 and 7. The total cost represents the total cost of the energy imported from the grid for both nodes, divided by the energy delivered to the EVs. Note that this cost is not only for the charging of the EVs but for the nodes as a whole; i.e. the local PV generation and local load are also taken into account because these are also part of the smart charging objective function as described in chapter 4. Because the load and generated PV energy is equal for all cases, it is possible to compare the costs. The departure SOC is the SOC of each EV at departure time, which ideally should be 100%. The minimum voltage is the lowest voltage reached at bus 5 during the simulation. The bus voltage is also analyzed further in figure 7.22. The peak overload is the maximum percentage by which the grid import constraint was exceeded. Note again that this constraint is not necessarily a limitation of the physical grid but a contractual limit. However, it could be enforced by, for example, a 50 A fuse. Such a fuse would have a certain I^2t rating (or a 'trip curve' in the case of a circuit breaker) and depending on this, it will allow a small overcurrent for a short amount of time without tripping. Therefore, when looking at the overload, it is important to take both the duration and severity of overload into account. So, to give a sense of how long the limit was exceeded, the overload energy is also given. This is the amount of energy that was imported from the grid, above the 50A constraint.

Table 7.1: Comparison of simulation results for different cases

	Total Cost [cent/kWh]	Departure SOC [%]				Min Voltage [p.u.]	Peak Overload [%]	Overload Energy [kWh]
		EV1	EV2	EV3	EV4			
Uncontrolled	10.71	100	100	100	100	0.909	56.3	54.1
1: Base Case	9.14	97	96	96	97	0.923	32.4	2.1
2: Inaccurate PV	9.17	98	96	96	96	0.918	39.8	4.2
3: Inaccurate Load	9.17	97	97	96	96	0.915	46.6	4.4
4: Setpoint - 10%	9.29	96	95	90	95	0.929	22.0	0.5
5: Capacity fade	9.79	100	100	100	100	0.924	30.0	1.5
6: Central Curt.	9.49	99	97	96	99	0.931	18.2	0.3
7: Decentral Curt.	9.15	98	97	95	96	0.926	27.2	1.5

Table 7.2 shows the same results as table 7.1, but as relative changes with respect to uncontrolled charging.

Table 7.2: Relative change w.r.t uncontrolled charging

	Total Cost per kWh	Voltage Deviation	Peak Overload	Overload Energy
Uncontrolled	0.0%	0.0%	0.0%	0.0%
1: Base Case	-14.7%	-15.3%	-42.5%	-96.1%
2: Inaccurate PV	-14.4%	-9.9%	-29.3%	-92.2%
3: Inaccurate Load	-14.4%	-6.6%	-17.2%	-91.9%
4: Setpoint - 10%	-13.3%	-22.0%	-60.9%	-99.1%
5: Capacity fade	-8.6%	-16.5%	-46.7%	-97.2%
6: Central Curt.	-11.4%	-24.2%	-67.6%	-99.4%
7: Decentral Curt.	-14.6%	-18.6%	-51.5%	-97.2%

The uncontrolled case is by far the most expensive. It also exceeds the 50A limit by the highest amount, 56.3%. It exceeds the limit for a long time as the overload energy is far higher than all other cases. This makes sense because the 50A limit is completely ignored during uncontrolled charging; all EVs simply charge at their maximum power regardless of the grid situation. As a result, its minimum bus voltage is also the lowest of all cases.

The base case is clearly an improvement with respect to uncontrolled charging. It results in a 14.7% cost reduction and 15.3% less voltage deviation from nominal. The maximum overload is reduced by 42.5% but, as the overloads are of much shorter duration, the total overload energy is reduced by 96.1%. The downside of this implementation of smart charging is that the EVs are not fully charged to the requested SOC of 100% because the algorithm does not take the EVs' constant-voltage charging regions into account.

The inaccurate PV and load forecasts have no significant influence on the cost. The main influence of the forecast inaccuracies is that the 50A constrain is exceeded more. This leads to a lower minimum bus voltage, higher maximum overload and higher overload energy.

Case 4, where the actual EV current is 10% less than the setpoint, results in the EVs being charged less than in the previous cases. Especially EV3 is only charged to 90% SOC. This could lead to inconvenience to the EV's owner.

The 30% battery capacity fade in case 5 results in the highest cost of all the smart charging cases as the EVs are not charging during the most cost-efficient periods. An unintended but positive effect is that the EVs, unlike in the base case, are fully charged because the charging started much earlier.

Lastly, cases 6 and 7 implement centralized and decentralized curtailment. The centralized curtailment performs well as the minimum bus voltage is the highest of all the cases, and the maximum overload and overload energy are the lowest overall. However, this comes at a cost increase with respect to the base case; cost reduction is reduced by 3.3%. The decentralized curtailment is unfortunately less effective as the improvement with respect to the base case is limited.

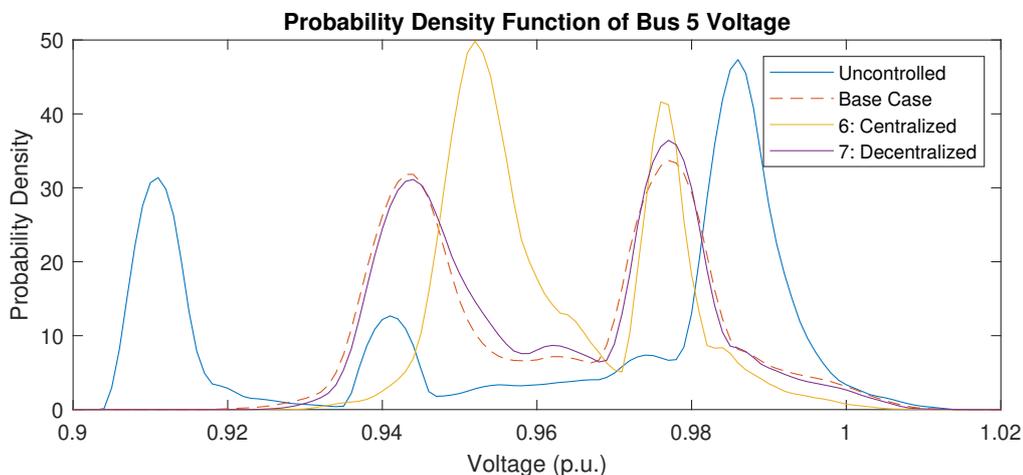
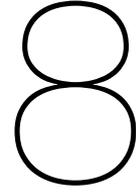


Figure 7.22: Probability density of bus 5 voltage

Figure 7.22 compares the probability density of bus 5 voltages during the simulation period of some key cases. The uncontrolled charging clearly has the most significant impact on the bus voltage with a lower peak at 0.91 p.u. The base case improves this by shifting the lower peak to 0.94 p.u. Cases 2, 3, 4 and 5 have a similar probability density function to the base case and so are not shown for clarity purposes. Decentralized curtailment also does not have a significant impact on voltage deviation compared to the base case. Case 6, with centralized curtailment, improves on the base case by further reducing voltage deviation. The smart charging algorithm with centralized curtailment is therefore the most effective at reducing voltage deviations.



Conclusions & Recommendations

In this thesis project, a Power Hardware-in-the-Loop (PHIL) testbed to experimentally validate the functionality of EV smart charging algorithms was developed. This testbed was then used to evaluate the performance of a given smart charging algorithm with respect to uncontrolled charging for different scenarios. This chapter describes the contributions of the work and concludes the main findings. Based on these findings, recommendations for improvement of the smart charging algorithm are made, as well as recommendations for future work.

8.1. Conclusion

The first objective was to mimic EV charging behaviour at a specific distribution grid node within the high power testbed. This was achieved through an EV-Emulator in combination with an AC load. Communication of the charging current setpoint was done using the IEC 61851 standard as is used in real EVs. The typical constant-voltage charging behaviour of an EV when its battery is nearly fully charged was also incorporated into the simulation as part of the Simulink model.

In addition to the implementation of EV charging behaviour, an emulated distribution grid node was also used. This was achieved with a power amplifier which amplified three sine waves generated by a Digital Real-Time Simulator (DRTS). The amplitude of these sine waves is determined by a Newton-Raphson power flow analysis running on the DRTS. The grid model used for this power flow analysis is based on a real Dutch rural grid with 19 nodes. The node at which the PHIL testbed is located also contains a local load, a 25 kWp PV installation and 2 virtual EVs. Furthermore, the grid also contains two more local loads and another virtual EV at different locations. These are all taken into account in the real-time power flow analysis in order to determine the voltage at the HIL node. The Newton-Raphson power flow analysis of the distribution grid was validated using DiGSILENT PowerFactory which gave positive results. The system is able to accurately emulate a 3-phase grid with variable variable voltage which responds to changes in the power flow in less than a second.

