UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

On the realization of a smart grid demo-site at Coteq in Almelo

Klaas Hoekstra

M.Sc. Thesis
Embedded Systems
August 2018

Supervisors:
dr. ir. M.E.T. Gerards (Marco)
prof. dr. ir. G.J.M. Smit (Gerard)
prof. dr. J.L. Hurink (Johann)
dr. ir. G. Hoogsteen (Gerwin)

Computer Architecture for Embedded Systems
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

The ongoing energy transition in the Dutch electricity grid may lead to grid capacity problems and mismatches between the production and consumption of electric energy. Renewable energy sources such as photovoltaic panels are introduced in residential areas and cannot be controlled. On the other hand, the increasing amount of electric vehicles and heat pumps increase the amount of energy that is consumed significantly. These new devices also provide a certain amount of flexibility to shift their consumption in time. By applying clever control, these devices can tackle the aforementioned problems. This is called decentralized energy management.

One of the stakeholders interested in controlling devices in a smart grid is Coteq, a distribution service operator in the Netherlands. To stimulate the development of smart grids, Coteq provides a demo-site containing solar panels, charging stations for electric vehicles and a stationary battery. This research focuses on the development of a smart grid for this demo-site.

The simulation and demonstration platform DEMKit is used as a basis to realize a smart grid demo-site. Within this platform we implemented and simulated the valley-filling algorithm, to determine the charging profiles of the electric vehicles. The valley-filling algorithm is split up into an offline part, which determines the fill-level using historic energy consumption profiles, and an online part, which determines the charging power of one or more electric vehicles using the determined fill-level and the current energy consumption of the demo-site. To enable the transition from a simulated environment to a real smart grid implementation, we extended DEMKit to acquire real-time input data for the algorithm. Measurement data is acquired from Influx databases in real time, and smart charging APIs are used to retrieve information about the electric vehicles, such as the state of charge of their batteries and when the charging has to be finished.

We tested the smart grid control in a pilot within the demo-site. The added features in DEMKit enabled a straightforward transition from a simulated environment to the pilot. We conducted an experiment where the developed algorithm is used to control the charging of multiple electric vehicles. The output data of the algorithm is communicated to the charging stations to control the charging profiles of the electric vehicles. The experiment resulted in reducing the imported energy by 15% and using 28% more of the energy produced locally instead of feeding it to the grid, when compared to a situation where no smart grid control is applied. Moreover, the peak power consumption is reduced by 23% and the peak power that is exported is reduced by 14%.

From the experiment, we conclude that using the valley-filling algorithm to control the flexibility of electric vehicles in a smart grid demo-site results in a reduction of stress on grid assets. The locally produced energy is utilized better and thus less energy is imported through the grid. Also, both the import and export peaks in the grid are reduced compared to a situation without control.

Samenvatting

De energietransitie die nu in volle gang is in Nederland, kan potentieel leiden tot overbelasting in het elektriciteitsnet en onbalans in de generatie en afname van elektrische energie. De productie van bronnen van groene stroom in het laagspanningsnet, zoals zonnepanelen kan niet worden gestuurd. Daarentegen, het stijgende aantal elektrische auto's en warmtepompen zorgt voor een aanzienlijke toename in de afname van elektriciteit. Echter bieden deze nieuwe apparaten een bepaalde mate van flexibiliteit, omdat ze hun energieconsumptie kunnen verschuiven over de tijd. Door slim met deze flexibiliteit om te gaan, kunnen we de bovengenoemde problemen aanpakken. Dit noemen we decentrale energiemanagement.

Een bedrijf dat geïnteresseerd is in het slim aansturen van apparaten in slimme netten is Coteq, een netbeheerder in Nederland. Om de ontwikkeling van slimme netten te stimuleren, stelt Coteq een demo opstelling beschikbaar, waar zonnepanelen, laadpalen voor elektrische auto's, en een batterij beschikbaar is. In dit onderzoek richten wij ons op het ontwikkelen van slimme aansturing voor deze demo opstelling.

Om deze slimme aansturing te realiseren, gebruiken we het simulatie en demonstratieplatform DEMKit. Met dit platform hebben we het zogeheten 'valley-filling' algoritme ontwikkeld en gesimuleerd, om de laadprofielen voor elektrische auto's te bepalen. We delen dit algoritme op in een offline deel, welke een 'fill-level' berekend op basis van historische consumptieprofielen, en een online deel, welke het laadprofiel van één of meer elektrische auto's bepaald aan de hand van het 'fill-level' en het huidige energieverbruik in de demo-opstelling. Om van de simulatie over te gaan naar het echt aansturen van apparaten, hebben we DEMKit uitgebreid met functionaliteit om real-time data te verwerven. Meetdata kan nu vanuit een Influx database verkregen worden, en API's voor slim laden worden gebruikt om informatie van de elektrische auto's te verkrijgen, zoals het percentage van de batterij en wanneer de auto opgeladen moet zijn.

We hebben de slimme aansturing getest tijdens een proef. De nieuwe functionaliteit in DEMKit zorgde voor een overgang van simulatie naar werkelijkheid die rechttoe rechtaan was. Een experiment is uitgevoerd waarin we het ontwikkelde algoritme gebruikten om meerdere elektrische auto's slim aan te sturen. De uitvoer van het algoritme werd naar de laadpalen gestuurd om zo de laadprofielen van de elektrische auto's te beïnvloeden. Gedurende het experiment konden we de hoeveelheid geïmporteerde energie verminderen met 15%, terwijl we 28% meer van de lokaal opgewekte energie gebruikt hebben, wanneer we het experiment vergelijken met een situatie waar geen slimme aansturing toegepast wordt. Daarbij komt ook nog eens dat de piekbelasting voor de geïmporteerde energie verlaagd is met 23% en de piekbelasting door lokale productie verlaagd is met 14%.

Dankzij dit experiment mogen we concluderen dat het gebruik van het ‘valley-filling’ algoritme voor het slim aansturen van de flexibiliteit van apparaten leidt tot een vermindering in de belasting op componenten in het elektriciteitsnet. Ook wordt de lokaal opgewekte energie beter benut, wat betekent dat er minder energie vanuit het net geïmporteerd hoeft te worden. Ook zorgt dit voor een vermindering in de piekbelasting voor zowel het importeren als het exporteren van energie, vergeleken met een situatie zonder slimme aansturing.

Contents

Chapter 1

Introduction

This thesis describes the final project of realizing a smart grid pilot at a demo-site located at Coteq in Almelo. The introduction motivates why to realize such a demo-site in Section 1.1, states the problems to overcome in order to realize the demo-site in Section 1.2, leading to the research questions to be answered, and gives an outline for the rest of the thesis in Section 1.3.

1.1 Motivation

Multiple stakeholders are interested in the realization of the smart grid demo-site at Coteq, and all have their own motivation. Here, the social importance of this research, the motivation for Coteq, and the motivation for the University of Twente are discussed.

1.1.1 Energy transition

The energy networks in the Netherlands consists of a nationwide electricity- and gas transmission and distribution network infrastructure [1]. These networks play an important role in the energy supply chain. The electricity transmission and distribution networks, jointly called *(electricity) grid* from now on, can be divided in an extra high voltage (EHV), a high voltage (HV), a medium voltage (MV), and a low voltage (LV) part, where the EHV and HV parts of the grid are used to transport electricity over large distances. In the *MV network*, the voltage is stepped down to connect local industry and large office buildings to the grid. Lastly, the *LV network* is used to provide availability of electricity in the residential areas [2].

Within the electricity grid, a clear distinction between producers and consumers was made in the past. Producers are power generators that insert energy in the grid and were historically situated within the HV grid. Consumers draw energy from the grid and range from large industries on the HV grid to the residential users connected to the LV grid. This supply chain was designed using a paradigm based on centralized energy production using fossil fuels, where electricity only flows in one direction, from producers in the HV grid to consumers in the MV and LV grids. The grid also has no means of storing energy, leading to the requirement that production and consumption of electricity in the grid must be balanced at all times.

In the Paris climate agreement that has been drafted in November 2015 and signed by 195 countries as of June 2018, it was agreed to limit the increase in the global average temperature to below 2 degrees, and try to limit this increase even further to 1.5 degrees [3]. Also, in the past, every newly built house in the Netherlands had the right to be connected to the gas grid. However, a change in the Dutch law implies that as of July 1018, no access to the gas grid is granted in applications for building permits for new houses or neighbourhoods [4]. Hereinafter, a trend is expected that existing neighbourhoods also lose this gas grid connection over time.

Because of the sketched changes resulting from the climate agreement, the situation in the energy supply chain has to change, on both the production and consumption side of the grid. The share of energy produced by renewable energy sources (RESs) increases and the number of devices that use electricity as their source of energy instead of other sources of energy increase as well. This change in the energy supply chain is called the *energy transition* and consists of the introduction of *renewable energy production* and the *electrification of the energy demand*.

Renewable energy production

To reduce the amount of fossil fuels used for electricity production, the share of energy produced by alternative RESs in the grid increases. This is a key aspect of the energy transition. A large part of renewable energy comes from wind parks, directly connected to the HV grid. In addition to the generation of electricity through wind, an increasing amount of renewable energy is produced by distributed producers (RESs) in the LV grid. These RESs are mainly photovoltaic (PV) panels that produce electricity from solar energy. Various facts and figures of the energy usage in the Netherlands are given in [5]. For example, in 2016, 5.6% of the produced energy in the Netherlands originates from a renewable source. In 2016, all households with solar panels could deliver a total peak power of 1407 MW, which is significantly more than the 651 MWp in 2014. Agreements such as the Paris climate agreement make it likely that the share of renewable energy continues to increase rapidly [3].

A characteristic of renewable energy is that most of its production is uncontrollable. This makes it challenging to balance the production and consumption of energy in the future electricity grid. Furthermore, when a household produces more energy than it consumes, it temporarily becomes a net producer in the electricity grid, leading to electricity flowing in the reverse direction. In the current LV grid setup, this may lead to protection issues when too much power flows from one house to another while bypassing overload protection assets.

Electrification of the energy demand

As mentioned, another aspect of the energy transition is the replacement of consuming assets that use non-renewable and non-electrical sources of energy (e.g., natural gas) by assets that use electricity or biogas as their source of energy. In this way they can ideally consume energy generated by RESs, as most RESs produce electricity. This electrification of the energy demand in the energy supply chain results in a shift from the distribution of energy using carriers other than electricity, such as natural gas, towards electricity as energy carrier. Because of this, the electricity grid faces a tremendous increase in electricity that it needs to distribute.

When looking at the total energy usage in the residential areas, we note that a large part of the total energy demand comes from heating households. In 2016, on average 43% of the total energy consumption of a household was used for heating, making it the largest energy consuming asset in the house [5]. Modern heating systems for buildings in the residential area, such as heat pumps, use electricity to replace natural gas used by traditional heating solutions as boilers. This increases the electricity demand for houses. In 2016, 85% of households in the residential area were heated with gas boilers, and only 1.5% used electric heat pumps. But the amount of installed heat pumps increased with 30% compared to the previous year [5]. Moreover, the aforementioned change in the Dutch law makes that using natural gas to heat houses may no longer be possible in all newly built houses.

The second largest energy consumer in a household is the car, being responsible for approximately 31% of the total energy consumption [5]. This means that together, the heating system and the car make up for 74% of the total energy consumption, which at this time does not come from the electricity grid, but from natural gas and fossil fuels. To use electricity instead of fossil fuels for the car, electric vehicles (EVs) are introduced. At the end of 2016, only 3% of the passenger cars in the Netherlands were (partly) powered by electricity, and 95% still used internal combustion engines for propulsion. Although 3% may not sound much, the Netherlands has the second-highest stock share of EVs in Europe [6]. Moreover, when comparing this to the amount of EVs in the Netherlands in the year before, a relative increase of 17% is observed [7].

The introduction of EVs causes an increase of electricity demand in residential areas due to the need to charge them. Currently, EVs are charged at home for a large part. The daily energy usage of a typical EV is about equal to the daily energy consumption of an average Dutch household [8]. As a consequence, installing charging stations in the LV grid in the residential area leads to a rapid increase of the energy consumption in this grid. This gets even worse when charging stations are used with a high output power, called fast chargers, For example, a Tesla Supercharger can charge a Tesla with a charging power of about 120 kW [9]. This is equivalent to the power that 120 households draw during the evening peak, or the average consumption of 240 households through the day. Such fast chargers are not installed in the residential area.

Another challenge is the simultaneity of EVs charging in the residential area. This causes a potential threat in violating grid capacity constraints, because most people start charging their EV around the same time. In a regular residential area this is after work. Furthermore, charging an EV takes relatively long, increasing the chance of simultaneity compared to an electric kettle for example.

One possibility to balance the consumption and uncontrollable production of renewable energy, and to make sure constraints on grid capacity are not violated, is to control the load on the grid. Many of the new assets provide a certain amount of flexibility on the demand side of the grid. Resources with such flexibility are called DERs. For example, the charging of an EV can be delayed or the amount of charging power can be reduced to some extent. Since the production of renewable energy is uncontrollable, this demand side flexibility may be used to keep production and consumption in the electricity grid in balance.

1.1.2 Interest Coteq

In the Netherlands, multiple stakeholders are responsible for operating and maintaining the electricity grids and to facilitate the distribution of electric energy to consumers. There are two types of these stakeholders: transmission system operators (TSOs), who are responsible for the transmission network (the HV grid), and distribution system operators (DSOs), who are responsible for the MV- and LV grids [1]. The problems caused by the energy transition in the Dutch electricity grid ask for a proper treatment by the involved stakeholders.

As one of the DSOs in the Netherlands, Coteq is such a stakeholder and is responsible for connecting producers and consumers to MV- and LV grids in some parts of the Netherlands. Coteq is interested in finding proper solutions for the challenges introduced in the grid due to the energy transition. To ensure connections in their part of the grid are maintained, constraints on minimum and maximum voltage in the grid, the deviation on the frequency and the power quality are defined [10]. Also, the load in the LV grid is constrained by the transformer rating, the currents that can flow through conductors, and the rating of protective fuses in the grid.

The introduction of PV in the residential area causes households in the LV grid to produce electricity during peak sun hours, leading to feedback of electricity to the grid. The peaks in the grid caused by this production may exceed the maximum capacity in the LV distribution grid, potentially causing transformers to overload and cables to overheat. Also, the increasing amount of EVs that are charged in the low voltage grid may cause load peaks leading to stress and overloading, resulting in faster degradation of assets in the distribution grid, or even a blackout [11]. Since the DSOs are responsible for keeping the connections in the LV grid uninterrupted, it is in their interest to find methods and solutions to avoid blackouts and severe ageing of grid assets.

1.1.3 Interest University of Twente

The Computer Architecture for Embedded Systems (CAES) chair and the Discrete Mathematics and Mathematical Programming (DMMP) chair of the University of Twente collaborate in solving energy related research questions [12]. The active research field of these groups is to provide intelligent smart grid control concepts and energy management methodologies with a focus on residential (micro) grids. Previous research is in creating demand side planning algorithms for assets in the low voltage grid, and the development of a flexible smart grid simulation and demonstration platform, called DEMKit [13].

This group already has knowledge concerning smart grid technologies and optimization algorithms, and DEMKit allows for detailed simulations of grid assets in a smart grid environment. In the past, implementations of smart grids have already been realized in field tests [14]. This provides a good basis to take the development of smart grid sites to a next level. However, the previous implementations were isolated and stand-alone pilots. The objective within this demo-site is to develop a system that is more general applicable and can easily be transferred to other demo-sites as well, by making use of DEMKit.

The interest of the University of Twente is to gain knowledge in how to move from a simulated environment to a real-life demo site using DEMKit. Moreover, they have the vision to merge the controlling of a real smart grid into DEMKit, by using hardware in the loop and co-simulation techniques. This way, the development time from a simulated smart grid to a real and working smart grid should decrease significantly. On the other hand, such a tool provides a more structural approach to deploying smart grid systems, in which it becomes easier to prevent issues.

1.2 Problem statement

The goal of this research is to realize a smart grid demo-site at the location of Coteq in Almelo. Multiple grid assets are already in place, such as PV panels on the rooftop of the buildings, EV charging stations to charge both EVs from Coteq's pool of vehicles and EVs owned by employees, and a battery to store (locally) generated electricity. However, no coordination mechanism to make these assets cooperate is in place. Coteq would like to implement a smart grid to control these DERs and to provide a demo for other stakeholders to see the potential of smart grid systems. Besides that, the University of Twente wants to use this opportunity to extend their DEMKit simulator to enable a robust transition from a simulation environment to a real-life smart grid system, for different cases and grid assets. These goals lead to the following research question:

> *How to design and implement a decentralized energy management (DEM) system at Coteq to control the flexibility provided by energy resources in a grid, with the goal of reducing stress on grid assets and balancing consumption and production, while providing insight on the effect of smart grid control to its stakeholders?*

To answer this research question, the following sub-questions are formulated:

1. *How to use DEMKit as a basis to implement a DEM system?*
2. *What energy resources at Coteq provide flexibility and to what extent do they provide this flexibility?*
3. *What optimization algorithm should be used to control the flexibility?*
4. *How can this algorithm be implemented in the demo-site at Coteq in Almelo?*
5. *How to provide stakeholders with easy insight on the effects from this DEM control system?*
6. *To what extent does DEM control reduce stress on grid assets and to what extent does it balance consumption and production?*

1.3 Outline of the thesis

The remainder of this thesis is organized as follows. In Chapter 2, background information related to the subject of the thesis is described. In Chapter 3, the hardware and software infrastructure of the demo-site at Coteq are elaborated on, and a model of the demo-site is created in DEMKit. Subsequently, DEM optimization algorithms for EV charging are introduced and explained in Chapter 4. The implementation of these algorithms in the demo-site is described in Chapter 5 and in Chapter 6 we discuss the experiments and results to test the DEM algorithms in the demo-site. Lastly, conclusions and recommendations are given in Chapter 7.

Chapter 2

Background

This chapter describes background information on subjects related to the research. First, the Dutch grid and its history is introduced in Section 2.1. Then, decentralized energy management (DEM) is explained and different classes of DEM approaches are introduced in Section 2.2. After that, the DEMKit simulator is explained in Section 2.3, and lastly we explain how charging an EV works in Section 2.4.

2.1 Grid infrastructure

The electricity grid in the Netherlands slowly evolved from many local electricity grids in the large cities, to a nationwide grid. It all started in 1886, when the first electricity network in the Netherlands was operational, powering 300 lights in Kinderdijk [15]. More and more local electricity grids arose in the Netherlands and were integrated to share their generators. Through governmental regulations, the rural areas later got connected to the grid as well. To facilitate the total demand, large national power lines were placed, which connected all generators to a nationwide power grid. This grid has been the basis for the centralized, alternating current (AC), 50Hz, 230V, three-phase LV electricity grid that we use today.

