

DELFT UNIVERSITY OF TECHNOLOGY

MASTER THESIS

Hardware-in-the-Loop Simulation of EV Smart Charging in a Distribution Grid

Authors: Junsheng Zhang
Student number 5228239

To obtain the degree of Master of Science in Electrical Power Engineering

August 22, 2022



Preface

I started this thesis in the department of DCE&S in November 2021 and back then, this final report seemed only a very far goal, now it is over. For the past 11 months, nearly a year, I have devoted all of myself to the topic of smart charging algorithms and HIL testbed. From the beginning, I didn't understand anything and now I have gained experience in implementing the algorithm and connecting hardware. In these 11 months, the biggest gain is not the knowledge or the experience I have accumulated, but the progress of my mentality. During these 11 months, I have been frustrated, broken down, and encountered many difficulties several times, but I overcame them one by one and got to where I am today. The master thesis made me understand that in academic research, only firm belief can overcome difficulties, and this is also what an engineer must have. As a graduating student who is about to step into society, the 11 months I spent on my master thesis will be an unforgettable memory for me. This thesis cultivated and improved my ability to apply knowledge, so that now I can enter the stage of combining theory with practice and actively analyzing and solving problems.

Here, I sincerely hope you will find the results I present here helpful to you, at the moment of the completion of my master thesis, I would like to express my sincere thanks to my supervisor, Ph.D. supervisor MSc. Y. Yu, who guided and helped me over the past 11 months. She has given careful guidance and selfless help in various aspects such as system design, on-site investigation, laboratory work, and thesis writing. I learned a lot from her profound knowledge, rigorous academic attitude, and selfless dedication. I also appreciate my supervisor Dr.ir. G. R. Chandra Mouli, who always helps me with the planning for the next stage of the project. Under the COVID-19 circumstances, their help is very important to me, without their help I would not be able to finish the master thesis, finally, I would like to thank my family, During the epidemic, being able to meet them in a short time every day can greatly inspire my, let me regain the confidence to overcome one difficulty after another.

Thank you all.

Abstract

With the development of sustainable energy technology and the increasing emphasis on environmental protection, the popularity of electric vehicles is increasing. However, As the number of electric vehicles increases, so does the pressure on the grid due to charging. If a large number of electric vehicles are charged at the same time, the grid will inevitably be overloaded. For this problem, smart charging can be a good solution. The smart charging algorithm determines when it is optimal to charge the electric vehicles(EVs) based on factors like Photovoltaic(PV) generation, local load consumption, and energy price. For customers, this means the total cost of charging is the lowest, for the power system it means charging when the network is not overloaded. Furthermore, Smart charging can reduce the need for grid infrastructure investments and reduce energy costs. From an energy-friendly and economical perspective, this is worth doing.

Up to now, there is a large amount of literature study about smart charging. However, the vast majority of literature is based on pure software analysis rather than hardware experiments. In this thesis, a hardware-in-the-loop based simulation is presented. The basic idea of hardware-in-the-loop simulation is to include a part of the real hardware in the simulation loop. The advantages of hardware-in-the-loop simulation are obvious. Compared with pure software simulation, hardware-in-the-loop(HIL) simulation has the real response of the hardware components, and it is not very complicated to set everything up. In this thesis, the testing and the evaluation of the system are carried out in real-time because the HIL setup includes an OPAL-RT OP4510 DRTS. The simulation of the distribution grid is running on the OPAL target by the software ePHASORsim. The rest part of the testbed consists of a Programmable Logic Controller (PLC) and two bidirectional Direct Current(DC) power supplies to form an Alternating Current(AC) source and an Alfen charger. The smart charging algorithm is written in Python code. By calling Gurobi, smart charging algorithms can calculate and decide the optimal current setpoints for each EV connected to the distribution grid in real-time.

Contents

1	Introduction	1
1.1	Motivation	1
1.1.1	Why smart charging	1
1.1.2	Hardware-in-the-loop simulation	2
1.2	Research Objectives	3
1.3	Outlines	3
2	Literature Review	5
2.1	Digital Real-Time simulation	5
2.2	Choosing Hardware-in-the-loop simulation	6
2.2.1	Reasons for applying Power Hardware-in-the-loop simulation	6
2.2.2	Introduction of Hardware-in-the-loop simulation	7
2.2.3	Real-Time simulator	7
2.2.4	Introduction of ePHASORsim	9
2.3	literature study of smart charging algorithm	10
2.4	Similar experiments	11
2.4.1	experiment 1	11
2.4.2	experiment 2	11
2.4.3	experiment 3	12
2.4.4	Summary	13
3	System design	15
3.1	The schematic of HIL setup	15
3.2	Preparation of model	17
3.2.1	Before the simulation	17
3.2.2	Configure the I/O interface	18
3.3	Build the simulation Model	18
3.3.1	Console	19
3.3.2	Master	19
3.3.3	Two "simulation time"	22
3.4	Functions of smart charging algorithm	23
3.4.1	objective functions	23
3.4.2	sub functions	24
3.4.3	Constrains	25
4	Integrating Algorithm with Model and Hardware	27
4.1	The adaption of algorithm	27
4.2	Integrate the hardware	29
5	Basic Case	37
5.1	Simulation scenario	37
5.2	Validation of ePHASORsim	39
5.3	Simulation results	39
6	Case study	41
6.1	Case 1	41
6.2	Case 2	43
6.3	Case 3	45
6.4	Summary	46

7 Conclusion & Future work	49
7.1 Conclusion	49
7.2 Recommendations and Future work	50
7.2.1 Recommendations	50
7.2.2 Future work	50
Appendices	53
A MATLAB Model	53
B Registers Tables	61
B.1 Product identification registers	61
B.2 Station status registers	62
B.3 SCN registers	63
B.4 Socket measurement registers	64

List of Figures

1.1	Growth in global car sales[5]	1
1.2	Charging during off-peak hour[8]	2
2.1	Off-line simulation VS Real-time simulation[13]	5
2.2	Different methods[14]	6
2.3	CHIL and PHIL[13]	6
2.4	Overview of HIL setup[19]	7
2.5	On host computer[22]	8
2.6	Part of the powerFactory model	9
2.7	The control loop of EV smart charging in experiment 1[35]	11
2.8	Experiment 2 setup[36]	12
2.9	Experiment 3 setup[39]	13
3.1	The schematic of HIL setup[40]	15
3.2	The observation on ACE Service Installer	17
3.3	OPAL-RT Board	18
3.4	Reference wave-forms	20
3.5	Constant voltage and constant current charging stage[26]	21
3.6	Grid Model	22
4.1	Algorithm functions	27
4.2	Flowchart of interaction	28
4.3	Photo of all components	29
4.4	Schematic without Delta	30
4.5	Type 2 plug[44]	30
4.6	Alfen live monitor	32
4.7	Modbus registers[46]	32
4.8	Modbus channel assistant	33
4.9	PLC Program	34
4.10	The screenshot of connection	34
4.11	Values in the PLC rungs	35
5.1	Prices of buying electricity, selling electricity and FCR	37
5.2	Grid model for basic case	38
5.3	Result under uncontrolled charging	39
5.4	Results of the base case	40
6.1	Simulation results of case 1	42
6.2	Evening prices	43
6.3	Simulation result of case 2	44
6.4	Simulation results of case 3	45
7.1	Power circulation[40]	49
A.1	MATLAB Model	53
A.2	SC_Console	54
A.3	SM_Master part1	55
A.4	SM_Master part2	56
A.5	Analog Out	57
A.6	SOC check and CV charging	58
A.7	SOC check and CV charging	59
A.8	SOC check and CV charging	60

List of Tables

2.1	Summary	13
3.1	I/O channels	18
3.2	Objective functions parameters	24
3.3	Sub functions parameters	25
4.1	Values of voltage and resistance for different state [45]	31
4.2	CP duty cycle to maximum charging current[45]	31
4.3	Memories used	33
5.1	Parameters used for basic scenarios	37
5.2	Parameters of EVs	38
6.1	Parameters of two groups of EVs	41
6.2	Departure SOC	43
6.3	Arrival and departure time	43
6.4	Summary	46
6.5	Voltage deviation	46

List of Abbreviations

AC Alternating Current.

CP Control Pilot.

CV Constant Voltage.

CPU Central Processing Unit .

CHIL Controller Hardware-in-the-Loop.

DRTS Digital Real-Time Simulator.

DC Direct Current.

PV Photovoltaic.

EU European Union.

EV Electric Vehicle.

EVSE Electric Vehicle Supply Equipment.

FPGA Field Programmable Gate Array.

GCC GNU Compiler Collection.

HIL Hardware-in-the-Loop.

I/O Input/Output.

KNMI Royal Netherlands Meteorological Institute.

NEDU Association for Dutch Energy Data Exchange

RCP Rapid Control Prototyping.

SIL Software-in-the-loop.

SOC State Of Charge.

PHIL Power Hardware-in-the-Loop.

PLC Programmable Logic Controller.

API Application Programming Interface.

PWM Pulse-Width Modulation.

PE Protective Earth.

1 Introduction

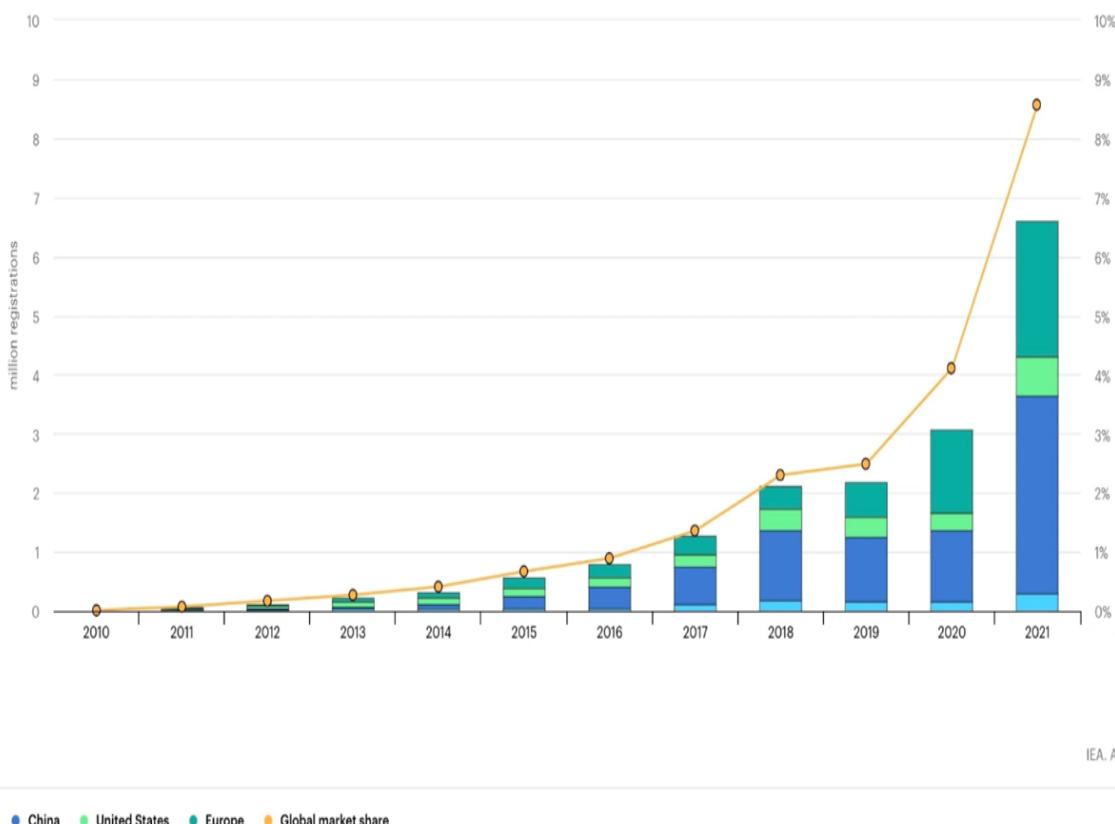
1.1 Motivation

1.1.1 Why smart charging

In response to the deteriorating environment, governments around the world are protecting the environment by reducing carbon emissions, and reducing carbon emissions in transportation is a top priority. The European Union (EU) and its member states are encouraging the development of sustainable energy alternatives to fossil fuels [1]. Due to the exhaust pollution caused by transportation, especially fuel vehicles, people tend to advocate low-carbon mobility to replace traditional fuel vehicles. Therefore, electric vehicles have received widespread attention as it provides many benefits, such as reducing fossil fuel dependence and improving the environment [2]. Electric vehicles are driven by power battery packs and electric motors. No waste gas is produced during work, and no waste gas pollution is emitted. Therefore, the environmental pollution caused by electric vehicles is very small [3].

The market share of electric vehicles has been growing over the past decade, especially with impressive growth rates over the past three years. As can be seen from Figure 1.1, 2.2 million EVs were sold in 2019, accounting for only 2.5% of global vehicle sales. But in 2020, the overall auto market was shrinking, and electric vehicle sales rose against the trend, rising to 3 million units, accounting for about 4% of total vehicle sales. By 2021, electric vehicles had reached 6.6 million units, accounting for nearly 9% of the global auto market, and the market share has tripled from two years ago [4].

Global sales and sales market share of electric cars, 2010-2021



IEA. All Rights Reserved

Others China United States Europe Global market share

Figure 1.1: Growth in global car sales[5]

Although the growth in the number of electric vehicles is good news for the environment, there are still some negative effects that deserve attention, especially the impact on the electrical distribution grid. If vehicle charging is not restricted, a large number of charging electric vehicles can affect the performance and efficiency of the grid. In addition to this, if only simple charging strategies are used, a large number of EVs will be charged at the same time, which means that the peak load will increase. The grid will require more generation and transmission capacity. To increase the power generation and transmission capacity of the grid, then the original transformers and cables must be replaced, which is very troublesome and expensive.[6]-[7]. The cost-effective solution is smart charging. Smart Charging delays charging and automatically starts charging in order to finish before departure time while also ensuring to charge the battery during off-peak hours as figure 1.2 shown below to reduce energy costs.

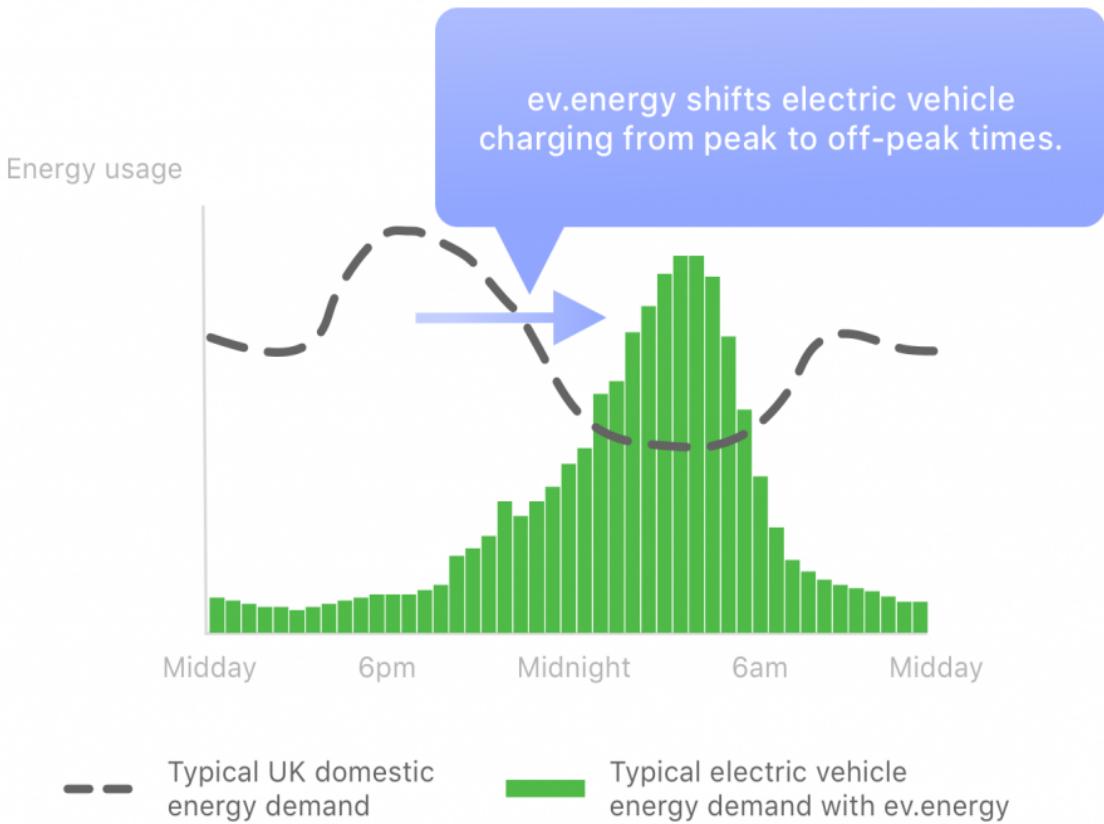


Figure 1.2: Charging during off-peak hour[8]

Charging during these off-peak hours avoids overloading the grid, and also keeps charging costs down by charging when electricity prices are lower.

1.1.2 Hardware-in-the-loop simulation

The smart charging algorithm is not a new concept. In fact, people have been researching smart charging for nearly ten years. At present, the smart charging algorithm has many applications in practice. However, researching the impact of smart charging algorithms on the distribution power grid is challenging. This is because parking several EVs in different physical locations in the lab is inconvenient. That is why many research about the smart charging algorithm on distribution grid is purely software-based. But if the simulation is purely software-based, then

the process would be simplified, and some factors will be ignored. Therefore, a Hardware-in-the-Loop (HIL) setup was adopted as a compromise solution in this thesis.

A HIL simulation is a type of simulation that combines software models or control algorithms with real physical hardware. When performing HIL simulation, the real distribution grid with EVs connected can be replaced by an equivalent computer model. Through inputs and outputs (I/Os) channels of the simulator, the model running on the real-time simulator is capable of interfacing with the external hardware. Especially for this thesis, a rural distribution grid and EVs connected to this grid will be replaced by an equivalent computer model. A total of four EV charging processes are emulated, but only one electric vehicle charging is emulated by hardware while the rest three are simulated by software.

1.2 Research Objectives

This thesis is built on the basis of the thesis of a student who has previously graduated. Compared with the previous thesis, the smart charging algorithm used in this thesis and the method used for grid simulation are all upgraded. In this thesis, the smart charging algorithm is combined with the HIL testbed to explore the impact of smart charging on the power grid under different scenarios. Therefore, the main research question is the following: How to evaluate the impact of EV smart charging on the distribution grid by implementing HIL real-time simulation?

The main objectives of this thesis are the following three aspects:

1. Hardware testbed: Mimic the charging behavior of a chosen EV on a developed hardware testbed.
2. Algorithm: Integrate the smart charging algorithm with the HIL testbed.
3. Grid simulation: Build a grid simulation model to simulate the power grid when the smart charging algorithm is applied to the distribution power grid.

1.3 Outlines

This report starts with a literature review that provides background information on real-time simulations, HIL setups, and similar cases. Chapter 3 then introduces how to build the model on the simulator, and the main function and sub-functions of the smart charging algorithm. Chapter 4 introduces the MATLAB Simulink model that runs on the DRTS and how it interacts with the smart charging algorithm. Chapter 5 shows the experimental results of the smart charging base case. Chapter 6 shows the experimental results of smart charging under different scenarios and analyzes the results, chapter 7 presents recommendations for improvement and work in the future.

2 Literature Review

The purpose of this chapter is to provide a review of the literature available on the main approaches, knowledge, and similar experiments. First subsection 2.1 introduces what is digital real-time simulation, in subsections 2.2 what is HIL real-time simulation has been introduced and the reasons for choosing HIL real-time simulation have been presented. Subsection 2.3 introduces how to combine smart charging with a hardware-in-the-loop test bed and in subsection 2.4 similar experiments will be presented and the research gap will be shown.

2.1 Digital Real-Time simulation

Digital Real-time simulation is a transient state simulation technology of a power system that is different from traditional offline simulation [9]-[11]. Traditional Offline simulation means that the response speed of the digital simulation model is different from the response speed of the actual system to observe the running state of the model for about 1s, the time required for simulation calculation may be much higher than 1s [12], depends on the complexity of the model. On the contrary, digital real-time simulation keeps models running in sync with reality, that is, the computer model runs at the same speed as the actual physical system, and if the model is simulated for 10 seconds, then it also takes ten seconds in the real world. The difference between off-line and real-time simulation can be observed in figure 2.1.

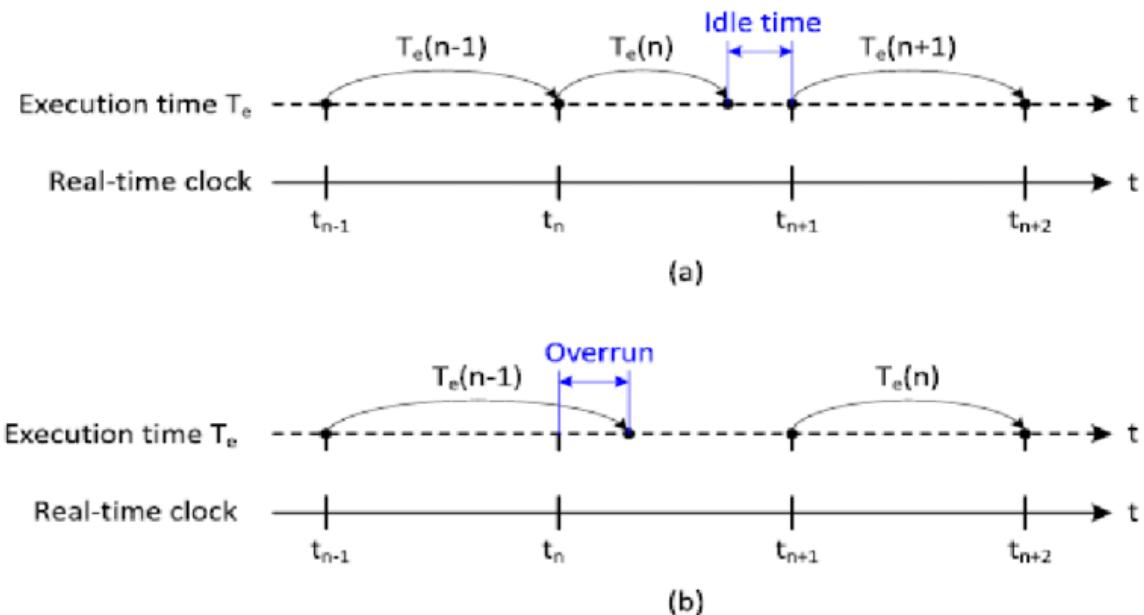


Figure 2.1: Off-line simulation VS Real-time simulation[13]

In the above figure, T_e is the execution time. If it is equal to or less than one real-time time step, then it is a real-time simulation. If T_e is larger than one time-step, then it is an off-line simulation [13]. In the field of power and power electronic systems, commonly used simulation software, such as PSCAD/EMTDC, MATLAB/Simulink, etc., are all running non-real-time digital simulations. While concepts like the HIL and RCP (Rapid Control Prototyping) belong to the real-time simulation.

2.2 Choosing Hardware-in-the-loop simulation

2.2.1 Reasons for applying Power Hardware-in-the-loop simulation

Various techniques can be used for real-time simulation [14], including HIL, RCP, Software-in-the-loop (SIL). The differences between these methods can be seen clearly in figure 2.2.

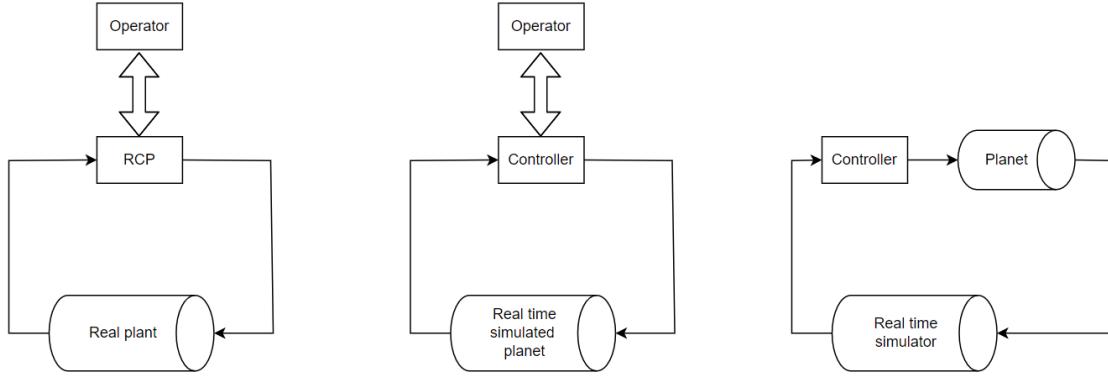


Figure 2.2: Different methods[14]

SIL is different from the other two methods because SIL means both the power system and the controller are simulated on the same real-time simulator, it seems good because outside interference will no longer be effective. However, this thesis is to simulate the real EV charging behavior. So outside interference should be included in the simulation, rather than simplifying them. The power system should be built by hardware and run in the real world rather than running on a real-time simulator.

The HIL real-time simulation can be Controller Hardware-in-the-loop (CHIL) or Power Hardware-in-the-loop (PHIL). The difference can be seen in the following figure.