A second objective was to validate the performance of an existing smart charging algorithm in the real-time environment for different scenarios. The Python algorithm was integrated into the real-time simulation using RTLab's API to exchange data between the Python algorithm and the real-time model. Simulations were run for various scenarios. Firstly a base case was run where the smart charging algorithm was used without modifications. It was found that the algorithm generally works satisfactorily and is able to reduce the total costs by 14.2%. However, there are areas of improvement. The EVs were not fully charged to the requested State Of Charge (SOC) of 100%, which means a penalty was incurred. Additionally, the 50 A constraint on grid import that was set was repeatedly exceeded. Recommendations for solving this in the future are stated in section [8.2](#).

Other scenarios were also considered. Inaccuracies in the local load and solar irradiance forecasts were not corrected by the algorithm and therefore caused the grid import constraint to be exceeded even more than during the base case. The algorithm's assumption that EVs always charge at their

setpoint caused issues in a scenario where the current was set to 10% less than the setpoint as one EV was only charged to 90% instead of the requested 100%. A scenario with battery capacity fading due to aging significantly reduced the performance of the smart charging algorithm and reduced cost savings to 9% instead of 14%.

The last objective was to compare the effect of uncontrolled and controlled charging on the distribution grid in terms of voltage deviation and grid connection overload. With respect to uncontrolled charging, it was shown that the implemented controlled charging method reduced the voltage drop by 15% and the maximum overload by 42%. Scenarios with both centralized and decentralized curtailment were also simulated to further reduce this. The centralized curtailment was most effective as it was able to further reduce the voltage drop by 9% and maximum overload by 26%. However, this came at a slightly increased cost with respect to the base case.

The usage of a hardware-in-the-loop testbed instead of a purely software-based simulation has offered several benefits for this project. Firstly, it has brought to light some practical problems that may otherwise not have been noticed. One of these is the fact that, when the algorithm sends a setpoint of 0A, the EVSE opens its relays to physically disconnect the power to the EV. This happens relatively often (e.g. 35 times in the base case) and these relays have a limited amount of cycles before failure. In the long-term, this may cause premature failure of these components within the EVSE. The HIL testbed also brought to light the fact that communication delays cause the grid import constraint to be exceeded for brief periods. Lastly, the HIL aspect of the testbed becomes even more valuable as the project will be further developed in the future. When the smart charging algorithm reaches a more mature stage of development, its integration into the HIL testbed can serve as a very valuable proof of concept to external partners. There is also the possibility of, with some upgrades, using the testbed to charge real EVs. This could provide very valuable real-world data on the charging behaviour of these EVs which could be used to improve mathematical models.

8.2. Recommendations for Improvement of Smart Charging Algorithm

The smart charging algorithm used in this project is currently still in an early stage of its development. Based on observations made about the algorithm's behaviour throughout this project, the author offers some recommendations for solving potential issues and improving future versions of the algorithm.

- **Include the effect of the constant-voltage (CV) charging region in the SOC prediction**
The current version of the algorithm assumes that the EV will always charge at the setpoint it is given. Based on this, it makes predictions of how the SOC will increase over time and therefore which charging profile is needed to fulfill the requested energy demand. However, when an EV's battery is almost full, it enters the CV charging region and its maximum charging current decreases with increasing SOC, so it is no longer able to follow the setpoint. Taking this into account on the SOC prediction would allow the algorithm to allocate enough charging time to fully charge the EV.
- **Trigger a new optimization whenever a constraint is violated**
In section 7.1 it was shown that the smart charging algorithm only checks if constraints are met at every full minute, which means that there are intra-minute violations of the constraints. A separate process could be used to trigger a new optimization whenever a violation is detected.
- **Take communication delays into account when changing setpoints**
In section 7.1 it was shown that communication delays can cause short spikes in a node's power consumption when changing EVs' charging current setpoints. This is detrimental to the distribution grid as it causes voltage fluctuations. While it is not possible to avoid this completely, this effect could be reduced by predicting the communication delay and taking this into account so that EVs change their setpoint at the same time.
- **Reduce the frequency of setpoint changes**
Currently setpoint changes are made almost every minute, but this could be significantly reduced.

If setpoint changes are made less frequently, this would also reduce the amount of spikes in power consumption described above. Additionally, when a setpoint is changed from zero to non-zero or vice-versa, the EVSE's relays are physically switched to (dis)connect the power to the EV. Reducing the frequency of these events would reduce wear on these components.

- **Use measurements of the local load power consumption and PV generation**

Load and solar irradiance forecasts will never be completely accurate, yet the current version of the algorithm relies heavily on them. It was shown in 7.2 and 7.3 that inaccuracies in the forecasts can cause system constraints to be exceeded. By using measurements in combination with the forecasts, the algorithm could adjust charging profiles when there is a large difference between the measurements and forecasts. This way, it can be ensured that all constraints are met.

- **Use machine learning to predict a specific EV's charging behavior**

In addition to the CV charging region mentioned above, there are also other reasons that an EV's charging behaviour may be different than expected. It was shown in sections 7.4 and 7.5 that battery degradation and discrepancies between the setpoint and actual charging current can cause negative effects on the charging performance. The extent to which these effects are present is dependent on many factors and unique to each EV, making it challenging to predict. However, if a certain EV has previously been charged using the smart charging algorithm, these measurements could be analyzed and used for future predictions of this specific EV's charging behaviour. This way, the algorithm will be able to compensate.

- **Use a moving average voltage measurement or hysteresis for decentralized curtailment**

In section 7.7 it was shown that the current implementation of decentralized curtailment causes a sort of yo-yo effect which in turn introduces low-frequency voltage oscillations and reduces the efficacy of this method of curtailment. This effect could be significantly reduced by basing the curtailment on a moving average of the measured bus voltage, or adding hysteresis to the curtailment.

8.3. Recommendations for Future Work

The PHIL testbed presented in this thesis report is a first version and will hopefully be further improved in the future. Some recommendations for future work are offered which were not implemented in this project due to practical and time constraints.

- **Implement changes to smart charging algorithm**

The smart charging algorithm which was implemented in this testbed is still under development and new features are still being added. Section 8.2 also offers some suggestions on possible improvements for this algorithm. Once the algorithm is further improved, this new version can be implemented and the experiments rerun.

- **Investigate the effect of more EVs and different grids**

This project used a relatively simple distribution grid with only 19 nodes and 4 EVs. In the future, it could be interesting to use the testbed to investigate the effect of uncontrolled and controlled charging on different types of grids. Additionally, the effect of greater EV penetration could also be included.

- **Investigate the effect of changing optimization parameters**

In table 6.1, a series of parameters was presented which formed the basis for the different controlled and uncontrolled charging scenarios. It would be beneficial to rerun the simulations with different parameters to see how the results are affected by the parameters. For example, a fixed electricity selling price was used in the simulations so it could be investigated what the effect would be of a variable selling price. This may result in more PV energy being sold to the grid at times of high selling prices.

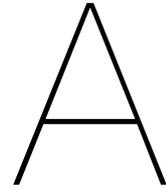
- **Implement communication over OCPP**

In the proposed testbed, the communication between the smart charging algorithm and the EVSE happens through the Modbus TCP protocol over the lab's local Ethernet network. However, this is not completely realistic as commercial EVSEs usually use the Open Charge Point Protocol for

communication. To make the testbed behaviour closer to reality, OCPP communication could be implemented instead of Modbus TCP.

- **Use real EVs**

The main strength of HIL simulations is that the benefits of software simulation can be combined with the actual response of real hardware. In this project, a real EVSE is used but the EV is emulated using a controllable load. Care has been taken to ensure that the load is controlled in such a way that its behaviour closely resembles an EV's charging behaviour. However, inevitably there are still some simplifications involved and the behaviour of a real EV is likely not 100% matched. With some small modifications (a more powerful power amplifier), this same testbed could be used to charge real EVs. By acquiring data on the charging behaviour of real EVs, mathematical EV models can be made more accurate and the testbed can be validated.



Appendix A: EVSE Modbus Mapping Table

The following pages are extracted from [54]. This document is produced by Alfen N.V. and accompanies the Alfen EVE Single Pro-Line EVSE. The tables show the different register locations of all variables that can be accessed using the Modbus protocol.

3 Modbus Register table

3.1 Product identification registers

The product identification registers can be reached using slave address 200.