2.1.1 The centralized electricity grid

When considering only large centralized producers in this centralized grid, energy flows in only one direction: from the producers to the consumers. In order to minimize transport losses between producers and consumers, the grid is divided into multiple voltage levels. Chapter 1 already mentioned these voltage levels. Table 2.1 shows the different grids and their corresponding voltages. The EHV and HV grid are called transmission networks and are mainly used to transport electricity on a national level, over large distances. Large power plants are connected to the HV grid, together with some heavy industrial consumers (See Figure 2.1). The Dutch TSO TenneT is responsible for maintaining the EHV and HV networks.

	Voltage Level	Operator
EHV	220 kV, 380 kV	TSO
HV	50 kV, 110 kV, 150 kV	TSO
MV	3 - 25 kV	DSO
LV	230/400 V	DSO

Table 2.1: Voltage levels in the Dutch electricity grid

The MV and LV grids are called distribution networks, and multiple DSOs in the Netherlands are responsible for the maintenance and availability of these grids. From the HV grid, the voltage is stepped down to the MV grids, as shown in Table 2.1. This grid supplies medium sized industry and enterprises, but also medium sized power plants can supply energy to this grid. Lastly, the voltage is stepped down once more to create the LV grids. Small business and residential consumers are connected to this network.

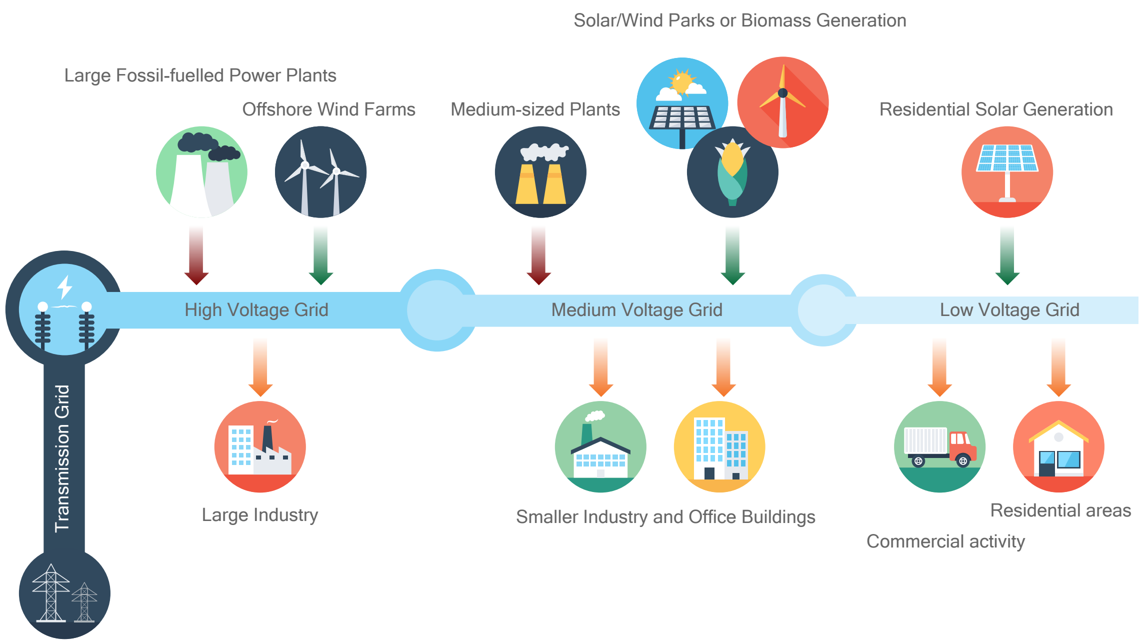


Figure 2.1: Overview of the European electricity grid levels ([16])

2.1.2 Shifting towards a decentralized electricity grid

As already sketched in Chapter 1, (decentralized) renewable energy sources (RESs) now also supply energy to the electricity grid, next to large centralized producers, as part of the energy transition. These RESs are connected to different levels of the electricity grid. Large wind farms can be connected directly to the HV grid, and smaller wind parks, solar parks, and biomass generators connect to the MV grid (see Figure 2.1). Lastly, residential PV production supplies energy to the LV grid. Because of the introduction of these RESs, energy is not only supplied to the grid from some centralized fossil fuelled generators in the HV grid, but also from many uncontrollable, decentralized RESs in all levels of the electricity grid. This is called a decentralized electricity grid and currently the Dutch electricity grid is ongoing a shift towards such a decentralized grid.

2.2 Decentralized energy management

Next to the introduction of uncontrollable RESs in all levels of the grid, also the electrification of the energy demand, as described in Chapter 1, plays an important role in the energy transition.

Both the introduction of RESs and the electrification of the energy demand introduce challenges in the distribution of electricity in the grid. The introduction of consuming assets that use electricity as their source of energy to replace assets that use non-renewable sources of energy (e.g., natural gas), increase the load on the electricity grid. The increased load can cause grid capacity problems, especially with the high probability of simultaneity in these assets.

A core aspect of the electricity grid is that there is no notable storage and thus production and consumption have to be equal at all times. In the current grid, mainly fossil fuelled generators supply energy to the grid, which are following the energy demand. However, the production of RESs is uncontrollable and thus can not follow the demand.

Some of the new assets (e.g., EVs or heat pumps) provide a certain amount of flexibility in the time at which they consume energy. Resources that provide such a flexibility and allow for use of local energy to reduce the need for distribution capacity are called distributed energy resources (DERs). The increased load on the grid caused by DERs is to some extent flexible, and by managing when and how DERs consume their energy, the demand can follow the available supply by RESs to some extent. This principle is called energy management (EM). Moreover, when also considering where in the grid energy is produced and consumed, one speaks of DEM [17]. When using DEM, transmission losses may be reduced, as generated energy is used locally. DEM takes into account both energy supplying assets (e.g. PV) and energy demanding assets (e.g. EVs) in the local grid.

2.2.1 Online and offline DEM

In [18] it is stated that two main classes of DEM approaches can be distinguished. Firstly, the online, or real-time, class of DEM approaches steer the system based on the current state of the grid and do not look ahead. An example of an online DEM approach is the auction-based PowerMatcher, as presented by Kok in [19], [20]. The second class of approaches is offline, or planning-based DEM, which uses predictions of the future to optimize the energy profile for an upcoming period of time (often a day) [18]. An example of a planning-based DEM approach is the profile steering approach as presented in [21], [22]. In more recent research, DEM approaches are presented that combine offline and online DEM approaches. The valley-filling algorithm is such a combined DEM approach [23]. More information about these and more approaches can be found in the literature study in Appendix A.

2.3 DEMKit simulator

To observe the behaviour of DERs in a smart grid environment, a simulation and demonstration platform called DEMKit is developed at the University of Twente [13], [17]. The goal of DEMKit is to enable efficient and effective research on DEM approaches. It is developed using the Python programming language.

2.3.1 Components of DEMKit

Different components are available in DEMKit to quickly prototype and test DEM approaches. These components represent devices or DERs, control logic, or grid infrastructure assets. Using these components, e.g. a scenario can be modelled that represents a LV grid in a residential area. In the following, the relevant components that are available in DEMKit are discussed.

Devices

Components that represent devices in and around the house are available in DEMKit. The device components are set up in a way that they represent the properties and behaviour of the devices and reflect the behaviour of the real device. For example, a device may have a consumption or production profile that states when and how much energy a device produces or consumes, or a device may have properties of its capacity to store energy. We define eight different classes of devices.

Firstly, uncontrollable devices represent a type of device where control is not possible, meaning that these devices are no DERs (i.e., lights, televisions, computers, etcetera). A variant of the uncontrollable device class is the class of curtailable devices. PV panels are modelled as curtailable devices, since the production of PV is only controllable in one direction: to produce less energy.

The second class of devices is called the buffer device. A buffer can store energy and feed it back later. A buffer is used to temporarily store energy produced locally, such that this energy can be consumed later. This may reduce transportation of energy. A buffer can also be used to perform peak-shaving, by storing or feeding back energy to the system at times of the highest peaks. In a sense, a buffer allows to decouple the time of production of energy and when it is consumed.

A converter is a device that converts energy from one form to another. For example, a combined heat power (CHP) system converts natural gas into electricity and heat. This electricity from the CHP system may be used by other devices. The energy usage of a converter depends on the demand on the secondary side of the converter, which must be fulfilled through energy conversion from the primary side. Often, a converter is coupled to a storage, e.g., a heat pump coupled to a heat store or thermal mass. This storage or buffer adds flexibility to a converter, because the demand of the secondary side does not have to be fulfilled immediately by the primary side. Therefore, another device called a buffer-converter is introduced, which inherits from both the buffer device and the converter device to model its combined behaviour.

The next class of devices is called a timeshiftable device. Such devices do not have any buffering capabilities, but have the ability to shift their consumption to a later moment. Examples of such devices are smart white goods (e.g., washing machines, dryers). A timeshiftable device has a certain consumption profile, but the time when it starts is flexible. An important aspect of timeshiftable devices is that they have a deadline for their operation, which is the latest time when the device must be finished. Just as with the converter class, there also exist timeshiftable devices that contain a buffer, called buffer-timeshiftable devices. Herein, the fixed consumption profile is replaced by a buffer that has to reach a certain state of charge (SoC) before the deadline. An example of a buffer-timeshiftable device is an EV, which has a buffer that must be charged. But besides that, an EV has a deadline when it must be fully charged or charged up to a certain level. This property makes that the timeshiftable class is used next to the buffer class as a base for this type of devices.

Lastly, a meter device is available in DEMKit as well. Other devices are connected to such a meter, and this meter aggregates the consumption of all the devices connected to it. This way, the energy consumption of a group of devices can be monitored.

Controllers

For every class of devices, a controller component is available as well. A controller gives possibility to control the flexibility of a device. If no controllers are added in a scenario, devices have a default behaviour, which usually boils down to consuming energy as soon as possible.

Controllers contain the logic of the DEM approach. A DEM approach has certain requirements it must meet, for example in not violating grid constraints or peak-shaving on a transformer level. The control actions of the controllers are based on these requirements. Next to device controllers, a group controller may be available, which aggregates the profiles of all device controllers connected to it. This enables the control logic to be hierarchical. Based on the optimization goal on the highest hierarchical level, group controllers request their device controllers to send certain control actions to their devices that change the state of the system towards this optimization goal. Figure 2.2 gives an overview of all classes of devices and their controllers in DEMKit.

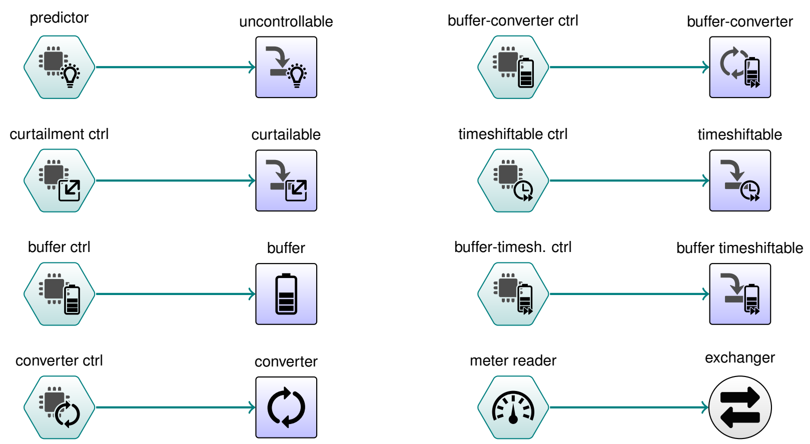


Figure 2.2: Devices and their controllers in DEMKit [17]

Grid infrastructure

To respect grid constraints in a simulation, a model of the grid infrastructure is needed. DEMKit can be used to model LV grids and simulate the current flow through and voltage drop over cables, using a load-flow solver. This information may be used to prevent overloading in the grid. Information about the topology of the grid and the length and diameter of the cables is required for these calculations. When too much power is expected to flow through a cable, controllers of devices connected to that cable section can control their flexibility to decrease the power.

2.3.2 Simulation logic

A scenario can be created using multiple components (devices, controllers, grid infrastructure) that are connected together. Such a scenario is then loaded in DEMKit to simulate the effect and behaviour of the scenario. Multiple steps are executed in the simulation. At the start of the simulation, each component within the scenario is initialized. After that, time progression is simulated based on discrete time intervals.

At every discrete time interval, the simulator performs six steps. First, the states of all devices are updated. These states depend on the class of device. It contains the current power consumption and can also contain the SoC of its internal buffer, etc. Based on the state of the total scenario, controllers then determine their control actions for the devices in the system. The devices then act upon those control actions, but also validate whether the desired control actions are actually possible. In the last steps, the consumption of all devices is aggregated, so a flow calculation can be executed. From the results of these calculations, it can be determined if the realized energy consumption in the scenario is within grid constraints.

Figure 2.3 shows the structure of DEMKit. The dotted lines indicate that objects can interact with each other. The host ties together all components within a scenario. Depending on the use, a different host object may be used, such as the simulation host or a clock host for real-time demonstrations. The output of the simulation/demonstration is written to a database. When the host is used to simulate discrete time steps, an arbitrary time is used and all discrete time steps are executed directly after each other. In a clock host, the current time is used and each time step is executed at the real time. For example, when discrete time steps of fifteen minutes are used, the simulator host executes all time steps directly after each other, but the clock host only executes a time step every fifteen minutes.

2.3.3 DEMKit work-flow

DEMKit has been developed to enable efficient and effective research on DEM approaches, so that these algorithms can be used in a smart grid. Using DEMKit for this purpose involves a so called work-flow, that starts by modelling a scenario, as mentioned above. Components of different classes and a control layer are added to the scenario. As stated, algorithms of DEM approaches are implemented in this control layer. Then, the scenario is simulated using the simulation logic in DEMKit. The simulation results are used to verify whether the requirements of the DEM approach are met. If not, the models and algorithms may be refined. This process is

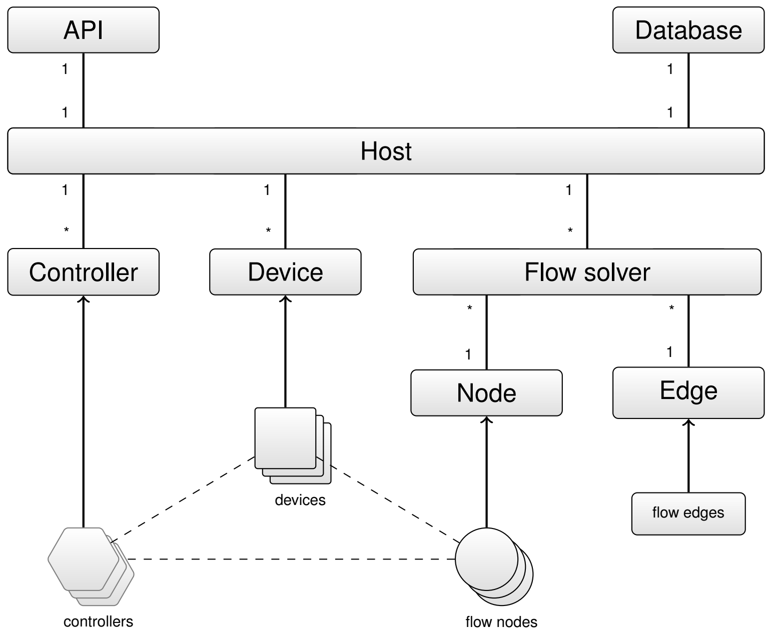


Figure 2.3: Components in DEMKit and their interconnections (From [17])

continued until the requirements are met. Then the resulting algorithms are ready to be tested in a smart grid demo-site.

One goal of this research is to extend this work flow, by enabling hardware-in-the-loop or co-simulations with real devices connected to DEMKit. This way, the algorithms can be validated in a smart grid using DEMKit. When an algorithm gets refined, it can first be tested in a simulation environment before deploying it in a real smart grid. Moreover, since these refinements are done in DEMKit as well, they are also used when the algorithm is deployed in another smart grid. Moreover, DEMKit has the ability to use historical measurement data as input for the models. This way, scenarios can be (re)simulated using realistic conditions for validation. Figure 2.4 shows a flowchart of the work flow in DEMKit. As can be seen, in the current situation DEM algorithms can be modelled, simulated and refined until the desired result is obtained. With this research, the work flow is extended such that the DEM algorithm is also implemented and validated in a smart grid. From these results, refinements in the algorithms can be made, simulated, and implemented until the requirements are met.

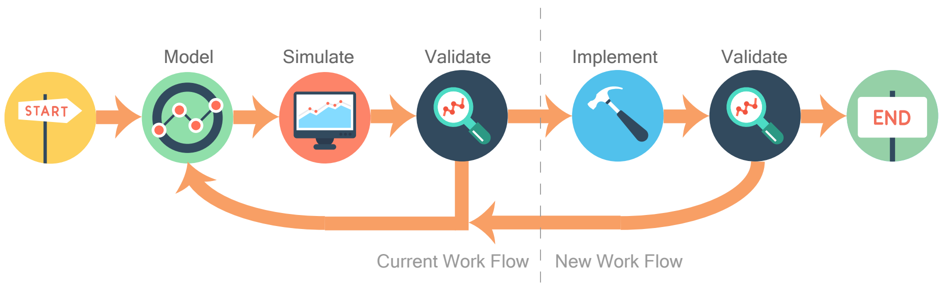


Figure 2.4: Flowchart of the work flow in DEMKit ([16])

2.4 Electric vehicle charging

As already mentioned, the introduction of a large number of EVs in the current grid has a risk to overload the cables if too many of them are charged at the same time. To reduce the simultaneity in when EVs are charged in a residential area, a DEM system can be used to control the flexibility of EVs. In order to implement a DEM system to charge EVs, it is required to know the methods, interfaces, and protocols used to charge an EV. This part of the thesis elaborates on these methods and techniques. Subsection 2.4.1 gives an overview of the different charging levels, modes and plug types, and Subsection 2.4.2 explains the communication protocol.

2.4.1 Charging levels, modes and plug types

To charge an EV, it must be connected to a charging station, or electric vehicle supply equipment (EVSE). The International Electrotechnical Commission (IEC) has described several standards for EV charging. Firstly, three levels of charging can be defined. Moreover, the IEC defines four modes for charging an EV in the IEC 62196 standard [24]. Also, four plug types to connect an EV can be distinguished. These aspects are presented in more detail in the following.

Charging levels

In the IEC 61851 standard, three charging levels are classified for the type of charger and the charging power [25]. The levels are described as normal power or slow charging, medium power or quick charging, and high power or fast charging respectively. The used connection, current and power for each level are given in Table 2.2.

	Connection	Max current	Power
Normal power	single phase AC	10 - 16 A	<3.7 kW
Medium power	single or three phase AC	16 - 32 A	3.7 - 22 kW
High power	three phase AC or DC	>32 A	>22 kW

Table 2.2: Charging levels

Charging modes

The four charging modes define how power is transferred to the EV and what communication takes place between the charging station and the EV. *Mode 1* describes slow charging through a regular household socket to the EV, at a maximum current of 16A. This can be either a standard single or three phase connection, so the maximum charging rate is either about 3.7 kW for a single phase connection, or about 11 kW for a three phase connection. An EV contains an on-board charger, to rectify the incoming AC supply. The first mode has no communication and the socket has to provide earth to the EV to have ground fault protection.