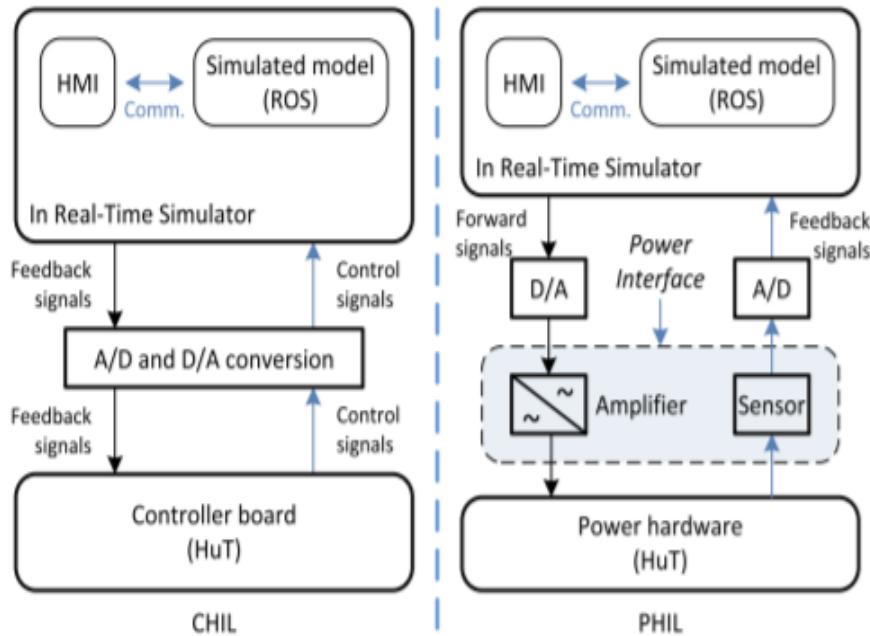


Figure 2.3: CHIL and PHIL[13]

Both control signal and feedback signal need to pass A/D and D/A conversion. However,

there are still some structural differences. CHIL means that the controller is able to interact with the rest of the system [14]. However, there will be no power transfer in the CHIL simulation loop, only signals can exchange between the simulator and controller through I/O channels. PHIL also has a controller, but between the controller and the simulator, there is a power interface that allows the power transfer to or from the Power hardware. Therefore, after comprehensive consideration, PHIL real-time simulation is used in this thesis to simulate the real EV charging behavior.

2.2.2 Introduction of Hardware-in-the-loop simulation

The rise of HIL is because of testing the system previously in the laboratory using a complete set of hardware. So, this approach can be very expensive, inconvenient, inefficient, and unsafe [15], and there is no guarantee of how accurate and how fast the test will be.

Therefore, HIL testing provides an alternative to the traditional method. The basic idea of the HIL simulation part of the simulation loop should consist of hardware [16]. In this way, instead of testing the control algorithm on a purely mathematical model, the simulation can be done with real hardware (if available). The benefits of the hardware-in-the-loop simulation are obvious:

1. Design and testing of the system without operating it[17], so that the system defects and software bugs can be found at the beginning.
2. Combine the flexibility and accuracy of software with the actual response of hardware.
3. Test hardware and software even under extreme conditions (e.g., high/low temperatures) [18]
4. The building of the whole system is very convenient and time-saving because there is no need to build a whole system with hardware.

The high-level overview of a HIL simulation setup is shown in Figure 2.3. Once the setup is completed, multiple scenarios can be tested easily without spending on actual physical tests. This makes the entire design and validation process thorough and cost-effective while saving testing time.

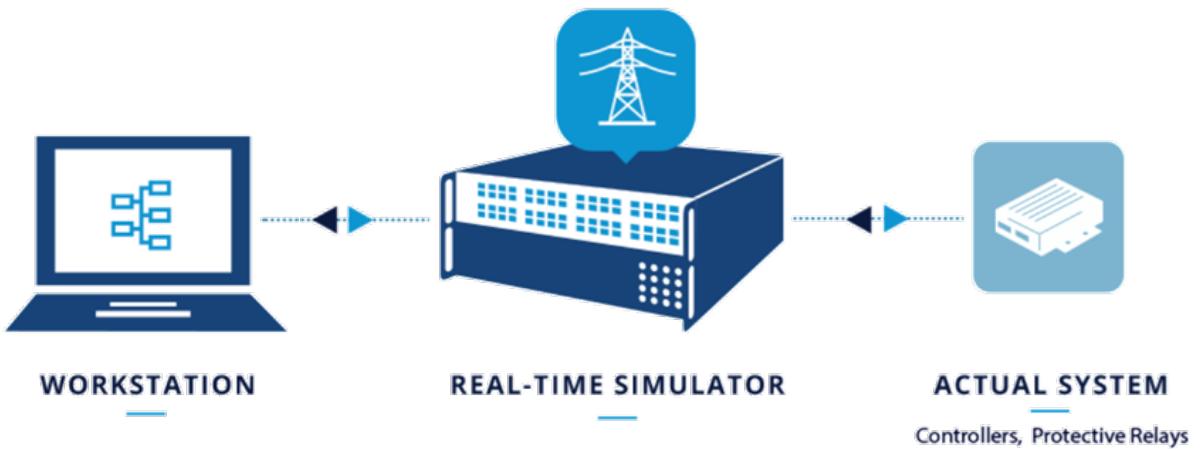


Figure 2.4: Overview of HIL setup[19]

2.2.3 Real-Time simulator

In the real-time simulation, due to a large amount of calculation of power and power electronic system models, it is often necessary to rely on a multi-core Central Processing Unit (CPU)

or Field Programmable Gate Array(FPGA) hardware with powerful computing power to perform calculations, and then perform some equivalence and optimization on real-time simulation modeling and calculation. Simulation is achieved. Therefore, in the real-time simulation system, the simulator is an important part, and the simulation scale and computing power of the real-time simulation system are determined by the hardware resources of the simulator.

There are several companies that are currently offering competitive HIL simulation capabilities as end-to-end solutions. Some of them (stated in no order) are Typhoon, OPAL-RT, dSPACE, Plexim's RT Box, Speedgoat, and NI. A vast majority of these solutions work with simulation models created by the MATLAB/Simulink software platform, thus making the integration and HIL validation process more convenient, intuitive, and user-friendly. Before building the testbed of HIL, the requirements of real-time target must be considered. There are 3 requirements that the real-time target must meet.

1. The HIL simulator must have the same time frame as the virtual model being simulated on it, and the real-time simulator that communicates with the outside physical devices must be able to communicate directly. Only in this way can the transmission of signals between the real-time simulator and other physical devices be real-time.

2. In HIL simulations, the real-time simulator must make sure that each run takes an integer multiple of the time step,[21].

3. A real-time simulator for HIL should ensure that what corresponds to that step is completed within each step, and it can be done early, but never with delays [21].

Usually, there will be software on the PC that can control the simulator, and the software will provide a simulation environment. The simulation environment provides the proper communication between the model and the IO. The following figure shows the simulation environment on the host PC.

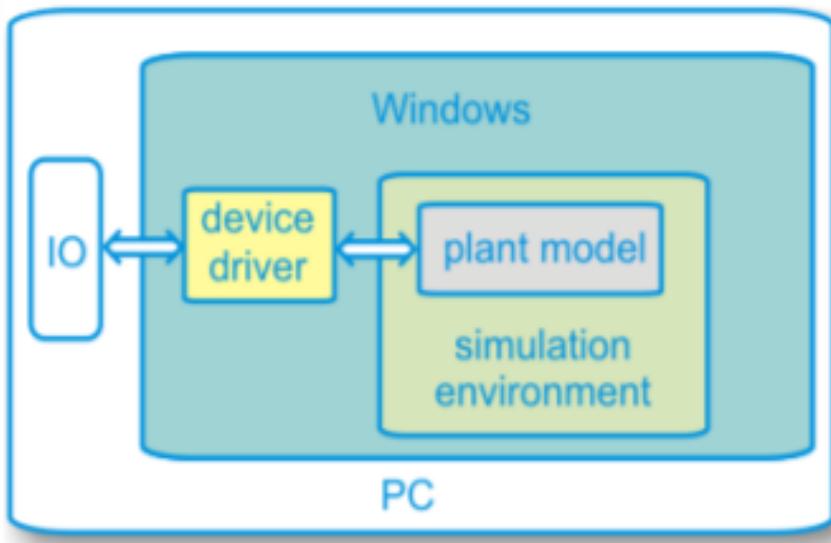


Figure 2.5: On host computer[22]

In this thesis, OPAL-RT has been selected, and RT-Lab shall be installed on the host PC. OPAL-RT HIL simulator offers offline simulation, real-time simulation, and real hardware test

capabilities on a single test bench. It promises open architecture, speed, accuracy, and overall high-performance, along with a wide range of demo models [19]. Eventually, OP4510 has been selected as the DRTS in this thesis.

2.2.4 Introduction of ePHASORsim

In addition to the advantages mentioned above, the OP4510 has another advantage, which is that it is compatible with the grid simulation tool ePHASORsim. ePHASORsim tool offers real-time dynamic simulations for the distribution grid. With the help of ePHASORsim, the state of each device, and various information including loss/load/phase shift in the grid during simulation can be observed [17]. There are several features of this simulation tool:

1. A large and convenient built-in positive sequence and 3-phase library. Depending on the native library, most of the distribution grid can be built and simulated. In this thesis, the following elements in the library will be used: 1) Controllable voltage source 2) Controllable current source 3) Transformer 4) Line 5) Load 6) Impedance

2. The flexible input data format: The user can choose between Excel, PSS/e, CYME and PowerFactory input file formats.

3. On-the-fly changes of parameters: During the real-time simulation the parameters can be changed by the external signals.

The rural grid model applied in this thesis is originally built by the software PowerFactory. In fact, ePHASORsim itself supports PowerFactory as input, this requires exporting the powerfactory model as a .xml file. Besides, an additional excel template needs to be created, which defines what the input and output of the grid model are. This excel template together with the exported to an XML file read by ePHASORsim block to run grid simulation. Part of the original PowerFactory model used in this thesis is shown below:

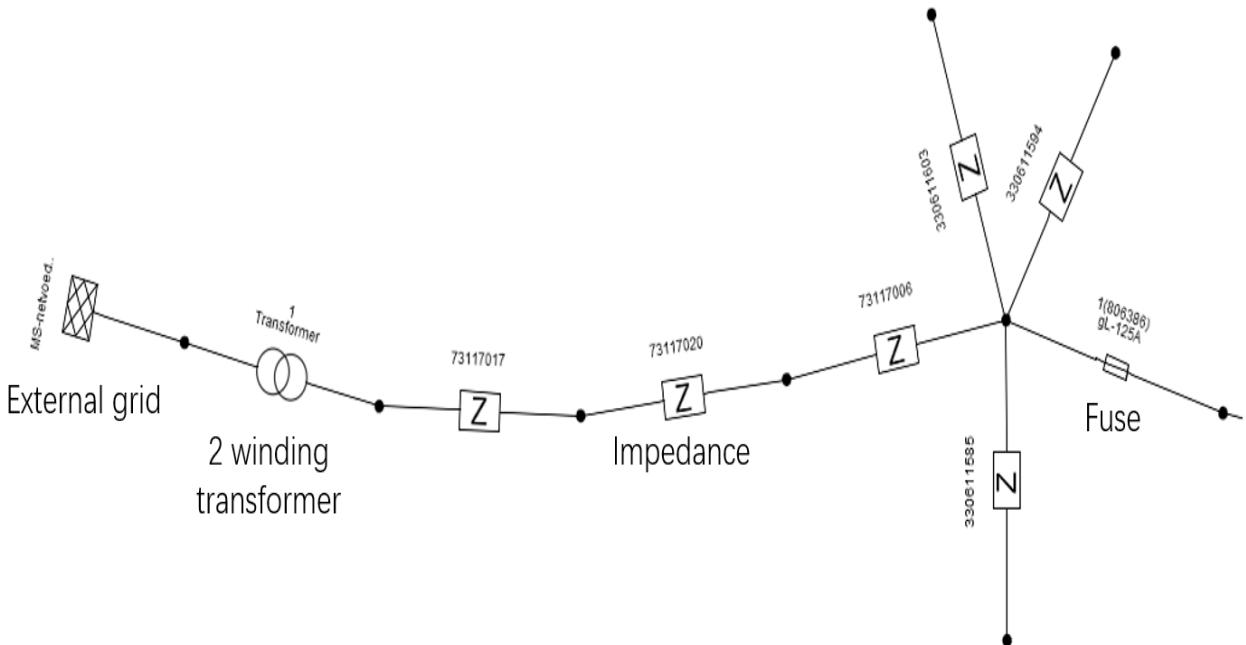


Figure 2.6: Part of the powerFactory model

2.3 literature study of smart charging algorithm

In many research studies, the main purpose of the smart charging algorithm is to maximize benefits for EV owners by reducing the charging cost or to maximize benefits for the electric utility by shifting the load demand to off-peak hours [23]-[25]. The smart charging algorithm controls the current flow in EV charging. It fully considers various parameters in EV charging, including energy prices, grid congestion, local generation, etc., and gives the optimal current set-point based on the current parameters through calculation. Smart charging is not a new topic, so far, the smart charging algorithm has been used in many practical cases:

1. Residential EV charging: The charging points of smart cars are generally divided into three categories: home, public and semi-public, this application is a typical example of charging at home. When calculating the optimal current setpoints, the load of household appliances is brought into the optimization calculation as local load consumption [26], the smart charging algorithm shifts the load demand to reduce the peak power and reduce the impact of EV charging on the distribution grid.
2. Smart charging for car park scenarios [27]: The smart charging algorithm will provide optimal charging scheduling based on factors such as electricity price, how many vehicles are parked, etc. to provide optimal charging scheduling. The charging schedule will benefit the users to charge their own electric vehicle (EV) at a lower cost.
3. EV smart charging with PV forecasting [28]: Including the prediction of PV generation in the smart charging algorithm [29]. Through the prediction of the solar radiation intensity, the optimization calculation is carried out by the smart charging algorithm, and the optimal current set-points are calculated so that the excess energy converted from the solar energy can be used for EV charging to the greatest extent.

In fact, the above are just a few examples of the practical application of the smart charging algorithm. By summarizing some existing applications, it can be found that the main benefits of the smart charging algorithm include the following points:

1. Minimize the cost of EV charging: Since the price of electricity is not fixed, but changes with different time periods, the intelligent charging algorithm allows charging to be concentrated in the time period when the electricity price is lower, so as to reduce the cost of importing energy from the grid.
2. Improving grid stability and safety: By time-shifting and avoid having a large number of EVs charging at the same time, the likelihood of grid overloading is greatly reduced.
3. Integrate the renewable energy sources: Excess energy provided by integrated renewable energy sources (PV, Wind turbine) can be used in EV charging to reduce the electricity consumption in the distribution grid [29],[30].
4. Saving energy loss: By time-shifting, avoiding having a large number of EVs charging at the same time can also reduce the energy loss caused by the excessive current in the grid.
5. Provision of ancillary services: Ancillary services like voltage regulation, power quality, and emission management [31] can be provided to the grid.

2.4 Similar experiments

Prior to this thesis, there has been a lot of research on implementing smart charging algorithms [32]-[34]. However, there are still some studies that experimentally simulate EV smart charging on the distributed grid. Several examples are presented below.

2.4.1 experiment 1

This experiment performed EV smart charging in SYSLAB, figure 2.6 shows the control loop of EV smart charging in the experiment.

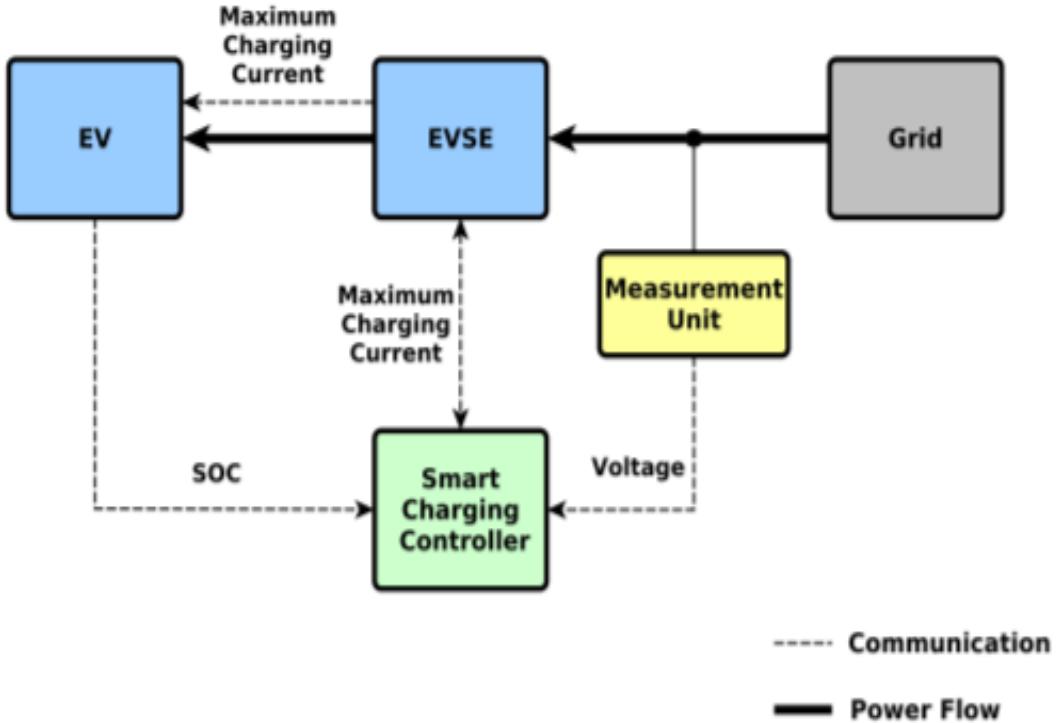


Figure 2.7: The control loop of EV smart charging in experiment 1[35]

In this experiment, the smart charging algorithm is used to improve the voltage quality in distribution grids, by measuring the voltage from the distribution grid and monitoring the state of charge (SOC) of the EV, the smart charging algorithm is able to give the maximum charging current limitation to the electric vehicles supply equipment (EVSE). However, the setup used in this experiment is not HIL. There is no real-time simulator in this setup. The components in this control loop only include a measurement device, EVSE, grid emulator, controller, and commercially available EVs. This also makes the whole set of equipment very expensive, and it will be very time consuming and laborious to build this set of equipment.

The result of this experiment proves that implementing the smart charging algorithm is able to improve power quality [31]. Nevertheless, it is also admitted that due to the complexity of the entire set of equipment, there is no way to test more scenarios.

2.4.2 experiment 2

This experiment performed EV smart charging based on the dynamic electricity price, figure 2.7 experiment setup. The setup used in this experiment also does not rely on the HIL method.

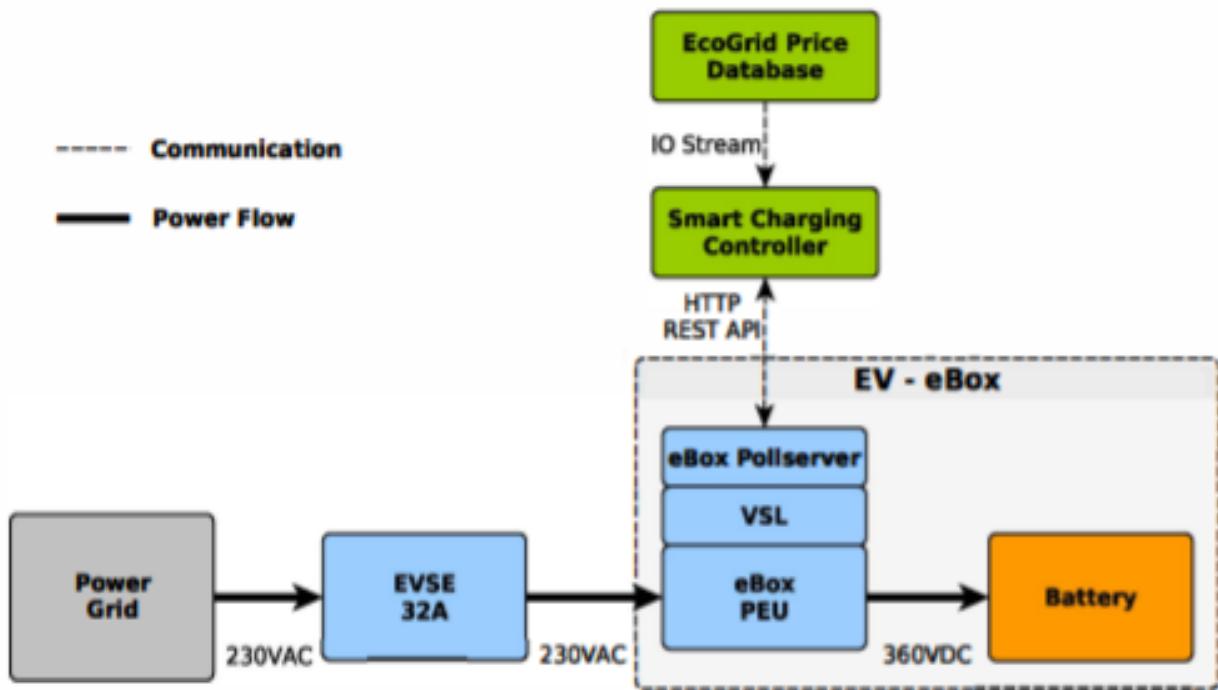


Figure 2.8: Experiment 2 setup[36]

In this experiment, the smart charging algorithm is used to reduce the EV owner's cost based on the dynamic energy price. Based on the Real-time price, Hour ahead price, and Day-ahead price the smart charging algorithm is able to predict the energy price variation in the next 24 hours [36][37]. One thing that needs to be noticed is the voltage provided by the power grid is 230V and the maximum current that the EVSE could provide is 32A. These two devices are not directly controlled by the smart charging algorithm, which means, the supplied voltage and current will not change due to the smart charging algorithm. The function of the smart charging algorithm in this experiment is to develop a charging scheme. The 'EV' in this experiment consists of a modified AC Propulsion eBox with a 35kWh battery and Power Electronics Units [38]. When the 'EV' can be charged depends on the scheme specified by the algorithm.

The result of this experiment proves that implementing the smart charging algorithm is able to minimize the charging cost. Nevertheless, The simulation of this model is not complete. In reality, the voltage of the grid cannot be maintained at 230v all the time, and the same as the current.

2.4.3 experiment 3

This experiment presents a decentralized scheduling algorithm for EV charging, figure 2.8 shows the experiment setup. It can be seen that the experiment setup is a HIL test bench. Part of the hardware is replaced by an equivalent computer model, and the model running on the computer has been ensured can interact with that the external hardware.

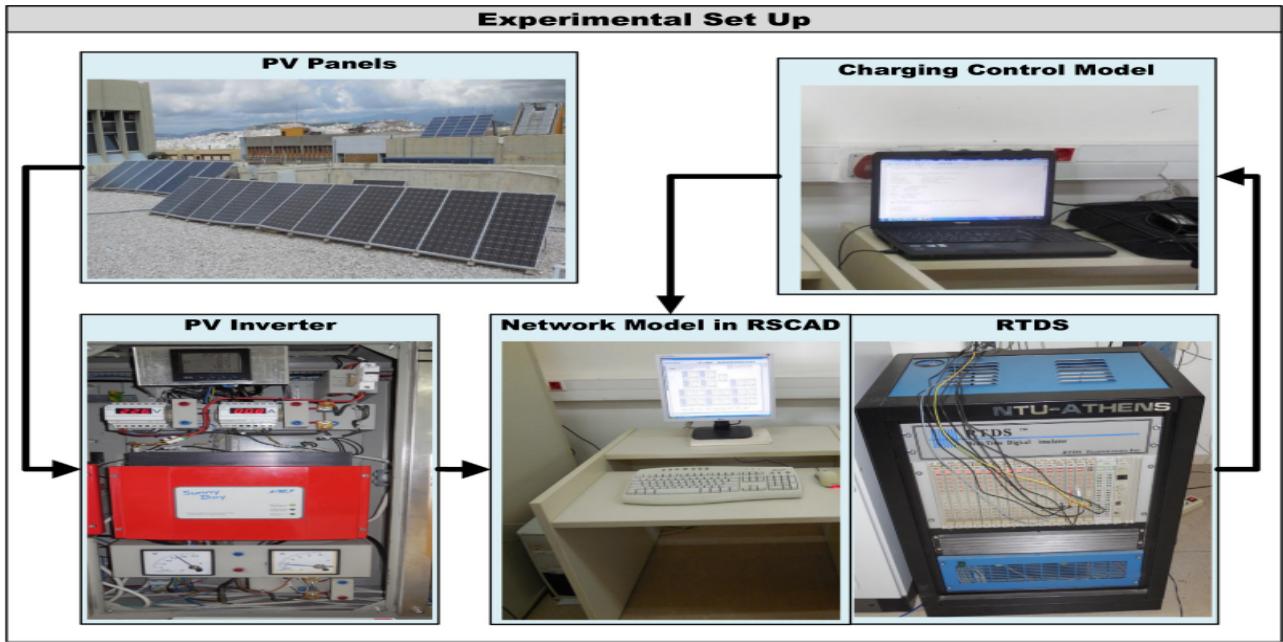


Figure 2.9: Experiment 3 setup[39]

In this experiment, EVs are divided into two categories: responsive and non-responsive, and the decentralized scheduling algorithm is implemented to formulate a charging strategy according to the current power generation demand and the forecast of future photovoltaic power generation. This charging strategy enables Responsive EVs adjust their charging schemes based on 'unresponsive' EVs need.

The HIL test bench consist of a real-time simulator, photovoltaic inverters, PV panels and RSCAD. In this setup, the real-time simulator and host computer are responsible to simulate the EVs charging model in real time and design the optimized charging schemes. The RSCAD is a user's interface which allows the user to design the model, run the simulation and analysis the HIL test. The PV inverter will input the real-time PV power generation into RSCAD in the form of signals, so that accurate prediction of future PV power generation can be made by the simulator and computer according to the real-time PV power generation. The forecast of future PV power generation will also be taken into account as an important factor when developing the optimal charging plan.

This experiment optimizes the charging of EVs through a decentralized algorithm, so that each EV owner can charge EVs at a lower electricity price, and PV generation is also considered when formulating the charging schemes. However, the advantage of this experiment is actually also the disadvantage. It is time-consuming and laborious to incorporate PV panels into the HIL test bench through PV inverters.

2.4.4 Summary

The following table summarizes the previous experiments and methodology in this thesis.

	Smart charging algorithm	Grid voltage fluctuation	HIL	PV&local load
experiment1	Yes	Yes	No	No
experiment2	Yes	No	No	No
experiment2	Yes	Yes	Yes	PV only
This thesis	Yes	Yes	Yes	Yes

Table 2.1: Summary

The experiment 1 and 2 do not use HIL real-time simulation, the hardware of experiment 1 is very expensive, and experiment 2 does not take into account the fluctuation of voltage with load changes. Experiment 3 uses HIL real-time simulation, however, the set up of the HIL test bench is challenging. Other studies related to HIL can be found through literature review, like [40][41]. Setup in [41] is also relatively costly and complicated. In [40], the grid simulation is achieved by the algorithm.