Description	Start address	End address	Number of 16 bit registers	Read or Write	Data Type	Step size & Units	Additional info
Name	100	116	17	R	STRING	n.a.	"ALF_1000"
Manufacturer	117	121	5	R	STRING	n.a.	"Alfen NV"
Modbus table version	122	122	1	R	SIGNED16	n.a.	1
Firmware version	123	139	17	R	STRING	n.a.	"3.4.0-2990"
Platform type	140	156	17	R	STRING	n.a.	"NG910"
Station serial number	157	167	11	R	STRING	n.a.	"0000R000"
Date year	168	168	1	R	SIGNED16	1yr	2019
Date month	169	169	1	R	SIGNED16	1mon	03
Date day	170	170	1	R	SIGNED16	1d	11
Time hour	171	171	1	R	SIGNED16	1hr	12
Time minute	172	172	1	R	SIGNED16	1min	01
Time second	173	173	1	R	SIGNED16	1s	04
Uptime	174	177	4	R	UNSIGNED6 4	0.001s	100
Time zone	178	178	1	R	SIGNED16	1min	Time zone offset to UTC in minutes

3.2 Station status registers

The station status registers can be reached using slave address 200.

Description	Start address	End address	Number of 16 bit registers	Read or Write	Data Type	step size & Units	Additional info
Station Active Max Current	1100	1101	2	R	FLOAT32	1A	The actual max current
Temperature	1102	1103	2	R	FLOAT32	1°C	Board temperature, does not reflect environment temperature
OCPP state	1104	1104	1	R	UNSIGNED16	N.A.	To verify whether back office is connected
Nr of sockets	1105	1105	1	R	UNSIGNED16	N.A.	Number of sockets

3.3 SCN registers

The SCN registers can be reached using slave address 200.

Description	Start address	End address	Number of 16 bit registers	Read or Write	Data Type	step size & Units	Additional info
SCN name	1400	1403	4	R	STRING	n.a.	
SCN Sockets	1404	1404	1	R	UNSIGNED16	1A	Number of configured sockets
SCN Total Consumption Phase L1	1405	1406	2	R	FLOAT32	1A	
SCN Total Consumption Phase L2	1407	1408	2	R	FLOAT32	1A	
SCN Total Consumption Phase L3	1409	1410	2	R	FLOAT32	1A	
SCN Actual Max Current Phase L1	1411	1412	2	R	FLOAT32	1A	
SCN Actual Max Current Phase L2	1413	1414	2	R	FLOAT32	1A	
SCN Actual Max Current Phase L3	1415	1416	2	R	FLOAT32	1A	
SCN Max Current per Phase L1	1417	1418	2	R/W	FLOAT32	1A	
SCN Max Current per Phase L2	1419	1420	2	R/W	FLOAT32	1A	
SCN Max Current per Phase L3	1421	1422	2	R/W	FLOAT32	1A	
Remaining valid time Max Current Phase L1	1423	1424	2	R	UNSIGNED32	1s	Max current valid time
Remaining valid time Max Current Phase L2	1425	1426	2	R	UNSIGNED32	1s	Max current valid time
Remaining valid time Max Current Phase L3	1427	1428	2	R	UNSIGNED32	1s	Max current valid time
SCN Safe current	1429	1430	2	R	FLOAT32	1A	Configured SCN safe current
SCN Modbus Slave Max Current enable	1431	1431	1	R	UNSIGNED16	n.a.	1:Enabled, 0: Disabled.

3.4 Socket measurement registers

The socket measurements show information regarding the energy meter that is connected to the only socket in case of a single socket charging station, or the left socket in case of a dual socket charging station and can be reached using slave address 1. In case of a dual socket station, the right socket related energy measurements can be reached using slave address 2.

Description	Start address	End address	Number of 16 bit registers	Read or Write	Data Type	Step size & Units	Additional info
Energy measurements							
Meter state	300	300	1	R	UNSIGNED16	n.a.	Bitmask with state: Initialised: 0x01 Updated: 0x02 Warning: 0x04 Error: 0x08
Meter last value timestamp	301	304	4	R	UNSIGNED64	0.001s	Milliseconds since last received measurement
Meter type	305	305	1	R	UNSIGNED16	n.a.	0:RTU, 1:TCP/IP, 2:UDP, 3:P1, 4:other
Voltage Phase V(L1-N)	306	307	2	R	FLOAT32	1V	
Voltage Phase V(L2-N)	308	309	2	R	FLOAT32	1V	
Voltage Phase V(L3-N)	310	311	2	R	FLOAT32	1V	
Voltage Phase V(L1-L2)	312	313	2	R	FLOAT32	1V	
Voltage Phase V(L2-L3)	314	315	2	R	FLOAT32	1V	
Voltage Phase V(L3-L1)	316	317	2	R	FLOAT32	1V	
Current N	318	319	2	R	FLOAT32	1A	
Current Phase L1	320	321	2	R	FLOAT32	1A	
Current Phase L2	322	323	2	R	FLOAT32	1A	
Current Phase L3	324	325	2	R	FLOAT32	1A	
Current Sum	326	327	2	R	FLOAT32	1A	
Power Factor Phase L1	328	329	2	R	FLOAT32	N.A.	
Power Factor Phase L2	330	331	2	R	FLOAT32	N.A.	
Power Factor Phase L3	332	333	2	R	FLOAT32	N.A.	
Power Factor Sum	334	335	2	R	FLOAT32	N.A.	
Frequency	336	337	2	R	FLOAT32	1Hz	
Real Power Phase L1	338	339	2	R	FLOAT32	1W	
Real Power Phase L2	340	341	2	R	FLOAT32	1W	
Real Power Phase L3	342	343	2	R	FLOAT32	1W	
Real Power Sum	344	345	2	R	FLOAT32	1W	
Apparent Power Phase L1	346	347	2	R	FLOAT32	1VA	
Apparent Power Phase L2	348	349	2	R	FLOAT32	1VA	
Apparent Power Phase L3	350	351	2	R	FLOAT32	1VA	
Apparent Power Sum	352	353	2	R	FLOAT32	1VA	

Reactive Power Phase L1	354	355	2	R	FLOAT32	1VAr	
Reactive Power Phase L2	356	357	2	R	FLOAT32	1VAr	
Reactive Power Phase L3	358	359	2	R	FLOAT32	1VAr	
Reactive Power Sum	360	361	2	R	FLOAT32	1VAr	
Real Energy Delivered Phase L1	362	365	4	R	FLOAT64	1Wh	
Real Energy Delivered Phase L2	366	369	4	R	FLOAT64	1Wh	
Real Energy Delivered Phase L3	370	373	4	R	FLOAT64	1Wh	
Real Energy Delivered Sum	374	377	4	R	FLOAT64	1Wh	
Real Energy Consumed Phase L1	378	381	4	R	FLOAT64	1Wh	
Real Energy Consumed Phase L2	382	385	4	R	FLOAT64	1Wh	
Real Energy Consumed Phase L3	386	389	4	R	FLOAT64	1Wh	
Real Energy Consumed Sum	390	393	4	R	FLOAT64	1Wh	
Apparent Energy Phase L1	394	397	4	R	FLOAT64	1VAh	
Apparent Energy Phase L2	398	401	4	R	FLOAT64	1VAh	
Apparent Energy Phase L3	402	405	4	R	FLOAT64	1VAh	
Apparent Energy Sum	406	409	4	R	FLOAT64	1VAh	
Reactive Energy Phase L1	410	413	4	R	FLOAT64	1VArh	
Reactive Energy Phase L2	414	417	4	R	FLOAT64	1VArh	
Reactive Energy Phase L3	418	421	4	R	FLOAT64	1VArh	
Reactive Energy Sum	422	425	4	R	FLOAT64	1VArh	
Status and transaction registers							
Availability	1200	1200	1	R	UNSIGNED16	n.a.	1: Operative, 0: inoperative
Mode 3 state	1201	1205	5	R	STRING	n.a.	61851 states
Actual Applied Max Current	1206	1207	2	R	FLOAT32	1A	Actual Applied overall Max Current for socket
Modbus Slave Max Current valid time	1208	1209	2	R	UNSIGNED32	1s	Remaining time before fall back to safe current
Modbus Slave Max Current	1210	1211	2	R/W	FLOAT32	1A	
Active Load Balancing Safe Current	1212	1213	2	R	FLOAT32	1A	Active Load Balancing safe current
Modbus Slave received setpoint accounted for	1214	1214	1	R	UNSIGNED16	n.a.	1:Yes, 0: No
Charge using 1 or 3 phases	1215	1215	1	R/W	UNSIGNED16	phases	1: 1 phase, 3: 3 phase charging

Note: Register 1214 'Modbus Slave received setpoint accounted for' indicates whether the received Max Current (registers 1210-1211), also called a setpoint, is taken into account to determine the Actual

B

Appendix B: Simulink Model

This appendix includes screenshots of all subsystems in the Simulink model which runs in real-time on the DRTS. Figure [B.1](#) shows the top-level layout of the three subsystems; Master, Slave and User Interface. Figures [B.2](#), [B.3](#), [B.4](#), [B.5](#), and [B.6](#) show the Master subsystem and all its subsystems. Figures [B.7](#), [B.8](#), [B.9](#), and [B.10](#) show the Slave subsystems and all its subsystems. Lastly, figure [B.11](#) shows the User Interface (console).