The second mode is similar to the first, since it uses a regular single- or three phase household socket. Opposed to the first mode, *Mode 2* uses a special cable with some electronics built into it. Because of this cable, the maximum current is now 32 A per phase. The electronics in the cable provide control and protection. Note that in order to supply 32 A, the socket must be suitable to do this as well. *Mode 1* and *Mode 2* are commonly used for domestic charging.

In *Mode 3*, specific EV charging sockets are introduced. *Mode 3* chargers cannot be connected to a regular household socket, but only to fixed EV charging stations, or EVSE, equipped with such an EV socket outlet. These charging stations have the capability to provide three phase charging, but the EV determines whether three phases or one phase is used for charging. The IEC standard describes a maximum charging current of 32 A, depending on the type of plug and cable used. The EVSE is not active by default, and a communication channel is used to activate charging. This communication channel is also used to determine the maximum charging power of the EV and cable used. *Mode 3* charging stations are commonly seen in public or semi-public parking lots in the residential area.

The last mode describes direct current (DC) fast charging. Here, the internal charger of the EV is not used, but a charger in the EVSE rectifies the AC mains supply outside of the EV and feeds the DC directly into the EV. With *Mode 4* charging, charge powers of up to 350 kW are possible. The charging modes can be linked to the charging levels as follows:

Mode 1 $\rightarrow$ Normal power or Medium power charging level
Mode 2 $\rightarrow$ Normal power or Medium power charging level
Mode 3 $\rightarrow$ Medium power or High power charging level
Mode 4 $\rightarrow$ High power charging level

Plug types

Lastly, four types of plugs are defined. These plugs are in the EVSE and connect to the cable from the EV. Setups using *Mode 3* or *Mode 4* charging use these EV specific plugs. The first plug type is for single phase charging, and can only supply AC mains. Its maximum current is rated at 32 A. The second plug type enables either single or three phase AC charging, and has a maximum current rating of 63 A per phase or 70 A when only one phase is used. Both plugs contain a proximity pilot pin and a control pilot pin. The first is used to inform the EV that it has been connected to an charging station and the latter is for communication between them. The *Type 2* plug also contains a protective earthing system.

For both the *Type 1* and *Type 2* plugs, a so called combined charging system (CCS) plug, is developed as well. These plugs combine the AC charging characteristics of the original plugs with DC fast charging pins in the same connector. This way the EV can use the same plug for both AC and DC charging. Note that when using AC charging, it is called *Mode 3* charging, but when the DC pins are used on the CCS plug, it is called *Mode 4* charging.

Plug *Type 3* also enables single and three phase AC charging, but up to 32 A per phase. The difference with the *Type 2* plug is that this plug has a protective shutter to make sure nobody can touch the potential hazardous pins when no EV is connected. However, these plugs are mostly used for *Mode 3* charging, and *Mode 3* requires the socket to be inactive when no EV is connected, so there is no hazard the shutter could protect from. Besides that, public charging stations expose the socket and plugs to a harsh environment, and it is expected that the shutter can easily malfunction. Therefore, the *Type 2* plug has become more popular than *Type 3*.

Plug *Type 1* and *Type 2*, and their CCS variants all use the power-line communication (PLC) protocol for communication between the EV and EVSE. For this protocol, a control pilot (CP) pin and a protective earth (PE) pin are present in the plugs.

Finally, the last plug type is designed for DC fast charging. Specifically, a charging method under the trade name CHAdeMO uses this plug. Since this plug is used for DC fast charging, it always uses *Mode 4* charging. The *Type 4* connection uses the controller area network (CAN) bus protocol for communication. Figure 2.5 shows the layouts of plug *Type 1*, *Type 2*, their CCS variants, and plug *Type 4*.

(a) Plug *Type 1* *(b) Type 1* CCS *(c)* Plug *Type 2* *(d) Type 2* CCS *(e)* Plug *Type 4*

Figure 2.5: EV plug types (From [26])

Charging scenarios

When combining the charging levels, modes and plug types, four different scenarios can be defined for charging an EV. Table 2.3 states these scenarios and shows the specifications for either single phase or three phase (ϕ) charging for the scenarios. One of the EVs available at the demo-site is the Opel Ampera-e that has a 60 kWh battery. The last column of the table shows the time it takes to charge this car by 75%, which means charging 45 kWh of energy. Note that in this range, the battery is charged using a constant current. When charging the battery to its full capacity, the charger switches to charging using a constant voltage at some point. This decreases the charging speed.

Scenario	ϕ	Level	Mode	Max Power	Socket Type	Ampera-e 75% (45 kWh)
Charging at home	1	Normal	Mode 1 / 2	3.7 kW	Domestic	12h
	3	Medium	Mode 1 / 2	11 kW	Domestic	4h
Semi-public charging	1	Medium	Mode 3	7.4 kW	Type 2 / 3	6h
	3	Medium	Mode 3	22 kW	Type 2 / 3	2h
Charging at a highway	3	Medium / High	Mode 3	43.5 kW	Type 2 / 3	1h
Public DC charging	-	High	Mode 4	120 kW	Type 4 / CCS	20m

Table 2.3: Different scenarios for charging an Ampera-e

2.4.2 Communication

In *Mode 3*, communication takes place between the charging station and the EV. In the protocol used, the charging station can detect the presence of an EV, limit the maximum amount of charging current, and the EV can request the charging station to supply power. This is done using a 12V, 1 kHz square wave signal on the control pilot pin, generated by the charging station.

In a standby situation, when no EV is connected to the charging station, the circuit from the control pilot (CP) pin to the protective earth (PE) is open, and the top voltage of the square wave signal between CP and PE is +12V. When a Mode 3 compatible vehicle is connected, a $2.7K\Omega$ resistor is connected between CP and PE, resulting in a maximum of +9V. The EV is not requesting power in this state. When the EV switches to a 880Ω resistor, the top voltage drops further to +6V, and the vehicle is ready to be charged.

The charging station communicates the maximum available current the EV can demand to the EV using pulse width modulation of the generated square wave signal. A duty cycle of 16% means the EV may request 10A of current at maximum, 25% corresponds to 16A of maximum current, and 50% to 32 A of maximum current. A duty cycle of 90% flags a fast-charge option. Since for AC charging, the charger is in the EV, the EV determines the amount of power the battery is charged with and thus the amount of power drawn from the charging station. The EVSE only communicates the maximum power that can be drawn.

Chapter 3

Demo-site at Coteq

The goal of this research is to realize a smart grid demo-site located at Coteq in Almelo. In order to implement a DEM system to control the DERs at the demo-site, it is of importance to first gather information about the demo-site. This chapter gives an overview of the demo-site at Coteq, the distributed energy resources (DERs) and renewable energy sources (RESs) that are available, and the hardware and software infrastructure. In this chapter we also discuss a model of the demo-site.

3.1 Overview

Coteq in Almelo has a demo-site, where a DEM system can be implemented. Different hardware assets are available in this demo-site. Furthermore, there are office buildings with their regular energy consumption, and some DERs and RESs. To control the hardware, a software stack is already in place.

3.1.1 Hardware infrastructure

Coteq has its own transformer that is connected to the MV grid. The LV side of he transformer is connected to a distribution cabinet. In this cabinet, multiple buildings of Coteq are connected. For each building, a meter is installed that measures the energy consumption of that building. One connection from the main distribution cabinet leads to the PV/EV distribution cabinet. Onto this cabinet, a total of 407 PV panels are installed. These panels are divided into four groups, each with its own inverter. In total, the PV panels can deliver a peak power of 101.75 kWp. Besides this, four EV charging stations are installed. One of these charging stations is an AL434 charging station that is located on the front side of the office buildings. The other three charging stations are EN-PRO44 charging stations, with each two EV connectors. These charging stations are located at the rear of the office buildings and are connected to the PV/EV distribution cabinet as well. Figure 3.1 shows a satellite overview of the demo-site. All charging stations make use of the Mode 3 charging standard for power transfer and communication between EVSE and EV, and Type 2 plugs to connect the EV to the EVSE, (see Chapter 2).

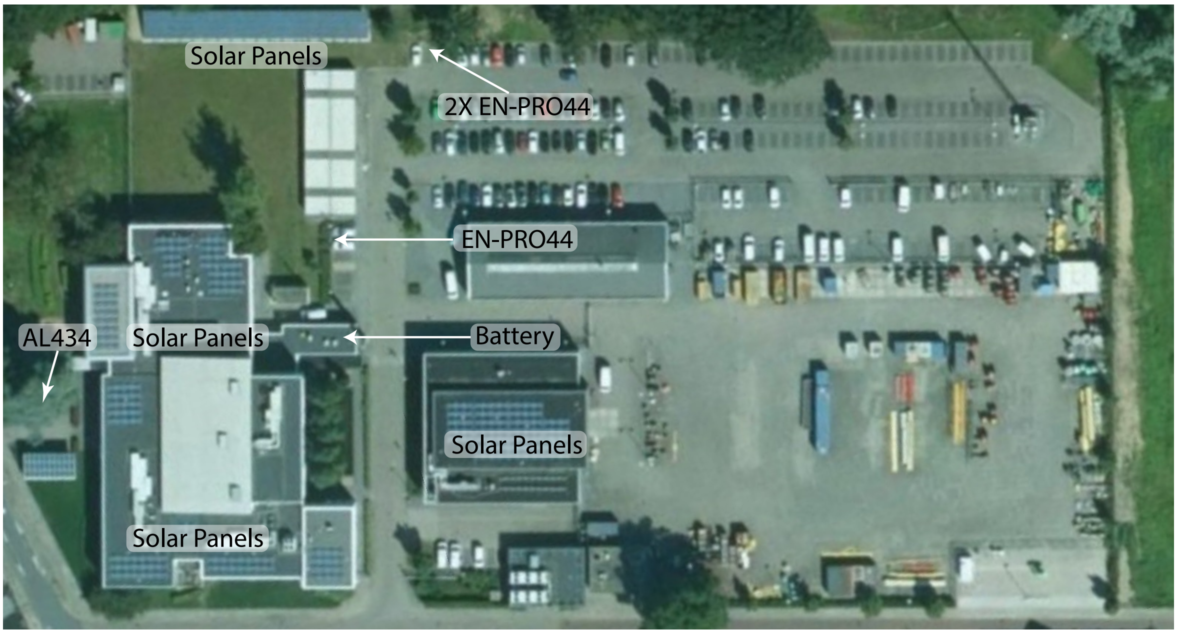


Figure 3.1: Satellite overview of the demo-site

To add an additional layer of flexibility, a battery is added to the demo-site. This battery consists of multiple battery cells with a total of 20 Ah of energy storage. The battery cells operate at a DC voltage of 400 V, making the capacity of it 8 kWh. An uninterruptible power supply (UPS) is used to charge and discharge the battery cells. A limitation of using the UPS, is that the battery can only switch between either charging from the grid or discharging. Unlike most stationary batteries, the implementation at this demo-site does not offer the opportunity to discharge stored energy back into the grid. Instead, two controllable EV charging stations are connected to the output of the UPS. This way, the battery can only be charged from the grid, and only discharged into EVs that are connected the specific EVSE. When the UPS is in charging mode, the battery cells are charged with a fixed charging current and any EVs connected to the EVSE are charged from the grid as well. Otherwise, when the UPS is in discharging mode, the battery cells cannot be charged and the discharge rate is dependent on the charging rate of the connected EVs. Figure 3.2 shows an overview of the complete hardware infrastructure of the demo-site.

3.1.2 Software infrastructure

Some software is already accompanied with the hardware. This software enables the communication between different assets in the demo-site. The charging stations use the Open Charge Point Protocol (OCPP) to communicate with a charge point management system (CPMS), which manages the charge points and authorizes users to use the charge points. This CPMS is running on a Raspberry Pi that is situated locally within the demo-site. Note that only the three charging stations at the rear of the buildings are connected to this CPMS. The AL434 charging station in the front communicates with a third-party CPMS through the cloud.

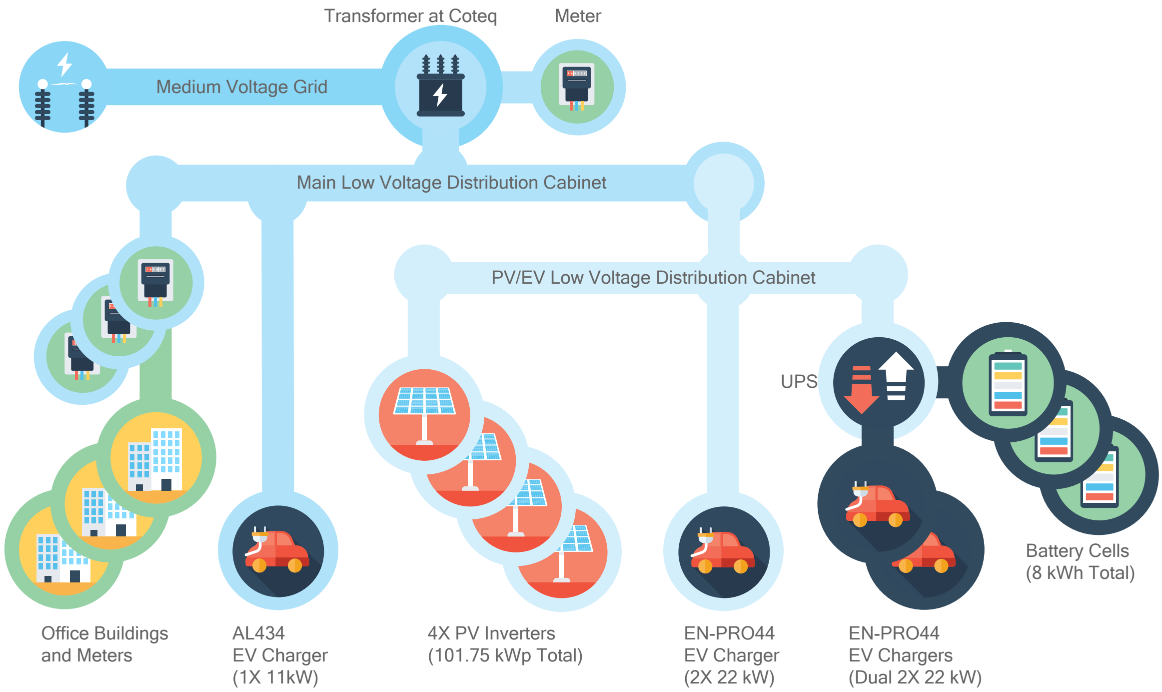


Figure 3.2: Overview of the hardware in the demo-site at Coteq ([16])

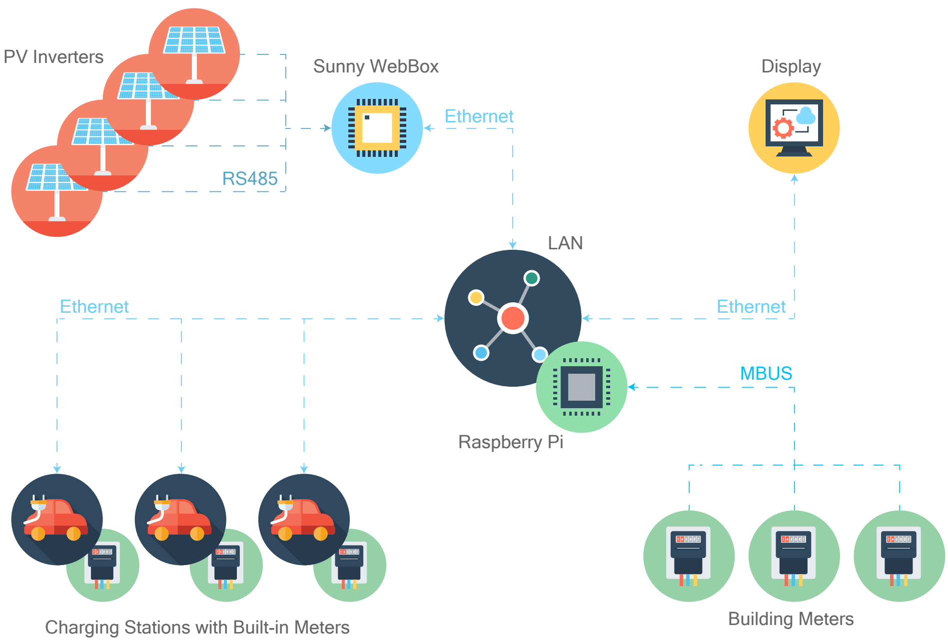


Figure 3.3: Overview of the Software infrastructure ([16])

A local area network (LAN) is set up to connect the rear EVSE to the Raspberry Pi. In the local CPMS, transactions are started to enable the charging of an EV at the EVSE. Also, the charging power that is measured by a meter in the EVSE is sent back and logged in the CPMS. All inverters of the PV panels communicate with a single Sunny WebBox, which is connected to the same LAN. In the Sunny WebBox, the energy produced by the PV panels is monitored and stored in a database. The meters of all buildings at the demo-site communicate their consumption through the MBUS protocol to the Raspberry Pi. This data is sent to a database as well.

In order to realize the smart grid pilot, the DEM system uses the CPMS on the Raspberry Pi. It controls the charging of EVs by setting the charging power through the CPMS. To give insight about the effectiveness of the DEM system, a display is connected to the LAN as well. The Raspberry Pi sends information about the state of the system to this display, so that stakeholders can see the effect of DEM in the demo-site. Figure 3.3 gives an overview of the software infrastructure and the communication protocols used between the different components.

3.2 Modelling the demo-site

We created scenario to simulate the demo-site in DEMKit. In the scenario, the following devices and controllers are instantiated. First, a meter device is instantiated, where all devices are connected to. This way, the total energy consumption or production of the entire demo-site can be monitored. A group controller that acts as the parent for all device controllers is instantiated in the scenario. Because this group controller can control all its device controllers, a DEM algorithm can control the DERs in such a way that e.g. a flat profile is obtained at transformer level.

A device of the class uncontrollable devices is added, which represents the loads of the office buildings. This device is called the static load and the consumption data for this device is retrieved by aggregating the values of all the office building meters. A controller is added to this device that retrieves this data from a database. A device of the class buffer-timeshiftable is added, to represent the AL434 EV charging station. Since this charging station is not connected to the local CPMS, it can not be controlled by the DEM system. However, to create a scenario that is as close to the real demo-site as possible, it is still added to the scenario. Moreover, its consumption is monitored to keep the values of the meter device accurate.

The following devices in the PV/EV distribution cabinet are added to the scenario. A device of the class uncontrollable devices is added that represents the four inverters of the PV panels. The production data is retrieved by aggregating the production of all four inverters. A controller is added to this device that retrieves this data from a database. Two devices of the buffer-timeshiftable class are added that represent the two connections on the first EN-PRO44 EV charging station. The behaviour of a buffer-timeshiftable model equals the behaviour of charging a single EV. Therefore, a device is added for every connector instead of every charging station. Controllers are added to these devices that communicate with the CPMS. This way, the charging profiles of the EVs can be set by the DEM algorithm.