The simulations of this study were carried out with algorithm interfaced DRTS and used ePHASORsim for grid simulations. An interface script was written to read signals, call the smart charging algorithm to calculate the current setpoints, and modify the parameters in the MATLAB Simulink model in every time step.

3 System design

3.1 The schematic of HIL setup

The HIL setup adopted in this thesis was built by a former student named Lode[40]. The schematic of the setup is shown in figure 3.1.

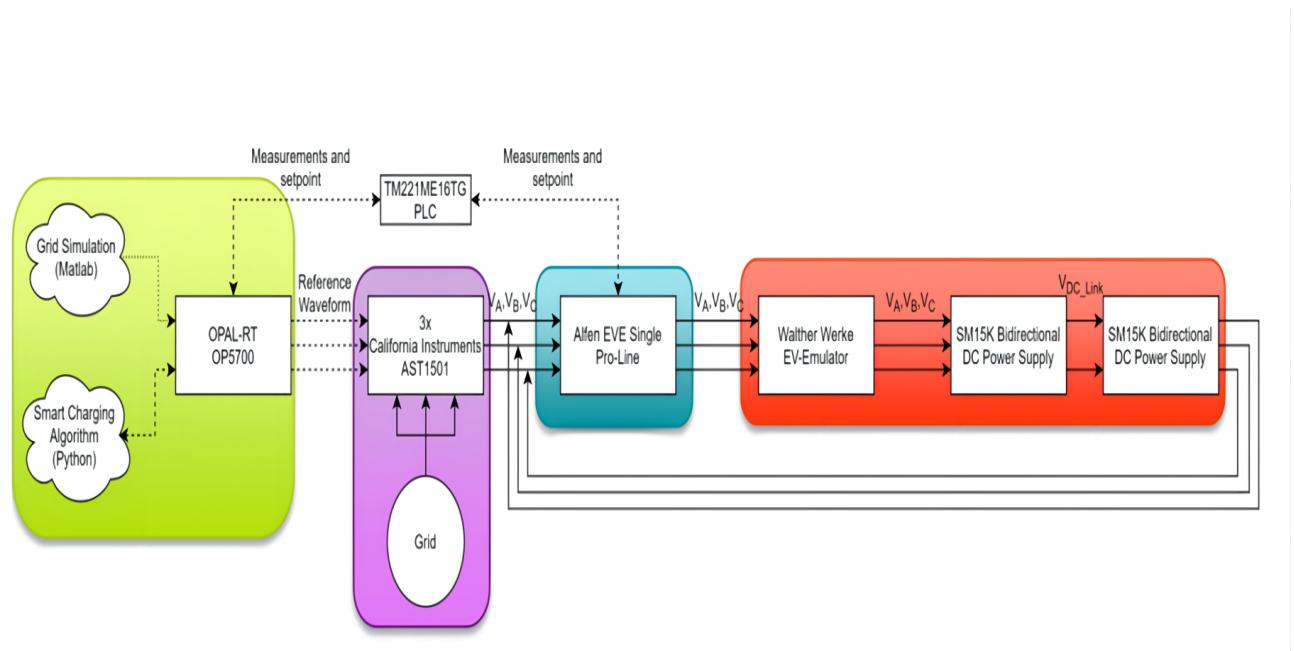


Figure 3.1: The schematic of HIL setup[40]

There are two kinds of flows in this schematic, one power flow, and the other is information flow. The solid lines in the schematic figure 3.1 represent power flows while the dotted lines represent the signal flow. The signal flow also contains analog signals and digital signals. The power in the power flows comes from two components: the California instrument and Delta DC power supply.

In the figure, all the hardware components involved in this schematic have been shown. The DRTS is OP4510 which has been mentioned in the previous part. The grid simulation model and the smart charging algorithm will run on it. More details will be discussed in section 3.2. The $3 \times$ California instruments AST1501 will be used to emulate a three-phase grid with controllable voltage, ranging from 0V to 400V. The voltage produced by California is determined by the grid simulation. The EVSE is connected to California. It can provide electricity to the EV in the real world. Lastly, there is the EV-emulator which determines whether the power can flow through the EVSE to the EV. The EV emulator used does not process any power by itself, and the power provided by California is insufficient, so an external load is needed. Therefore, two Delta are needed and used as an AC power source

Details about the function of each component are as follows:

(1) **OPAL4510**: As discussed in section 2.2.3, there are several requirements for DRTS, and the final decision is OP4510. The OP4510 is designed to run models on FPGA and CPU, it contains multiple cores, which can handle multiple subsystems at the same time. The OP4510's FPGA-based multi-rate architecture enables user subsystems to operate at sub- $7\mu s$ time steps. Therefore, as long as the time step is greater than $7\mu s$, the completion of every integration step within the real-time step can be ensured and the model will be executed with an integer multiple of a time step. The subsystems of a virtual model are running on OPAL4510, the details of this part will be introduced in section 4.2. Overall, the OP4510 is an affordable and

high-performance real-time simulator, and it is important that the OP4510 is ePHASORsim compatible, which means that grid simulation can be run on that real-time simulator.

(2) 3×California instruments AST1501: To simulate a three-phase distribution network, three California Instruments AST1501 power supplies were used. A California Instruments AST1501 output voltage is controllable between 0-400 V (RMS), in this thesis three AST1501s are required to output AC sine waves. The magnitude of the grid voltage should be calculated by the ePHASORsim in the model, and once the value has been calculated, the DRTS will tell this voltage value to AST1501 sources by using the analog output channels of the DRTS. To simulate a three-phase distribution grid, there should be a 120-degree phase difference between the three sine waves output by the three AST1501s. The amplitude of the three sine waves output by the DRTS is between 0 - 10 V, but the maximum output voltage of California is 400V (RMS). So, there will be a scale factor of $10 \times 230/400 = 5.75$, the analog output value multiplied by this scale factor should be the output voltage of the AC sources

(3) Schneider Electric TM221ME16TG PLC: In this experimental setup, the optimal current setpoints calculated by the smart charging algorithm on the host computer need to be sent to the DRTS, and then sent to the EVSE from the DRTS. But the EVSE and DRTS cannot communicate directly with each other because the OP4510 does not support Modbus TCP. Therefore, a third-party device is required to convert the current setpoints output from the DRTS to the form of Modbus TCP, which is able to be read by the EVSE. Therefore, Schneider Electric's Programmable Logic Controller (PLC) is used for this conversation between the EVSE and DRTS.

(4) Walther-Werke Type 2 EV-Emulator and Delta SM15K 15kW Bi-directional DC Power Supplies: Charging an EV is complicated because it needs to consider the cable type the charger type, and the communication protocol. This process is inevitable and that is why the EV emulator is necessary because the EV emulator is the device that can simulate the real communication behavior during the EV charging. As part of the experimental setup, it connects the EVSE and load. Although the EV emulator itself cannot provide any power, it decides whether the Power from the EVSE can flow into the load. In this thesis, a commercially available EV-emulator from Walther-Werke has been applied. This EV emulator is built into a handy and robust hand-held case and is provided with an integrated charging cable with a connected type 2 plug. It has a total of four states A,B,C,D, and the rated current corresponding to each state is 13 A, 20 A, 32 A, or 63 A. Therefore, the maximum current 16A in the cable will not be a problem.

The EV emulator described above is only used to mimic the communication between the EV and EVSE, and in a real EV, the power flows through the EV emulator and heads to the load. So, if real EV behaviors will be emulated, then besides the previously mentioned hardware, in the HIL testbed, some hardware needs to act as a load. However, the real EV battery is costly and unpractical, so the two Delta Elektronika SM500-CP-90 DC Power Supply are used as a load to draw the power which acts like a real EV. These are bidirectional DC Power Supply, voltage ranges from 0V to 500V, and current ranges from -90A to 90A. The maximum power that this DC Power supply can provide is 15kw. Since they are bidirectional DC power sources, in this, they are connected back-to-back to form an AC load that can circulate power back to the (emulated) grid.

Another reason for using these two DC loads is because the California AC sources provide insufficient power, the AC source can only provide energy of no more than 10kw, and these two power supplies are connected back-to-back to form an AC load that can provide enough power and circulate power back to the (emulated) grid.

(5) Alfen EVE Single Pro-line: It is an AC charger that supports single-phase and three-phase charging and provides a voltage between 230V and 400V. In the lab, the EVSE can be controlled by software named ACE Service Installer. Once the charger and the computer with the software installed are connected as a local network, the information of the charger can be observed on the software, as shown in the following figure. The charging process can also be observed through the software.

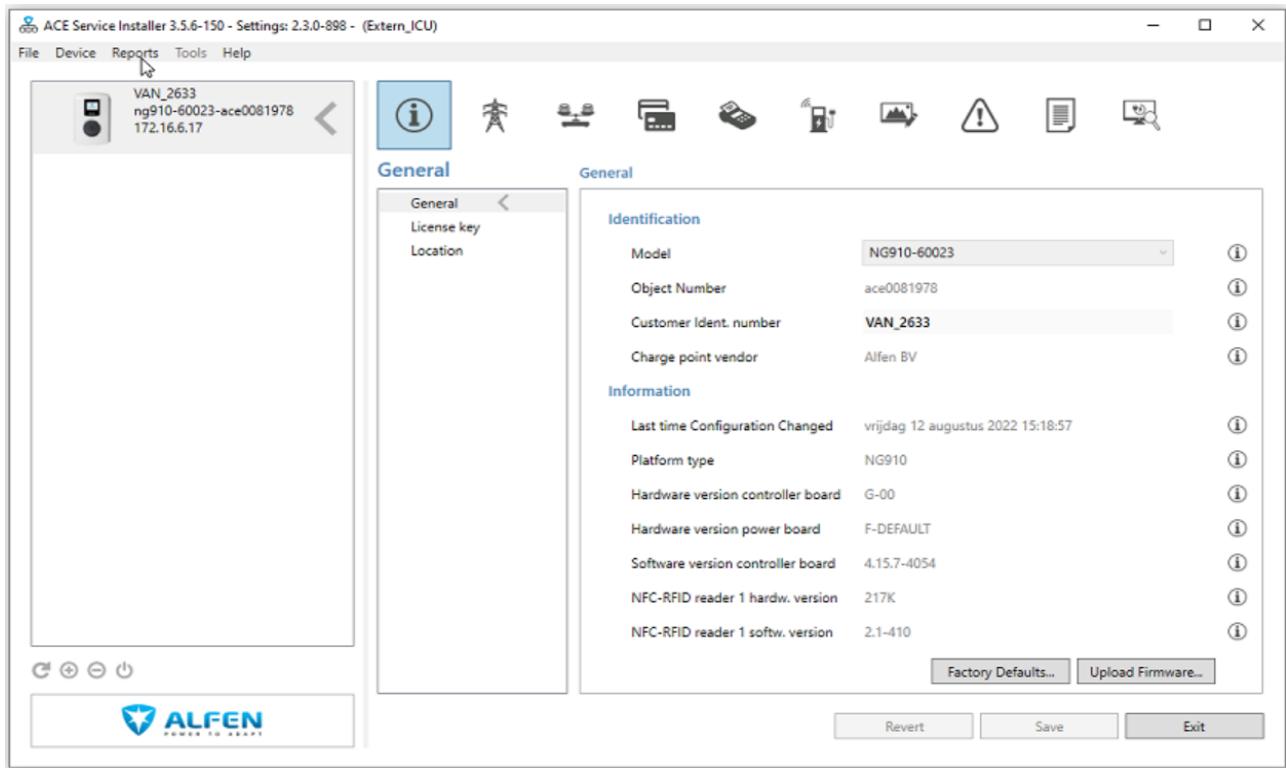


Figure 3.2: The observation on ACE Service Installer

However, although the Alfen Eve Single Pro-line is used for smart charging, it cannot talk to the DRTS directly, therefore, there must be a third-party device that can act as a "bridge" between DRTS and EVSE. Luckily, this EVSE has been designed with smart-charging applications and thus can communicate with 3rd party devices.

3.2 Preparation of model

3.2.1 Before the simulation

How to install the DRTS physically and how to configure the opal file will not be introduced too much here, but to run the model normally on DRTS, then there are several points that need to be noticed before the simulation. The first thing is to set the default GNU Compiler Collection (GCC) compiler which is a compiler used to compile C programs. Since the Intel compiler is not available, the GCC compiler will need to be set, otherwise, the model cannot be compiled by the DRTS. In order to select the GCC compiler, a user variable can be set in the RT-LAB variables list under the "Variables" tab of the model. Secondly, upload the PV profile and local profile under the "File" tab of the model before the execution, since the OPfromfile block will be used to read the real historical data of local load consumption and PV generation. The last thing is setting the number of ePHASOR threads in the RT-LAB variables list. The number of ePHASOR threads needs to be manually allocated to the model.

3.2.2 Configure the I/O interface

As discussed in the previous section, the DRTS needs to output the analog out signals and give the signals to the California instrument. Therefore, an I/O interface must be established in the RT-Lab project. The type of I/O interface in this thesis is OPAL-RT Board because the OPAL-RT Board provides the Analog Out functionality and Digital In functionality, which allows the OPAL-RT to output analog signals and input digital signals. When the OPAL-RT Board is configured, manually add the channel number and assign which signal is input and output from which channel.

The I/O location is identified by a group number. The information about OP4510 I/O channels is listed below. The DRTS used in this thesis has 16 input and output channels for the analog signal and 32 input and output channels for the digital signal.

I/O channels	Group/Number
32 digital inputs	IO GROUP 1 / SECTION A (CH 00-31)
32 digital outputs	IO GROUP 1 / SECTION B (CH 00-31)
16 analog inputs	IO GROUP 2 / SECTION A (CH 00-15)
16 analog outputs	IO GROUP 2 / SECTION B (CH 00-15)

Table 3.1: I/O channels

The configuration of OPAL-RT Board has been shown in the following figure.

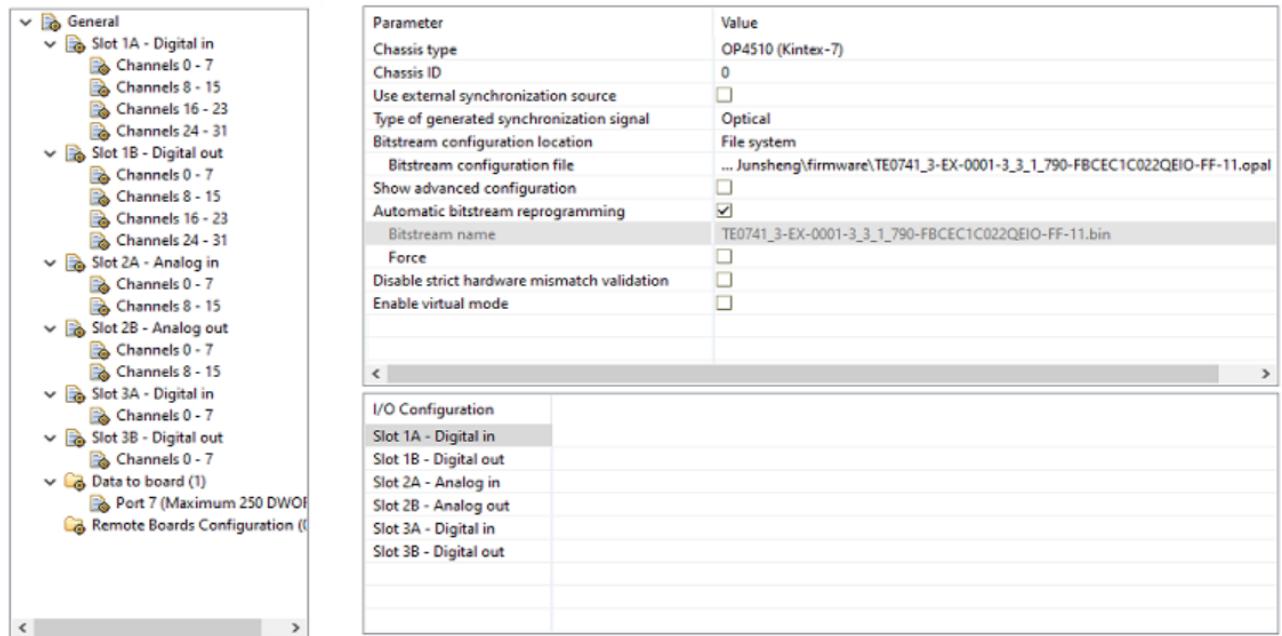


Figure 3.3: OPAL-RT Board

3.3 Build the simulation Model

Simulink modules running on DRTS are composed of three types of subsystems, and all content must be placed in these three types of subsystems. They are listed below:

1. Console: There is always one and only one Console subsystem inside a model. The Console subsystem is the subsystem operating on the host computer that enables users to interact with the system. Users can also view data in this subsystem by using blocks like scope or display.

2. Master: There is always one and only one Master subsystem inside a model. The main calculation in this model should be carried out in this subsystem. OPAL-RT will definitely allocate a core for the calculation in the Master

3. Slave: This subsystem is an optional subsystem and there can be several Slave subsystems inside the model. If the calculation in the Master subsystem is too complicated, part of the calculation can be placed in the slave subsystem to ensure that the calculation is finished in time.

During the real-time simulation, the Simulink model will be converted to the C code, the Console will run on the host computer while the other subsystems run on the DRTS. In this thesis, the whole model consists of only two subsystems, SC_Console and SM_Master. The reason for not using a slave subsystem is that the DRTS in the lab only has 2 cores, so one core can be used by the master and the other will be occupied by the ePHASORSim. Although using a slave subsystem can allocate another core to deal with part of the calculation, however, since one core has been occupied by the ePHASORsim, there are no available cores for the slave subsystem. The screenshots of the subsystems can be seen in appendix A.

In the following essay, what is inside the SC_Console and SM_Master will be introduced in detail.

3.3.1 Console

The console in the model has the following key functions:

- 1) Monitoring of signals on a scope, the observed signals include Bus voltage, EV SOCs, Current, Grid import, Local load consumption, PV generation, EV power, and analog output signals.
- 2) Possibility to manually set the grid voltage by using the block slider gain.
- 3) Possibility to manually set the charging current of all EVs, but this only happens under an uncontrolled charging scenario.
- 4) Activate the Smart charging or deactivate it.

3.3.2 Master

Since there are only two cores allowed to be used, and one of them has been allocated to ePHASOSsim, then there is only one core available for the subsystems. Therefore, there is only one subsystem exists in the model, which is SM_Master. All the computation, input from the console, and I/O signals must be handled by this subsystem. The specific functions of the Master are as shown below:

key functions include:

1. Generating three sine waves based on the bus voltage derived by the grid simulation from ePHASORsim. The three sine waves will be sent to California as reference waves. The three sine waves are generated by the master subsystem with a frequency of 50 Hz and 120-degree phase shift. The amplitude of the sine waves sent to California is determined by the results of the grid simulation multiplied by 5.75 (as explained in section 3.1)

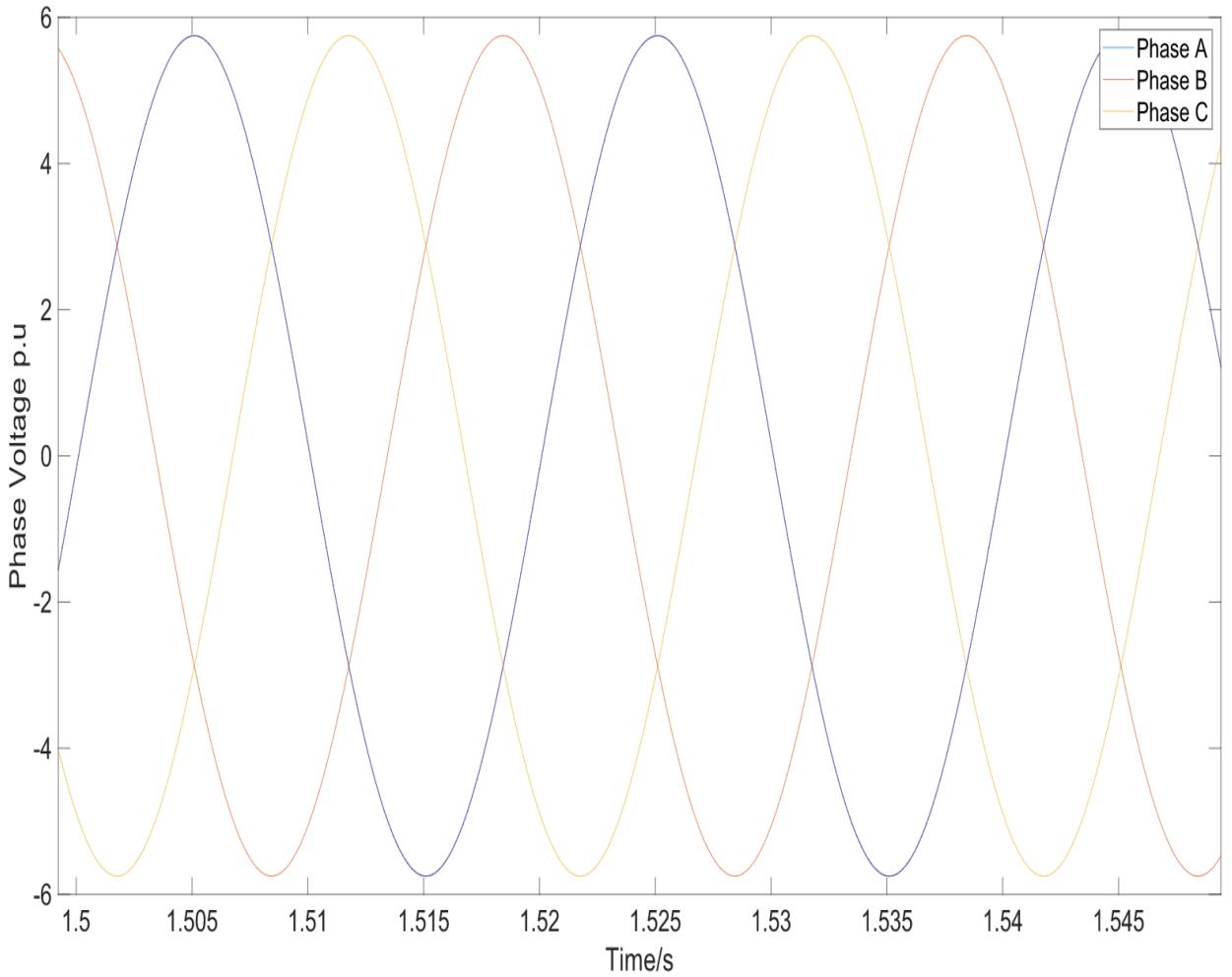


Figure 3.4: Reference wave-forms

2. Emulating the charging behavior of an EV.

In order to realistically emulate an EV, then there are some points that must be paid attention to. The first is about the charging current of the EV. When the SOC of the electric vehicle is in different stages, the charging current value is also different. The EV charging process is mainly divided into three stages, $SOC=1$ (when the battery is already full), $1 \geq SOC \geq 0.9$ (Constant Voltage stage), and $0.9 \geq SOC$. If the battery is full ($SOC=1$), the EV will stop charging. Second, for the battery's Constant-Voltage charging stage, the current set points will be lower than the given set points. The current set-points will be reduced as a function of SOC.

The following formulas describe the functional relationship between the current value and SOC at different stages of SOC.

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

The following figure introduces the constant voltage and constant current charging stage during EV charging.

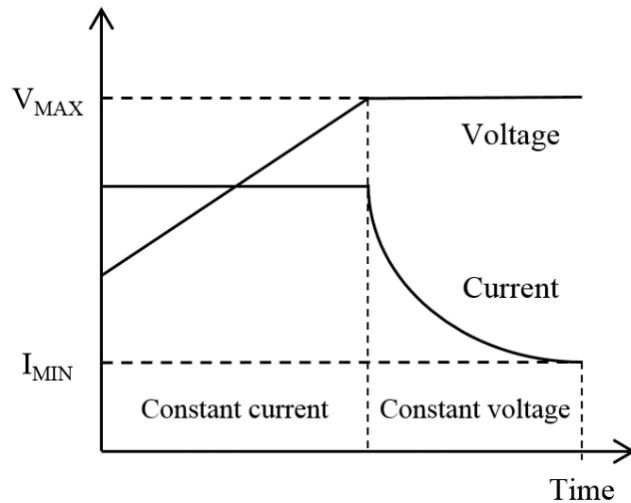


Figure 3.5: Constant voltage and constant current charging stage[26]

3. Reading the load and PV profiles by using the OPfromfile block. The load and PV profiles read by the OPfromfile are created to simulate the local load consumption and PV generation during the EV charging:

For some EV-connected points, the influence of local load and PV generation on grid voltage also needs to be considered. In this thesis, real historical data is used. Before executing the model, the real historical data shall be saved in a .mat file and uploaded on the DRTS. During the simulation, data will be read by Opfromfile block.

4. Calculating the current SOC of the EVs connected to the distribution grid:

Once the code detects that a new EV is connected to the grid, the EV's SOC, battery size, and other parameters should be updated immediately. And once the EV leaves, all parameters must be cleared in the model. The SOC is calculated as the integral of power, divided by the battery size multiplied by 3600(The unit is kwh)

5. Running the grid simulation by using the ePHASORsim solver mask block. Get the bus voltage values:

The grid simulation is run using the ePHASORsim solver block, the input of this block is the load power, and the output is the voltage at every node in the grid. The original plan is using Powerfactory together with ePHASORsim to run grid simulation on DRTS. However, this plan has several disadvantages. Firstly, an export definition file is missing. To export the data from the PowerFactory model to a compatible .xml file, A definition file is required, which specifies which components and which parameters of these components will be exported. Unfortunately, the definition file officially provided by OPAL is only applicable to the PowerFactory version 2019, so a definition file needs to be created for this thesis. Secondly, ePHASORsim has limitations on the version of PowerFactory. The XML version compatible with ePHASORsim is the 2019 version, but the latest PowerFactory version is 2022, and the oldest .xml version that can be exported in PowerFactory 2022 is the version 2020. So, the existing PF version is not compatible with ePHASORsim at all. Last but not least, not all components in the PowerFactory model are supported by ePHASORsim. In the model shown in Figure 3.2, including external grid, fuse and PV system are not supported by ePHASORsim. Even if these components are successfully exported to the XML file, the grid simulation will fail in the end.