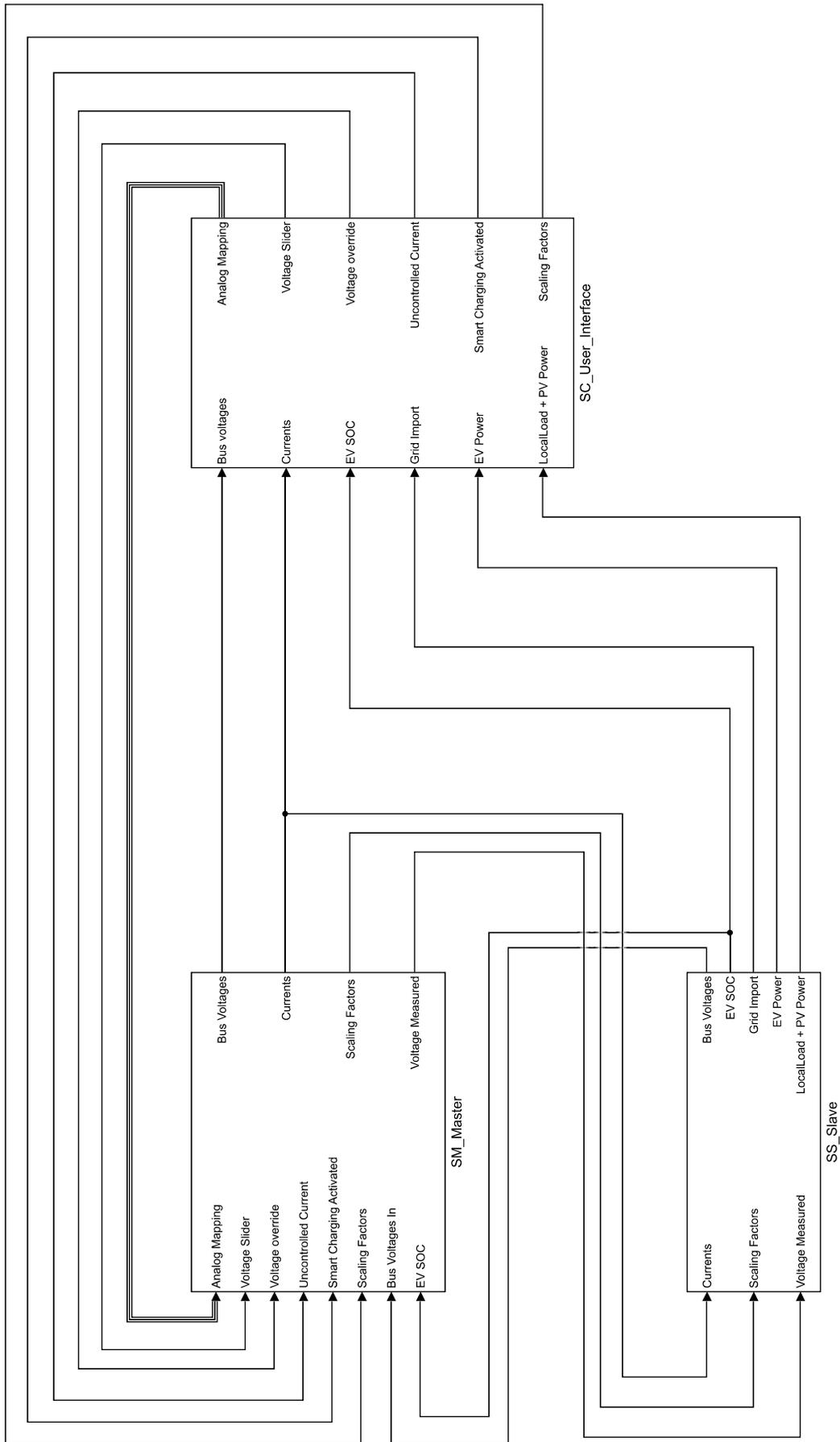


Figure B.1: Top-Level overview of Simulink model

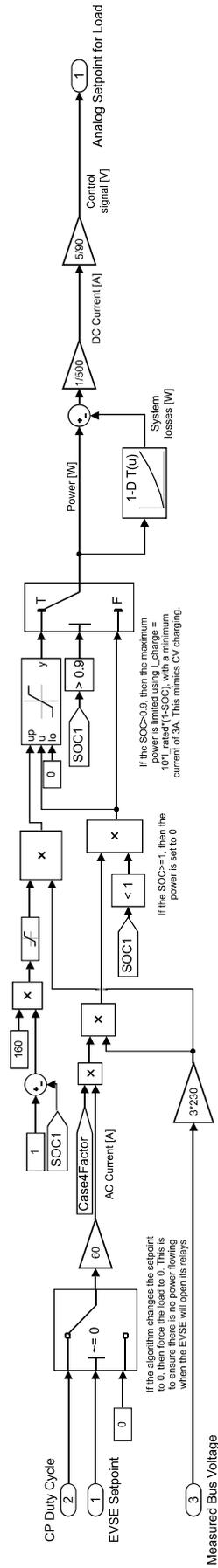


Figure B.3: Subsystem within Master which converts the Control Pilot duty cycle to an analog setpoint for the AC load.

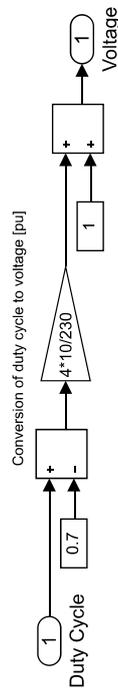


Figure B.4: Subsystem within Master which converts a PWM duty cycle received from the PLC into the measured phase voltage

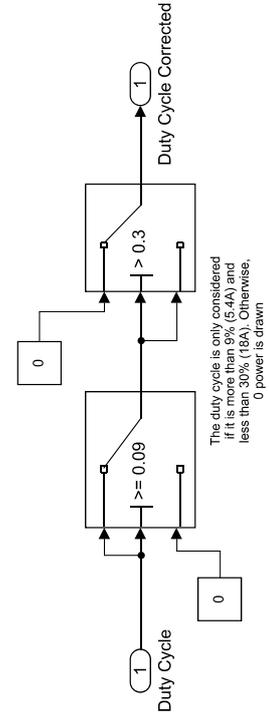


Figure B.5: Subsystem within Master which performs sanity checks on the incoming duty cycle of the control pilot signal. For safety reasons, if the received signal is not within expected bounds, the AC load will not draw any power.