3.2.1 Modelling the battery

As explained in Chapter 2, the buffer device in DEMKit is used to model the behaviour of a battery. However, the implementation of the battery in the demo-site is different from a regular implementation, making it complicated to use a buffer device to model the battery. The buffer device in DEMKit feeds energy back into the grid, but the battery in the demo-site can only discharge to the EVSE. Therefore, a different approach is used in the scenario. Instead of using a buffer device, a device of the buffer-converter class is added. A buffer-converter device has a certain energy demand is must provide at its output, but also has a buffer to decouple the input and output power in time. The size of the buffer is equal to the size of the battery, and the energy request on the output of the converter is set to the energy that the connected EV charging stations require. A controller is added to the buffer-converter device, where a DEM algorithm can control when the battery should charge and discharge.

In order to instantiate the buffer-converter device, the energy consumption of the connected EVs must be known. This is realized in the scenario by developing a new group controller. To this group controller the controller of the buffer-converter device and multiple controllers of buffer-timeshiftable devices are added. The group controller is used to first optimize the energy consumption profiles of the charging stations connected to the battery. This information is communicated to the buffer-converter device controller. This controller may then use an optimization algorithm to optimize its buffer, using the requested output power of the EVs. In the sequence diagram in Figure 3.4 it is explained in what steps the profile of the UPS is retrieved. Figure 3.5 shows an overview of all DEMKit components used in the scenario and how they correspond to the real hardware.

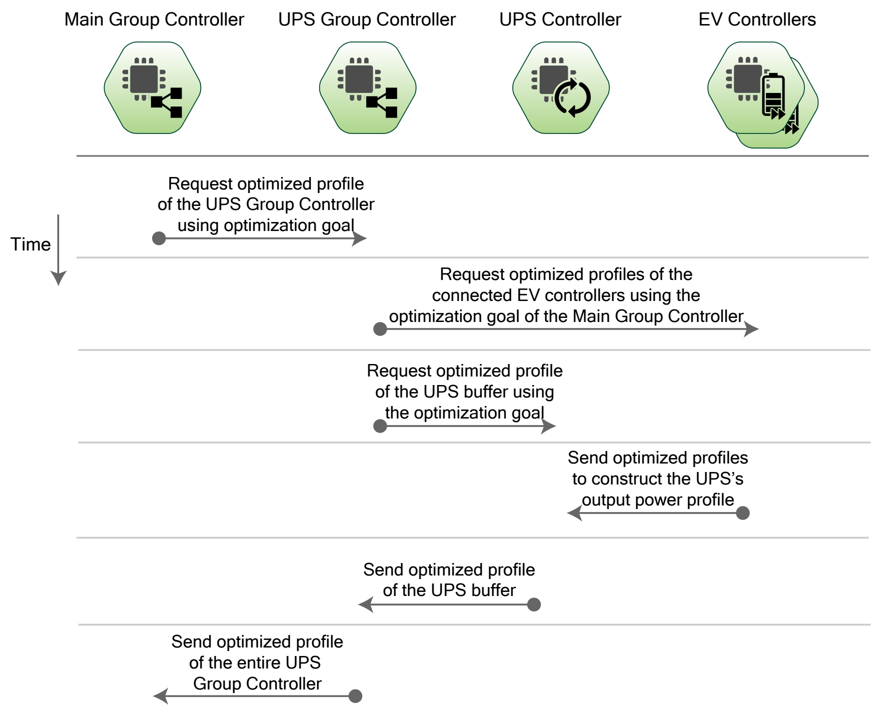


Figure 3.4: A sequence diagram on how the UPS battery profile is optimized

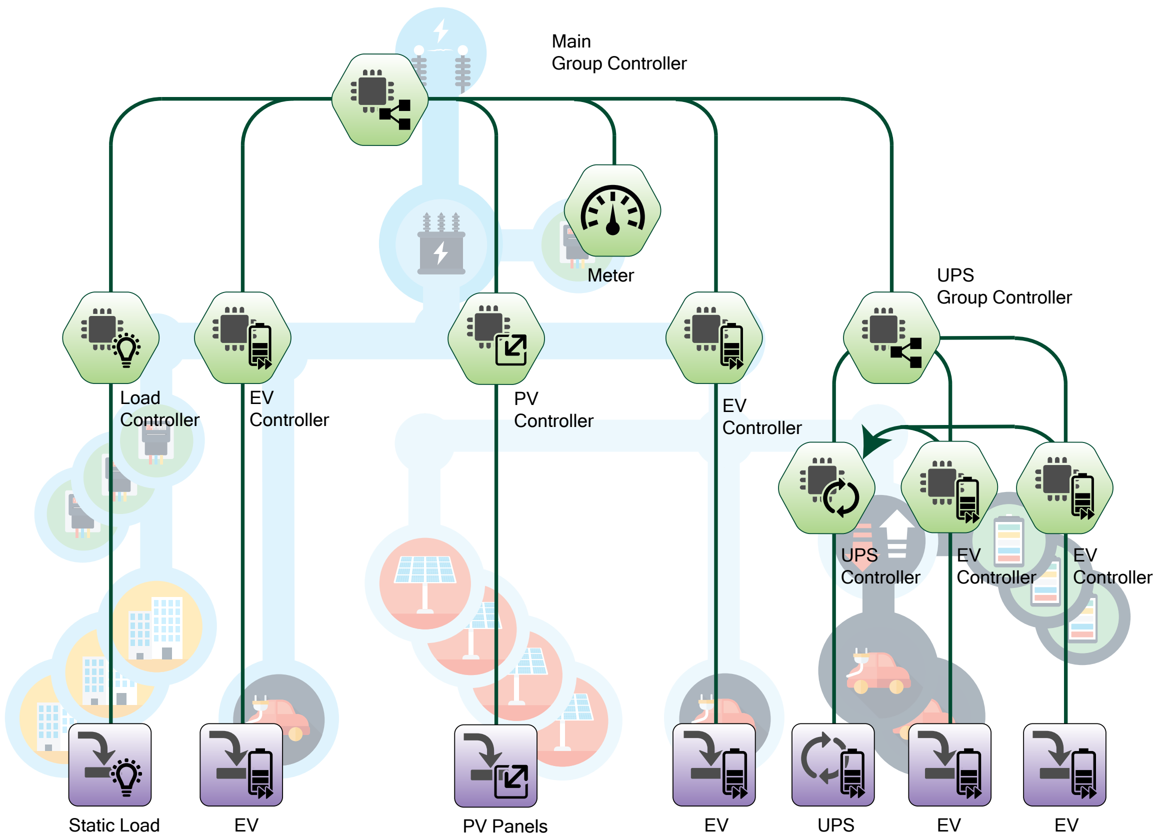


Figure 3.5: The scenario modelled in DEMKit ([16])

Chapter 4

Optimizing the charging of EVs

In this chapter we focus on the control logic that is added to control the flexibility of the DERs in a scenario. More precisely, we describe the valley-filling approach that is used to control the charging of EVs. First, we introduce the valley-filling approach in Section 4.1 and describe how it has been implemented in DEMKit in Section 4.2. Next, we explain how the fill- and charge levels are determined in Section 4.3 and we explain the charge-level auction in Section 4.4. The latter is an extension on the algorithm to divide the charge-level over the connected EVs using a double-sided auction. Lastly, the algorithms are tested and simulated in Section 4.5

4.1 The valley-filling approach

The decentralized energy management (DEM) approach that is used to control the charging of EVs in this project is called the valley-filling approach. This approach is explained in detail in Section 2.2.2 of the literature study (Appendix A). The approach was originally presented in [23], where the charging power of an EV is determined at the beginning of each interval using only a prediction of the *fill-level* and the power consumption of the rest of the system in that interval. From here on, we call the latter the *baseload*. The fill-level represents the sum of the baseload and the EV charging power for every interval. It is the aim to have this fill-level the same for all intervals and it is chosen in such a way that an EV is able to charge its battery within a specified time range, while keeping the total energy profile flat. Figure 4.1 shows the charging of an EV using the valley-filling algorithm, where the fill-level is Z. The baseload is shown in white and the charge power of the EV in grey.

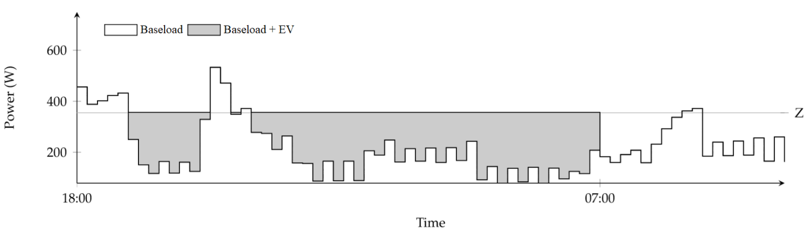


Figure 4.1: Optimal EV planning for peak shaving using valley-filling (Image from [23])

In the approach, an EV charges in every interval where the fill-level exceeds the baseload at a corresponding *charge level*. This charge level is the difference between the fill-level and the baseload and therefore in general different at each interval. In other words, the charging power of the EV fills the valleys in the baseload profile, hence the name valley-filling. With an 'optimal fill-level', an EV finishes charging exactly when the EV has to leave (this time is called the *deadline*). However, if on beforehand the profile of the baseload is not known for all time intervals until the deadline, it is in general not possible to determine the optimal fill-level.

As in general the baseload is not known exactly for the future intervals, the approach presents a method to predict the fill-level using historic data of baseloads of multiple days in the past. In order to predict the fill-level, the EV is fictionally charged on each of these historic days. Since the profiles of the baseloads of the historic days are known, an optimal fill-level can be determined for each day in the past. To fictionally charge the EV, also the deadline should be known, as well as how much the EV has to be charged. In [23], it is shown that in practice, the fill-levels from the historic days are often comparable to the fill-level of the current day, since the baseload is quite stable for consecutive days. Therefore, the fill-level that should be used to calculate the real charge levels for the EV can be approximated using the optimal fill-levels of the fictional charging jobs from the historic days. According to [23], it is best to use the maximum of all candidate fill-levels that follow from the fictional charging jobs, because this decreases the chance of high charging peaks due to prediction errors. In [27], where the prediction of the fill- and charge levels is elaborated, it is suggested that the candidate fill-levels should be determined for ten to fifty historical days, to incorporate both short-term and long-term behaviour.

In [27], the authors state that when the baseload contains PV production, the similarity between the baseloads of historic days and the baseload of the current day decreases. To increase the accuracy of the fill-level predictions, a separate prediction of the PV production of the current day should be used in the historic data of the baseloads. In order to realize this, we split up the baseload into two parts: the PV production and the power consumption of the other devices, called *remaining load* in this thesis. The predictions of the PV production of the current day are added to historic data of the remaining load to create a combined baseload. Subsequently the fill-level is determined by fictionally charging the EV with this combined baseload.

In the valley-filling approach, we make a distinction between the *offline* valley-filling algorithm and the *online* valley-filling algorithm for this thesis. Offline valley-filling describes how to predict the fill-level using the baseloads of historic days, and online valley filling uses this predicted fill-level and the baseload of the current time interval to determine the charge level at every time interval. Section 4.3 gives more details on how the fill- and charge levels are determined.

4.2 Valley-Filling in DEMKit

To realize the valley-filling approach in DEMKit, we developed four different controllers. First, we developed a controller for the buffer-timeshiftable class of devices, which raises an event when an EV wants to start a charging job. This synchronous event contains information about the deadline and the amount of energy to be charged. Also, this controller has information on the current SoC of the EV and it can send control signals for the charging level to the device it is connected to.

Secondly, we developed two different controllers for the uncontrollable device class. The reason for this is that both the PV panels and the remaining load are devices of the class uncontrollable devices. The controller of the PV panels gives a prediction of the PV production profile for the current day and the controller of the remaining load has information on historical data of the energy consumption.

Lastly, we developed a group controller that contains the algorithms to determine the fill- and charge levels. This group controller acts as the parent for the buffer-timeshiftable device controllers and the uncontrollable device controllers. When a buffer-timeshiftable device controller raises an event, the group controller predicts the fill-level using the offline valley-filling algorithm. This algorithm first obtains the predicted PV production and the historical data of the remaining load. For each day in the historical data, the group controller aggregates the predicted PV production and the remaining load, and then determines the candidate fill-levels using the historical baseloads, the deadline, and the required charge of the job.

Subsequently, the group controller determines the charge level at every time interval as long as the EV is connected, using the online valley-filling algorithm. This algorithm uses the highest fill-level of all candidate fill-levels to determine the charge level during the charging job. At every discrete time interval, the group controller executes this algorithm to determine the charge level. The uncontrollable device controllers send their current measured production or consumption to the group controller, which calculates the current baseload and determines the charge level for the next interval. In the case that only one EV is charging, this EV charges at the determined charge level. However, when multiple EVs are connected and charging, the total charge level must be divided over the EVs. In Section 4.4 we discuss how to divide the charge level. Every time a new EV arrives or when an EV finishes its charging, the fill-level is predicted again using the offline valley-filling approach. The pseudo code in Algorithm 4.1 shows an overview of all the steps that are executed in the group controller.

4.3 Determine the fill- and charge levels

The offline valley-filling algorithm approximates the optimal fill-level by fictionally applying the charging job on the baseload of days in the past. In this thesis, three alternative algorithms are introduced to approximate the optimal fill-level. In the following, these three algorithms are described.

Algorithm 4.1: Pseudo code of the steps that the group controller executes at every time interval

```
function time interval pre-tick (time);
Input: The time of the current discrete interval
FillLevel = 0
EVsConnected = 0
// Decrease EVConnected counter to stop the online algorithm when all EVs are finished
if EVDisconnectedEvent then
    EVsConnected- = 1
end
// Execute the offline algorithm when a new EV is connected
if EVConnectedEvent then
    EVsConnected+ = 1
    PVPredictions = getPVPredictions(time, DeadLine)
    RemainingLoads = getHistoricalLoad(time, DeadLine, HistoricalDays)
    BaseLoads = PVPredictions + RemainingLoads
    FillLevel = doOfflineValleyFilling(BaseLoads, DeadLine, RequiredCharge)
end
// Execute the online algorithm as long as at least one EV is connected
if EVsConnected >= 1 then
    CurrentBaseLoad = getCurrentBaseLoad(time)
    ChargeLevel = doOnlineValleyFilling(CurrentBaseLoad, FillLevel)
end
```

4.3.1 Bisection search algorithm

The first offline valley-filling algorithm is based on a bisection search algorithm. It determines the fill-level by iterating over possible fill-levels until a proper value is found. At first, the algorithm determines the possible minimum and maximum fill-levels and then sets the initial fill-level at halfway. The corresponding charge levels are determined for the historic baseloads for every time interval until the deadline. When the resulting total charged amount is too high, the algorithm sets the next fill-level at halfway between the minimum and the previous fill-level. When the resulting total charged amount is too low, the algorithm sets the next fill-level at halfway between the maximum and the previous fill-level. The algorithm continues to change the fill-level in this way until the energy that is charged during the job is (almost) equal to the required charge of the job, before the deadline is reached. However, this algorithm does not find the optimal fill-level, but converges to an approximation with a pre-specified error bound. When multiple EVs require energy, the required charge of all charging jobs is summed before starting the algorithm. Algorithm 4.2 shows the pseudo-code for this algorithm.

Algorithm 4.2: Using bisection search to approximate the optimal fill-level in an offline approach

function doOfflineValleyFilling $(BaseLoads, DeadLine, RequiredCharge)$;
Input : Array of historic baseload profiles,
The amount of energy to charge
Output: Approximated fill-level

$FillLevels = [\,]$
foreach $BaseLoad$ **in** $baseLoads$ **do**
    $MinFillLevel = findMinFillLevel(BaseLoad, RequiredCharge)$
    $MaxFillLevel = findMaxFillLevel(BaseLoad, RequiredCharge)$
    $FillLevel = (MinFillLevel + MaxFillLevel)/2$
    $FillLevelRange = abs(FillLevel - MinFillLevel)$
    $Charged = 0$
    while $abs(Charged - RequiredCharge) > ErrorBound$ **do**
        $Charged = 0$
        foreach $PowerConsumption$ **in** $BaseLoad$ **do**
            $ChargeLevel =$
             $min(MaxPower, max(0, FillLevel - PowerConsumption))$
            $Charged \mathrel{+}= ChargeLevel$
        end
        if $Charged > RequiredCharge$ **then**
            $FillLevel = FillLevel - (FillLevelRange/2)$
        end
        if $Charged < RequiredCharge$ **then**
            $FillLevel = FillLevel + (FillLevelRange/2)$
        end
        $FillLevelRange = fillLevelRange/2$
    end
    $FillLevels.append(FillLevel)$
end
return $max(FillLevels)$

Since the absolute error between the desired fill-level and the fill-level used in the iteration is halved at each iteration, the rate of convergence of the bisection search algorithm is exponential in the range of the fill-levels. Furthermore, the time needed to approximate the optimal fill-level with the bisection search algorithm depends on the allowed error in the approximation. Note that the minimum and maximum fill-level bounds and the number of time intervals in the charging job also influence the time needed to approximate the optimal fill-level.

4.3.2 Breakpoint search algorithm

To overcome the mentioned issues of the bisection search algorithm, we use an algorithm to determine the optimal fill-level based on a breakpoint search algorithm. This algorithm is presented and explained in [28]. A variant that uses discrete charging power steps also exists. The time complexity of this algorithm is $\mathcal{O}(n \log n)$, where n is the number of time intervals. This means that the time this algorithm takes to find the optimal fill-level only depends on the number of time intervals, as opposed to the bisection search algorithm, which depends on both the fill-level range, the pre-specified error bound, and the number of time intervals.

The breakpoint search algorithm of [28] returns an optimal charging profile for an EV charging job using a baseload as input. Internally, the algorithm uses a fill-level to determine this charging profile. This algorithm is already implemented in DEMKit. Therefore, we decided to adapt the algorithm to also return the fill-level it uses internally next to the charging profile. This enables us to use the algorithm as an offline valley-filling algorithm. Since the algorithm determines the optimal fill-level for one EV charging job, it is executed for each EV that is currently connected to the system. Every time the breakpoint search algorithm is executed, the charging profile that belongs to the obtained fill-level is added to the baseload before the algorithm is used to find the optimal fill-level of the next EV. This is done because the algorithm should take the charging jobs of the previous EVs into account. The fill-level that is returned for the last job is used as the fill-level for that baseload profile. This process is repeated for all historic baseload profiles and, just as with the bisection search algorithm, the maximum of all fill-levels is used for the online valley-filling algorithm. Algorithm 4.3 shows how this breakpoint search algorithm is implemented.