Therefore, in this thesis, the input data format would be Excel only. ePHASORsim's strict restrictions on the version of PowerFactory and definition file will no longer be a problem. However, it is still necessary to find a way to replace those components that are not supported

by ePHASORsim. Finally, the external grid is decided to be replaced by a voltage source, the fuse is replaced by a wire, the net value of PV system and the local load consumption is replaced by one load. The value of the load in the grid simulation can be negative, which also solves the problem that sometimes the PV generation is larger than the local load consumption. One thing to note, since the parameters of the wires given in the PF model are the parameters of the positive sequence instead of ordinary parameters when creating the excel sheet, the type of wire must be selected as Three Phase Line with Sequential Data. In this way, the rural grid can be accurately simulated

There are two types of chargers in this grid, charger at home and charger at semi-public. The difference is that the charger at home and photovoltaic power generation, the local load is connected at one point in the grid, and the charger at semi-public is connected at one point in the grid alone. The modified grid model is shown in the following figure 3.3, it can be seen that Chr_Home_0,Chr_Home_1,Chr_Home_2(Charger0,1,2 at home) are located at node nd_5355967 and Chr_Semi_Public0(Charger0 at semi-public) is located at 310387526.

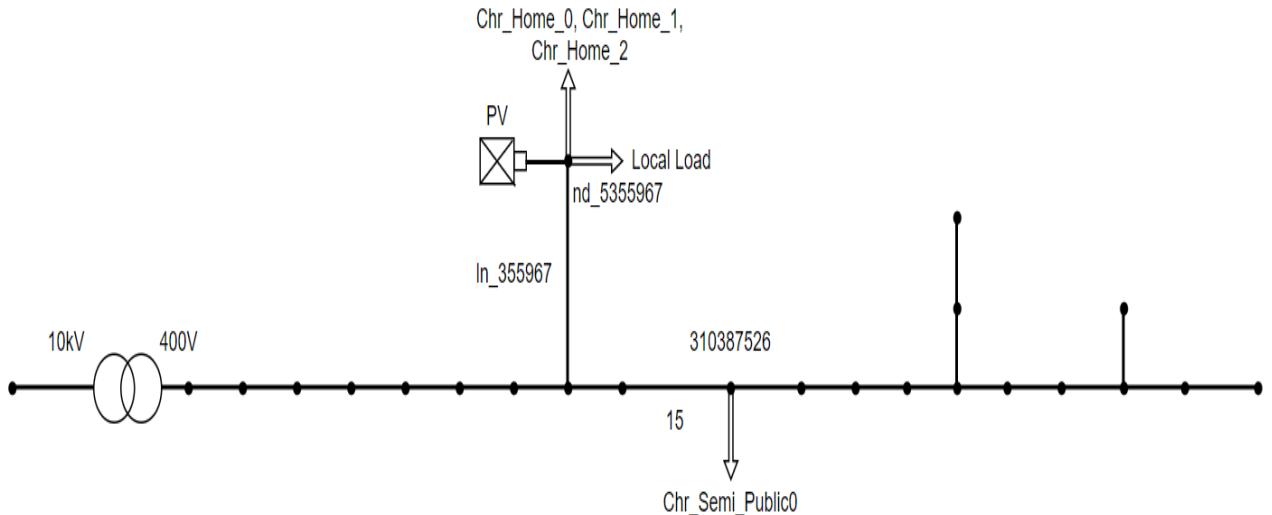


Figure 3.6: Grid Model

In fact, ePHASORsim can also simulate the current between various points in the power grid, the losses on the line, but these are not used in this thesis.

6. Record the simulation result every second, and save it into a .mat file on the host computer:

During the real-time simulation, important data needs to be recorded and saved. The OpWriteFile block has the function of recording various signals once per second. After the simulation, all data will be recorded in a .mat file for data analysis.

3.3.3 Two "simulation time"

The top-level of the Simulink model has two subsystems: master, and console (user interface). All the calculations, grid simulation as well as the I/O signals handling are carried out in the master system. One thing that needs to note is during the simulation there will be two 'simulation time', one for the host computer and the other one for DRTS. This is because the

SM _ subsystem and inside any other SS _ subsystems are run in real-time in the simulator while the SC _ console is on the host computer.

During the build process, RT-LAB converts what is inside the SM _ subsystem and other SS _ subsystems into C code sends it to the simulator, and runs it in real-time. But this is different from what is inside the SC _ subsystem. This subsystem is not involved in the building process so is not converted into C code for real-time simulation. This one stays on Windows as a graphical user interface and does not run in real-time (it runs as fast as possible in Windows). However, the data that can be seen in the SC _ subsystem are correct (amplitude, phase, frequency, etc) but are not synchronized with the real-time simulation. The connection between the simulator and the Windows computer is done through an ethernet cable which sends/receives data through TCP/IP and data buffering. So, the data received or sent in the SC _ subsystem are correct but are asynchronous compared to the real-time simulation. There is no possibility to make them synchronous because of the type of connection. In conclusion, the console allows users to monitor the results of the real-time simulation, but the data are sent/received asynchronously. To get real-time data, the best way is to connect the simulator to an oscilloscope. It is also possible to record data in real-time using the DataLogger or the OpWriteFile block.

3.4 Functions of smart charging algorithm

The implemented smart charging algorithm is from previous work in project OSCD [42]. As mentioned in the previous chapter, the smart charging algorithm written in Python needs to adapt to the grid simulation model. The smart charging algorithm runs on the host computer and sends the optimized current set-point to the Simulink model which runs on the DRTS. The smart charging algorithm calls Gurobi, a mathematical optimization solver named to calculate the optimal setpoint for each EV connected to a node.

3.4.1 objective functions

The main objective function of the algorithm is as follows. The ultimate purpose of the smart charging algorithm is to minimize the total cost during the charging. The current set-point would be given based on that principle. Another important principle of smart charging is to minimize the unfinished charging which means all the EVs connected to the nodes in the distribution grid should be charged as much as possible, otherwise there will be a penalty for the charger point.

$$\begin{aligned}
 Min.C_n^{opt} = & \sum_{j=1}^J \left(B_{n,j}^a + d_{n,j} - B_{n,j,T_j^d} \right) C_{n,j}^p + \Delta T \sum_{t=1}^T p_{n,t}^{PV} C^{PV} \\
 & + \Delta T \sum_{t=1}^T \left(p_{n,t}^{g(imp)} C_t^{e(buy)} - p_{n,t}^{g(exp)} C_t^{e(sell)} \right) \\
 & - \Delta T \sum_{t=1}^T \sum_{j=1}^J p_{n,j,t}^{r(up)} C_t^{r(up)} + p_{n,j,t}^{r(dn)} C_t^{r(dn)}
 \end{aligned} \tag{2}$$

In the following table, all the parameters of objective functions have been listed, based on [43]:

From the above formula, we can see that the penalty consists of four parts[43]:

Parameters	Meaning	Unit
C_n^{opt}	EV charging cost at node n	€
$B_{n,j}^a$	Energy in the battery when j^{th} EV connects to a car park	kwh
$d_{n,j}$	The demanded energy of j^{th} EV	kwh
B_{n,j,T_j^d}	Energy in the battery when j^{th} EV leaves at T_j^d	kwh
$C_{n,j}^p$	Penalty for EV not fully charged	€/kwh
$p_{n,t}^{PV}$	PV energy generated at time t at node n	kwh
C^{PV}	Cost for buying third-party PV power	€/kwh
$p_{n,t}^{g(imp)}, p_{n,t}^{g(exp)}$	the EV at node n import or export power	kw
$C_t^{e(buy)}, C_t^{e(sell)}$	Cost of buying or selling power from/to the grid	€/kw
$p_{n,j,t}^{r(up)}, p_{n,j,t}^{r(dn)}$	Reserve power capacity for up and down regulation by j^{th} EV at time t, node n	kw
$C_t^{r(dn)}, C_t^{r(up)}$	Cost for offering capacity reserves for up and down regulation at time t	€/kw

Table 3.2: Objective functions parameters

1.The penalty paid to the customers if the energy demand $d_{n,j}$ is not met by the departure time.

2.Sometimes the PV generation power might be provided by thirty-party. Therefore, if this thirty-party provided PV power to charge the EV. Then pre-determined contractual cost of C^{PV} will be charged.

3.The cost of buying and selling energy from the grid or to the grid based on the energy prices. Buy energy from the grid as much as possible when the price is low and sell the power to the grid when the price is high.

4.Income obtained from offering reserve capacity

3.4.2 sub functions

This section explained how the parameters/variables are defined or how are they related to each other.

$$B_{n,j,t} = B_{n,j}^a + \Delta T \sum_{t=T_j^a}^t (p_{n,j,t}^{e+} \eta_{n,j}^{ev}) \quad \forall t \in n, j, [T_{n,j}^a; T_{n,j}^d] \quad (3)$$

$$B_{n,j,T_{n,j}^d} = B_{n,j}^a + \Delta T \sum_t (p_{n,j,t}^{e+} \eta_{n,j}^{ev}) \quad \forall t \in n, j, [T_{n,j}^a; T_{n,j}^d] \quad (4)$$

$$p_{n,j,t}^{e+} = i_{n,j,t}^{e+} \times V_{n,t} \quad (5)$$

$$S_{n,j,t} = \frac{B_{n,j,t} - B_{n,j}^{\min}}{(B_{n,j}^{\max} - B_{n,j}^{\min})} \quad \forall t \in n, j, [T_{n,j}^a; T_{n,j}^d] \quad (6)$$

$$p_{n,t}^{g(imp)} = V_{n,t} \times i_{n,t}^{g(imp)}, p_{n,t}^{g(exp)} = V_{n,t} \times i_{n,t}^{g(exp)} \quad (7)$$

The parameters of the sub functions are listed below

Parameters	Meaning	Unit
$B_{n,j,t}$	Energy in battery of j^{th} EV at node n at time t	kwh
$B_{n,j}^a$	Energy in the battery when j^{th} EV connects to a car park	kwh
$\eta_{n,j}^{ev}$	charging efficiency EV	
B_{n,j,T_j^d}	Energy in the battery when j^{th} EV leaves at T_j^d	kwh
$p_{n,j,t}^{e+}$	Charging power	kw
$i_{n,j,t}^{e+}$	Charging current	kw
$S_{n,j,t}$	SOC of j^{th} EV at node n, time t	
$B_{n,j}^{\min}$	Maximum power in the battery of j^{th} EV at node n	kwh
$B_{n,j}^{\max}$	Minimum power in the battery of j^{th} EV at node n	kwh

Table 3.3: Sub functions parameters

3.4.3 Constraints

In optimization, some constraints are imposed on the variables of the objective functions. The Gurobi shall consider those constraints when it runs to calculate the optimal current set-points during the optimization.

Constraints on EV battery, When EV is connected:

$$B_{t,j,n} \geq B_{j,n}^a \quad \forall t \in n, j, [T_{n,j}^a; T_{n,j}^d] \quad (8)$$

$$B_{t,j,n} \leq \text{Min} \{ d_{j,n} + B_{j,n}^a, B_{j,n}^{\max} \} \quad \forall t \in n, j, [T_{n,j}^a; T_{n,j}^d] \quad (9)$$

when EV is not connected:

$$i_{n,j,t}^{e+}, p_{n,j,t}^{e+}, p_{n,j,t}^{r(up)}, p_{n,j,t}^{r(dn)}, B_{n,j,t} = 0 \quad \forall t < T_{n,j}^a \& t \geq T_{n,j}^d \quad (10)$$

Constraints on EV charger: For each EV charger, the lowest allowed charging current is 6A because of AC charging standards (IEC 61851) and the current value should be integer with step of 1A ($i_{n,j,t}^{e+}$ is the charging current)

$$(i_{n,j,t}^{e+} = 0) \text{ OR } (i_{n,j,t}^{e+} \geq 6) \quad \forall t \in n, j, [T_{n,j}^a; T_{n,j}^d] \quad (11)$$

Constraints on prices: Buying energy costs more than selling it.

$$C_t^{e(\text{buy})} \geq C_t^{e(\text{sell})} \quad (12)$$

4 Integrating Algorithm with Model and Hardware

In the previous section, the functions and constraints of the algorithm have been introduced [43]. In this section, how to adapt the smart charging algorithm written in Python to the Simulink model on DRTS, and how to integrate the system will be introduced.

4.1 The adaption of algorithm

The smart charging algorithm used in this project is already well established. The summary of functions of the smart charging algorithm is shown in the following figure.

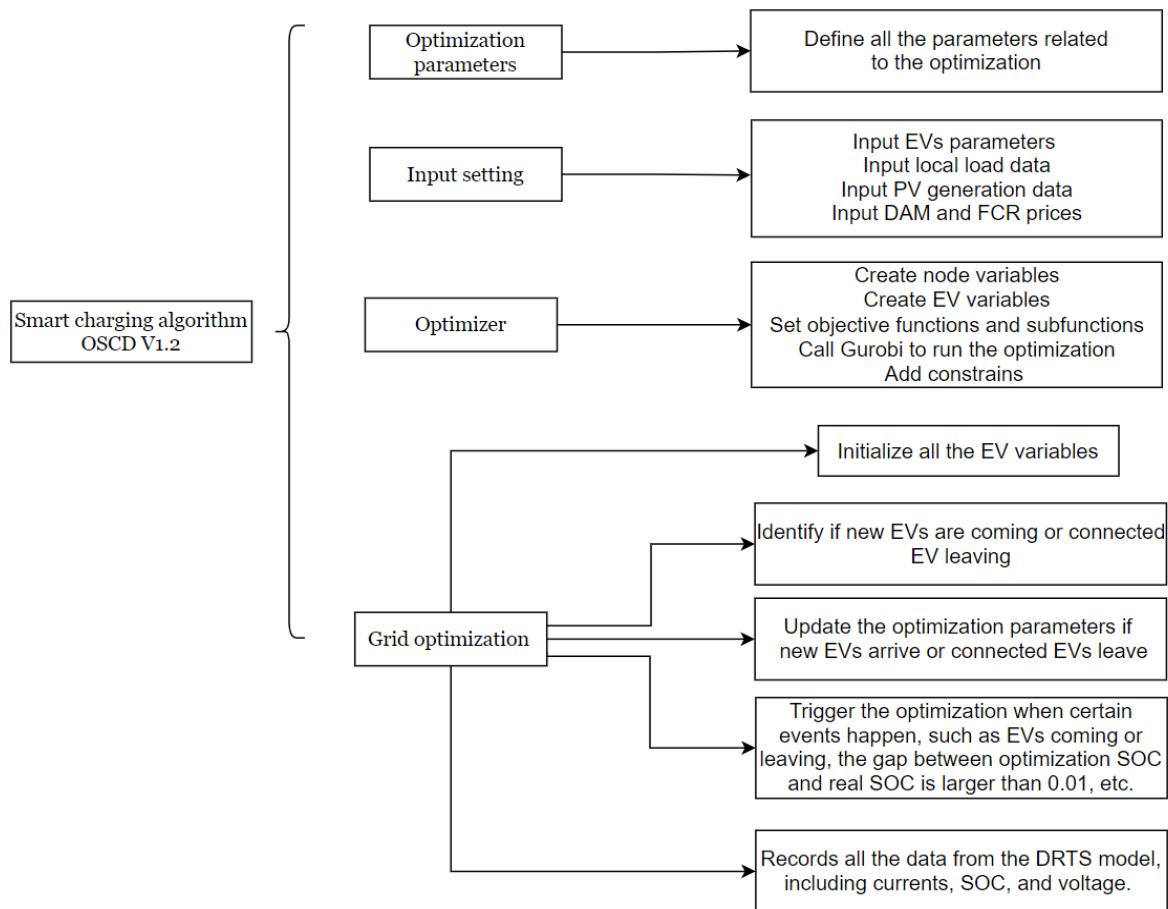


Figure 4.1: Algorithm functions

This smart charging algorithm was originally used to optimize the charging behavior of all electric vehicles connected to the grid within a week, according to their arrival SOC, connection time, local load consumption, photovoltaic generation, and other factors to calculate and give the optimal charging current point. However, this algorithm cannot fit in the HIL real-time simulation. Therefore, another script needs to be written, so that the intelligent charging algorithm and its functions can be called at any time, and the data simulated by the model on the DRTS can also be read by the script. Luckily, The DRTS OP4510 has a set of Application Programming Interfaces (API) called RT-Lab API. Once the RT-Lab API is called by the python script, a series of operations can be performed by this script based on the simulated model. In this thesis, Only the following functions are called:

1. Read the value of a signal in the model in the simulation
2. Set the parameters of the model in the simulation

3. Read the simulation time of the model to make the code run in sync with the model
4. After the simulation is over, the DRTS will be automatically recharged, and the model will be closed to record the data

The following flowchart explains in detail how the smart charging algorithm interacts with the DRTS model. They run in parallel, synchronously, only variables and parameters are exchanged through the interface script.

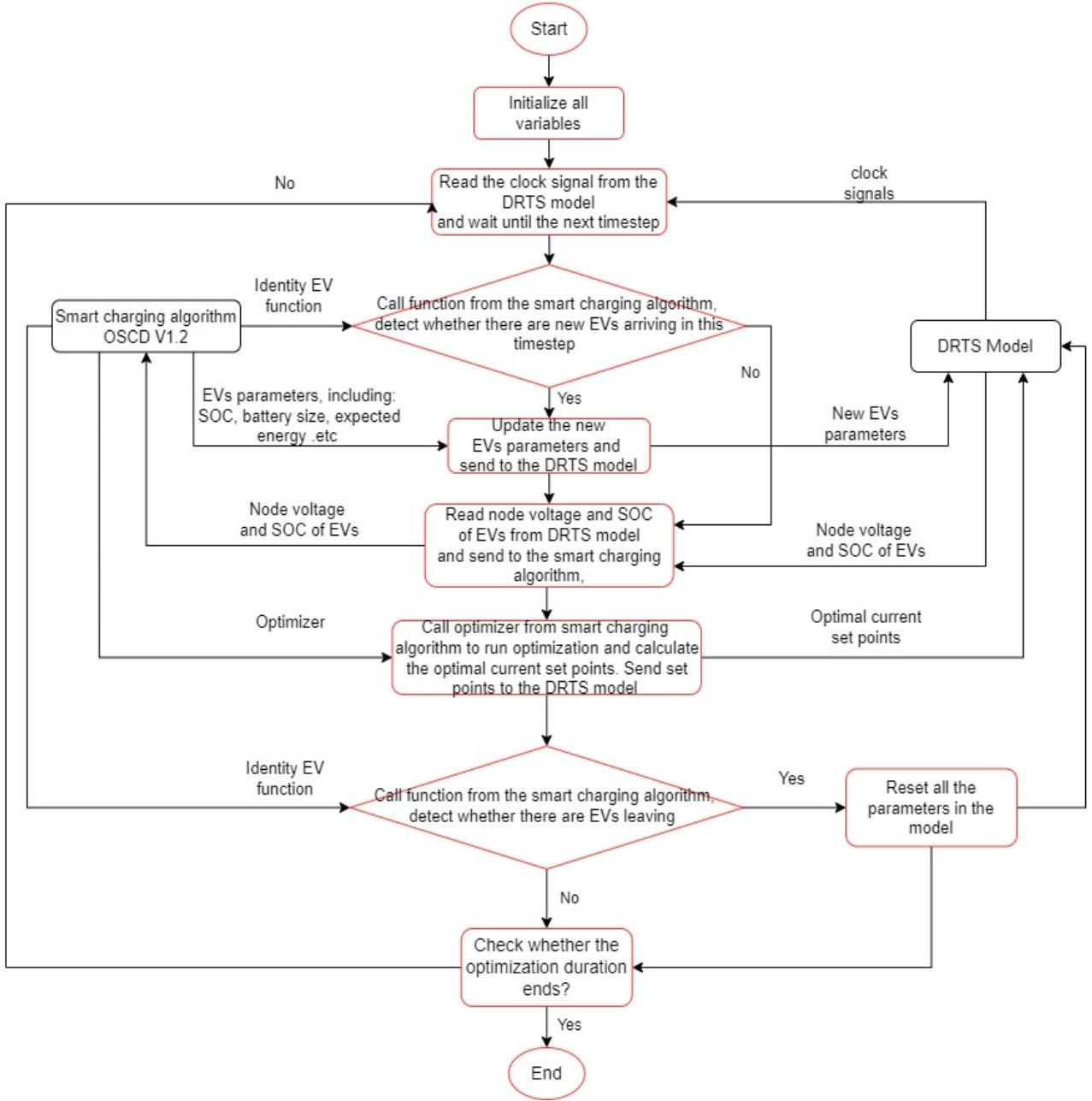


Figure 4.2: Flowchart of interaction

The red blocks in the middle represent the functions of the interface script, and the arrow in the flow chart and arrows in the flow chart indicates the direction of information transfer. First of all, the script will initialize all the parameters related to optimization. After that, the script will first read the clock signal from the DRTS model, wait for dozens of seconds to enter the next timestep, and then start to detect whether there are new EVs connected to the grid. This is because the running order of python and the model is to run the model on the

DRTS first, and then run the script, so the running of the script will be a few seconds behind from the beginning. In order to make them run synchronously, the script needs to wait for the model. Then, the updated EV parameters including SOC, battery capacity, etc. will be sent to the optimizer of the smart charging algorithm and DRTS model, and the node voltage at this moment will be sent to the optimization script from the DRTS model. Based on parameters like SOC, node voltage, and energy prices, the smart charging algorithm will calculate the current set-points. After all, optimizations finish, the algorithm sends the calculated current setpoints to the model on DRTS, which results in a change in grid voltage. The time step is one minute, and this loop is repeated every time step.

In this way, the interaction of data between the model and the intelligent charging algorithm is realized.

4.2 Integrate the hardware

In the lab, real devices are connected together as shown in the figure below:

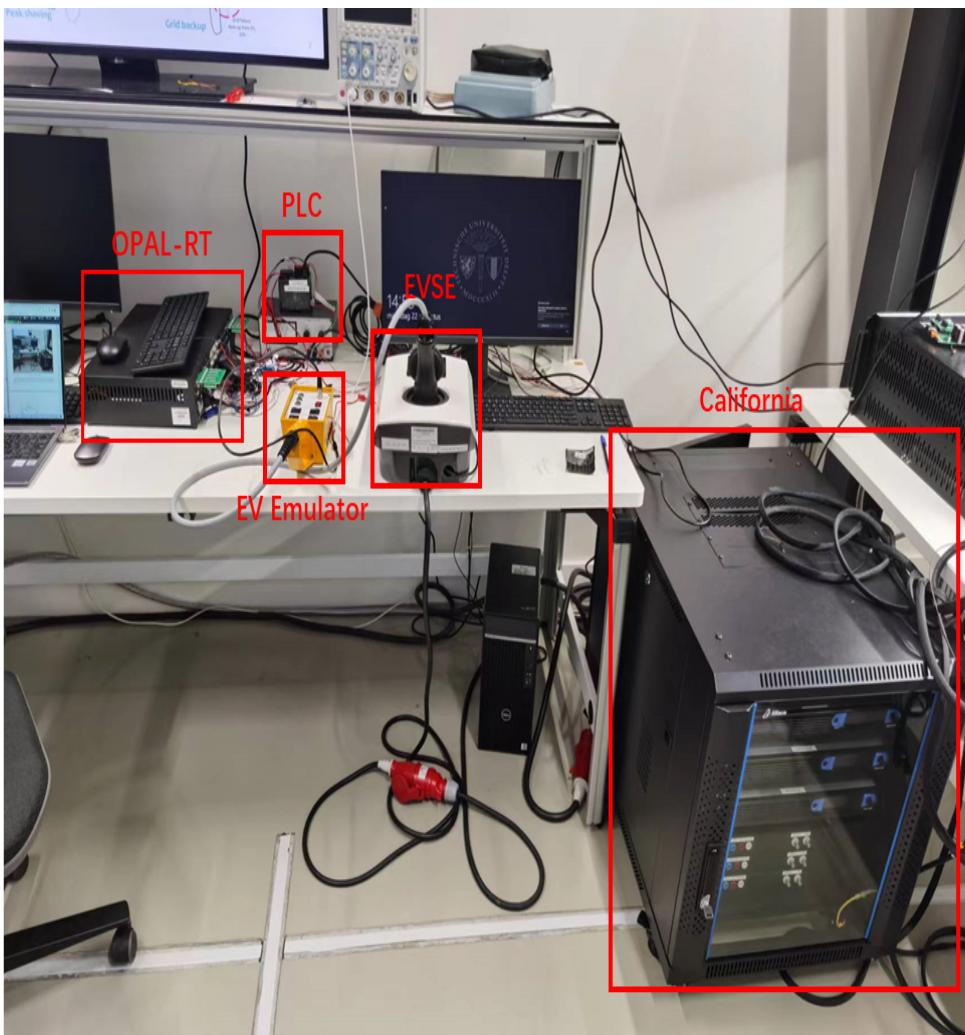


Figure 4.3: Photo of all components

Because of the lack of Delta-Elektronika SM500-CP-90 Bidirectional DC Power Supply, the power circulation cannot be achieved, so the remaining part only consists of PLC, OPAL-RT, EVSE, 3xCalifornia instruments, and EV emulator, the schematic is shown in the following figure.