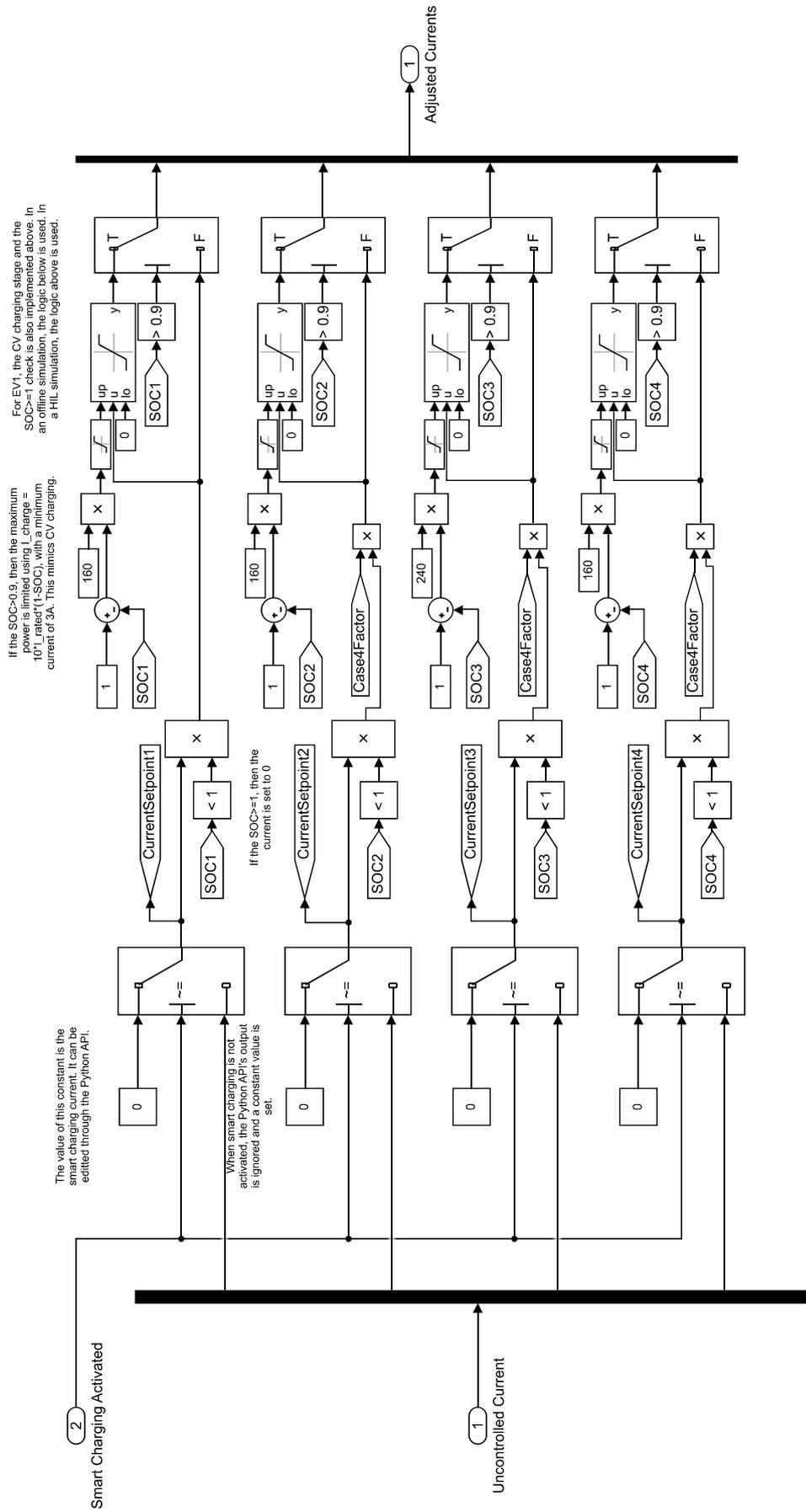


Figure B.6: Subsystem within Master which implements EV charging behaviour; it stops the charging when SOC≥1 and implements constant-voltage charging.