Algorithm 4.3: Use the breakpoint search algorithm to find the optimal fill-level in an offline approach

function doOfflineValleyFilling ($BaseLoads, ActiveJobs$);
Input : Array of baseload profiles for a number of historic days,
Active charging jobs
Output: Optimal fill-level
$FillLevels = [\,]$
foreach $BaseLoad$ **in** $BaseLoads$ **do**
  foreach Job **in** $ActiveJobs$ **do**
    $Profile, FillLevel = breakpointSearchAlgorithm(BaseLoad, Job)$
    $BaseLoad+ = Profile$
  end
  $FillLevels.append(FillLevel)$
end
return $max(FillLevels)$

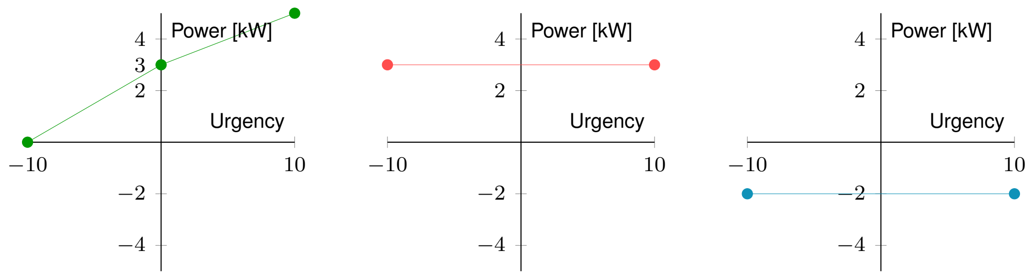
4.3.3 Minimum charging threshold algorithm

In a realistic EV charging scenario, not only an upper bound exists for the charging power, but a minimum bound as well. EVs mostly require at least 6 A, or 1380 W, from the charging station before they can start charging. From this point, the charging power can increase almost continuously until the maximum bound for the charging power. Therefore, an algorithm that still uses a continuous charging power, but with a minimum charging threshold is required. Such an algorithm is presented in [29]. Here, the breakpoint search algorithm is extended to determine the optimal charging profile of an EV, when taking a minimum charging threshold into account. The resulting charging profile contains charging powers that are either zero, or between the minimum and maximum charging bounds. For more details on this algorithm we refer to [29]. Just as with the breakpoint search algorithm, this algorithm is already available in DEMKit. We also adapted this algorithm to return the fill-level used internally next to the resulting charging profile. This enables us to implement the algorithm as an offline valley-filling algorithm in the exact same way as presented for the breakpoint search algorithm in Algorithm 4.3. The only difference is that instead of calling the breakpoint search algorithm in line 5, the algorithm that respects the minimum charging threshold is called.

4.4 The charge-level auction

Based on the offline valley-filling algorithms that approximate the fill-level using optimal fill-levels determined with historical data, we now present an online valley-filling algorithm which is executed at each time interval, to determine the used charge level. This algorithm determines the charge level by subtracting the (predicted) baseload at that time interval from the fill-level. When only one EV charges, this EV can charge using this determined charge level. However, when multiple EVs charge, the charge level should be divided over the EVs. Note that, when all EVs receive the same share of the total charge level, it may be that some EVs can not fully charge their battery in time, while other EVs are allowed to charge more than they have to. Therefore, an online valley-filling algorithm that determines the total charge level and divides this over the available EVs in a fair way is developed. This algorithm is developed using a double-sided auction, as described in [17]. In an auction, a centralised auctioneer divides the available power over all connected devices. At every time interval, all devices submit a so-called bid function to the auctioneer, in this case the group controller. The bid function describes the flexibility of a DER. In our case it contains one or more points that specify the amount of power a device is going to consume at this time interval, for some urgency levels. The *urgency level* is a number whereby a device is willing to draw more power at a higher urgency level. In other words, when the urgency of charging is higher, the device will draw more power. A line is drawn between the points so that the bid function describes a power level for every possible urgency level.

Figure 4.2a shows a bid function for an EV. In this bid function, the EV charges at its maximum charging rate when the urgency level is maximal. For a minimal urgency level, the EV charges nothing. Halfway the urgency level, the optimal charge level of the EV is specified. This optimal charge level implies that the EV is fully charged exactly at its deadline if the EV charges at this level for every time interval until the deadline. These three points are connected to create a complete bid function. Figure 4.2b and Figure 4.2c show the bid functions for the remaining load and PV production respectively. Although there is no flexibility in these devices, they also submit a bid function to the auctioneer, because in practice, this creates the baseload. Since these devices submit their consumption and production to the auctioneer, it does not have to predict the baseload for the coming time interval. Instead it uses these bid functions. Both these devices have no flexibility in their production or consumption and therefore they consume or produce the same for all urgency levels.



(a) EV *(b)* Remaining load *(c)* PV

Figure 4.2: Examples of bid functions of the different devices

The goal of the auctioneer is to clear the market in such a way that the resulting power consumption is equal to the fill-level. It does this every auction interval by aggregating all submitted bid functions and then selecting an urgency level where the amount of consumed power is equal to the fill-level that has been determined by one of the offline algorithms. An example is shown in Figure 4.3, where the bid functions of the EV, the remaining load, and the PV panels of Figure 4.2 are aggregated. If we take a fill-level of *5 kW* in the example, the auctioneer determines the urgency level for this power to be *5*, as seen in Figure 4.3. The auctioneer then sends this urgency level to all device controllers, which use their own bid function to determine their consumption. In the example, the EV starts charging at 4 kW, since the charge power of the EV bid function in Figure 4.2a that corresponds to an urgency level of *5* equals 4 kW. When there is no urgency level corresponding to the fill-level, the auctioneer clears the market at an urgency level that is closest to the corresponding fill-level, to ensure that the market is always cleared so that the connected devices can determine their consumption.

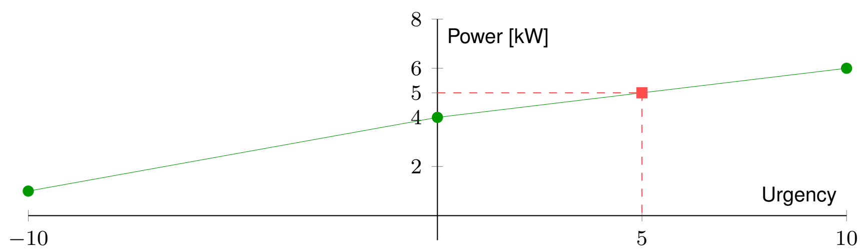


Figure 4.3: Market clearing by the auctioneer

This auction-based algorithm has as advantage that, instead of allocating equal shares of the total available charging level to all EVs, an EV that has a higher priority automatically receives a larger share of the available power. This is because the bid function of an EV that still has to charge a lot is more greedy than the bid function of an EV that has a lot of time to charge.

4.4.1 Minimum charging threshold auction

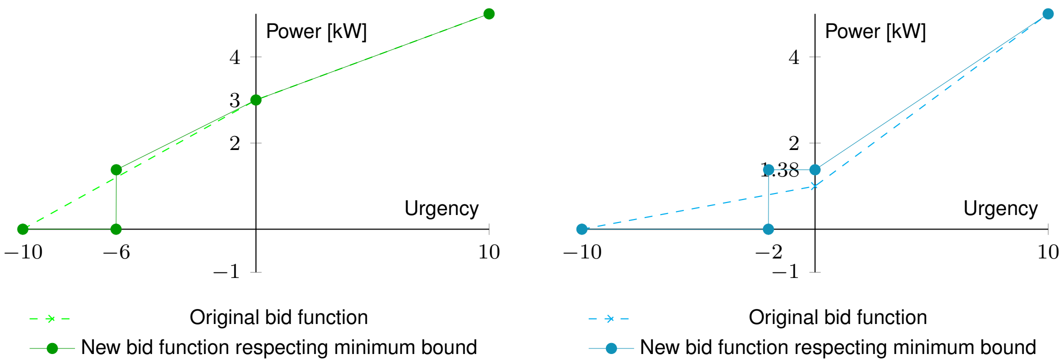
The explained auction-based online valley-filling algorithm works well with the offline breakpoint search algorithm. This is because the breakpoint search algorithm determines the fill-level using all possible charge levels between zero and the EV's maximum charge level, and the presented bid function of the EV also has urgency levels for all charging levels between zero and the maximum charge level. However, we stated that in a realistic EV charging scenario, not only an upper bound exists for the charging power, but a minimum bound as well. The offline valley-filling algorithm that respects this minimum charging bound to determine the fill-level is already presented. But when we combine this offline algorithm with the presented auction, a charge level that is lower than this minimum bound may still be chosen in the market clearing. Therefore, we developed an EV bid function to work with the offline minimum charging threshold algorithm. This bid function ensures that an EV charges either nothing or between its minimum and maximum charging power.

An EV bid function that respects the minimum charging threshold should have urgency levels that correspond to charging powers between the maximum and minimum bounds, and to a charging power of zero. There should be no urgency level that corresponds to a charging power between zero and the minimum bound. When transforming the original EV bid function to one that respects the minimum charging threshold, two situations may occur. In the first situation, the determined optimal charge level is higher than the minimum charging threshold. Just as with the original bid function, this optimal charge level implies that the EV is fully charged exactly at its deadline if the EV charges at this level for every time interval until the deadline. In the second situation, the optimal charge level is lower than the minimum charging threshold of the EV.

Figure 4.4a shows the first situation, where the optimal charge level is higher than the minimum charging threshold. The dashed line represents the original bid function and the solid line is the bid function adapted to respect the minimum charging threshold. In this bid function, the maximum charging and optimal charge level are not changed. However, the in left-hand part of the bid function we see a 'cut-off' at the minimum charge level threshold. The urgency level of this so-called cut-off point is determined as in Equation 4.1.

$$\mathrm{Cutoff} = 0 - \left(\frac{\mathrm{OptimalChargeLevel}}{\mathrm{MaxChargeLevel}} \times \mathrm{MaxUrgencyLevel}\right) \tag{4.1}$$

Figure 4.4b represents the second situation that may occur. Here, the optimal charge level is lower than the minimum charging threshold. To adapt this bid function, the point in the bid function that represents the optimal charge level is raised to the minimal charging threshold. The bid function defines multiple urgency levels for which the charge level is zero, until the urgency level cut-off point is reached. At this cut-off point, the charge level goes to the minimum bound until the point where the optimal charge level used to be. This cut-off point is also determined using Equation 4.1.



(a) Threshold 1 *(b)* Threshold 2

Figure 4.4: Examples of EV bid functions using a minimum charging threshold

As example, an auctioneer aggregates the EV bid functions incorporating the minimum charging threshold in Figure 4.4, and the bid functions of the the remaining load and the PV production in Figure 4.2b and Figure 4.2c respectively. This aggregation would result in the bid function in Figure 4.5. In the example the fill-level is *2.92 kW*, so the auctioneer clears the market at an urgency level of *-4*. Subsequently, the EV of Figure 4.4a would start charging at *1.92 kW* and the EV of Figure 4.4b would not charge.

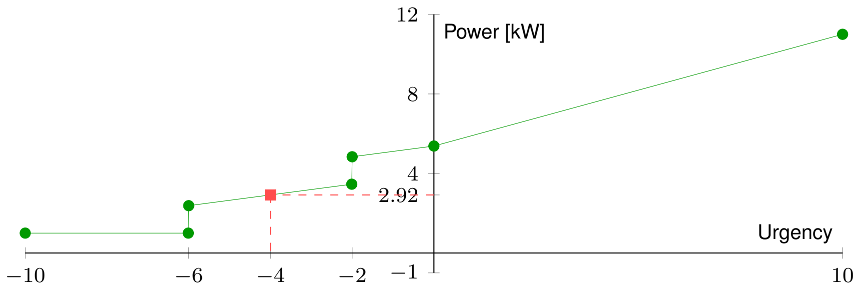


Figure 4.5: Market clearing by the auctioneer

4.5 Simulations

The valley-filling algorithms are tested in a simulated environment in DEMKit. The test scenario contains the remaining load, PV panels, and EV charging stations. All EV charging stations are connected directly to the main group controller. The input data for the remaining load and PV panels are generated using the Artificial Load Profile Generator (ALPG) [30]. The output data of the ALPG is saved to files, which are loaded into DEMKit when starting the simulation. The charging powers and capacity of the EVs are derived from the specifications of the EVs at the demo-site. For this project, two Opel Ampera-e EVs are available. These have a maximum charging power of 32A and a capacity of 60 kWh. In the simulations, the required charge of a charging job is either 40% or 90% of the total capacity of the EV. In total, four different combinations of settings are simulated, which are discussed in the following subsections.

4.5.1 Simulation setup

Depending on the simulation, one or two charging jobs are performed in a 24 hour window. In the test scenario, a meter device is added, which measures the total power consumption/production of the scenario. The PV panels and the remaining load, which together form the baseload, are aggregated by the meter, as well as the charging powers of the EVs. The power profile as measured by the meter device is denoted as *Measured* in the simulations.

Two charging jobs are defined for the simulations. The first job specifies that the EV arrives at 04:00 and must be finished by 15:00. In this time, 40% of the 60 kWh battery, or 24 kWh has to be charged. The EV of the second job arrives at 10:30 and must be finished by 19:30. The required charge for this job is 90% of the 60 kWh battery, or 54 kWh. Table 4.1 gives an overview of the charging jobs used in the simulations.

	Start Time	Deadline	Required Charge
Job 1	04:00	15:00	24 kWh
Job 2	10:30	21:30	48 kWh

Table 4.1: Charging jobs for the simulations

Four different combinations of settings are simulated. For every simulation, the same baseload profile is used. In the simulations, different combinations of offline and online valley-filling algorithms are simulated to see the effect of using the different algorithms. Table 4.2 shows the combination of which charging jobs are simulated and what algorithms are used for the simulations.

	Jobs	Offline algorithm	Online algorithm
1	Only Job 1	Bisection search	None
2	Job 1 and Job 2	Breakpoint search	Equal Shares
3	Job 1 and Job 2	Breakpoint search	Charge level auction
4	Job 1 and Job 2	Breakpoint search with minimum charging threshold	Charge level auction with minimum charging threshold

Table 4.2: The configurations of all simulations done

4.5.2 Reference simulations

To asses the effect of the algorithms in the simulations, we compare the simulation results to a reference simulation. In this reference simulation, no DEM control is applied and therefore the EVs start charging at their maximum charging power as soon the charging job starts. We call this behaviour the no-control behaviour of a charging job. In Figure 4.6 we see the resulting profile of the no-control behaviour for *Job 1*, as measured by the meter device in DEMKit.

The measured power consumption/production is equal to the baseload for most intervals, except for when the EV is charging. A positive measured value means that energy is subtracted from the grid. When the value is negative, energy is fed to the grid. In the result, we see that the EV starts charging at its maximum charging power of 32 A, or 7.4 kW, when the job commences. It continues to charge at its maximum charging power until its battery is fully charged. The results of *Simulation 1* are compared to this reference simulation, since only *Job 1* is used in this simulation.

When both *Job 1* and *Job 2* are executed without using DEM control, the measured power profile results in the profile seen in Figure 4.7. We see that next to the increased energy consumption of the first job, the second job now also causes a rise in the power profile starting at 10:30. Just as with *Job 1*, this EV starts charging immediately at 7.4 kW, until its battery is fully charged. The results of *Simulation 2*, *Simulation 3*, and *Simulation 4* are compared to this reference simulation, since both *Job 1* and *Job 2* are used in these simulations.

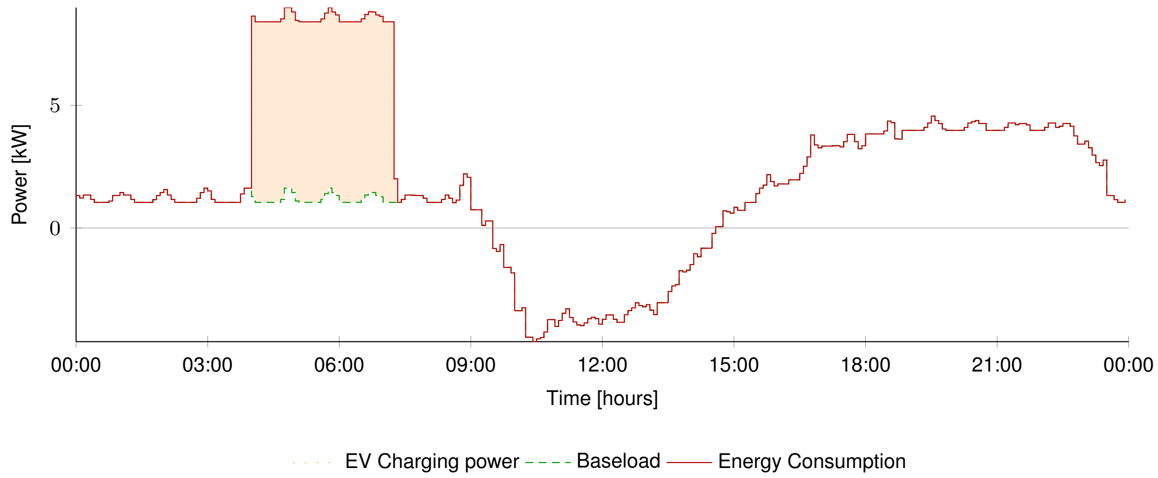


Figure 4.6: No-control behaviour of charging job 1

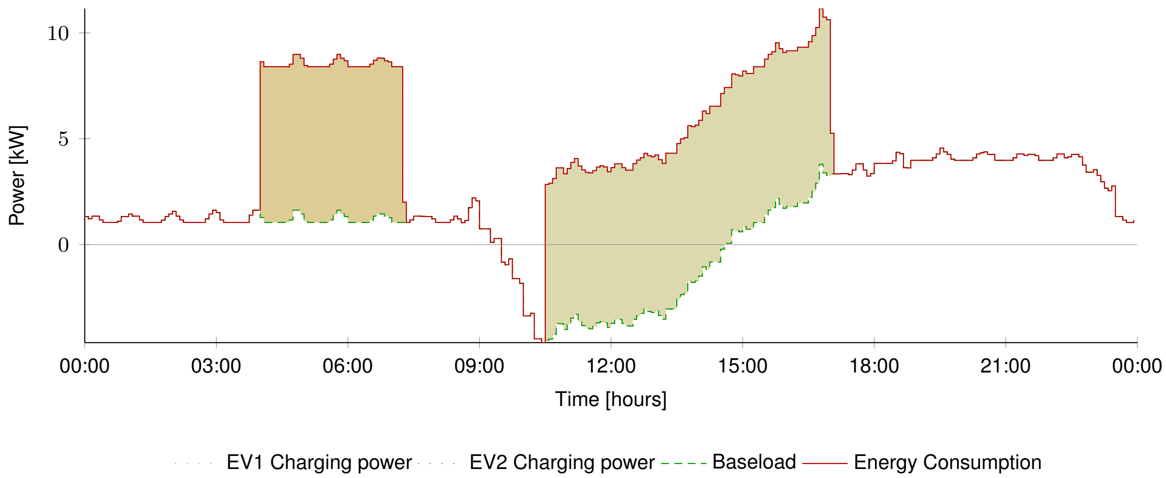


Figure 4.7: No-control behaviour of charging jobs 1 and 2

4.5.3 Simulation 1

In the first simulation, the bisection search algorithm is used to approximate the fill-level for *Job 1*. The algorithm uses the baseload profiles of ten historic days to determine the fill-level. These baseloads use the remaining load of the historic days and a perfect prediction of the PV production of the simulated day. Since only one job is executed, the charge level does not have to be divided over multiple running jobs. Therefore, no online algorithm is used to divide the charge level, but the group controller determines the charge level based on the fill-level and the current baseload every five minutes. The EV of *Job 1* then uses this level as its charging level. Summarizing, these settings are used for *Simulation 1*:

Charging jobs: *Only Job 1*
Offline algorithm: *Bisection search algorithm*
Online algorithm: *None*

Figure 4.8 shows the resulting measured power consumption for *Simulation 1*, as measured by the meter device. Furthermore, the fill-level, baseload, and the charging power of the EV are shown. As can be seen, a fill-level is determined by the bisection search algorithm when the job commences. Every five minutes, the group controller determines the charge level and the EV charges at this level. Because the baseload fluctuates a little in these five minutes, we see that the total power consumption is not exactly equal to the fill-level, but also fluctuates a bit. When the deadline of the job is reached, the EV stops charging and the fill-level is reset to zero.