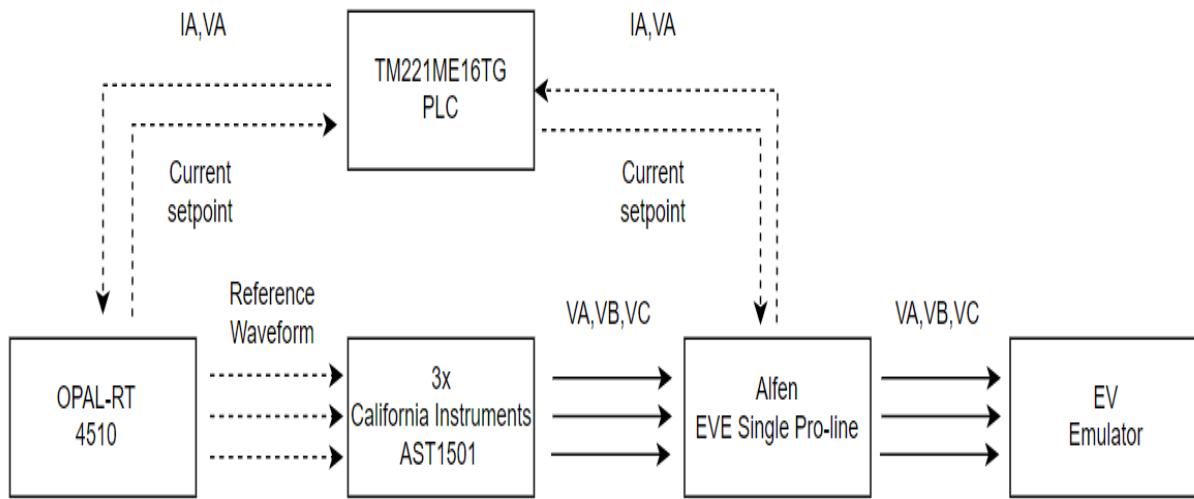


Figure 4.4: Schematic without Delta

According to Figures 4.5 and 4.6, there are 5 connections in the schematic, the connection between OPAL-RT 4510 and California instruments, the connection between California instruments and EVSE, the connection between EVSE and EV emulator, the connection between EVSE to PLC and the connection between PLC and OPAL-RT. The connection between California and OPAL-RT, as mentioned in 4.24, is for OPAL to transmit the reference voltage values to California. The connection between OPAL-RT and PLC is similar to the former connection. During the real-time simulation, the OPAL needs to send the current setpoints calculated by the smart charging algorithm to the PLC. These two signals are both analog signals output by OPAL-RT.

The EVSE connects to the EV emulator with a type 2 AC charging cable which as shown in the following figure.



Figure 4.5: Type 2 plug[44]

As described in chapter 3. The reason for connecting EVSE to an EV emulator is to mimic the communication behavior during smart charging. By monitoring Control Pilot (CP) signal, the charging state and the maximum charging current from the EVSE can be known during smart charging.

A voltage of 12 V will be applied to the CP and Protective Earth (PE) when the EV emulator detects that the cable has been plugged into EVSE. And the resistor between CP and PE will be activated in order to reduce the voltage. The value of the activated resistance and the applied voltage will be decided by the charging state of the EV, as shown in Table 4.2.

State	Resistances	Voltage
EV not connected	$\infty \Omega$	+12V
EV connected, not ready	2740 Ω	+9±1V
Ready	822 Ω	+6±1V
Ready, ventilation required	246 Ω	+3±1V
Shut off		0V
Error		-12V

Table 4.1: Values of voltage and resistance for different state [45]

So, it can be seen from the previous table that When the EV is in the state "EV connected, not ready", "Ready" and "Ready, ventilation required", the CP signal will be a pulse-width modulated (PWM) signal. In the lab, the CP signals can be clearly observed by an oscilloscope. There is a relationship between the duty cycle of the PWM signals and the maximum current, as shown in Table 4.3.

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

Table 4.2: CP duty cycle to maximum charging current[45]

The most important connection of the system integration is the connection between PLC and EVSE. In section 3.1, Schneider Electric Programmable Logic Controller (PLC) is briefly introduced, it is a bridge between the DRTS and EVSE, because these two devices cannot directly communicate with each other. In this thesis, as a bridge, PLC firstly needs to read the analog signal coming from the DRTS, convert this to Modbus TCP format, and send it to the EVSE. Secondly, the PLC needs to read the current and phase voltage values from EVSE and convert them from a Modbus TCP format to digital signals which can be read by the DRTS. The voltage and current can also be monitored by the software ACE service installer, as shown in the following figure.

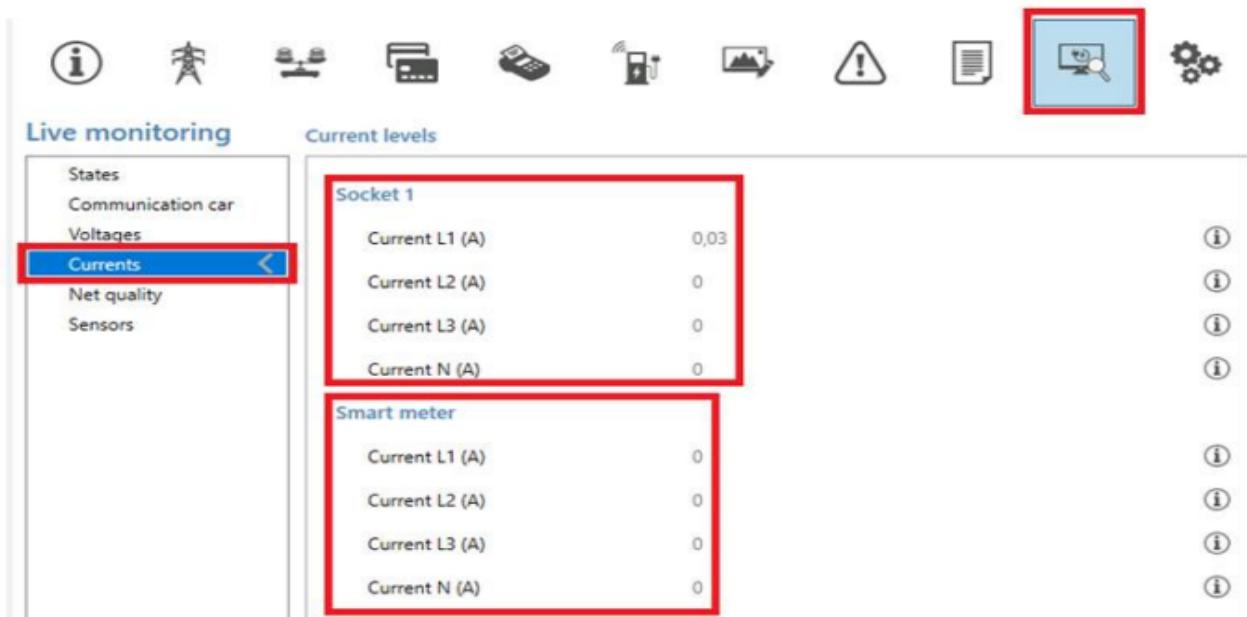


Figure 4.6: Alfen live monitor

The PLC can be programmed by using Schneider Electric's EcoStruxure Machine Expert software. No matter whether the PLC is ready to send setpoint to EVSE or read current value and voltage value from EVSE, it needs to be programmed to read/write to the correct Modbus registers of the EVSE. The address of the EVSE registers can be found in a Modbus Register table in appendix B. The part of the figure can be seen in the following figure.

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 4.7: Modbus registers[46]

In the Channel Assistant of Schneider Electric's EcoStruxure Machine Expert software, devices that want to send or receive Modbus TCP signals to communicate with PLC must be set here, as shown in the following figure 4.10. The IP address and device type must be set manually. The read and write offset here means the address that needs to be read from or

written to the EVSE respectively, the details about the offset can be seen in the manual. The length here is the length of the values read from or sent to the EVSE.

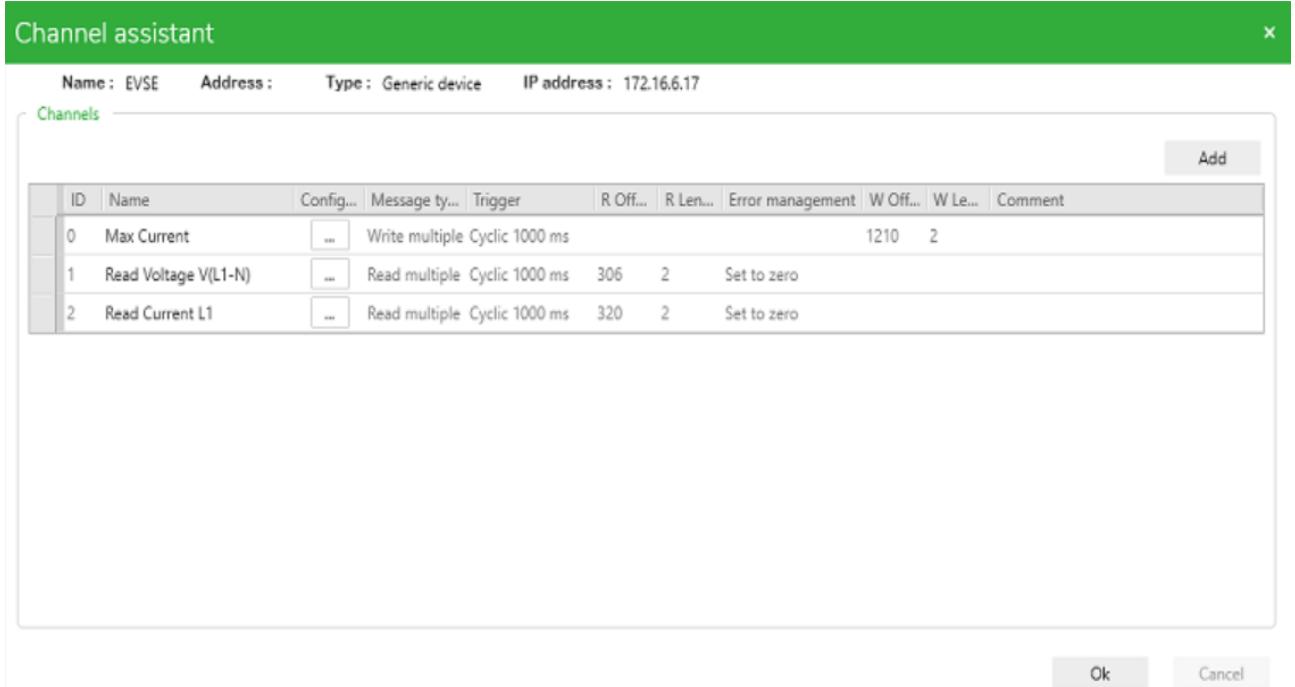


Figure 4.8: Modbus channel assistant

The PLC needs to be programmed after the Channel Assistant is configured so that the variables being read into the registers can be manipulated correctly. Through programming, the PLC can make the variables read from the input/output registers to be stored in the memories and use those memories to generate PWM signals or do calculations. There are several kinds of memories used in this thesis, and they have been listed in the following table.

Object Type	Object	Object Functions	Description
Object Type	%MFi	Memory floating point	Stores memory floating point
	%MDi	Memory double words	Stores 32-bit memory word
	%MWi	Memory words	Stores 16-bit memory word
	%Mi	Memory bits	Stores memory bit.

Table 4.3: Memories used

The PLC program includes 14 rungs, and these 14 rungs can be divided into three stages in total. Rung 0 3 is to read the analog signals from OPAL-RT and transfer the current setpoints to EVSE, rung 4 8 is to read the voltage value of EVSE first, store the voltage value in two memory words, then convert it and output to OPAL-RT in the form of PWM. Rung 9 13 is to read the current value of EVSE first, which is also stored in two memory words, and then converted into PWM and output to OPAL-RT. The reason why both the current value and the voltage value are stored in two memory words is that these two values are in the form of 32-bit floating point format, so both of them need two 16-bit memory words to be temporarily stored. One thing worth noting, that 32 is a floating point number, so the last 16 bits are the value of voltage or current, and the first 16 bits are not. Therefore, Operations are mainly focused on the last 16 bits. Part of the PLC program can be seen in the following figure.

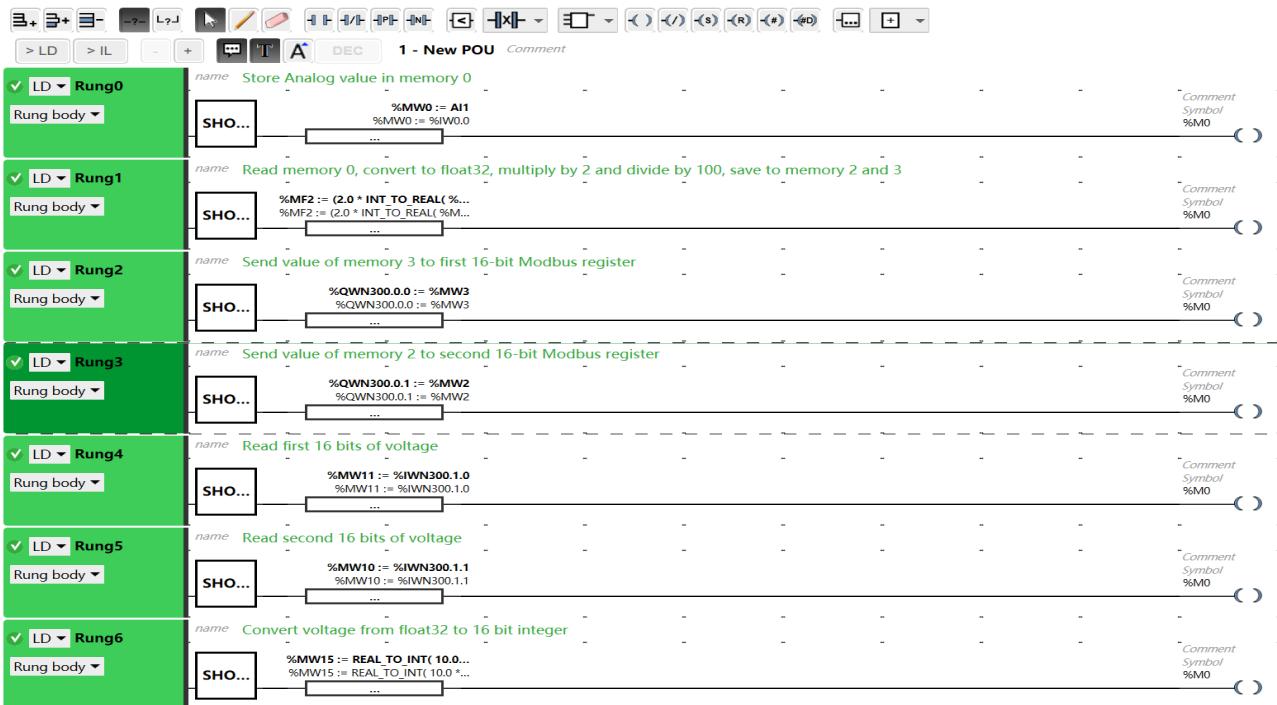


Figure 4.9: PLC Program

When the communication is successfully established, it can be observed from the ACE service installer software and values can be seen in the PLC programming rungs. The observations are shown in the following figure.

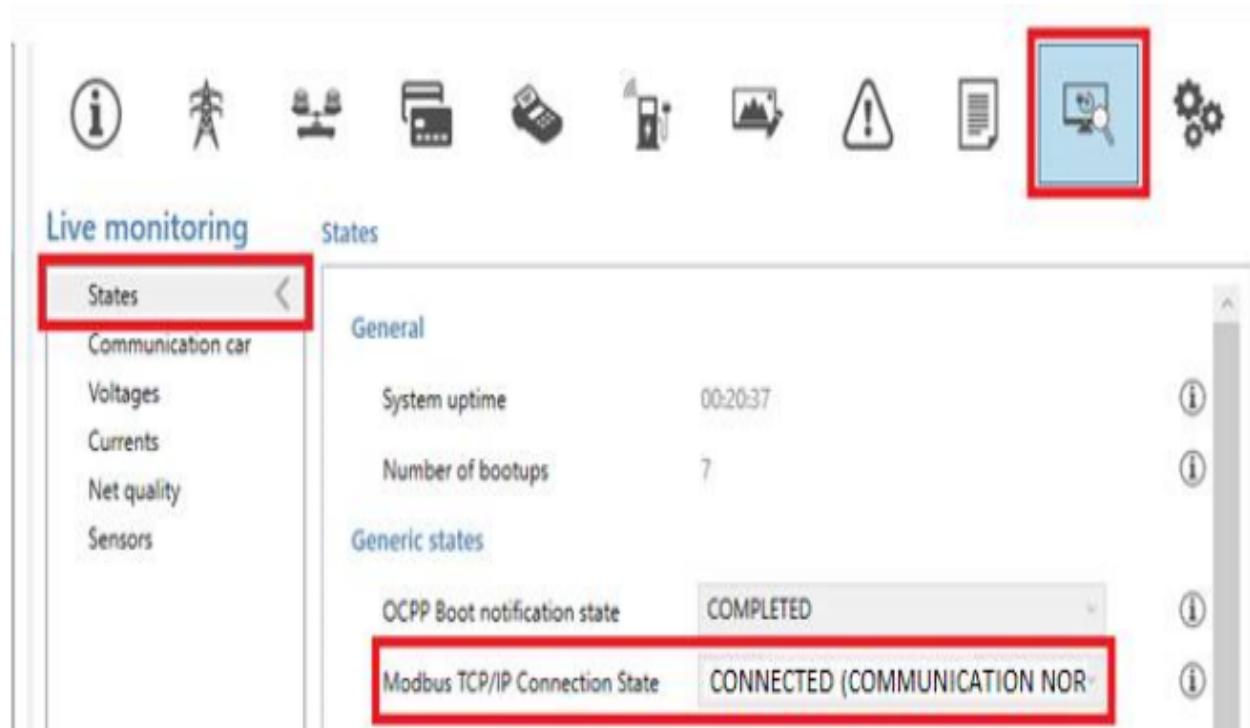


Figure 4.10: The screenshot of connection

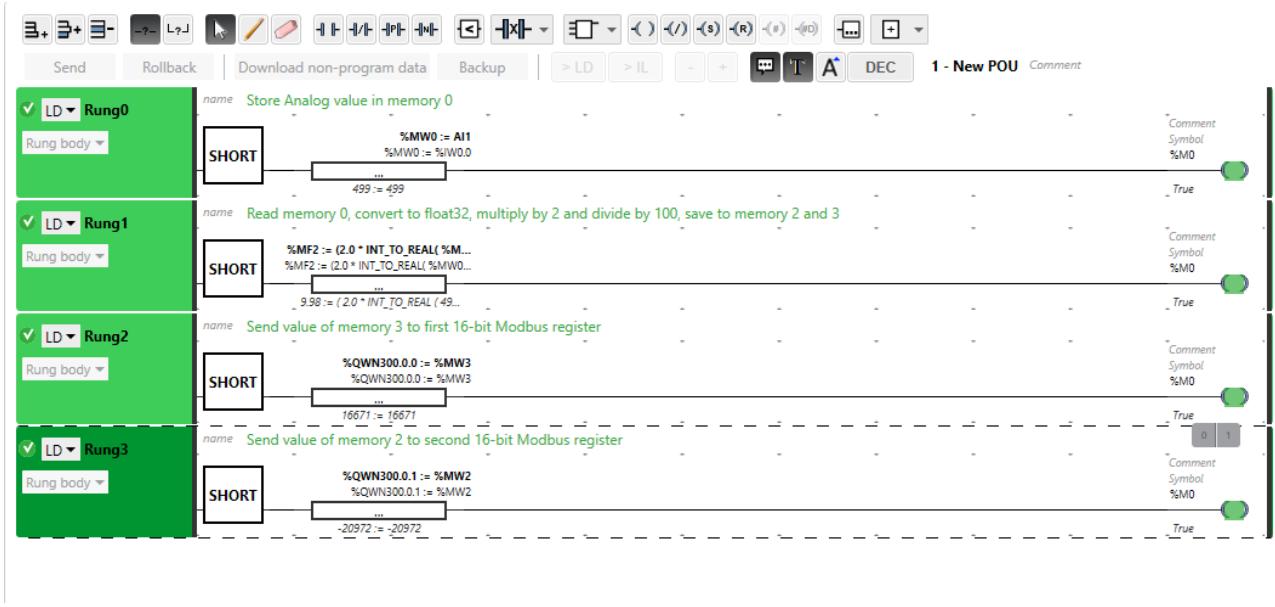


Figure 4.11: Values in the PLC rungs

5 Basic Case

This base case is a benchmark for all other cases without any changes to parameters. Other cases will have some changes in parameters or time with respect to the base case. In this smart charging case, the smart charging algorithm described in the previous chapter was implemented.

5.1 Simulation scenario

Table 5.1 shows the parameters used for this scenario. This scenario is based on EV owners charging the car at noon when PV generation is at its maximum and leaving at night when PV generation is almost at its lowest. Because of this, the simulation time is set to be from 13:00 to 19:00; six hours in total. Data from 10/06/2018 was randomly selected. The normalized PV profile and load profile over the selected day are shown in figure 5.1

Overview	Time of simulation	13:00 to 19:00
	Number of EVs (J)	4
	Number of nodes (N)	2
	Electricity Buying Price $C_t^{e(buy)}$	Day-ahead prices [47], 10/06/2018
	Electricity Selling Price $C_t^{e(sell)}$	0.02 €/kWh
	Variable PV Cost C_{PV}	0€/kWh
	Penalty $C_{n,j}^p$	0.1€/kWh
	Normalized PV Profile $p_{n,j}^{PV,norm}$	KNMI data [48], 10/06/2018
	Normalized Load Profile $p_{n,t}^{LL,norm}$	E2B 5000 NEDU Profile[49], 10/06/2018
	FCR	0.03648€/kWh , 10/06/2018

Table 5.1: Parameters used for basic scenarios

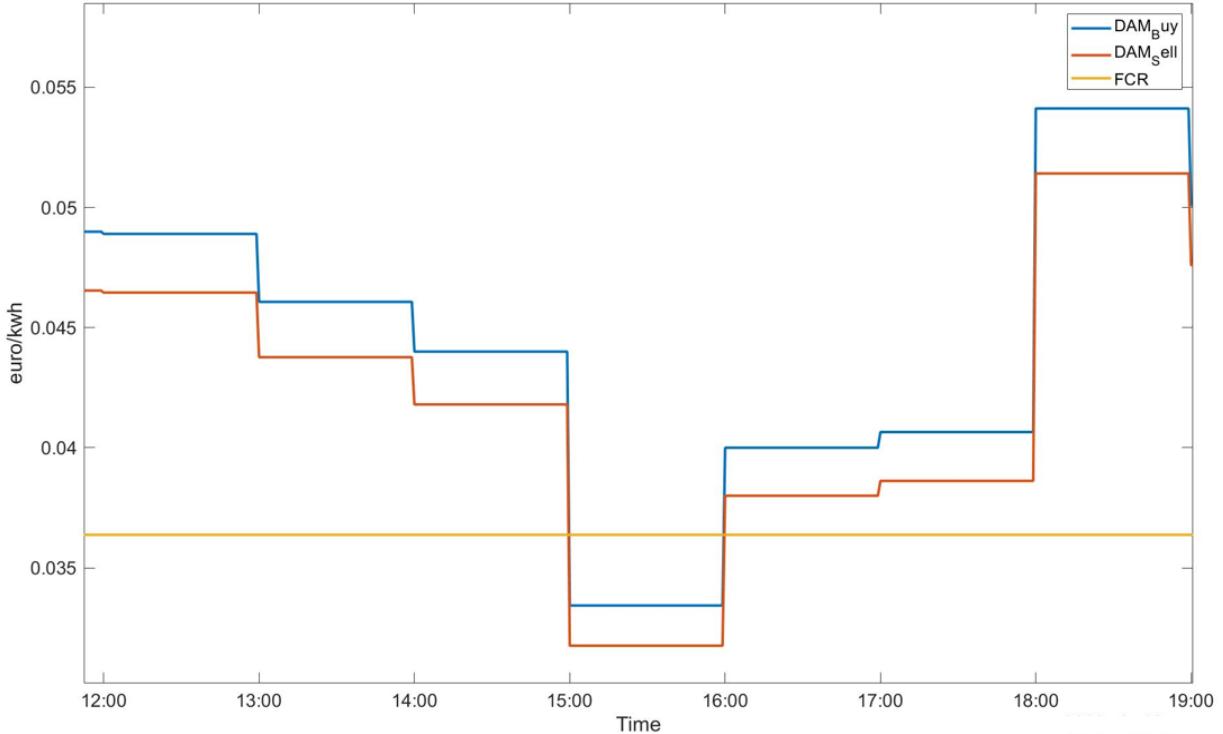


Figure 5.1: Prices of buying electricity, selling electricity and FCR

As can be seen in figure 5.2, there are 4 EVs in total; 3 EVs(EV1,2,3) connect to node

nd_53559675 and 1(EV4) connects to node 310387526. Among the four vehicles, EV1 is simulated by PHIL hardware, and the other three are virtually simulated. Grid node nd_53559675 has a local load and also has local PV installation, while node 310387526 has nothing.

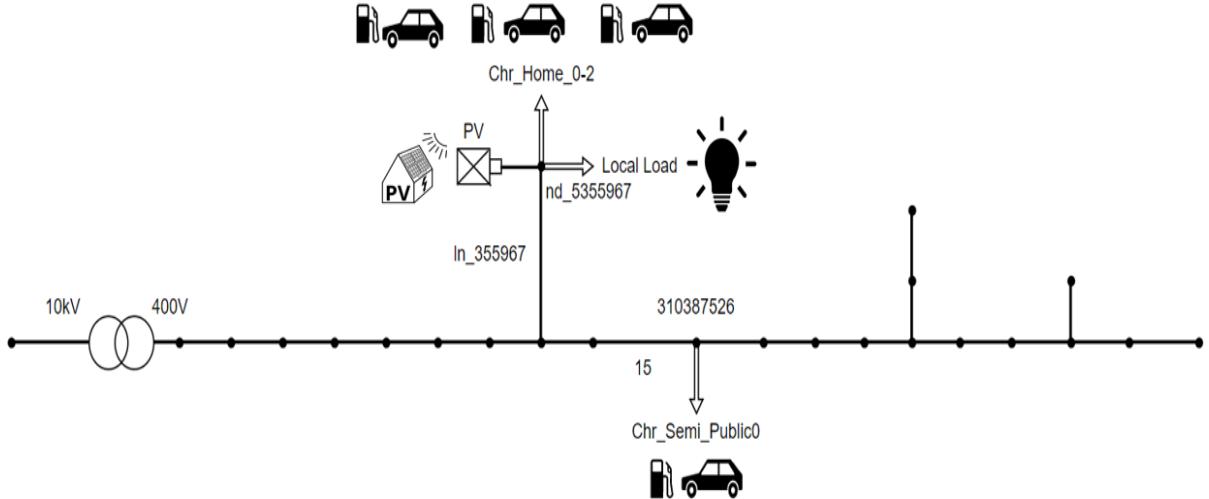


Figure 5.2: Grid model for basic case

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). The profile can be seen in the figure 5.3.