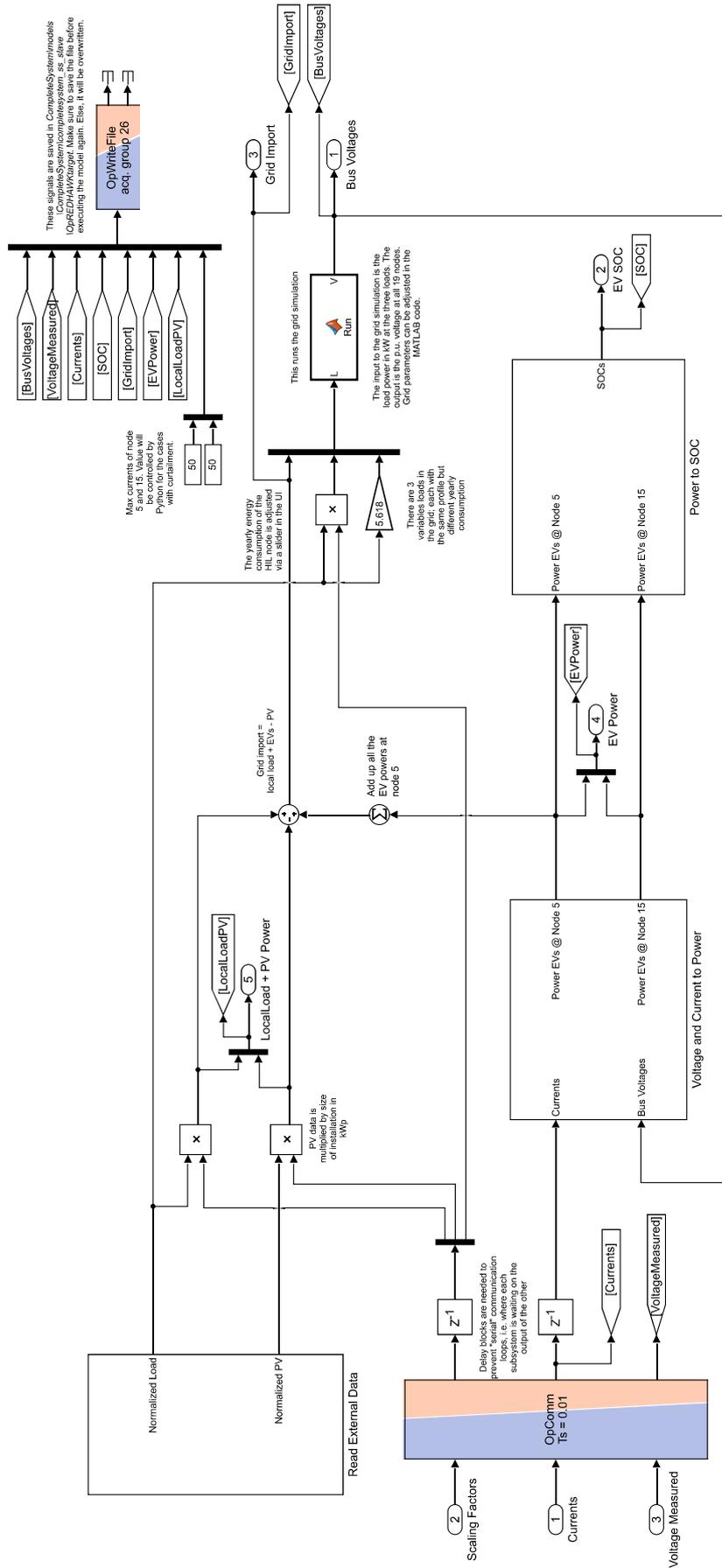


Figure B.7: Slave Subsystem

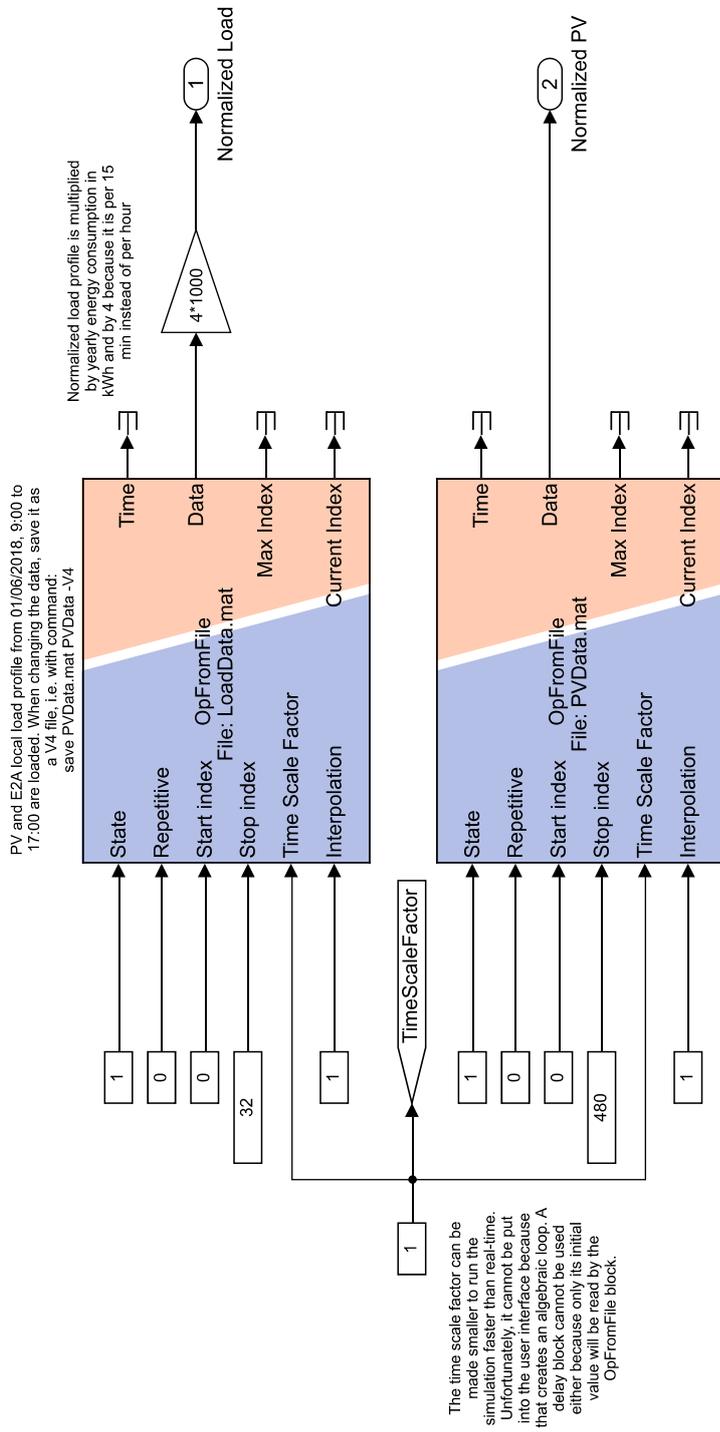


Figure B.8: Subsystem within Slave which reads data from external files

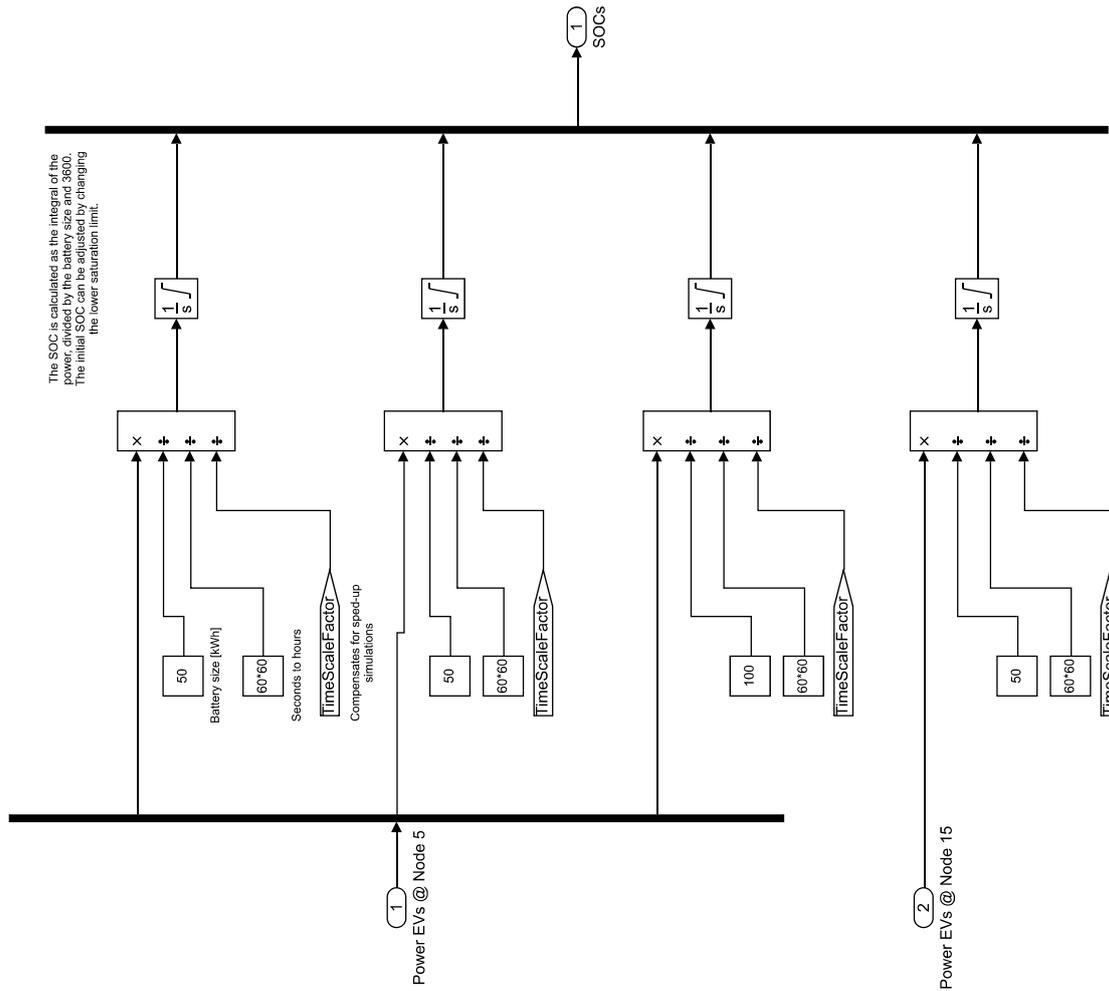


Figure B.9: Subsystem within Slave which calculates the EVs' SOC by integrating their charging power.

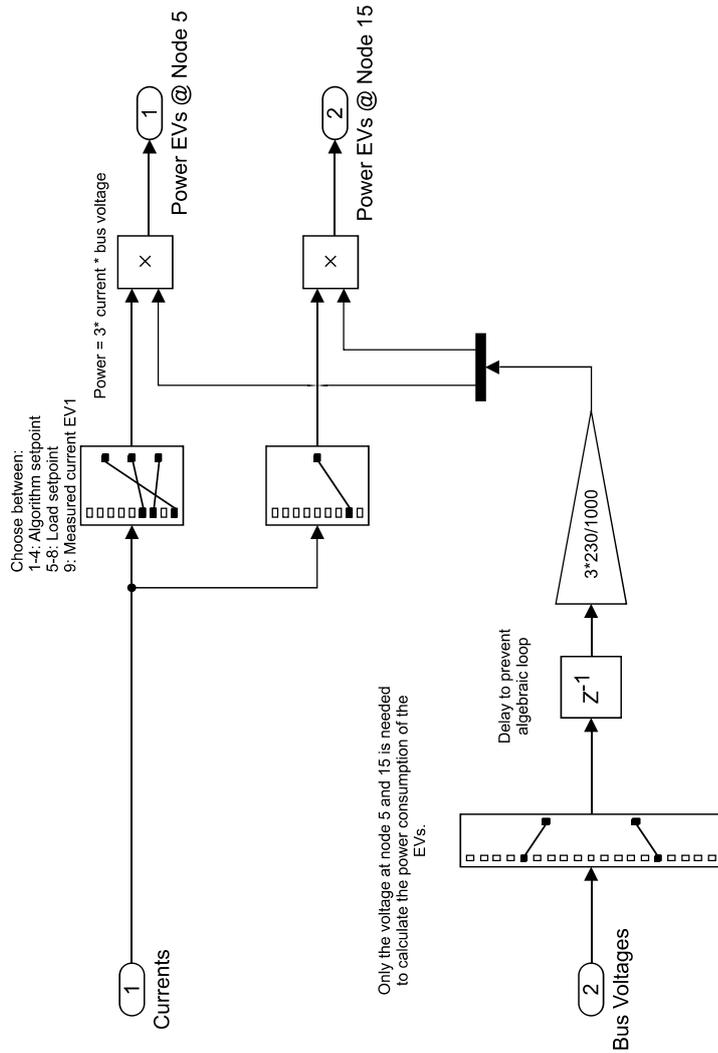


Figure B.10: Subsystem within Slave which calculates power going to each EV based on current and bus voltage