A more detailed view of the charging power of the EV is given in Figure 4.9. Here, the line *Available* shows when the EV was available, and *State of Charge* shows the SoC of the battery in the EV. We see here that the SoC does not reach 100% before the deadline. Instead, the EV is charged up to 98%. A probable explanation for this is that the determined fill-level was too low. This can be because the remaining load of the historic days was less than for the simulation day. Moreover, the bisection search algorithm approximates the fill-level instead of determining the optimal fill-level.

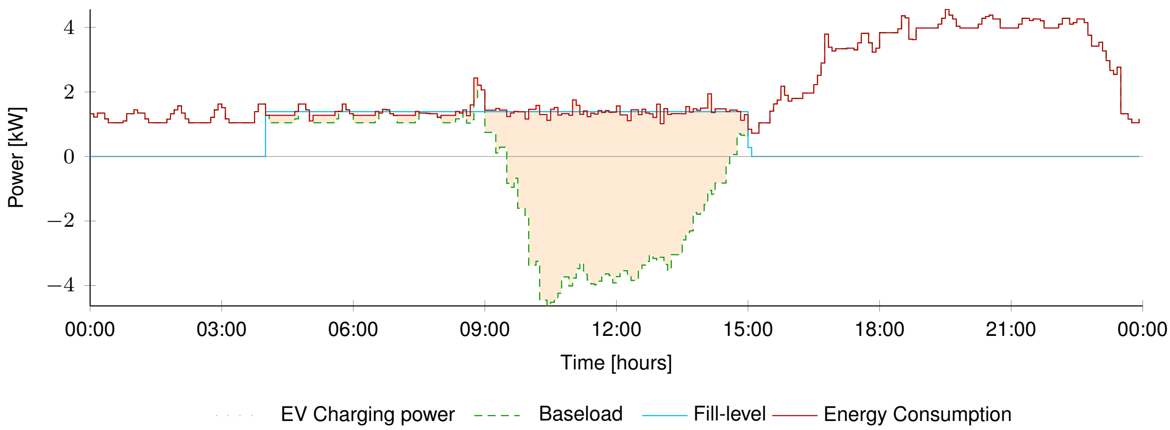


Figure 4.8: Results of Simulation 1

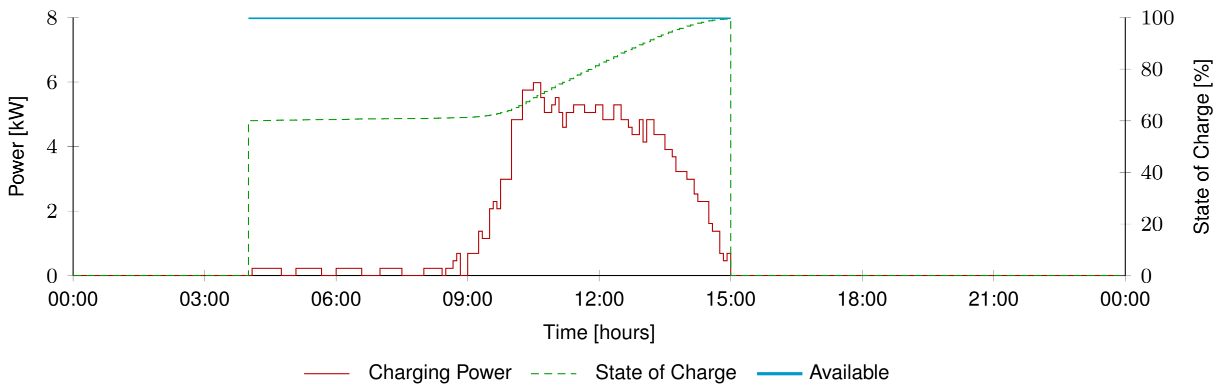


Figure 4.9: Charging power of the EV from Simulation 1

To quantify the effects of DEM control in this simulation, the results are compared to the results of the no-control behaviour in the reference simulation. We compare the total amount of energy that is imported from the grid and the total amount of energy that is fed to the grid, or exported. Moreover, the highest import and export peaks are compared as well. In *Simulation 1*, 49.65 kWh of energy is consumed from the grid and nothing is fed to the grid. From the no-control behaviour of the reference simulation, a total energy consumption of 64.97 kWh and a total energy production of 15.02 kWh is retrieved. Therefore, using the bisection search algorithm to charge an EV results in consuming about 24% less energy from the grid and using all locally produced energy. However, since the EV did not fully charge in the simulation, it is not entirely fair to compare these values in this way.

When we look at the peaks in the grid, we see that the highest import peak is reduced by 49%, from 8.99 kW to 4.56 kW. The highest export peak, or negative peak, is reduced from 4.63 kW to none, meaning that all produced energy is used locally. Table 4.3 gives an overview of the results retrieved from *Simulation 1* when compared to the reference simulation without control.

	Control behaviour	No-control	Improvement	Percentage [%]
Imported energy [kWh]	49.65	64.97	15.31	23.57
Exported energy [kWh]	0.0	15.02	15.02	100.00
Import peak [kW]	4.56	8.99	4.43	49.25
Export peak [kW]	0.00	4.63	4.63	100.00

Table 4.3: Quantification of the grid improvements for Simulation 1

4.5.4 Simulation 2

For the second simulation, the breakpoint search algorithm is used as the offline valley-filling algorithm, to determine the fill-level. The breakpoint search algorithm also uses the same baseload profiles of ten days as the bisection search algorithm in *Simulation 1*. Both jobs are executed in this simulation. Every five minutes, the group controller determines the charge level based on the fill-level and the current baseload. Since two EVs are connected in this simulation, the charge level must be divided over the active jobs. This is done by giving all EVs an equal share of the total charge level. We give all EVs equal shares of the charge level in this simulation, to motivate why an online valley-filling algorithm such as the charge level auction should be used instead.

Charging jobs: *Job 1 and Job 2*
Offline algorithm: *Breakpoint search algorithm*
Online algorithm: *Equal Shares*

Figure 4.10 shows the measured power consumption for *Simulation 2*. The charging powers of the EVs are stacked to provide insight in how they jointly contribute to the measured profile. Since the second charging job has a higher required charge, the fill-level is determined to be much higher as soon as this job commences. Since the charge level is equally divided, EV1 is charged at a higher rate than the fill-level that was determined when only EV1 was connected suggests. As a result, EV1 is fully charged significantly earlier than its deadline. When both EVs are charging, the part of the charge level that EV1 receives too much, is absent in the charge level of EV2, causing that once EV1 is charged, EV2 starts charging at its maximum power to make up for this. Figure 4.11 and Figure 4.12 give a more detailed view of the charging powers of EV1 and EV2 respectively. Here it can be seen that the charging powers of both EVs are equal when they are both charging.

We compare *Simulation 2* to the no-control behaviour of the reference simulation where also both charging jobs are executed. Since the no-control behaviour already utilizes a large part of the locally produced energy, we see that using the algorithms of *Simulation 2* only increases the use of locally produced energy by 2.58 kWh. Since the total amount of energy used during the simulation is equal, we see that the same 2.58 kWh of energy is consumed less from the grid. Due to the high fill-level of the job of EV2, we see that the highest consumption peak in the grid is decreased only by 3.25 kW when using these algorithms. Table 4.4 gives an overview of the results retrieved from *Simulation 2*.

	Control behaviour	No-control	Improvement	Percentage [%]
Imported energy [kWh]	97.94	100.53	2.58	2.57
Exported energy [kWh]	0.0	2.58	2.58	100.00
Import peak [kW]	7.90	11.15	3.25	29.14
Export peak [kW]	0.00	4.63	4.63	100.00

Table 4.4: Quantification of the grid improvements for Simulation 2

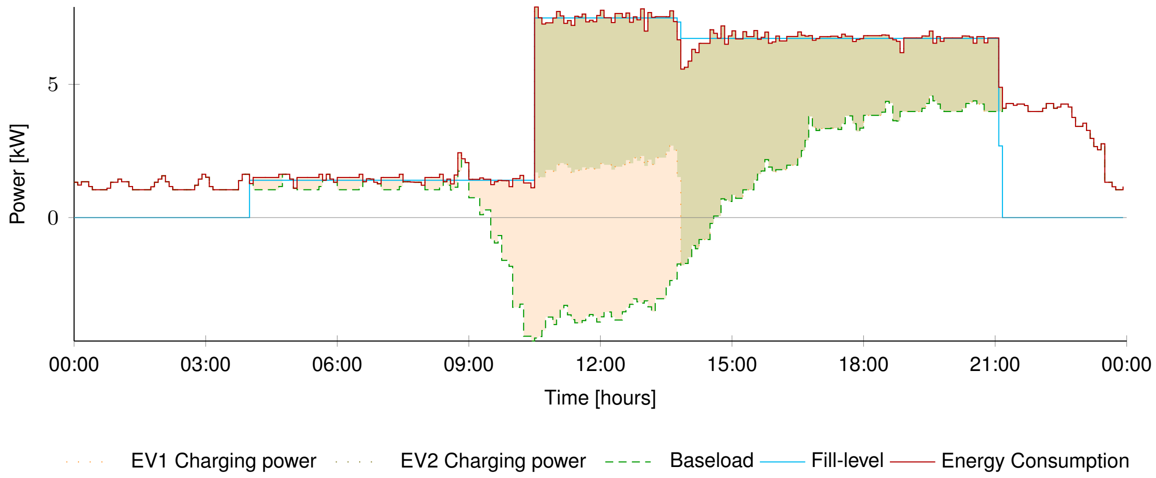


Figure 4.10: Results of Simulation 2

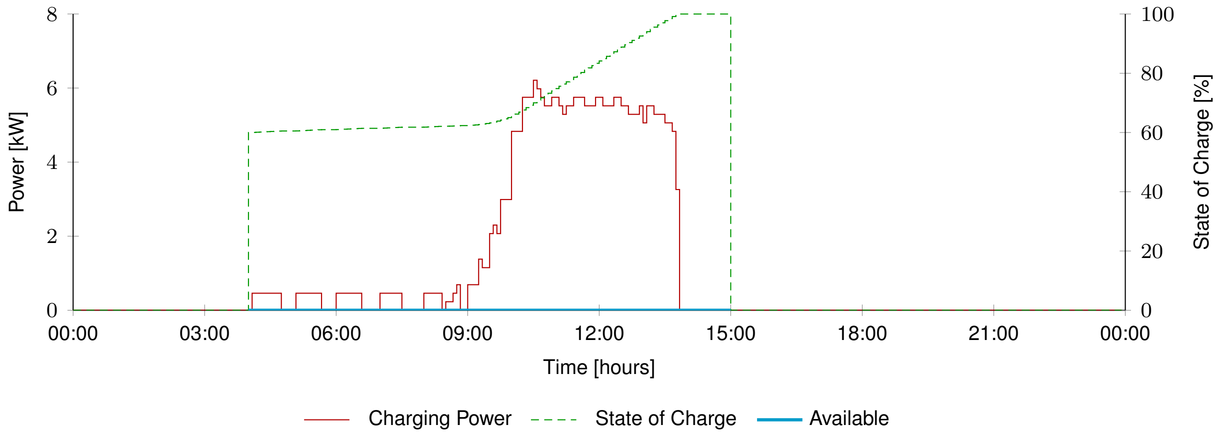


Figure 4.11: Charging power of EV1 from Simulation 2

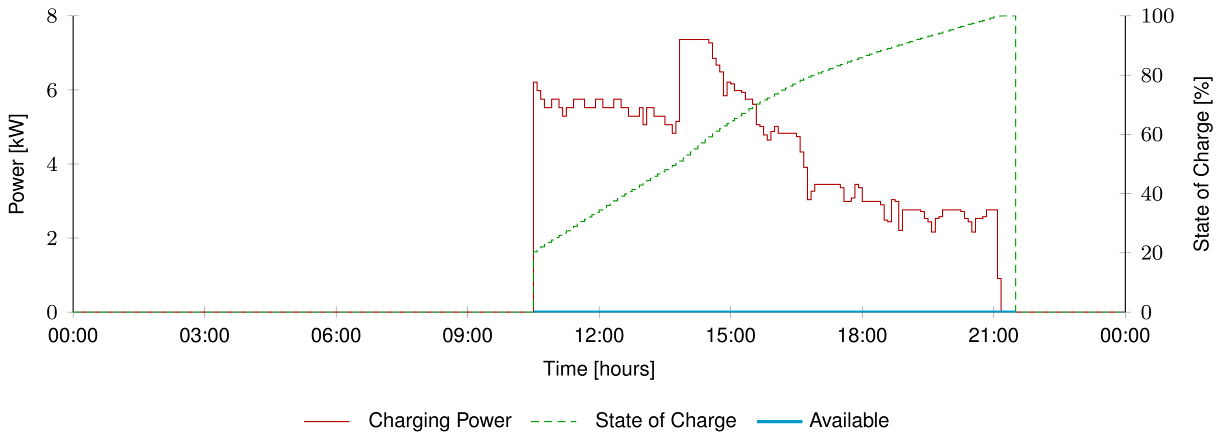


Figure 4.12: Charging power of EV2 from Simulation 2

4.5.5 Simulation 3

Simulation 3 is almost identical to *Simulation 2*. Again, the breakpoint search algorithm is used as the offline valley-filling algorithm and both charging jobs are executed. The only difference is that *Simulation 3* uses the charge level auction to divide the total charge level over the charging EVs. Every five minutes, all connected devices submit a new bid function to the group controller, which in turn clears the market at a certain urgency level. The EVs determine their charge levels based on this urgency level and their own bid functions. Summarizing, the following settings are used for *Simulation 3*:

Charging jobs: *Job 1 and Job 2*
Offline algorithm: *Breakpoint search algorithm*
Online algorithm: *Charge level auction*

Figure 4.13 shows the measured power consumption for *Simulation 3*. Again, the fill-level, baseload, and the charging powers of the EVs are shown. Once more the charging powers are stacked to emphasize how they contribute to the measured profile. The result of this simulation is very similar to that of *Simulation 2*. Here EV1 first fills the power consumption to the fill-level as expected. When EV2 arrives, the breakpoint search algorithm determines a new fill-level around 10 kW, similar to *Simulation 2*. However, now we see that due to the high charge required of the job of EV2, it receives a larger share of the total charge level than EV1. For this reason, EV2 charges more in the time that both EVs are connected in this simulation compared to *Simulation 2*. This results in a lower predicted fill-level once EV1 finishes its charging, since the remaining required charge of the job of EV2 is lower than in *Simulation 2*. Figure 4.14 and Figure 4.15 give a more detailed view of the charging powers of EV1 and EV2 respectively, where it can be seen that the charge level is not divided equally, but EV2 charges at a higher rate than EV1.

We compare the results of this simulation to the no-control behaviour of the reference simulation where both jobs are executed. Here we see that the amount of energy consumed from the grid is reduced by 2.58 kWh, which implies that 2.58 kWh of the produced energy is used more locally. Since all produced energy is utilized locally, the production peak is decreased from 4.63 kWh to none. The consumption peak is reduced by 3.39 kW, or 30%. This is where the charge level auction proves itself, since the consumption peak of this simulation, with auction, is reduced more compared to *Simulation 2*, where no charge level auction is used.