The parameters of the EV1, EV2, EV3 and EV4 can be seen in table 4.2, There are two types of EV parameters used in this case, Tesla Model 3 and Kona. Both these two types of EVs have the same rated current 16A. However, they have different battery sizes, and the EVs arrive at different times with different SOC and depart at different times.

nd_53559675	Parameters	EV1	EV2	EV3
Type	Model 3	Model 3	Model 3	
Battery size	47.5	47.5	47.5	
Rated power	11.04kW	11.04kW	11.04kW	
Arrive time	2018-6-10 13:10	2018-6-10 13:15	2018-6-10 13:16	
Departure time	2018-6-10 16:10	2018-6-10 16:00	2018-6-10 16:05	
Initial SOC	0.786358	0.770863	0.8483377	
Local Load yearly consumption	88.779 MWh			
PV generation	25 kWp			
310387526	Parameters	EV4		
Type	Kona			
Battery size	39.2			
Rated power	11.04kW			
Arrive time	2018-6-10 13:30			
Departure time	2018-6-10 18:58			
Initial SOC	0.680612			
Local Load yearly consumption	0 MWh			
PV generation	0kwp			

Table 5.2: Parameters of EVs

5.2 Validation of ePHASORsim

In order to validate the reliability of the results of ePHASORsim, it can be assumed that all chargers are in an uncontrolled charging state, which means that the EV charging current in this state is the maximum value of 16A. Moreover, the PV power generation and local load consumption are not considered under uncontrolled charging. The remaining parameters are the same as the parameters of the base case. The simulation results are shown in figure 3.4 below

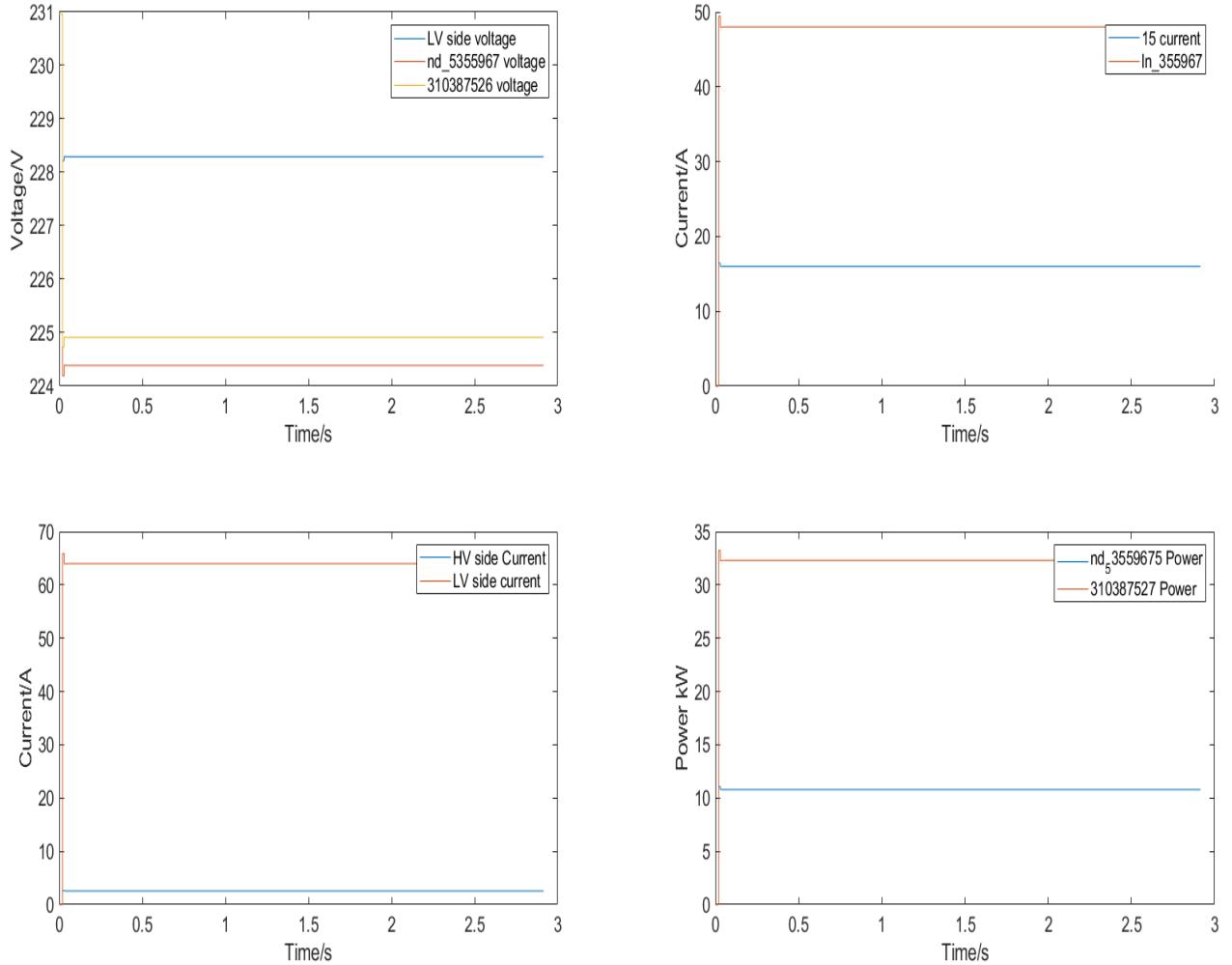


Figure 5.3: Result under uncontrolled charging

From the above graph, it can be seen that the voltage decreases as the load increases. When the voltage is stable, the current flowing to node `nd_5355967` is exactly 48A, because there are three chargers located at this point, and the other node is 16A because only one charger is located at this node. The current at the low voltage side is 64A, which is the sum of the 16A and 48A. The current at high voltage is 2.56A, this is because the transformer turns ratio is 25:1.

5.3 Simulation results

The results of the base case are shown in figure 5.2. This figure not only shows the changes in charging current, charging power, and SOC of EVs during charging but also shows the voltage changes of the distributed grid nodes as the load changes.

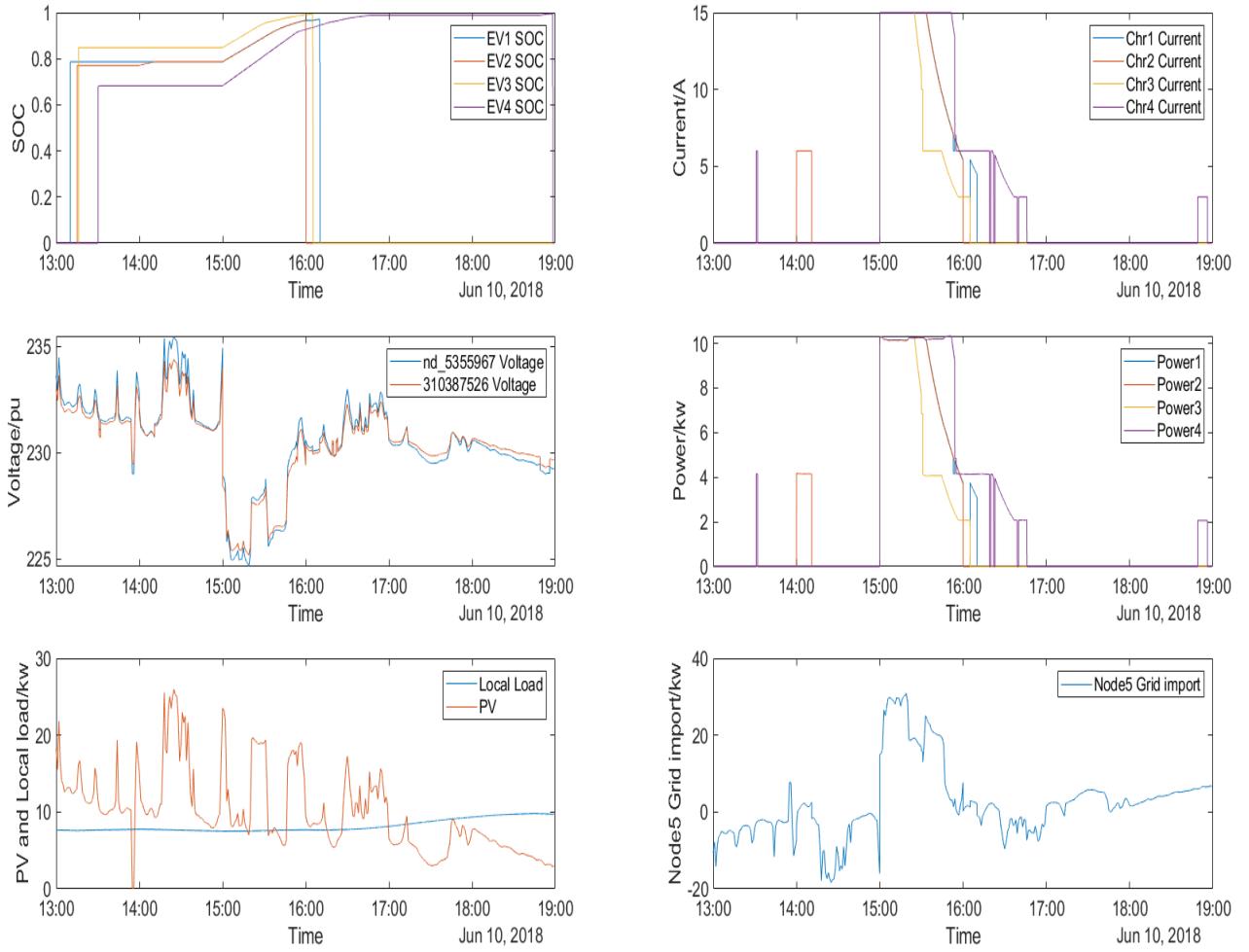


Figure 5.4: Results of the base case

It can be observed from the results that except for EV2 and EV4 have a few minutes of charging before 3 o'clock, all EVs start charging at 15:00, and charging mainly happens from 15:00 to 17:00, even if some EVs have been connected to the grid for almost two hours. There are two reasons for this; the first can be seen in figure 5.1: the price of buying electricity from the grid is the cheapest from 15:00 to 16:00. Because of this, the smart charging algorithm decides to let EVs import electricity from the grid as much as possible when electricity prices are low, and not charge when electricity prices are high (taking all constraints into account) in order to minimize cost. The second reason is that from 15:00 to 16:00, the PV generation is higher than the local load, then the EVs would have been charged with this excess energy. Since the price of buying electricity is always higher than the price of selling electricity, it is not worthwhile to export electricity to the grid, so all the electricity will be used for EV charging.

6 Case study

Since the goal of this study is to access the impact of smart charging algorithms on distributed power grids. It is unlikely to clearly see the full impact of a smart charging algorithm just by studying a basic case. Therefore, this section proposes several other cases. By testing the performance of the smart charging algorithm in other scenarios, its functional role and its impact on the distributed power grid will be further studied.

There are three cases presented in this subsection:

Case 1 is to test the performance of the smart charging algorithm when multiple groups of EVs are connected to the grid at different times.

Case 2 is to evaluate the impact of PV generation on the smart charging algorithm.

Case 3 is to evaluate the impact of charger efficiency on the smart charging algorithm.

6.1 Case 1

In real life, electric vehicles generally come and go, so in this section, two groups of EVs are set for node nd_53559675(the location of the nodes can be seen in figure 5.2), which come and go at different time periods respectively. The parameters of these two groups of EVs are shown in Table 6.1. It can be seen that the parking time of the first group of EVs is only about 1 hour, and the parking time of the second group of EVs is only about two hours. This also means that the EV charging time will be greatly shortened compared to the base case.

nd_53559675(g1)	Parameters	EV1	EV2	EV3
Type	i3	Kona	Kona	
Battery size	37.9	39.2	39.2	
Rated power	11.04kW	11.04kW	11.04kW	
Arrive time	2018-6-10 13:01	2018-6-10 13:05	2018-6-10 13:10	
Departure time	2018-6-10 13:55	2018-6-10 14:00	2018-6-10 16:05	
Initial SOC	0.83504	0.87306	0.82612	
Local Load yearly consumption		88.779 MWh		
PV genaration		25 kWp		
nd_53559675(g2)	Parameters	EV1	EV2	EV3
Type	Model 3	Model 3	Model 3	
Battery size	47.5	47.5	47.5	
Rated power	11.04kW	11.04kW	11.04kW	
Arrive time	2018-6-10 14:10	2018-6-10 14:15	2018-6-10 14:16	
Departure time	2018-6-10 16:10	2018-6-10 16:00	2018-6-10 16:05	
Initial SOC	0.786358	0.770863	0.8483377	
Local Load yearly consumption		88.779 MWh		
PV genaration		25 kWp		
310387526	Parameters	EV4		
Type	Kona			
Battery size	39.2			
Rated power	11.04kW			
Arrive time	2018-6-10 15:30			
Departure time	2018-6-10 18:58			
Initial SOC	0.680612			
Local Load yearly consumption	0 MWh			
PV genaration	0kwp			

Table 6.1: Parameters of two groups of EVs

The simulation results of this case can be seen in figure 6.1. As in the previous base case, the figure shows the changes in charging current, charging power, and SOC of EVs during charging, the voltage changes of the distributed grid nodes as the load changes.

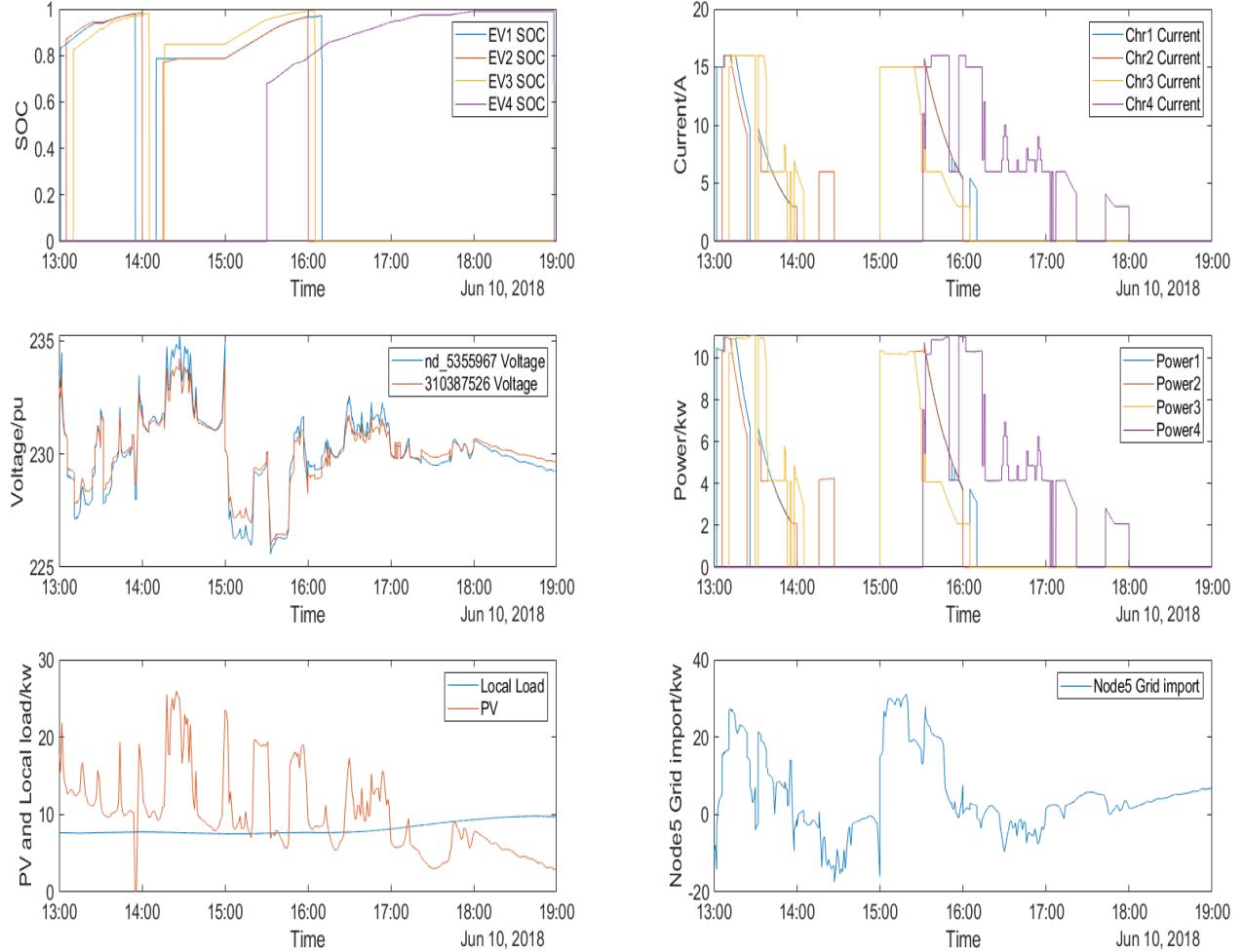


Figure 6.1: Simulation results of case 1

For the first set of EVs, charging begins the moment when they are connected to the charger. Unlike the base case, since the total charging time for this group of EVs is only about one hour, in order to fully charge the EV as much as possible, it is impossible for EVs to wait until the electricity price becomes lower. Therefore, even if the buying price of electricity will drop from 14:00 according to figure 5.1, the charging mainly happens between 13:00 to 14:00. And for the second group of EVs, since they are basically charging from 14:00 until they leave around 16:00. They all have a charging time of about two hours, and since the time span of these two hours exactly includes 15:00 to 16:00, the purchase price of electricity at this time happens to be the lowest, so it can be seen that the second Although the group EVs have been connected to the charger at 14:00, charging starts at 15:00.

For point 31038752, the result is also somewhat different from the base case. First, since the arrival time of EV4 is 15:30, which happens to be the time when the purchase price of electricity is the lowest, charging starts immediately, and from the very beginning, the EV4 starts to be charged at nearly the maximum power. From Figure 5.1, it can be seen that from 16:00, the buying electricity price begins to rise gradually, reaching the peak during the period from 18:00 to 19:00. Therefore, the charging of EV4 basically ends at 18:00. At around 15:50,

it can also be seen that the current is not the maximum current of 16A. This is due to the rapid increase in voltage at this moment. The smart charging algorithm recalculates a new charging plan based on the current voltage, so as the voltage increases, the optimal current point does not need to be that large, which is why the current decreases.

The final SOC results of EVs are shown in Table 6.2 below, none of the EVs are fully charged, that is because the smart charging algorithm hasn't considered the constant voltage charging when the EVs are charged with current lower than setpoints.

	EV1	EV2	EV3	EV4
Departure SOC	97.9% (group1) 97.2% (group2)	98.3% (group1) 96.7% (group2)	98.3% (group1) 99.4% (group2)	99.9%

Table 6.2: Departure SOC

6.2 Case 2

This case is to study EV charging at night when local load and PV generation will change. In this way, the voltage of the grid will be different compared to the base case. The parameters of the EVs are the same compared with the base case, the only difference is the arriving time and departure time. The arrival time and departure time are shown in Table 6.3

nd_53559675	Parameters	EV1	EV2	EV3
	Arrive time	2018-6-10 18:10	2018-6-10 18:15	2018-6-10 18:16
	Departure time	2018-6-10 21:10	2018-6-10 21:00	2018-6-10 21:05
310387526	Parameters	EV4		
	Arrive time	2018-6-10 18:30		
	Departure time	2018-6-10 23:58		

Table 6.3: Arrival and departure time

The DAM price and FCR price can be shown in figure 6.2

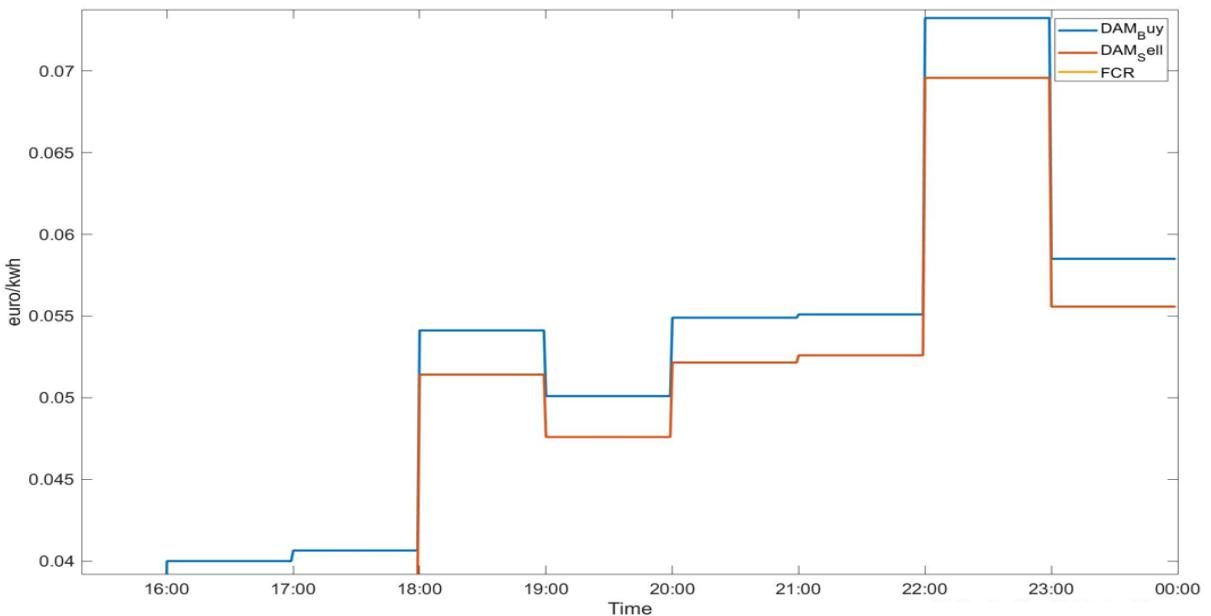


Figure 6.2: Evening prices

The simulation results are shown in figure 6.3.

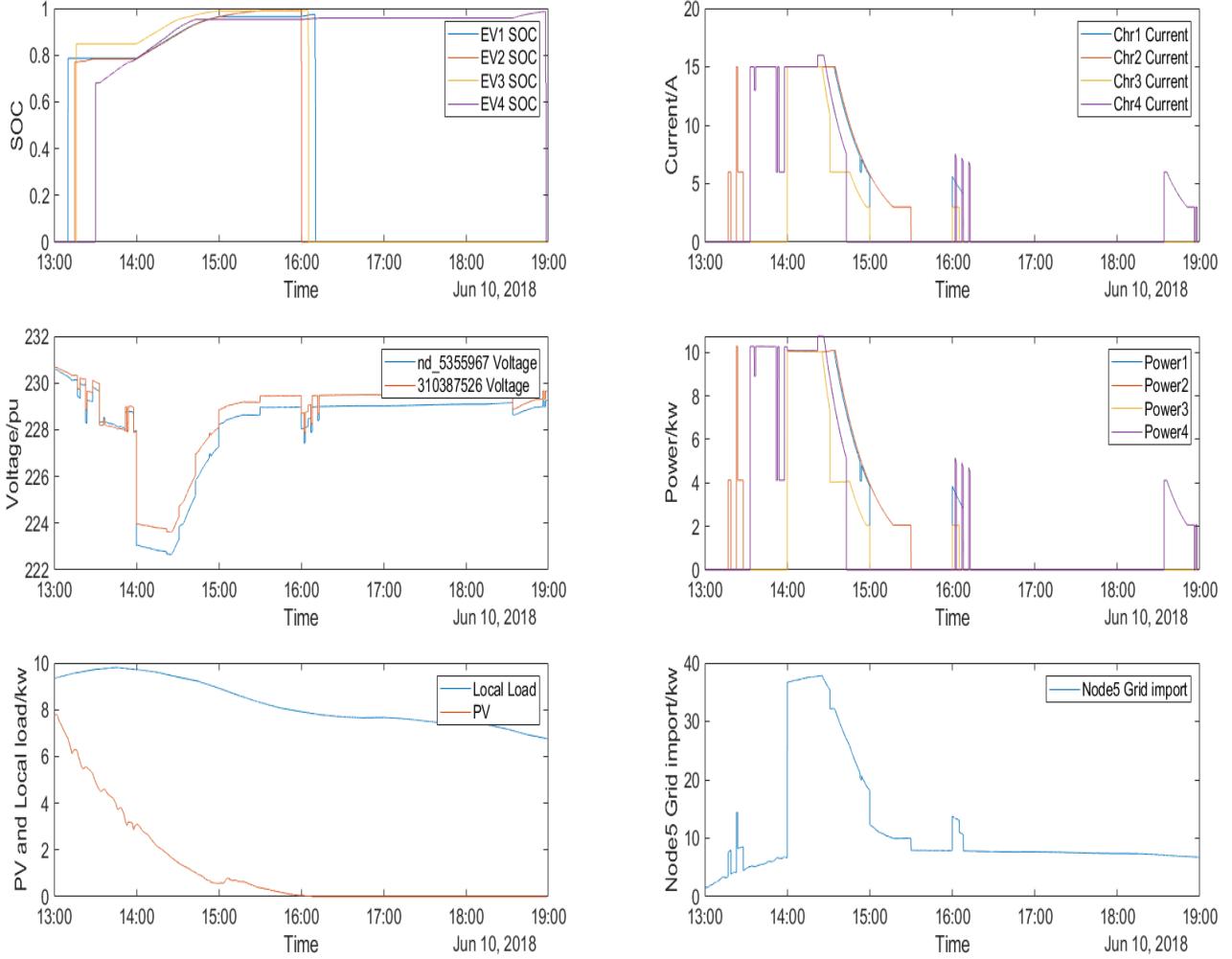


Figure 6.3: Simulation result of case 2

The depart EVs SOC are 97.4%, 98.92%, 99.28% and 99.6% respectively. It can be from the 6.3 that EV1,2, and 3 waits until 19:00 to start charging, the same reason as the previous case because the buying electricity price during 19:00 and 20:00 is the lowest over the whole 6 hours optimization duration. Even if the gap between local load and PV generation gets bigger and bigger as time goes by. In the base case, it was learned that the smart charging algorithm tends to give current set points in the time period when PV generation is larger than local load consumption because there can be additional energy from EV charging. However, in this case, the smart charging algorithm takes PV, local load, and electricity price into consideration and decides to let the charging happen when the electricity price is the lowest, in order to minimize the overall charging cost. As for the EV4, because the initial SOC of the EV is very low, and the gap between PV generation and local load is not that big at the beginning, the charging starts immediately when the EV4 is connected to the node.