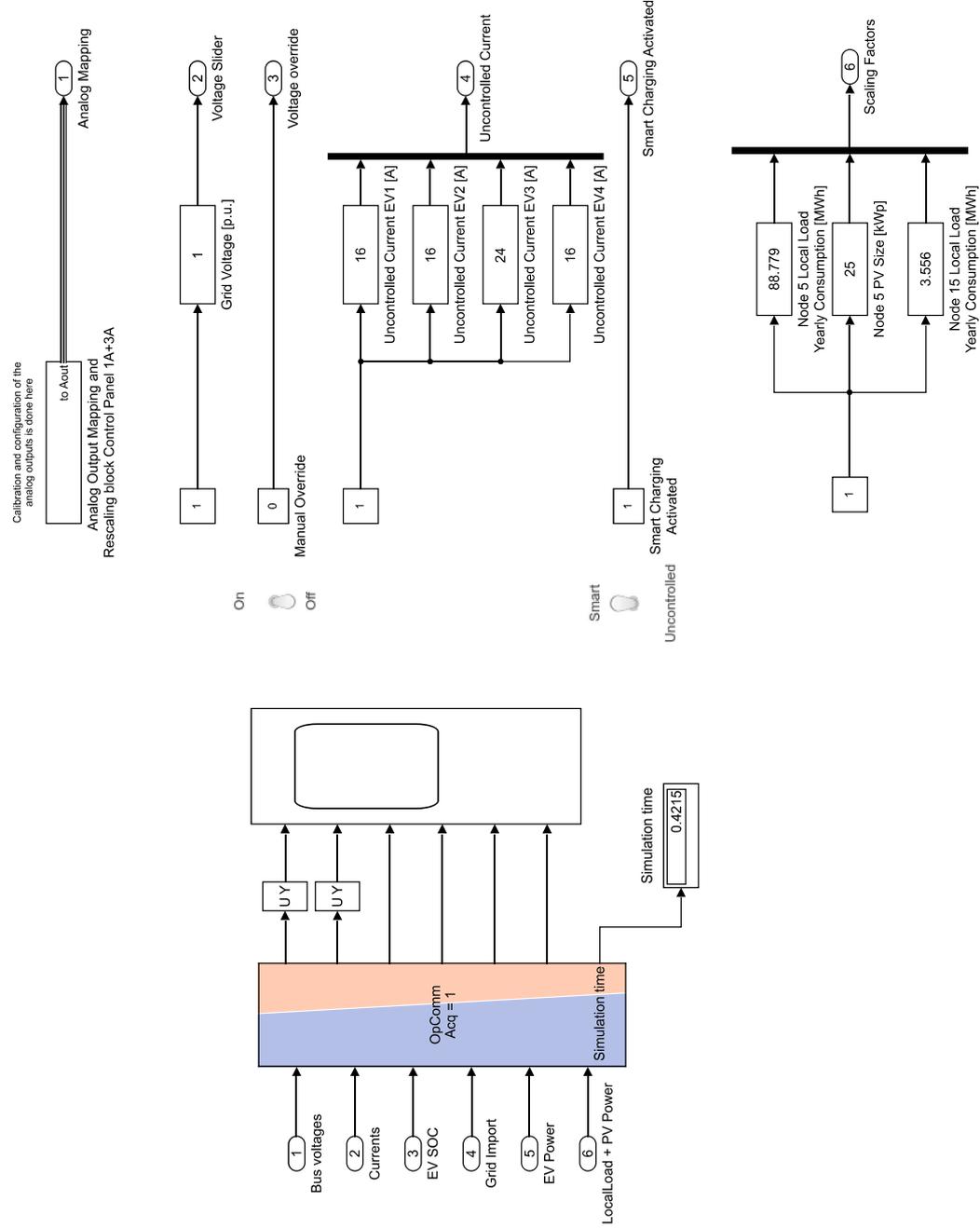


Figure B.11: User Interface (Console) Subsystem

Bibliography

- [1] European Environment Agency, “Greenhouse gas emissions from transport in Europe,” 2019. [Online]. Available: <https://www.eea.europa.eu/data-and-maps/indicators/transport-emissions-of-greenhouse-gases/transport-emissions-of-greenhouse-gases-12>
- [2] International Energy Agency, “Global EV Outlook 2020,” IEA, Tech. Rep., 2020.
- [3] P. Denholm, M. O’connell, G. Brinkman, and J. Jorgenson, “Overgeneration from Solar Energy in California: A Field Guide to the Duck Chart,” National Renewable Energy Laboratory (NREL), Tech. Rep., 2015. [Online]. Available: www.nrel.gov/publications.
- [4] International Renewable Energy Agency, “Innovation landscape brief: Electric-vehicle smart charging,” IRENA, Abu Dhabi, Tech. Rep., 2019. [Online]. Available: www.irena.org
- [5] G. Fitzgerald, C. Nelder, and J. Newcomb, “Electric Vehicles as Distributed Energy Resources,” Rocky Mountain Institute, Boulder, Tech. Rep., 2016. [Online]. Available: http://www.rmi.org/pdf_evs_as_DERs.
- [6] A. Chiş, J. Lundén, and V. Koivunen, “Reinforcement learning-based plug-in electric vehicle charging with forecasted price,” pp. 3674–3684, 5 2017.
- [7] D. Van Der Meer, G. R. Chandra Mouli, G. M. E. Mouli, L. R. Elizondo, and P. Bauer, “Energy Management System with PV Power Forecast to Optimally Charge EVs at the Workplace,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 311–320, 1 2018.
- [8] G. R. Chandra Mouli, M. Kefayati, R. Baldick, and P. Bauer, “Integrated PV charging of EV fleet based on energy prices, V2G, and offer of reserves,” *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1313–1325, 3 2019.
- [9] N. I. Nimalsiri, C. P. Mediwaththe, E. L. Ratnam, M. Shaw, D. B. Smith, and S. K. Halgamuge, “A Survey of Algorithms for Distributed Charging Control of Electric Vehicles in Smart Grid,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19, 10 2019.
- [10] C. P. Mediwaththe and D. B. Smith, “Game-theoretic electric vehicle charging management resilient to non-ideal user behavior,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3486–3495, 11 2018.
- [11] A. H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, “Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid,” *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, 12 2010.
- [12] G. R. Chandra Mouli, “Charging electric vehicles from solar energy Power converter, charging algorithm and system design,” Ph.D. dissertation, Delft University of Technology, Delft, 2018. [Online]. Available: <https://doi.org/10.4233/uuid:dec62be4-d7cb-4345-a8ae-65152c78b80f>
- [13] Z. Xu, Z. Hu, Y. Song, W. Zhao, and Y. Zhang, “Coordination of PEVs charging across multiple aggregators,” *Applied Energy*, vol. 136, pp. 582–589, 12 2014.
- [14] M. Nour, S. M. Said, A. Ali, and C. Farkas, “Smart Charging of Electric Vehicles According to Electricity Price,” in *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering, ITCE 2019*. Institute of Electrical and Electronics Engineers Inc., 2 2019, pp. 432–437.
- [15] Elaad NL, “EV Related Protocol Study,” Elaad NL, Arnhem, Tech. Rep., 2016. [Online]. Available: <https://www.elaad.nl/research/ev-related-protocol-study/>

- [16] Open Charge Alliance, "OCPP 2.0.1," 2020. [Online]. Available: <https://www.openchargealliance.org/protocols/ocpp-201/>
- [17] J. Lohse, "A New Chapter of EV Smart Charging by Introducing OCPP 2.0 and its Advantages to OCPP 1.6," 2020. [Online]. Available: <https://www.amppcontrol.io/ocpp-2-vs-ocpp-1-6>
- [18] Open Charge Alliance, "OCPP 2.0.1: Part 2 - Specifications," Open Charge Alliance, Arnhem, Tech. Rep., 2020.
- [19] M. Multin, "ISO 15118 as the Enabler of Vehicle-to-Grid Applications," in *2018 International Conference of Electrical and Electronic Technologies for Automotive, AUTOMOTIVE 2018*. Institute of Electrical and Electronics Engineers Inc., 10 2018.
- [20] V2G-Clarity, "What is ISO 15118?" 2019. [Online]. Available: <https://v2g-clarity.com/knowledgebase/what-is-iso-15118/>
- [21] M. O. Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. Strunz, M. Saeedifard, X. Wang, D. Shearer, M. Paolone, R. Brandl, M. Matar, A. Davoudi, and R. Iravani, "Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis," *IEEE Power and Energy Technology Systems Journal*, vol. 2, pp. 63–73, 2015.
- [22] S. Ledinger, D. Reihls, D. Stahleder, and F. Lehfuss, "Test Device for Electric Vehicle Grid Integration," in *Proceedings - 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe, IEEEIC/I and CPS Europe 2018*. Institute of Electrical and Electronics Engineers Inc., 10 2018.
- [23] F. Lehfuss, G. Lauss, C. Seitzl, F. Leimgruber, M. Noehrer, and T. I. Strasser, "Coupling of real-time and co-simulation for the evaluation of the large scale integration of electric vehicles into intelligent power systems," in *2017 IEEE Vehicle Power and Propulsion Conference, VPPC 2017 - Proceedings*, vol. 2018-January. Institute of Electrical and Electronics Engineers Inc., 4 2018, pp. 1–5.
- [24] E. de Jong, R. Graaff, P. Vaessen, P. Crolla, A. Roscoe, F. Lehfuss, G. Lauss, P. Kotsampopoulos, and F. Gafaro, *European White Book on Real-Time Power Hardware-in-the-loop testing*. European Distributed Energy Resources Laboratories, 1 2012.
- [25] M. Dargahi, A. Ghosh, G. Ledwich, and F. Zare, "Studies in power hardware in the loop (PHIL) simulation using real-time digital simulator (RTDS)," in *PEDES 2012 - IEEE International Conference on Power Electronics, Drives and Energy Systems*, 2012.
- [26] T. Hatakeyama, A. Riccobono, and A. Monti, "Stability and accuracy analysis of power hardware in the loop system with different interface algorithms," in *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics, COMPEL 2016*. Institute of Electrical and Electronics Engineers Inc., 8 2016.
- [27] W. Ren, M. Steurer, and T. L. Baldwin, "Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms," *IEEE Transactions on Industry Applications*, vol. 44, no. 4, pp. 1286–1294, 2008.
- [28] M. Shin, H. Kim, H. Kim, and H. Jang, "Building an interoperability test system for electric vehicle chargers based on ISO/IEC 15118 and IEC 61850 standards," *Applied Sciences (Switzerland)*, vol. 6, no. 6, 6 2016.
- [29] S. Martinenas, K. Knezovic, and M. Marinelli, "Management of Power Quality Issues in Low Voltage Networks Using Electric Vehicles: Experimental Validation," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 971–979, 4 2017.
- [30] Y. Zou, J. Zhao, X. Gao, Y. Chen, and A. Tohidi, "Experimental results of electric vehicles effects on low voltage grids," *Journal of Cleaner Production*, vol. 255, 5 2020.