	Control behaviour	No-control	Improvement	Percentage [%]
Imported energy [kWh]	97.94	100.53	2.58	2.57
Exported energy [kWh]	0.0	2.58	2.58	100.00
Import peak [kW]	7.76	11.15	3.39	30.39
Export peak [kW]	0.00	4.63	4.63	100.00

Table 4.5: Quantification of the grid improvements for Simulation 3

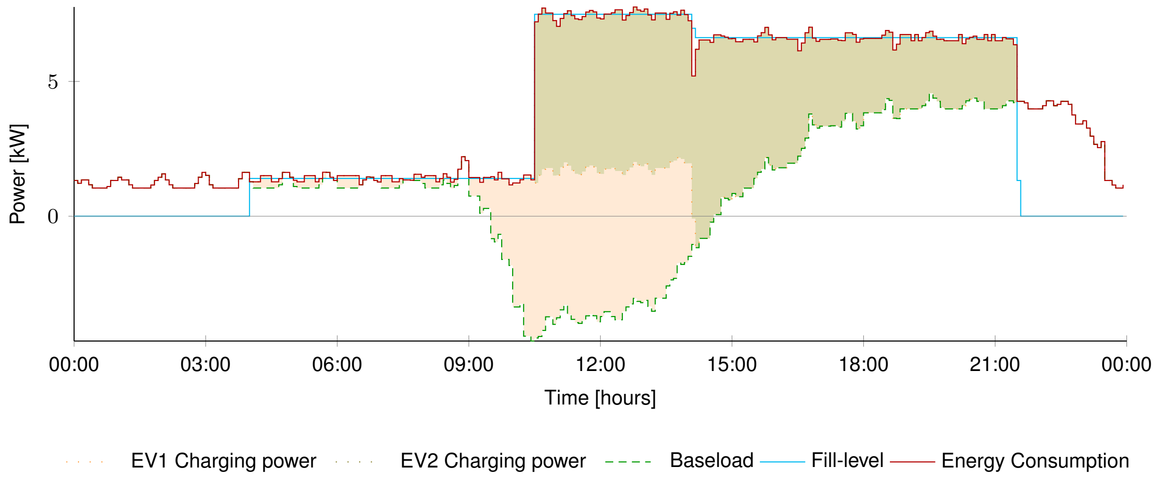


Figure 4.13: Results of Simulation 3

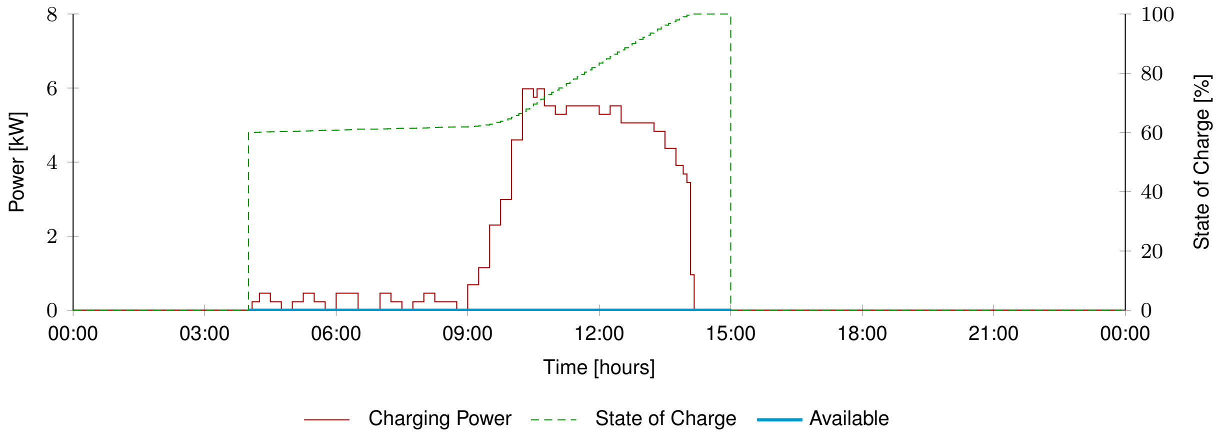


Figure 4.14: Charging power of EV1 from Simulation 3

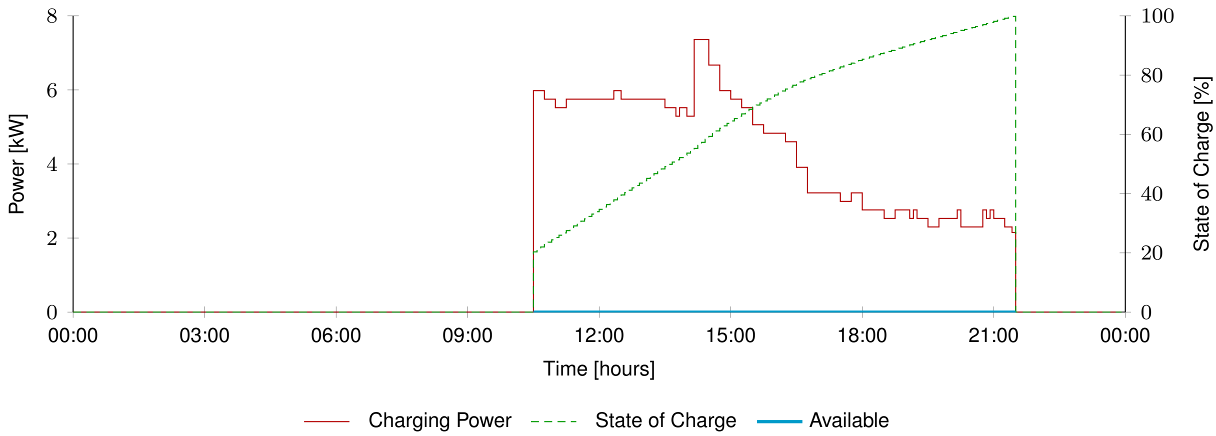


Figure 4.15: Charging power of EV2 from Simulation 3

4.5.6 Simulation 4

In the last simulation, the breakpoint search algorithm that respects the minimum charging threshold of the EVs is used for the offline valley-filling algorithm. This algorithm is accompanied by the online charge level auction that incorporates the minimum charging threshold in its bid functions as well. Both charging jobs are executed in this simulation. When a job commences, a fill-level is predicted and the online algorithm clears the market at a certain urgency level every five minutes. The EVs determine their charge levels based on this urgency level and their own bid function, which incorporates the minimum charging threshold. Summarizing, the following settings are used for *Simulation 4*:

Charging jobs: *Job 1 and Job 2*
Offline algorithm: *Valley-filling algorithm respecting minimum charging threshold*
Online algorithm: *Charge level auction respecting minimum charging threshold*

The resulting total power consumption obtained by this simulation is shown in Figure 4.16. Furthermore, the fill-level, baseload, and the charging powers of the EVs are shown. When the job of EV1 commences, the fill-level is determined to be higher compared to the fill-level of *Simulation 3*. The reason for it is that this offline algorithm incorporates the minimum charging threshold when it determines a fill-level. Figure 4.17 and Figure 4.18 show the charging powers in more detail. We see here that between the start of the job of EV1 and around 09:00, when the PV panels start producing energy, the charge level of EV1 behaves different compared to *Simulation 3*. Where in *Simulation 3* all possible charge levels between zero and the maximum charging power level are allowed, this simulation only allows a charge level of zero, or between the minimum and maximum charging powers. This results in the charge level being equal to the minimum charging power for some intervals, and zero in the other.

When comparing this simulation to the no-control behaviour of the reference situation, nearly identical results are obtained as for *Simulation 3*. This is expected, since both simulations are very similar. The only difference is that the improvement in the consumption peak is a bit lower for this simulation. A possible explanation is that the fill-level is determined to be higher by the offline valley-filling algorithm used here. The higher fill-level causes the EVs to charge at a higher rate, which results in the improvement being less for this simulation. However, the difference is only 30 W.

	Measured	No-control behaviour	Improvement	Percentage [%]
Imported energy [kWh]	97.94	100.53	2.58	2.57
Exported energy [kWh]	0.0	2.58	2.58	100.00
Import peak [kW]	7.79	11.15	3.36	30.17
Export peak [kW]	0.00	4.63	4.63	100.00

Table 4.6: Quantification of the grid improvements for Simulation 4

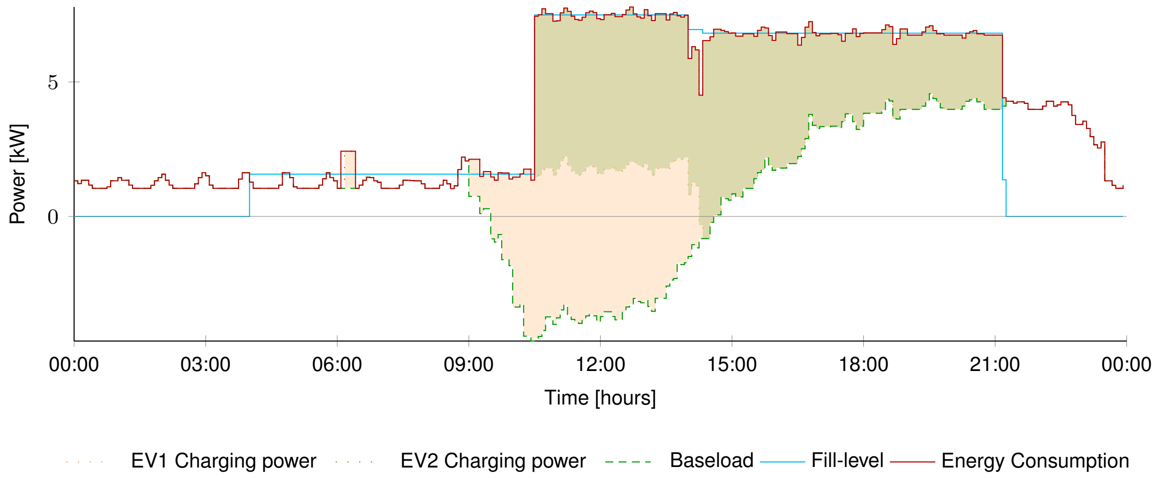


Figure 4.16: Results of Simulation 4

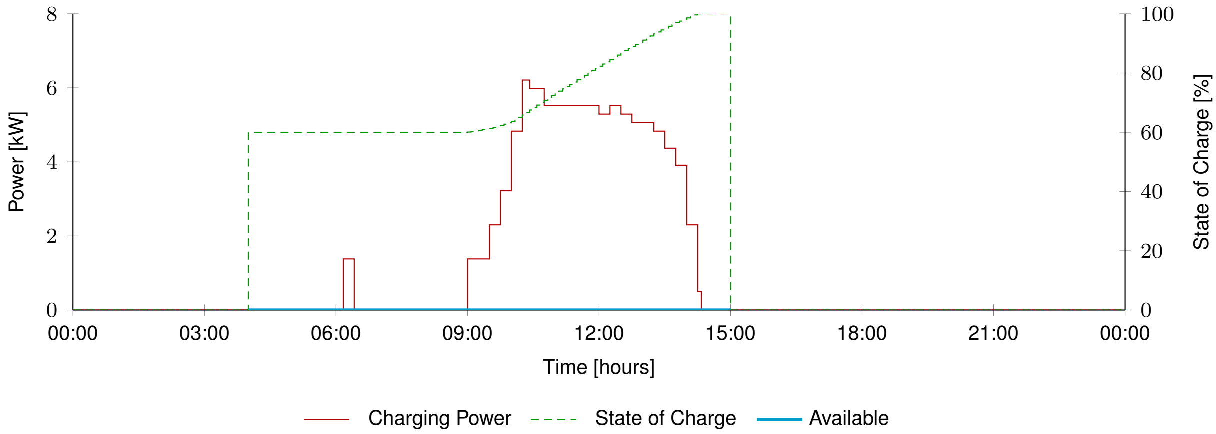


Figure 4.17: Charging power of EV1 from Simulation 4

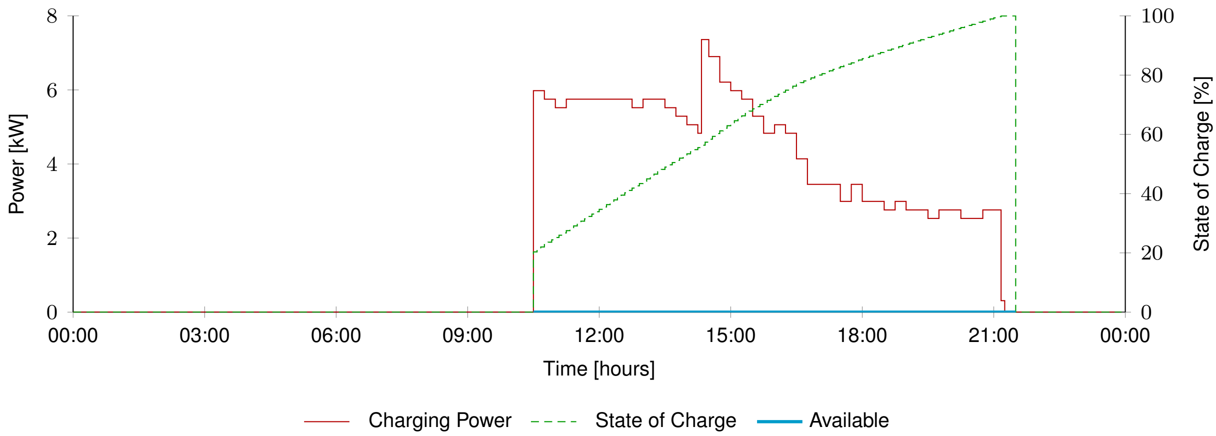


Figure 4.18: Charging power of EV2 from Simulation 4

An observation we do in all simulations where multiple EVs must be charged at the same time (*Simulation 2*, *Simulation 3* and *Simulation 4*) is that both EVs finish quite some time before their deadlines. The reason for this is that the fill-level is determined to be too high by the way we implemented the breakpoint search algorithm in the offline valley-filling algorithm. In this implementation, the breakpoint search algorithm first determines the fill-level for the first EV. Next to the optimal fill-level, the breakpoint search algorithm also returns the charging profile that corresponds to this fill-level. Note, that this charge profile cannot be used as the real charging profile for the EV since it is a charge profile that is optimal for one of the historic baseload profiles.

In the next step, the offline valley-filling algorithms applies this charge profile to all baseload profiles, since the breakpoint search algorithm should take into account the charging of the first EV when it determines the fill-level for the second EV. However, the fill-level that is then determined for the second EV, which is in this case the eventual fill-level, is too high. This causes both EVs to charge at a power level that is higher than the optimal charging power. As a consequence, a large drop is seen in the total energy consumption when the first EV stops its charging.

Chapter 5

Implementation

One of the goals of this research is to enable a transition from a simulation environment to implementing DEM control in a demo-site. To achieve this, we have added two aspects to DEMKit, which we explain in this chapter. First, we explain how we acquire the data the algorithm needs as input in Section 5.1. Secondly, we explain how we send the output of the algorithms, or the control signals, to the real hardware in Section 5.2.

5.1 Data acquisition

Randomly generated profiles are used as an input for simulations. However, real-time data is needed once the system is deployed in the demo-site. Moreover, this data should be updated automatically in order for the system to work autonomously. In the valley-filling algorithm for charging EVs as explained in Section 4.3 and Section 4.4, the offline valley-filling algorithm needs historical measurement data of the remaining load and a prediction of the PV production. The online valley-filling algorithm furthermore needs measurement data of the remaining load at the current time. For simulations, these profiles are generated in the ALPG and loaded into DEMKit using a CSV reader. However, when deploying the algorithm, this information must be acquired from a different online source. Information about the availability of the EVs that have to be charged are needed as well. This section describes the acquisition of historical measurement data, the prediction of the PV production, and the SoC and deadline of the charging job.

5.1.1 Database reader

At the demo-site at Coteq, historic measurements of the remaining load and the PV production are stored in an Influx database, a time-series database [31]. To acquire data from the database, we extended DEMKit with a database reader that retrieves data from Influx databases. Figure 5.1 shows a unified modelling language (UML) class diagram of the *CsvReader* and the *InfluxReader* classes, and the general *Reader* class. In the *Reader* class, methods are available to read one or multiple values. These methods call the *retrieveValues* method of one of the implementations, making the *CsvReader* and *InfluxReader* interchangeable. The *sendRequest* method of the *InfluxReader* performs a http request to the database to retrieve the values.

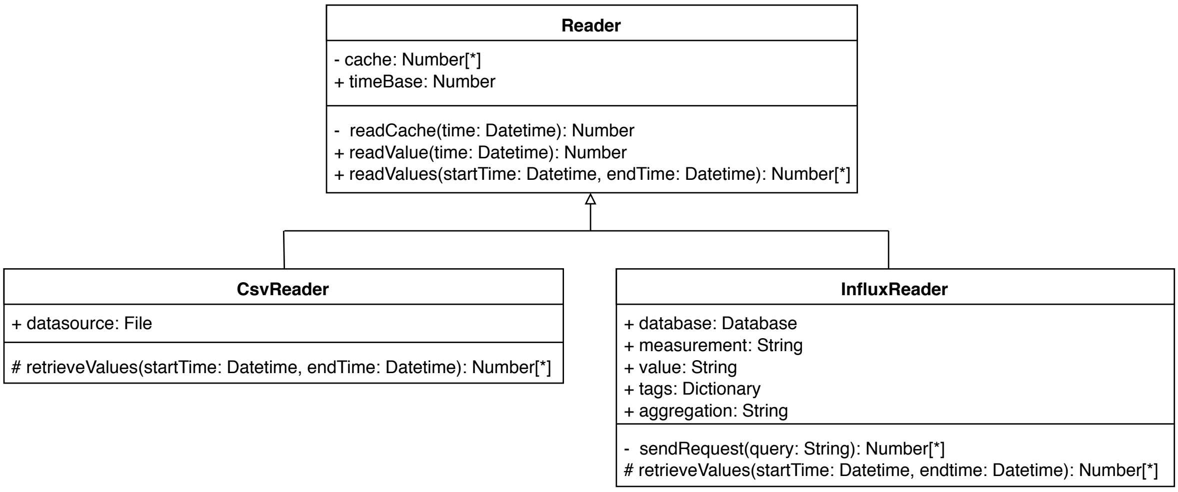


Figure 5.1: UML diagram of the CSV and Influx Reader classes

Influx uses the line protocol to insert new measurements. Source code 5.1 shows the format to do this. Measurements are saved for a certain moment in time. A measurement has a name and one or multiple values, each assigned to a so-called field. Moreover, tags may be added to a measurement, to distinguish properties of a measurement. Both fields and tags are assigned using a so-called key/value pair. The line protocol distinguishes the tags and fields by an empty space in between them. Source code 5.2, shows an example, where a three-phase power measurement in building number five is saved in an Influx database with measurement name *Active Power*. A tag key *Building* is set to the value *5*. Since the building has a three-phase connection, a field key for every phase is defined. In the field values, the respective measured active powers are saved. Lastly, the date and time of the measurement is passed in the unix nano timestamp format.

```
<measurement>
[,<tag-key>=<tag-value>...]
<field-key>=<field-value>[,<field2-key>=<field2-value>...]
[unix-nano-timestamp]
```

Source code 5.1: Format to add measurements to an Influx database

```
Active Power                            <measurement>
,Building=5                             ,<tag-key>=<tag-value>
phase1=5kW,phase2=3kW,phase3=1kW        <field-key>=<field-value>...
1532443453338000000000                  [unix-nano-timestamp]
```

Source code 5.2: Example of how to add measurements to an Influx database

To read values from an Influx database, we must know the name of the measurement. Also, the field keys and possibly tag key/value pairs need to be provided to find specific measurements. Moreover, we need to know for what time we want the measurement. In DEMKit, the host provides the internal time for the simulations. Instead of an arbitrary time starting at zero, the current system time may be used as well. When a component in DEMKit requests historic data of one day back, it subtracts one day of the current time in the simulator and passes this to the Influx database reader. The Influx database reader in DEMKit can either read one value for a moment in time, or multiple values for a given time range.

When a read request without tag key/value pairs is passed to the database, it may happen that it returns multiple measurements. For example, if we want to read the measurement that is added to the database in Source code 5.2, but no tag key/value pair are passed to the read request, the database returns the active power measurements of all buildings. The reader takes the average of these values by default, but can also be configured to return e.g. the sum or other ways of aggregating supported by Influx. The reader returns either a single value or an array of values for the requested measurement.

5.1.2 Predictions of PV production profiles

As stated in Section 4.1, the base-load measurement used for the offline valley-filling algorithm is split up into historic measurement data of the remaining load and a prediction of the PV production. However, the database only contains historic measurement data of the remaining load and the PV production. Therefore, a component is developed that gives a prediction of the PV production. The approach to compute the predictions is based on the algorithm explained in [32] and [33].

In this approach, the PV production is predicted based on the solar irradiation. Predictions of the solar irradiation are in general based on weather predictions and can be acquired from multiple sources. For this project, a prediction of the global horizontal irradiation (GHI) is obtained from Solcast [34]. Their service provides an application programming interface (API) to retrieve these predictions for a given location.

To derive the predicted PV production from the GHI predictions, we first have to determine a correlation between the production of the PV panels at the demo-site and the GHI. To do this, we compare the historic PV production with the historic GHI at the demo-site. The production of PV panels at a certain moment of the day depends on the irradiation, but e.g. also on whether there are objects blocking the sun from the panels. However, note that a drop in the PV production due to a static object blocking the sun happens approximately at the same time when using executive days. Therefore, such 'regular' information is already embedded in the historic PV production data, if this data is from recent periods.