Overall, charging mainly occurs from 18:30 to 20:30, which is also expected, because the buying electricity price is already the lowest during this period. The EV4 is also charged for tens of minutes at the minimum current in the final stage. This is because the price of electricity starts to increase sharply from 22:00 and drops until 23:00. In order to reduce the cost of charging, the charging period from 23:00 to 24:00 is selected .

6.3 Case 3

In the previous case, the charging efficiency of EVs in the code is 97% by default, but in practice, the charging efficiency of EVs is not so high. According to [50], the EV charging efficiency can be around 83.8% which has a significant gap compared to the default value. To test the effect of the EV charging efficiency on the implemented smart charging algorithm, in this case, the charging efficiency has been set to 83.8%. The other EVs parameters are the same as in figure 6.1. The results of this case are shown in figure 6.4.

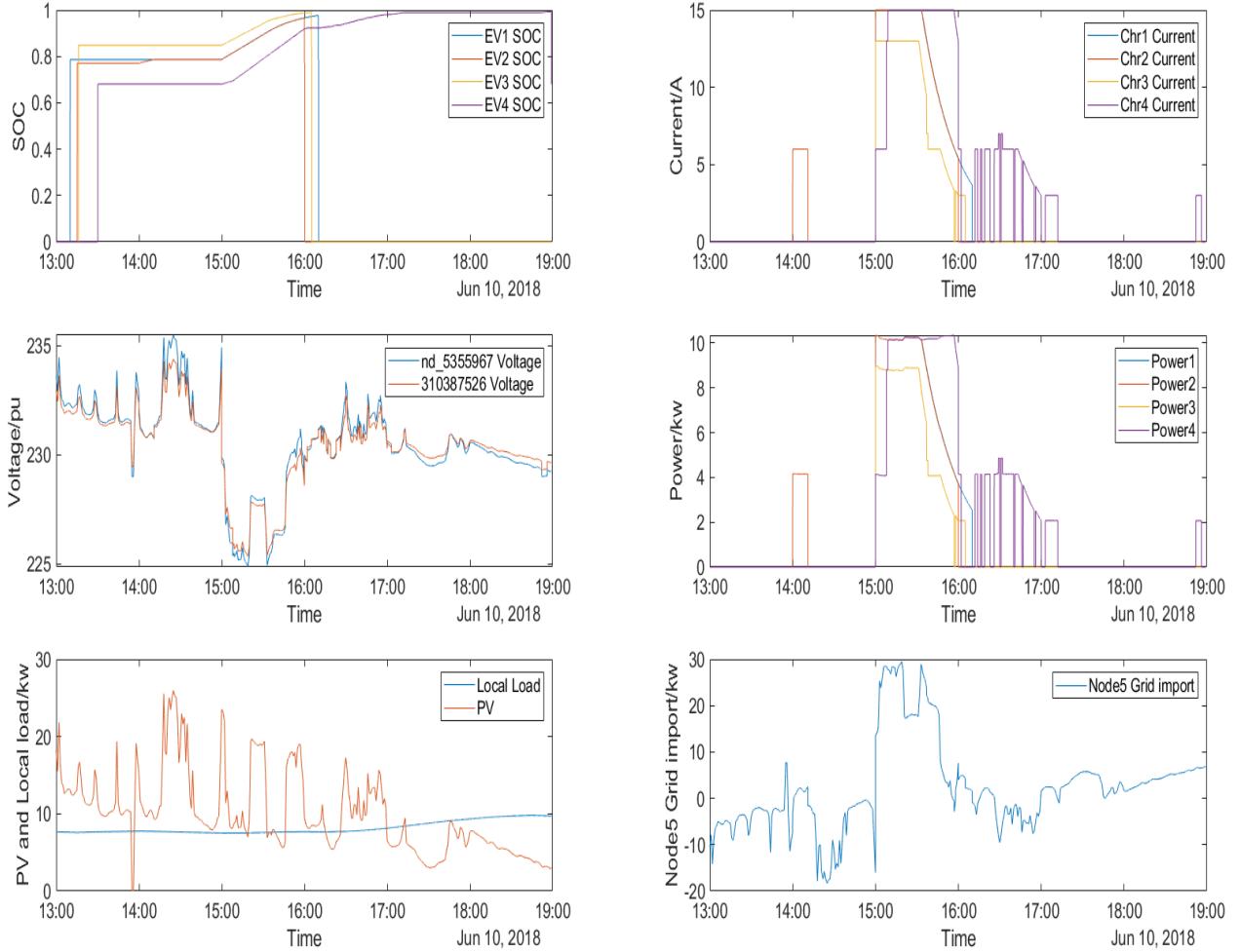


Figure 6.4: Simulation results of case 3

On the whole, charging still occurs in the time period from 15:00 to 16:00. The reason has been mentioned in the previous case and will not be repeated here. But in this time period, there are many differences from the base case. First, for EV1 and EV3, these two EVs are higher than EV2 due to the initial SOC and parking time. When the efficiency of charging decreases, their charging current in the interval from 15:00 to 16:00 will drop significantly. This is because as the efficiency decreases, the grid import limitation also decreases, and the SOC and parking time of EV1 and EV3 are higher than EV2, so the charging current of these two vehicles will decrease. As for EV2, due to the short initial SOC and parking time, the charging current is almost close to the maximum, which is consistent with the base case.

For EV4, in the interval from 15:00 to 16:00, the charging behavior is consistent with the base case, but in the interval from 16:00 to 17:00, the charging time is longer. The reason is similar to that of EV2. Since the initial SOC is low and EV charging efficiency decreases, it is

necessary to charge as much as possible in the interval from 16:00 to 17:00, where the electricity price is the second lowest, in order to reduce the penalty brought by unfinished charging. The depart EVs SOC are 96.65%, 95.6%, 98.7% and 99.1% respectively. Compared with the base case, there is still a slight gap, which proves that even if the smart charging algorithm adjusts the charging scheme, the impact of lower charging efficiency cannot be completely eliminated.

6.4 Summary

Table 6.4 summarizes a total of four cases that appear in chapters 5 and 6, showing the differences in some key features among the four cases.

Arrival SOC	EV1	EV2	EV3	EV4
Base case	78.6%	77.1%	84.8%	68.1%
Case 1	83.5%(group1) 78.6%(group2)	87.3% (group1) 77.1% (group2)	82.6% (group1) 84.8% (group2)	68.1%
Case 2	78.6%	77.1%	84.8%	68.1%
Case 3	78.6%	77.1%	84.8%	68.1%

Charging time	EV1	EV2	EV3	EV4
Base case	13:10-16:10	13:15-16:00	13:16-16:05	13:30-18:59
Case 1	13:01-13:55(group1) 14:10-16:10(group2)	13:03-14:00 (group1) 14:15-16:00(group2)	13:05-14:05(group1) 14:16-16:05 (group2)	15:30-18:59
Case 2	18:10-21:10	18:15-21:00	18:16-21:05	18:30-21:59
Case 3	13:10-16:10	13:15-16:00	13:16-16:05	13:30-18:59

Departure SOC	EV1	EV2	EV3	EV4
Base case	97.63%	96.56%	99.36%	99.5%
Case 1	97.9%(group1) 97.2%(group2)	98.3% (group1) 96.7% (group2)	98.3% (group1) 99.4% (group2)	99.9%
Case 2	97.4%	98.92%	99.28%	99.6%
Case 3	96.65%	95.6%	98.7%	99.1%

Minimum Voltage	nd_53559675	310387526
Base case	225.19	224.67
Case 1	225.92	225.57
Case 1	223.62	222.66
Case 1	225.36	224.87

Table 6.4: Summary

Since the Minimum voltage in the smart charging can be observed, and the voltage provided by the grid under normal conditions is about 230.94V, then the maximum voltage deviation can be calculated.

Maximum Voltage deviation	nd_53559675	310387526
Base case	2.49%	2.71%
Case 1	2.17%	2.32%
Case 1	3.17%	3.59%
Case 1	2.41%	2.63%

Table 6.5: Voltage deviation

It can be observed that the higher the load when the electric vehicle is charging, the lower the voltage provided by the grid. During smart charging, the maximum voltage deviation is 3.59%.

Another observation is that none of the EVs are fully charged, only to 97.63%, 96.56%, 99.36%, and 99.5% SOC. The reason is that the constant voltage charging stage is not considered. The IEC61851-based charger can give a minimum current setting of 6A, but the EV can be charged with a lower current during the constant charging stage. In addition, when calculating the optimal current setting value, the smart charging algorithm uses the grid voltage read by the interface script at a certain moment, but the grid voltage is different at any moment because the values of PV generation and local loads consumption are changing at any time [51][52]. Once the EV starts charging, the bus voltage changes and the smart charging algorithm calculates the optimal current setpoints every minute and doesn't take voltage deviation into account, so the results are inaccurate.

7 Conclusion & Future work

In this Master thesis, the impact of a smart charging algorithm on the distribution power grid is verified. Besides, a smart charging algorithm has been applied in different scenarios in order to evaluate the performance of a smart charging algorithm. This chapter includes the contributions made in this thesis and the main findings in this thesis. Based on the problems that occurred in the thesis, the recommendation for improvement had been made.

7.1 Conclusion

There are three objectives mentioned in Chapter 1, and the conclusion will be made based on those objectives. The first objective of this thesis is to mimic the EV charging behavior on the HIL testbed. In theory, the smart charging algorithm should send the current setpoints to the PLC by outputting analog signals, after PLC converts the analog signals to Modbus/TCP signal, the current setpoint will be sent to the EVSE. Based on the current setpoints and the reference waveform from the grid simulation, there should be an amount of power circulating in the whole system. The schematic of the power circulation is shown in the following figure.

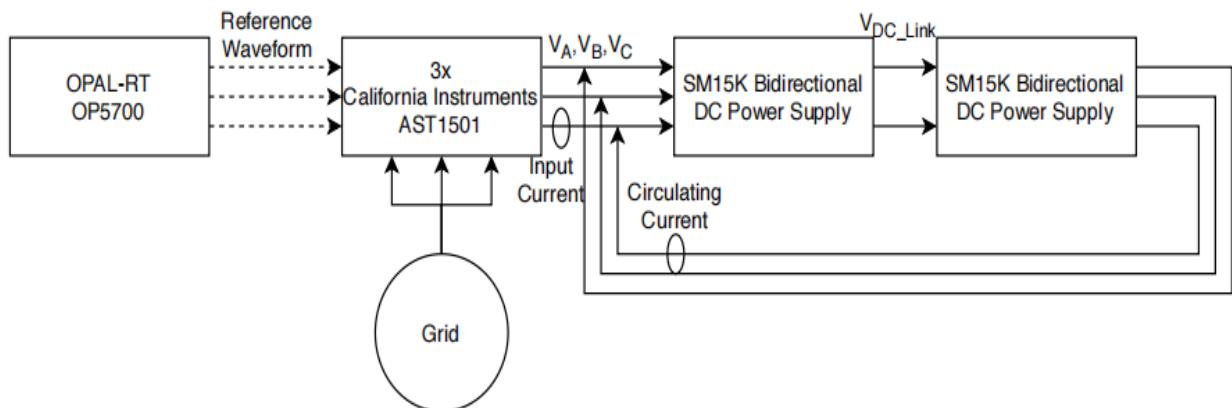


Figure 7.1: Power circulation[40]

With the help of two AC loads and an EV emulator, the EV's real charging behavior can be simulated on the testbed. However, because of some hardware were missing and time constraints, in this thesis, the connection between the hardware was only half done.

The second objective is to integrate the smart charging algorithm with the HIL testbed and run it under different scenarios in real-time. The smart charging algorithm is able to integrate with the HIL testbed with the help of OPAL-RT, the DRTS. The algorithm is required to run in real-time when it is integrated into the HIL testbed. With the help of RTLAR API, the Python algorithm is able to exchange the data with a model in real-time and run the real-time simulation. In this thesis, the smart charging algorithm ran under different scenarios. Firstly, a base case was run as a benchmark. In the base case, the smart charging algorithm optimized four EV charging, three of which are connected to one point of the distribution grid and one to another point of the distribution grid. It was found that smart charging was able to minimize the total cost of charging and reduce the voltage deviation. However, the algorithm also had the disadvantage of not being able to fully charge the car.

The third objective is grid simulation, building a grid simulation model to simulate the voltage changes at various points of the power grid when the smart charging algorithm is applied to the distribution power grid. In this thesis, the grid simulation is achieved by ePHASORsim. Depending on the local load, photovoltaic generation, and consumption of EV charging, the

grid simulation model can simulate the voltage value of each point in the distribution grid in real-time.

7.2 Recommendations and Future work

The smart charging algorithm applied in this thesis was developed a year ago, so there are still some aspects that can be improved. Based on some phenomena in the experiment, some suggestions are put forward so that the smart charging algorithm can be better improved. In addition to this, the HIL testbed also has some areas that can be improved.

7.2.1 Recommendations

1. The smart charging algorithm takes the communication delay caused by the commands into consideration. In Chapter 6, it has been mentioned that the reason EVs are not fully charged is the time delay caused by communication and command execution. Since the delay caused by communication is inevitable and compared with the delay caused by executing commands, the communication delay is not obvious, so it can be ignored. As for the delay caused by executing the command, it can be considered to reduce the impact of the delay on the setpoint switching through prediction.
2. Change the frequency of giving current setpoints from every minute to every five minutes. In the meantime, the smart charging algorithm changes from giving an optimal setpoint every minute to giving an optimal point every five minutes. Giving an optimal current setpoint every minute is actually a great challenge for the computer. In this thesis, because the number of EVs involved is small and the time is relatively short, the smart charging algorithm can run the optimization within one minute and calculate the optimal current point of each EV in the next minute. But in reality, the smart charging algorithm will have to deal with the situation where a huge number of EVs are connected to the distribution grid at the same time. It is difficult to calculate so many optimal current setpoints in one minute, so running the optimization every five minutes might be more realistic.
3. Consider combining real-time local load consumption and PV generation with the smart charging algorithm. The smart charging algorithm used in this topic is based on the historical data of PV generation and local load consumption to obtain the optimal current setpoints. Is it possible to predict the future load through real-time photovoltaic data and local load consumption data, and run the optimization base on the prediction?
4. Simplify the HIL testbed and remove the "bridge" PLC between EVSE and DRTS. The reason why a PLC is needed is that the Alfen EVE single proline cannot communicate directly with the DRTS. But there are other types of chargers that can directly receive analog signals. Therefore, if the type of EVSE is changed, it is possible to remove the PLC. In addition, the programming of the PLC itself is quite complicated. Once the IP address changes, everything needs to be recompiled. So it is best to remove the PLC.

7.2.2 Future work

1. Complete the hardware connection. In this thesis, as mentioned before, the hardware is only half connected and power circulation is not completed. Therefore, in the future, it is recommended to connect the two deltas to complete the power circulation. In this thesis, the measurement of current and voltage of EVSE has not been completed, in the future, if power circulation can be completed. Observations made on the DRTS can be done by measuring the voltage and current values of the EVSE and transmitting them to the DRTS.
2. The smart charging algorithm which was integrated into the HIL testbed in this thesis is still under development. In the future, there might be new features added to this smart charging

algorithm. In the previous section, some suggestions have been given for further improvement. If there is a new, improved smart charging algorithm, then it can be completely re-integrated with the testbed and rerun.

3. The connection between PLC and EVSE is still not established, if it is established, then the maximum allowed current setpoints would be able to send to EVSE from PLC.