- [31] Q. Xie, H. Hui, Y. Ding, C. Ye, Z. Lin, P. Wang, Y. Song, L. Ji, and R. Chen, "Use of demand response for voltage regulation in power distribution systems with flexible resources," *IET Generation, Transmission and Distribution*, vol. 14, no. 5, pp. 883–892, 3 2020.
- [32] Y. Xiang, Z. Jiang, C. Gu, F. Teng, X. Wei, and Y. Wang, "Electric vehicle charging in smart grid: A spatial-temporal simulation method," *Energy*, vol. 189, 12 2019.
- [33] M. H. Mobarak and J. Bauman, "Vehicle-Directed Smart Charging Strategies to Mitigate the Effect of Long-Range EV Charging on Distribution Transformer Aging," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, pp. 1097–1111, 12 2019.
- [34] D. Stahleder, D. Reihls, and F. Lehfuss, "Lablink - A Novel Co-Simulation Tool For the Evaluation of Large Scale EV Penetration Focusing On Local Energy Communities," Austrian Institute of Technology, Tech. Rep., 2018.
- [35] D. Stahleder, M. Nöhrer, F. Lehfuss, and H. Müller, "Implementation of a real time capable, flexible and accurate Electric Vehicle model to holistically evaluate Charging Services and Methods," in *7th Transport Research Arena*, Vienna, 4 2018.
- [36] A. Alyousef, D. Danner, F. Kupzog, and H. De Meer, "Design and validation of a smart charging algorithm for power quality control in electrical distribution systems," in *e-Energy 2018 - Proceedings of the 9th ACM International Conference on Future Energy Systems*. Association for Computing Machinery, Inc, 6 2018, pp. 380–382.
- [37] M. Rata, G. Rata, C. Filote, M. S. Raboaca, A. Graur, C. Afanasov, and A. R. Felseghi, "The electricalvehicle simulator for charging station in mode 3 of IEC 61851-1 standard," *Energies*, vol. 13, no. 1, 12 2019.
- [38] OSCD, "Infographic OSCD, The Project," 2020. [Online]. Available: <https://www.oscd.eu/about-us/testpage-1/>
- [39] OPAL-RT, "Power hardware in the loop." [Online]. Available: <https://www.opal-rt.com/power-hardware-in-the-loop/>
- [40] OPAL-RT Technologies, "OP5700 User Manual," OPAL-RT Technologies, Montreal, Tech. Rep., 11 2018. [Online]. Available: www.opal-rt.com
- [41] CompuMess Electronics GmbH, "AST1501 Product Page." [Online]. Available: <https://compumess.de/ast1501-ac/ac-california-instruments-ametek/>
- [42] AMETEK Programmable Power, "Asterion Series AC/DC Power Source 1U Models User Manual," Ametek, Tech. Rep., 2 2018.
- [43] O. Ramírez Barrón, "Powerflow (Newton Raphson)," 2017. [Online]. Available: <https://nl.mathworks.com/matlabcentral/fileexchange/60988-powerflow-newton-raphson>
- [44] A. Vijayvargia, S. Jain, S. Meena, V. Gupta, and M. Lalwani, "Comparison between Different Load Flow Methodologies by Analyzing Various Bus Systems," Rajasthan Technical University, Tech. Rep. 2, 2016. [Online]. Available: <http://www.irphouse.com>
- [45] B. Muruganantham, R. Gnanadass, and N. P. Padhy, "Performance analysis and comparison of load flow methods in a practical distribution system," in *2016 National Power Systems Conference, NPSC 2016*. Institute of Electrical and Electronics Engineers Inc., 2 2017.
- [46] Rexel, "Alfen Eve Single Pro-line." [Online]. Available: <https://www.rexel.nl/nln/Rexel/Duurzame-energie/EV-Laadunits/EV-Laadunits-onderdelen/Oplaadpunt-elektrisch-voertuig/Alfen-ICU-Eve-Single-Pro-line-Socket-3-fase-11kW-Plug-%26-Charge-%28excl-kabel%29/p/2850371586>
- [47] Scheinder Electric, "TM221ME16TG Product Page." [Online]. Available: <https://www.se.com/ww/en/product/TM221ME16TG/controller-m221-16-io-transistor-pnp-ethernet-spring/>

- [48] Mennekes Elektrotechnik GmbH & Co. KG, "MENNEKES - My power connection: Intelligent charging stations as a key to electric mobility," 2011. [Online]. Available: [http://www.mennekes.be/index.php?id=latest0&L=5&tx_ttnews\[tt_news\]=842&cHash=d1913d792b323a652d63cb9eafd1ffb](http://www.mennekes.be/index.php?id=latest0&L=5&tx_ttnews[tt_news]=842&cHash=d1913d792b323a652d63cb9eafd1ffb)
- [49] International Electrotechnical Commission, "IEC 61851: Electric vehicle conductive charging system," IEC, Tech. Rep., 2019.
- [50] C. Lewandowski, S. Gröning, J. Schmutzler, and C. Wietfeld, "Interference analyses of Electric Vehicle charging using PLC on the Control Pilot," in *2012 IEEE International Symposium on Power Line Communications and Its Applications, ISPLC 2012*, 2012, pp. 350–355.
- [51] Walther-Werke, "Walther-Werke: EV-Tester." [Online]. Available: https://www.walther-werke.de/nc/products-page/filter/ctrl/product/action/show/?tx_waltherwerkeproducts_productfilter%5Bartikelnummer%5D=780001502&cHash=60131651af5e62745adde8aa0ef2f33a
- [52] Delta Elektronika B.V., "SM15K Data Sheet," Delta Elektronika B.V., Zierikzee, Tech. Rep., 6 2020. [Online]. Available: www.DeltaPowerSupplies.com
- [53] Y. Yu, "OSCD Energy management system for smart charging of EVs V2.2," TU Delft, Delft, Tech. Rep., 2019.
- [54] T. Nederlof, "Implementations of Modbus Slave TCP/IP for Alfen NG9xx Platform," Alfen N.V., Tech. Rep., 2020.
- [55] Gamry Instruments, "Waveform Generation and Frequency Resolution." [Online]. Available: <https://www.gamry.com/application-notes/EIS/waveform-generation-and-frequency-resolution/>
- [56] N. Leemput, "Grid-supportive Charging Infrastructure for Plug-in Electric Vehicles," Ph.D. dissertation, KU Leuven, Heverlee, 11 2015. [Online]. Available: https://www.researchgate.net/publication/290358853_Grid-supportive_Charging_Infrastructure_for_Plug-in_Electric_Vehicles
- [57] Netherlands Enterprise Agency (RVO), "Statistics Electric Vehicles in the Netherlands," RVO, Tech. Rep., 8 2020. [Online]. Available: <https://www.government.nl/ministries/ministry-of-infrastructure-and-water-management;>
- [58] EV Database, "Tesla Model S Long Range Specifications," 2020. [Online]. Available: <https://ev-database.nl/auto/1194/Tesla-Model-S-Long-Range>
- [59] —, "Tesla Model 3 Standard Range Plus Specifications," 2020. [Online]. Available: <https://ev-database.nl/auto/1177/Tesla-Model-3-Standard-Range-Plus>
- [60] Liander N.V., "Typen aansluitingen," 2020. [Online]. Available: <https://www.liander.nl/consument/aansluitingen/typen>
- [61] European Network of Transmission System Operators for Electricity (ENTSO-E), "Day-ahead Prices," 2018. [Online]. Available: <https://transparency.entsoe.eu/transmission-domain/r2/dayAheadPrices/show>
- [62] Koninklijk Nederlands Meteorologisch Instituut (KNMI), "CESAR Database," 2018. [Online]. Available: <https://ruisdael-observatory.nl/cesar/>
- [63] Nederlandse Energie- Data Uitwisseling (NEDU), "Verbruiksprofielen," 2018. [Online]. Available: <https://www.nedu.nl/documenten/verbruiksprofielen/>
- [64] R. Blaga, A. Sabadus, N. Stefu, C. Dughir, M. Paulescu, and V. Badescu, "A current perspective on the accuracy of incoming solar energy forecasting," *Progress in Energy and Combustion Science*, vol. 70, pp. 119–144, 1 2019.
- [65] D. Wang, J. Coignard, T. Zeng, C. Zhang, and S. Saxena, "Quantifying electric vehicle battery degradation from driving vs. vehicle-to-grid services," *Journal of Power Sources*, vol. 332, pp. 193–203, 11 2016.