Appendices

A MATLAB Model

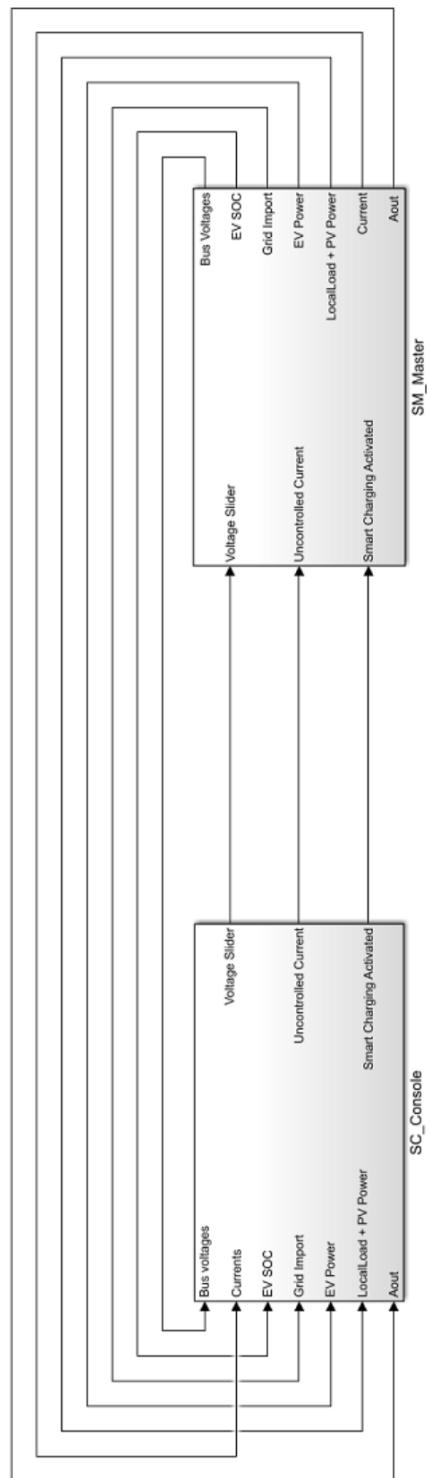


Figure A.1: MATLAB Model

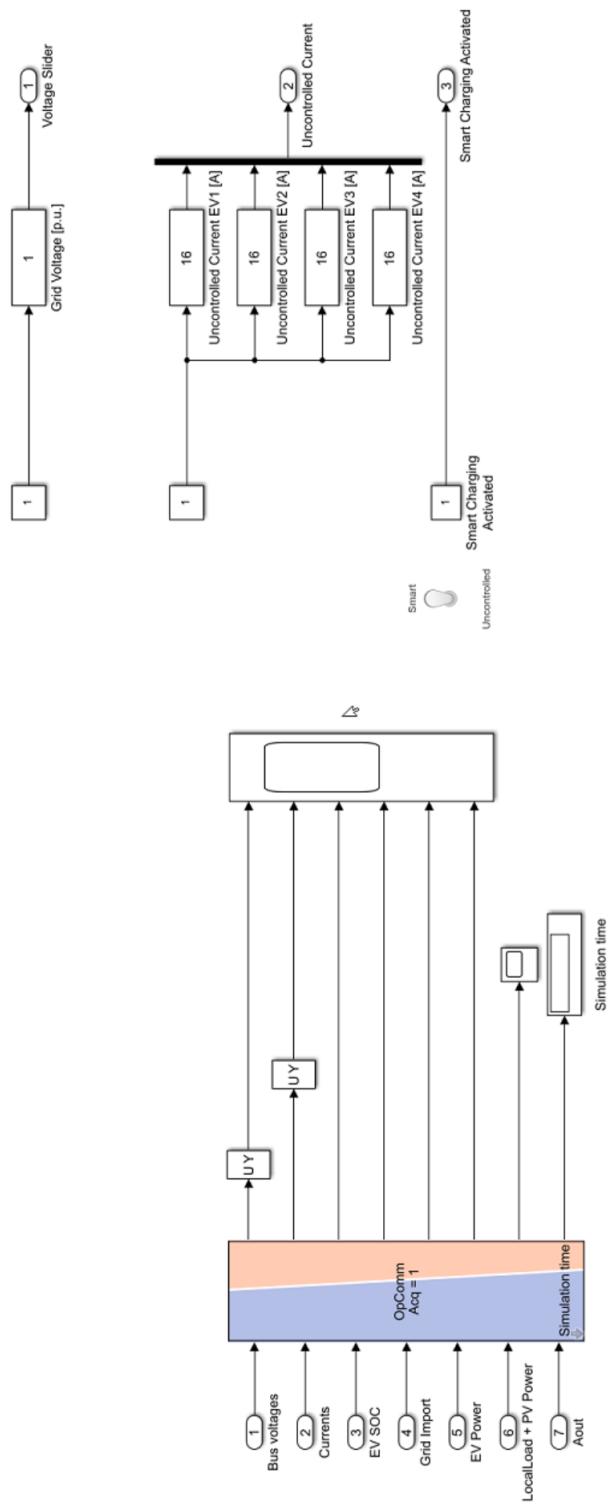


Figure A.2: SC_Console

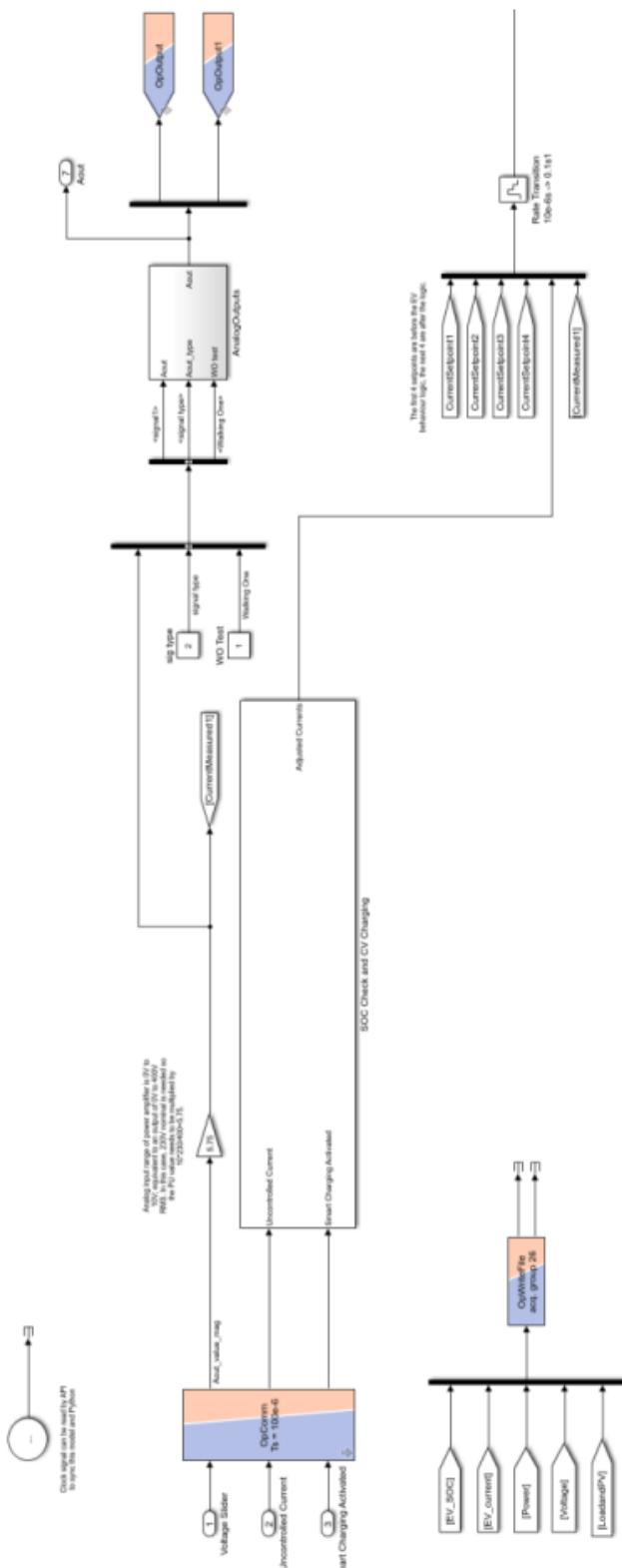


Figure A.3: SM_Master part1

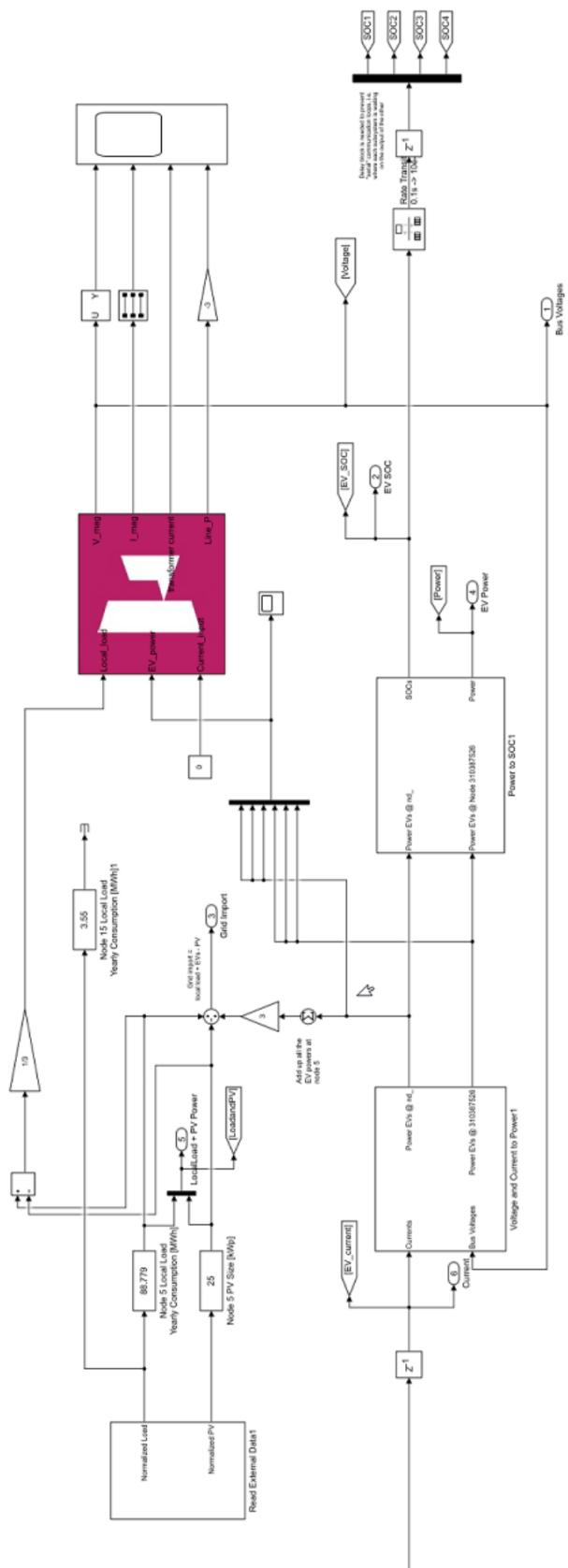


Figure A.4: SM_Master part2

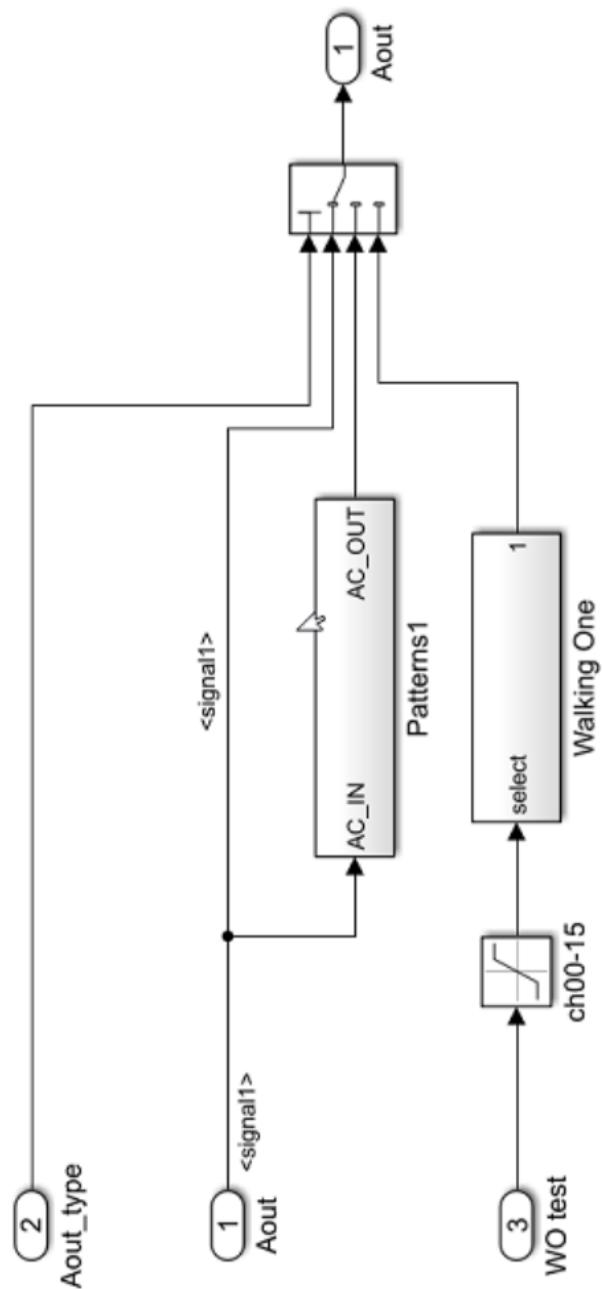


Figure A.5: Analog Out

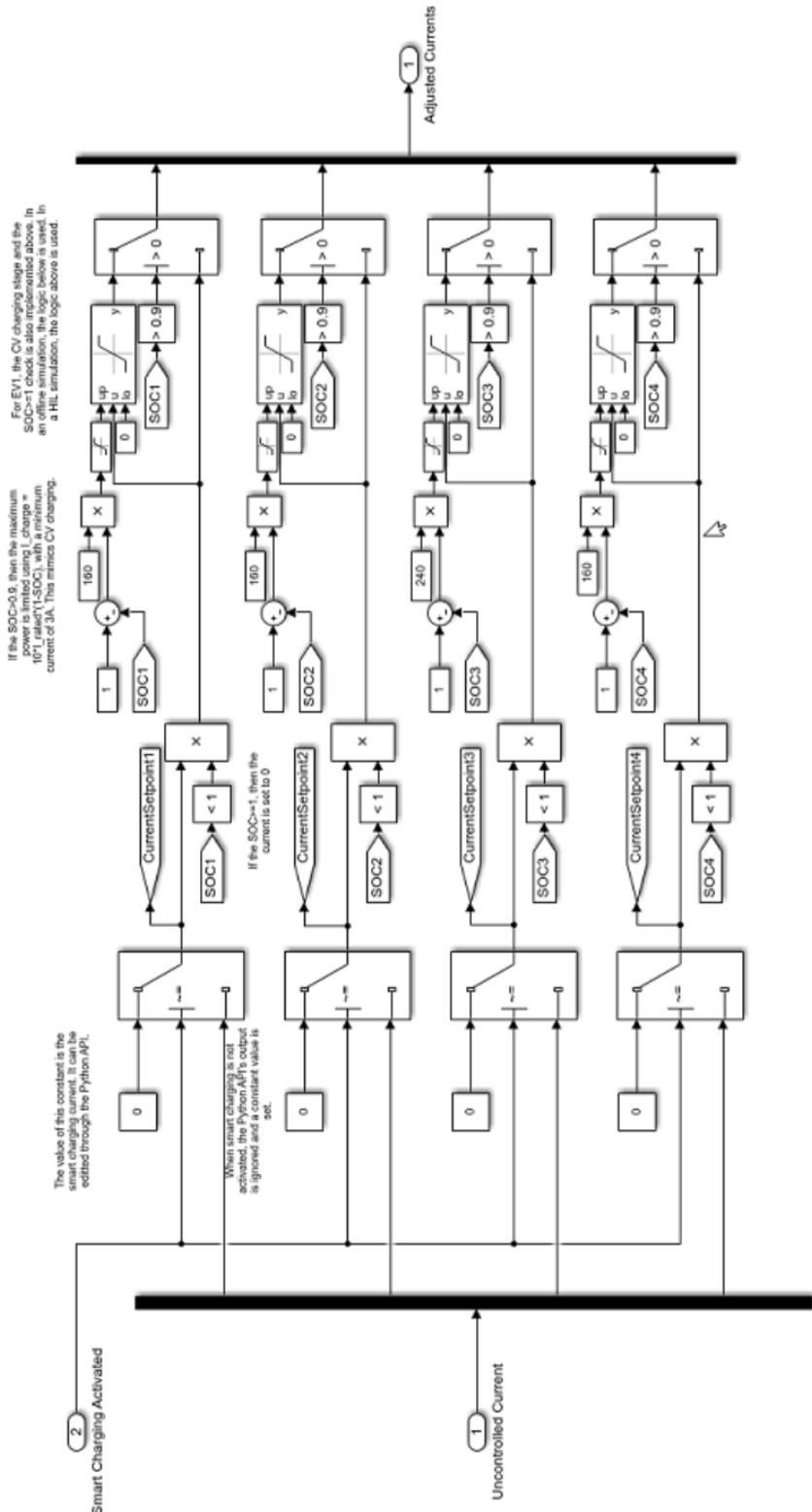


Figure A.6: SOC check and CV charging

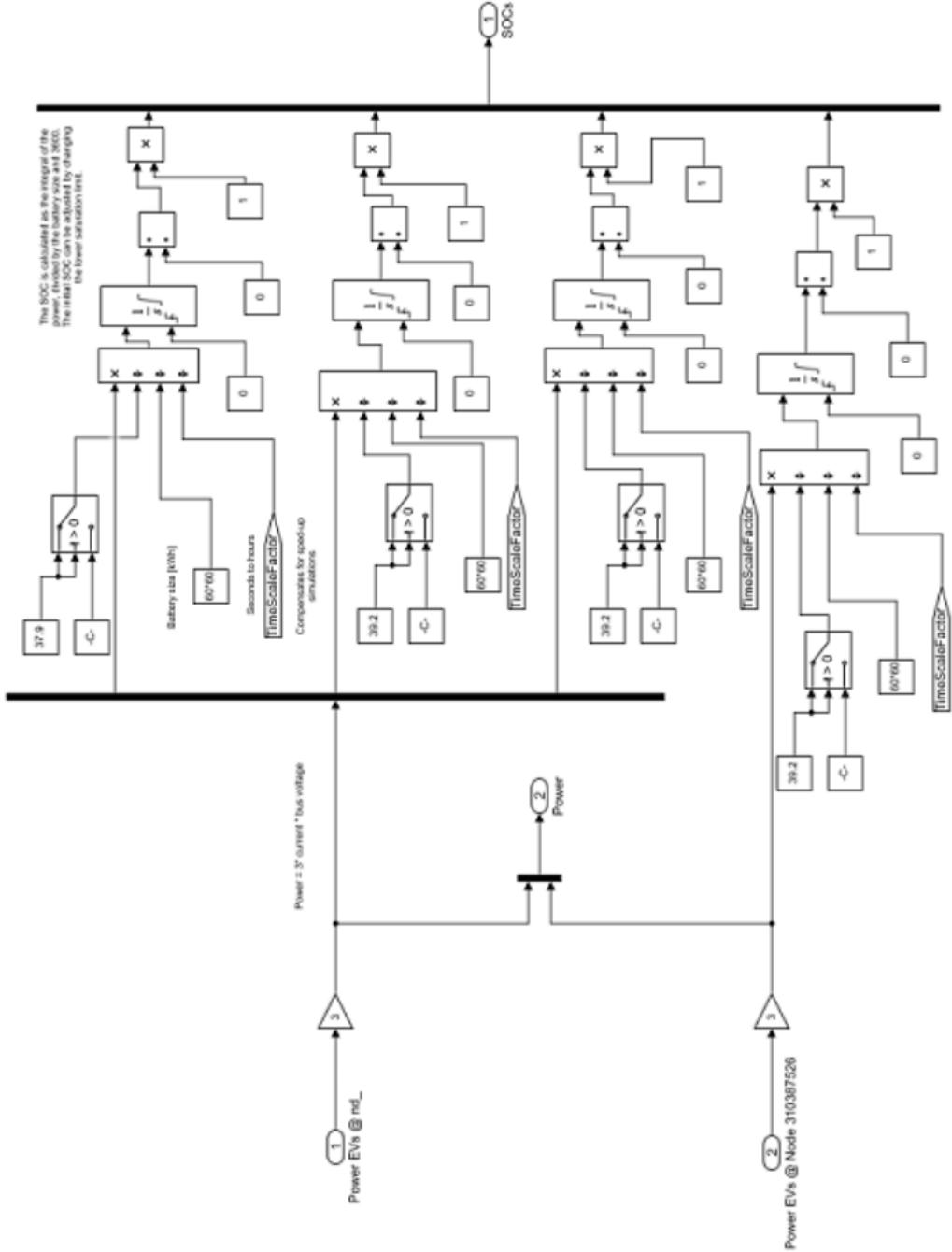


Figure A.7: SOC check and CV charging

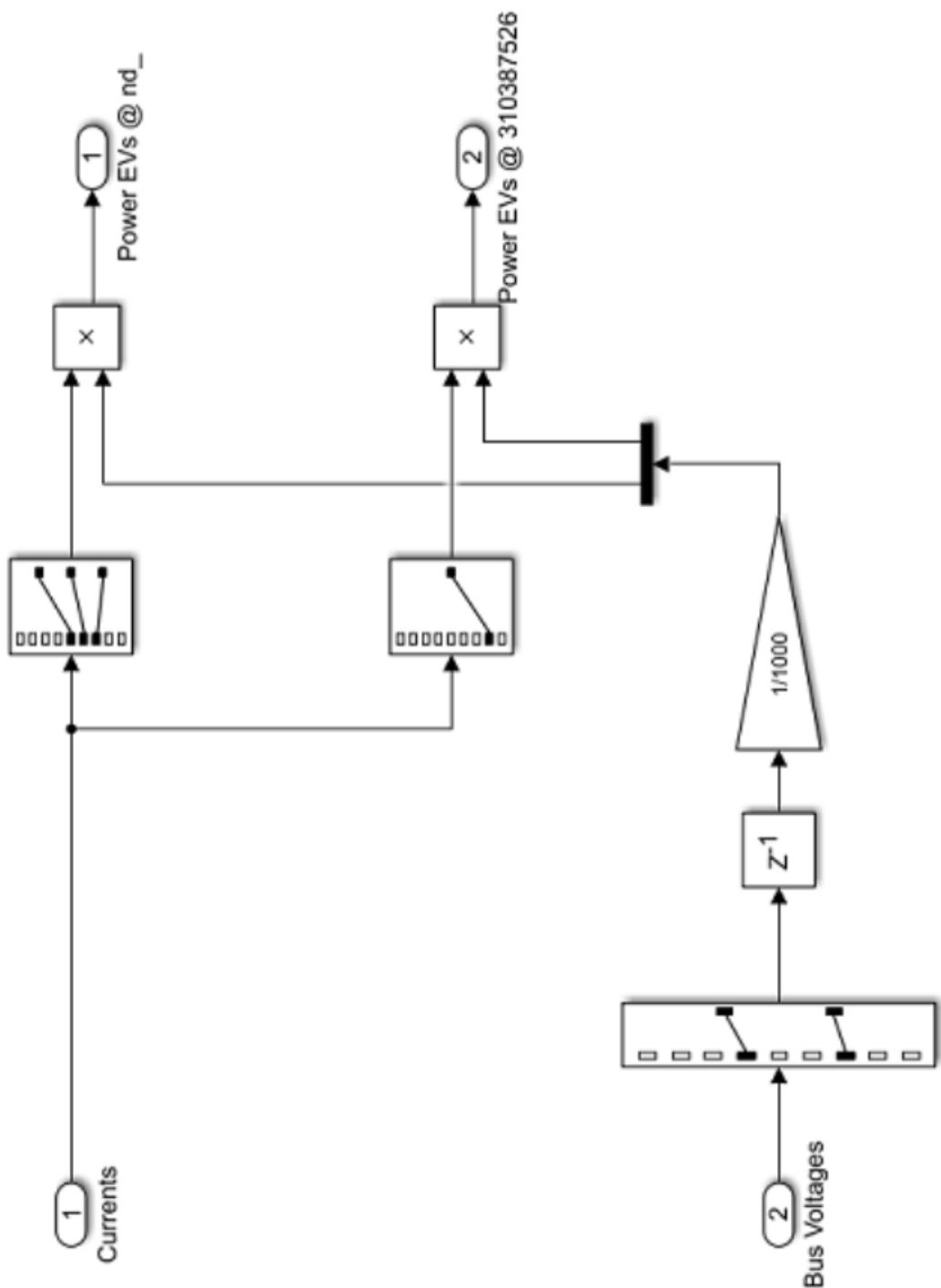


Figure A.8: SOC check and CV charging

B Registers Tables

B.1 Product identification registers

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.	"00000R000"
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

B.2 Station status registers

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

B.3 SCN registers

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.

B.4 Socket measurement registers

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

References

- [1]. Wolfram, Paul, and Nic Lutsey. "Electric Vehicles: Literature review of technology costs and carbon emissions." Working Paper 2016-14 (2016).
- [2]. Sierzchula, W.; Bakker, S.; Maat, K.; van Wee, B. The Influence of Financial Incentives and Other Socio-Economic Factors on Electric Vehicle Adoption. *Energy Policy* 2014, 68, 183–194. [Google Scholar] [CrossRef]
- [3]. Faria, M.V.; Baptista, P.C.; Farias, T.L. Electric Vehicle Parking in European and American Context: Economic, Energy and Environmental Analysis. *Transp. Res. Part A Policy Pract.* 2014, 64, 110–121. [Google Scholar] [CrossRef]
- [4]. IEA(2022), Electric cars fend off supply challenges to more than double global sales, IEA, Paris <https://www.iea.org/commentaries/electric-cars-fend-off-supply-challenges-to-more-than-double-global-sales>
- [5].Global-ev-sales-more-than-doubled-in-2021-vs-2020-tripled-vs-2019[Online].Available: https://www.greencarreports.com/news/1134999_global-ev-sales-more-than-doubled-in-2021-vs-2020-tripled-vs-2019
- [6]. M. Masoum, S. Deilami and S. Islam, “Mitigation of harmonics in smart grids with high penetration of plug-in electric vehicles”, Proc. IEEE Power and Energy Soc. Gen. Meet., Jul. 2010.
- [7].Rizvi, S.A.A.; Xin, A.; Masood, A.; Iqbal, S.; Jan, M.U.; Rehman, H. Electric Vehicles and their Impacts on Integration into Power Grid: A Review. In Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018; pp. 1–6. [CrossRef]
- [8]."What is smart EV charging?"<https://ev.energy/blog/the-benefits-of-smart-charging-with-ev-energy/>
- [9].H. W. Dommel, “Digital computer solution of electromagnetic transients in single-and multiphase networks,” *IEEE Trans. Power App. Syst.*, vol. PAS-88, no. 4, pp. 388–399, Apr. 1969.
- [10].H. W. Dommel, “Nonlinear and time-varying elements in digital simulation of electromagnetic transients,” *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 4, pp. 2561–2567, Nov. 1971.
- [11].C. Dufour, J. Mahseredjian, and J. Bélanger, “A combined state-space nodal method for the simulation of power system transients,” *IEEE Trans. Power Del.*, vol. 26, no. 2, pp. 928–935, Apr. 2011.
- [12].J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian, and L. Gérin-Lajoie, “On a new approach for the simulation of transients in power systems,” *Electr. Power Syst. Res.*, vol. 77, no. 11, pp. 1514–1520, 2007
- [13].M. D. Omar Faruque et al., "Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis," in *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 2, pp. 63-73, June 2015, doi: 10.1109/JPETS.2015.2427370.
- [14].P. M. Menghal and A. J. Laxmi, "Real time simulation: A novel approach in engineering education," 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 215-219, doi: 10.1109/ICECTECH.2011.5941592.

- [15].Hamed, Ali Hazzab, Abdeldjebar. (2018). Modeling and Real-Time Simulation of Induction Motor Using RT-LAB. International Journal of Power Electronics and Drive Systems(IJPEDS). 9. 1476. 10.11591/ijpeds.v9.i4.pp1476-148
- [16].M. Bacic, "On hardware-in-the-loop simulation," Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 3194-3198, doi: 10.1109/CDC.2005.1582653.
- [17].R. Isermann, J. Schaffnit, S. Sinsel, Hardware-in-the-loop simulation for the design and testing of engine-control systems, Control Engineering Practice, Volume 7, Issue 5, 1999,Pages 643-653,ISSN 0967-0661, [https://doi.org/10.1016/S0967-0661\(98\)00205-6](https://doi.org/10.1016/S0967-0661(98)00205-6).
- [18].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.
- [19].OPAL-RT, "Power hardware in the loop." [Online]. Available: <https://www.opal-rt.com/power-hardware-in-the-loop/>
- [20].Kleijn, C. "Introduction to hardware-in-the-loop simulation." Control Lab 7521 (2014).
- [21].Fathy, H., Filipi, Z., Hagena, J., Stein, J.: Review of hardware-in-the-loop simulation and its prospects in the automotive area. In: Proceedings SPIE 6228, Modeling and Simulation for Military Applications (2006)
- [22].Jae-Cheon Lee and Myuug-Won Suh, "Hardware-in-the loop simulator for ABS/TCS," Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328), 1999, pp. 652-657 vol. 1, doi: 10.1109/CCA.1999.806729..
- [23].C. Jin, J. Tang, and P. Ghosh, "Optimizing Electric Vehicle Charging: A Customer's Perspective," IEEE Trans. Veh. Technol., vol. 62, no. 7, pp. 2919–2927, Sep. 2013.
- [24].J. de Hoog, T. Alpcan, M. Brazil, D. A. Thomas, and I. Mareels, "Optimal Charging of Electric Vehicles Taking Distribution Network Constraints Into Account," IEEE Trans. Power Syst., vol.30, no. 1, pp. 365–375, Jan. 2015.
- [25].E. Sortomme, M. M. Hindi, S. D. J. MacPherson and S. S. Venkata, "Coordinated Charging of Plug-In Hybrid Electric Vehicles to Minimize Distribution System Losses," in IEEE Transactions on Smart Grid, vol. 2, no. 1, pp. 198-205, March 2011, doi: 10.1109/TSG.2010.2090913.
- [26].M. Blonsky, P. Munankarmi and S. P. Balamurugan, "Incorporating Residential Smart Electric Vehicle Charging in Home Energy Management Systems," 2021 IEEE Green Technologies Conference (GreenTech), 2021, pp. 187-194, doi: 10.1109/GreenTech48523.2021.00039.
- [27].Ma, Tan, and Osama A. Mohammed. "Optimal charging of plug-in electric vehicles for a car-park infrastructure." IEEE Transactions on Industry Applications 50.4 (2014): 2323-2330.
- [28].R. Fachrizal, D. Van Der Meer and J. Munkhammar, "Direct forecast of solar irradiance for EV smart charging scheme to improve PV self-consumption at home," 2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), 2021, pp. 1-5, doi: 10.1109/ISGTEurope52324.2021.9640149.
- [29].R. Fachrizal, M. Shepero, D. van der Meer, J. Munkhammar, and J. Widen, "Smart charging of electric vehicles considering photo-voltaic power production and electricity consumption: a review," eTransportation, vol. 4, p. 100056, mar 2020

- [30].D. B. Richardson, "Electric vehicles and the electric grid: A review of modeling approaches, impacts, and renewable energy integration," Renewable and Sustainable Energy Reviews, vol. 19, pp. 247–254, Mar. 2013.
- [31].K. Zhang, D. Recalde, T. Massier and T. Hamacher, "Charging Demonstrator for Ancillary Service Provision in Smart Grids," 2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), 2018, pp. 988-993, doi: 10.1109/ISGT-Asia.2018.8467795.
- [32].Alyousef, Ammar, et al. "Design and Validation of a Smart Charging Algorithm for Power Quality Control in Electrical Distribution Systems." Proceedings of the Ninth International Conference on Future Energy Systems. 2018.
- [33].Flack, Calvin & Ucer, Emin & Smith, Charles & Kisacikoglu, Mithat. (2022). Controller Hardware-in-the-loop (C-HIL) Testing of Decentralized EV-Grid Integration.
- [34].R. Leou, C. Su and C. Lu, "Stochastic Analyses of Electric Vehicle Charging Impacts on Distribution Network," in IEEE Transactions on Power Systems, vol. 29, no. 3, pp. 1055-1063, May 2014, doi: 10.1109/TPWRS.2013.2291556.
- [35].S. Martinenas, K. Knežović and M. Marinelli, "Management of Power Quality Issues in Low Voltage Networks Using Electric Vehicles: Experimental Validation," in IEEE Transactions on Power Delivery, vol. 32, no. 2, pp. 971-979, April 2017, doi: 10.1109/TPWRD.2016.2614582.
- [36].Martinenas, Sergejus & Pedersen, Anders Bro & Marinelli, Mattia & Andersen, Peter & Træholt, Chresten. (2015). Electric vehicle smart charging using dynamic price signal. 2014 IEEE International Electric Vehicle Conference, IEVC 2014. 10.1109/IEVC.2014.7056150.
- [37].D. Gantenbein, C. Binding, B. Jansen, A. Mishra, and O. Sundstrom, "EcoGrid EU: An efficient ICT approach for a sustainable power system," in Sustainable Internet and ICT for Sustainability (SustainIT), 2012, Oct 2012, pp. 1-6
- [38]."AC Propulsion eBox, " <http://www.futurecars.com/reviews/ebox.html>.
- [39].Erotokritos Xydas, Charalampos Marmaras, Liana M. Cipcigan,A multi-agent based scheduling algorithm for adaptive electric vehicles charging,Applied Energy,Volume 177,2016,Pages 354-365,ISSN 0306-2619, <https://doi.org/10.1016/j.apenergy.2016.05.034>.
- [40].De Herdt, Lode, et al. "Power Hardware-in-the-Loop Demonstrator for Electric Vehicle Charging in Distribution Grids." 2021 IEEE Transportation Electrification Conference & Expo (ITEC). IEEE, 2021.
- [41].S. Martinenas, K. Knežović, 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.
- [42].Orchestrating Smart Charging in mass Deployment (OSCD), available: <https://oscd.eu/deliverables/>
- [43].Y. Yu, "OSCD Energy management system for smart charging of EVs V2.2," TU Delft, Delft, Tech.Rep., 2019.
- [44].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=d1913d792b323a652d63cb9eaf1ffbe](http://www.mennekes.be/index.php?id=latest0&L=5&tx_ttnews[tt_news]=842&cHash=d1913d792b323a652d63cb9eaf1ffbe)

- [45].International Electrotechnical Commission, “IEC 61851: Electric vehicle conductive charging system,” IEC, Tech. Rep., 2019.
- [46].T. Nederlof, “Implementations of Modbus Slave TCP/IP for Alfen NG9xx Platform,” Alfen N.V.,Tech. Rep., 2020.
- [47].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>
- [48].G. R. C. Mouli, P. Bauer, and M. Zeman, “System design for a solar powered electric vehicle charging station for workplaces,” Appl. Energy, vol. 168, pp. 434–443, Apr. 2016.
- [49].Nederlandse Energie- Data Uitwisseling (NEDU), “Verbruiksprofielen,” 2018. [Online]. Available: <https://www.nedu.nl/documenten/verbruiksprofielen/>
- [50].Sears J, Roberts D, Glitman K. A comparison of electric vehicle Level 1 and Level 2 charging efficiency. In2014 IEEE Conference on Technologies for Sustainability (SusTech) 2014 Jul 24 (pp. 255-258). IEEE.
- [51].J. García-Villalobos, I. Zamora, J.I. San Martín, F.J. Asensio, V. Aperribay, Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches, Renewable and Sustainable Energy Reviews, Volume 38, 2014, Pages 717-731, ISSN 1364-0321, <https://doi.org/10.1016/j.rser.2014.07.040>.
- [52].Q. Wang, X. Liu, J. Du and F. Kong, "Smart Charging for Electric Vehicles: A Survey From the Algorithmic Perspective," in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1500-1517, Secondquarter 2016, doi: 10.1109/COMST.2016.2518628.