In [32] and [33], the authors correlate the PV production to the GHI at one moment in time, using data from the same time for a number of historic days. For each of these days, a value pair (GHI, PV) at that specific moment of the day exists. When taking these value pairs for all the historic days, a correlation between the PV production and the GHI at that moment of the day is determined using linear regression on the value pairs. This results in a linear equation that returns a PV production based on a GHI input. Results in [33] suggest that using the data of ten historic days for the linear regression gives already good results for the PV production.

The above mentioned linear correlation between the PV production and the GHI for a certain moment in time can be used to obtain a prediction of the PV production. For this, the predicted GHI, which is obtained using the Solcast API, is used as an input in the obtained regression equation. Subsequently, this is stored in a database, so that the algorithm in DEMKit is able to retrieve these values.

Algorithm 5.1 shows pseudo code of this approach. The algorithm takes as input the time for which the PV production should be predicted, and a predicted GHI for this time. The algorithm first obtains the GHIs and PV productions at this time of the day, for ten historic days. This data is retrieved from databases that contain the historic data. Every obtained PV/GHI pair is added to the list of regression points. Once all historic data is acquired, the regression points are used to define a regression function using linear regression. This regression function returns the predicted PV production for the specified time, using a prediction of the GHI as its input.

Algorithm 5.1: Algorithm to predict PV production for a time interval, based on a GHI prediction and historic data of the PV production and the GHI.

function get PV prediction $(time, predictedGHI)$;
Input : The time for which the PV is predicted,
The predicted GHI at this time from Solcast
Output: Predicted PV production profile
// For ten historic days, a PV/GHI value pair is obtained as regression point
for day **from** 0 **to** 10 **do**
 $historicTime = time - (day + 1)$
 $historyPV = databasePV.readValue(historicTime)$
 $historyGHI = databaseGHI.readValue(historicTime)$
 $regressionPoints.append(HistoryPV, HistoryGHI)$
end
// Using the acquired regression points, a function is obtained
// that returns the PV production based on a GHI input
$regressionFunction = doLinearRegression(regressionPoints)$
// The PV production is predicted using the obtained function and the predicted GHI
$predictedPV = regressionFunction(predictedGHI)$
return $predictedPV$

The PV prediction algorithm has been tested for the PV production of the demo-site at Coteq in Almelo. Historic data of the PV production is available through a database, and the historic GHI are previously predicted GHI values for Almelo, from Solcast. Figure 5.2 shows resulting predicted PV production for July 13, 2018. The real measured PV production of this day is also shown. For the test, predictions are done on a day-ahead basis and on an intraday basis. The day ahead prediction is done at midnight, whereas the intraday prediction is updated every hour, using new and better predictions of the GHI in Almelo. The figure shows that the short drops in the production are not predicted by this method, but that the over-all prediction is quite good. Note that the drops are due to clouds covering the sun. However, for the valley-filling algorithm it is of importance that the total amount of energy produced is estimated quite well, so prediction errors in the short drops do not create significant prediction errors for the fill-level.

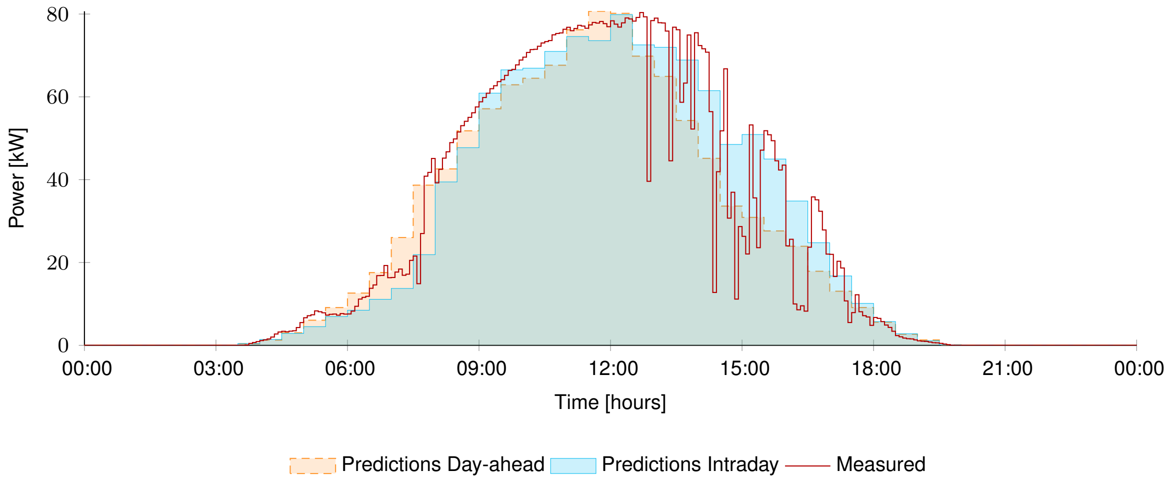


Figure 5.2: The predicted PV production compared to the real measured PV production

The total amount of produced energy that day was 582.08 kWh. We compare both the intraday and day-ahead predictions of the PV production to the real PV production in Table 5.1. Here we see that the total predicted production of the day-ahead predictions was 32.92 kWh too low and that of the intraday predictions was 1.69 too high, giving a total prediction error of 5.65% and 0.29% respectively. The average absolute error of the power is 7.14 kW for the day-ahead prediction and 4.44 kW for the intraday prediction. Lastly, we see that the maximum error in the predicted power is 33.16 kW for the day-ahead prediction and 23.23 kW for the intraday predictions.

	Day-ahead	Intraday
Total Production [kWh]	549.16	583.77
Absolute Production Error [kWh]	32.92	1.69
Total Production Error [%]	5.65	0.29
Average Absolute Error [kW]	7.14	4.44
Maximum Absolute Error [kW]	33.16	23.23

Table 5.1: Quantifying the PV predictions

5.1.3 EV and job properties

If we want to make reliable predictions of the fill-level, information about the required charge and the deadline of the job are needed. These properties are called the required charge and deadline respectively. In the demo-site, they are collected from multiple sources. To get these properties in DEMKit, the buffer-timeshiftable device class is extended with a data class. This class contains and retrieves properties of the EV and the charging jobs. Figure 5.3 shows a UML class diagram of the buffer-timeshiftable class and its data extension. Since the required information is gathered from different sources, the class contains unique identifiers for every data source. The *cpmsId* is used to distinguish the EVs in the CPMS. Every EV connected to the CPMS has a unique identifier, which can be used to detect when a job commences for this EV or to set the charging power of the EV. The *onStarId* is used to acquire the SoC of the EV through OnStar, and the *FleetsterId* is used to acquire the end time of the charging job. How this information is acquired is explained later on in this section.

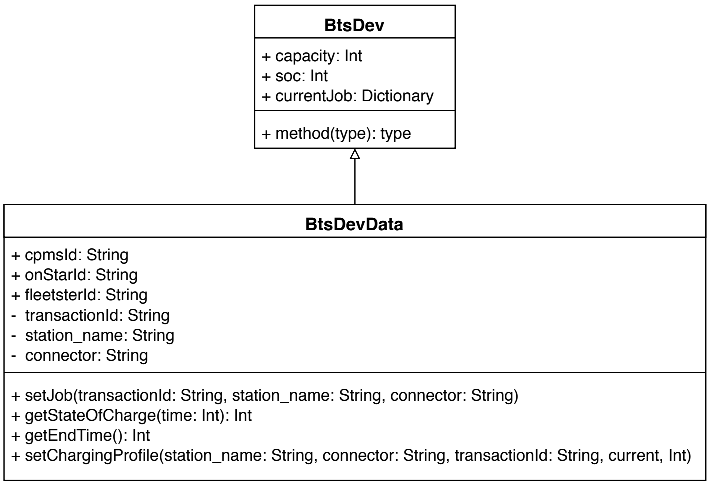


Figure 5.3: UML diagram of the data extension on the buffer-timeshiftable device class

To know when a charging job starts, the algorithm requests the CPMS to retrieve the state of all its connected charging stations at every time interval. The stations return whether an EV is connected and if so, it returns the *cpmsId*, information about which connector it is connected to, and the unique identifier of the transaction. Source code 5.3 shows the method that is used to request this information from the charging stations. When a new EV is connected, the setJob method is executed to set the transaction, station name and connector in the data object.

The *getEndTime* method in Figure 5.3 is executed to find the deadline for an active charging job. The EVs in the demo-site are pool cars of the company used for business trips. Employees have to make a reservation to use such a pool car. The reservation system used is called Fleetster [35]. In Fleetster, users can create a booking, which reserves a car for a certain time. An API is available for Fleetster that enables the automation of gathering data. With the API, a list of bookings is retrieved. Each booking has an identifier specifying which car is reserved. Moreover, in the booking it is specified when the EV has to leave. This information is used as the deadline for a charging job. So, when a charging job is started at the EVSE, the CPMS knows which EV

```python
def getStationInformation(station_name):
  url = "http://localhost:8080/api/stations"

  payload = {}
  payload['station_name'] = station_name

  r = request.get(url, data=json.dump(payload))
  if r.status_code == 200:
    result = r.json()
    return result
  else:
    return r.status_code
```

Source code 5.3: Python method to get information about whether EVs are connected to a station

is connected. Subsequently, the deadline for this job is determined by finding the first booking for this EV through the Fleetster API. Source code 5.4 shows the method to retrieve the deadline using the Fleetster API.

```python
def getEndTime():
  token = loginFleetser(username, password)
  url = "https://develop.fleetser.de/bookings

  header = {}
  header['accept'] = "application/json"
  header['Authorization'] = token

  r = request.get(url, headers=header)
  if r.status_code = 200:
    result = r.json()
    bookings = json.loads(result, object_hook=lambda d:
                    namedtuple('X', d.keys())(*d.values()))
  else:
    return r.status_code

  for booking in bookings:
    if booking.vehicleId = fleetsterId:
      endTime = booking.initialStartDate

  return endTime
```

Source code 5.4: Python method to get the end time for a charging job through Fleetster

It now remains that we describe how to acquire information about the required charge of a job. For this thesis we assume that for an EV, the desired SoC of the EV at the end of the charging job is 100%. Therefore, we can say that the required charge of a job is equal to the difference between the SoC of the EV at the start of the job and the total capacity of its battery (Equation 5.1). This means that we need information about the capacity of the EV and the SoC to predict the fill-level, instead of the required charge.

$$\text{RequiredCharge} = \text{TotalCapacity} - \text{StateOfCharge} \tag{5.1}$$

The Opel Ampera-e cars that are available in the demo-site, are connected to the Opel OnStar service [36]. This means that they have an active internet connection and that vehicle diagnostics can be requested over the air. To obtain this information, Grzegorz Szostak developed an open-source Python script that retrieves this diagnostic data from the OnStar website [37]. This script is used as a basis to retrieve the current SoC from the Ampera-e's in the demo-site when a job starts. Moreover, at every time interval in DEMKit, the SoC is retrieved again, to keep the values up to date. Source code 5.5 describes the *getStateOfCharge* method in the data class in Figure 5.3 to acquire the SoC from OnStar. The OnStar service may be used to retrieve vehicle diagnostics directly from the EV, but the last retrieved diagnostics are also saved on the OnStar server. To reduce the amount of data that is retrieved directly from the EV, the *getDiagnosticsReport* request is used first, which obtains the vehicle diagnostics from the OnStar server. We check at what time these diagnostics are saved to the server and when it is too long ago, we use the *performMetricsRefresh* request to acquire the SoC directly from the EV.

5.2 DER control

Using the acquired data from the different sources, the offline valley-filling algorithm determines the fill-level. Subsequently, the online valley-filling algorithm uses this fill-level to determine the charge levels for every EV. For executing the algorithm, these charge levels need to be sent to the respective EVs. In the following we explain how the determined charge levels are send to the EVs. Moreover, a battery is available at the demo-site, which consists of a UPS with battery cells connected to it. The reason why the UPS is not controlled is also discussed.

5.2.1 Control of the charging levels

To set the charging power of an EV, we use the CPMS, since it can manage the maximum charging currents for every connector of all connected charging stations. This maximum charging current limits the amount of power an EV can draw from the grid. However, the EV can still decide to charge less, so the maximum charge current is no guarantee that this amount of power is actually charged. For example, when the CPMS sets the maximum charging current to 32 A, but the battery of the EV is already full, the EV will charge at 0 A instead. To set the maximum charging current to the determined charge level, we use an API provided by the CPMS. The CPMS then sets the maximum charging currents for the respective connectors, causing the connected EVs to charge with no more than the maximum current set. Source code 5.6 shows the method used to set the charge current for a specific charging job.

```python
def getStateOfCharge(time):
  token = loginOnStar(username, password)
  url = "https://gsp.eur.onstar.com/gspserver/services/vehicle/"

  header = {}
  header['X-GM-token'] = token

  payload = {}
  payload['vehicleId'] = onStarId

  r = requests.get(url + "getDiagnosticsReport.json",
                  data=payload, headers=header)
  if r.status_code == 200:
    result = r.text()
    diagnostics = json.loads(result, object_hook=lambda d:
                  namedtuple('X', d.keys())(*d.values()))
  else:
    return r.status_code

  reportData = diagnostics.results[0].reportData
  if reportData.debug.completedOn < time - timeThreshold:
    url = "https://gsp.eur.onstar.com/gspserver/services/vehicle/"

    header = {}
    header['X-GM-token'] = token
    header['Content-Type'] = 'application/x-www-form-urlencoded'

    payload = 'vehicleId=' + onStarId

    r = request.post(url + "performMetricsRefresh.json",
                    data=payload, headers=header)
    if r.status_code == 200:
      result = r.text()
      diagnostics = json.loads(result, object_hook=lambda d:
                    namedtuple('X', d.keys())(*d.values()))
    else:
      return r.status_code

    reportData = diagnostics.results[0].reportData
  return reportData.metrics.soc * self.capacity
```

Source code 5.5: Python method to get the state of charge of an Ampera-e connected to a station

```python
def setChargingProfile(station_name, connector, transaction, current):
  url = "http://localhost:8080/api/charging_profile"

  payload = {}
  payload['station_name'] = station_name
  payload['connector_id'] = connector
  payload['transaction_id'] = transaction
  payload['amperage'] = current

  r = request.get(url, data=json.dump(payload))
  if r.status_code == 200:
    result = r.json()
    return result
  else:
    return r.status_code
```

Source code 5.6: Python method to set the charging current of an EV

5.2.2 The UPS

In Section 3.1.1, we discuss the hardware in the demo-site and explain how a battery is realized in the demo-site. This battery provides an extra layer of flexibility. The UPS of this battery can be set to either charge from the grid with a fixed charging power or discharge into the EVs. However, the UPS has fuses of 25 A per phase on its in- and outputs. Therefore, not even a single Opel Ampera-e can be charged at its maximum charging power before overloading the fuses. Moreover, two charging stations are connected to the battery and thus four EVs can potentially be connected to it simultaneously. In this case, every EV can charge at most 6 A. For this reason, we chose to switch the battery to a bypass mode, where the EVSE that is normally connected to the battery, is now connected directly to the grid. In this bypass mode, the battery is effectively removed from the system.

Chapter 6

Experiments and results

6.1 Introduction

To validate whether the system works and how it performs, we tested it within a pilot. The data of this pilot is logged to compare the behaviour of the system to the simulation results. Moreover, after the experiment we used the data from the pilot as an input to the simulator to further improve the algorithms.

6.2 Pilot

Both the online and offline valley-filling algorithms are implemented in a demo-site and tested during a pilot. More specifically, an experiment is conducted where two Opel Ampera-e EVs are charged. For offline valley-filling, the minimum charging threshold algorithm is used (Section 4.3). The minimum charging threshold auction from Section 4.4 is used as the online valley-filling algorithm. The online algorithm is executed every fifteen minutes.

Offline algorithm: *Valley-filling algorithm respecting minimum charging threshold*
Online algorithm: *Charge level auction respecting minimum charging threshold*

The first EV was plugged in at 05:30 and had to charge about 50% of its battery capacity before its deadline at 13:30. The second EV was plugged in at 06:30 and had to charge about 20% of its capacity before its deadline at 14:00. Table 6.1 shows an overview of these two charging jobs that are executed during the experiment.

	Start Time	Deadline	Required Charge
2018 Opel Ampera-e (EV1)	05:30	13:30	30 kWh
2017 Opel Ampera-e (EV2)	06:30	14:00	12 kWh

Table 6.1: Charging jobs during the experiment

For the used algorithms, the PV production and the remaining load are important inputs. Figure 6.1 shows the measured PV production and the power consumption of the remaining load during the experiment. However, note that these profiles are not known beforehand, but are measured during the experiment. For the PV production we may use the predicted PV production, which is known beforehand, as input. It is determined using the PV prediction algorithm as presented in 5.1.

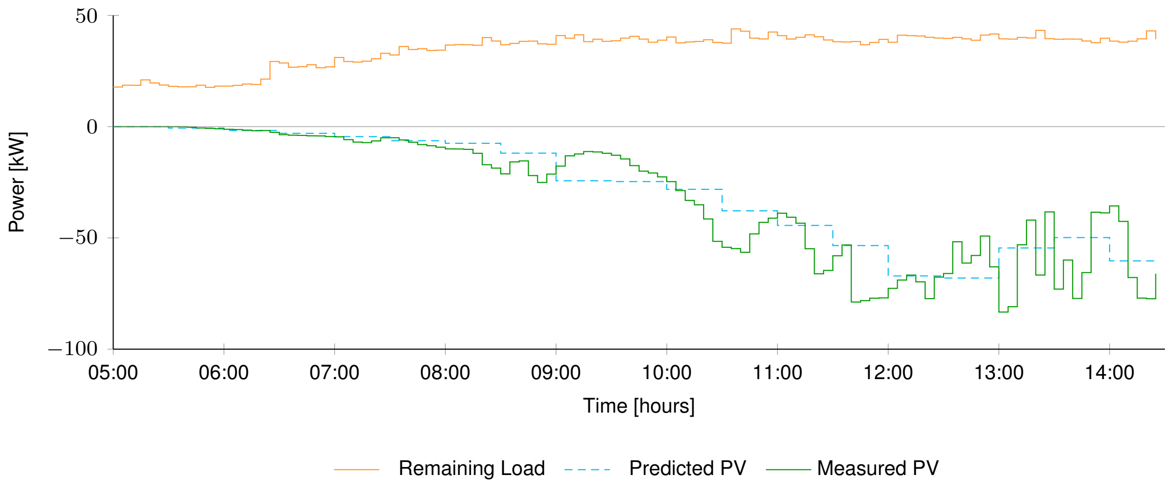


Figure 6.1: The PV production and power consumption of the remaining load during the pilot

We use ten historic load profiles of the past days to determine the fill-level when a charging job starts. These baseload profiles consist of the prediction of the PV production as seen in Figure 6.1 and the historic load profiles of the past days. The online algorithm clears the market using the baseload profile of the day of the experiment. Figure 6.2 shows this baseload. However, since the online algorithm clears the market every fifteen minutes, it takes the average baseload of those fifteen minutes to determine the charge level.

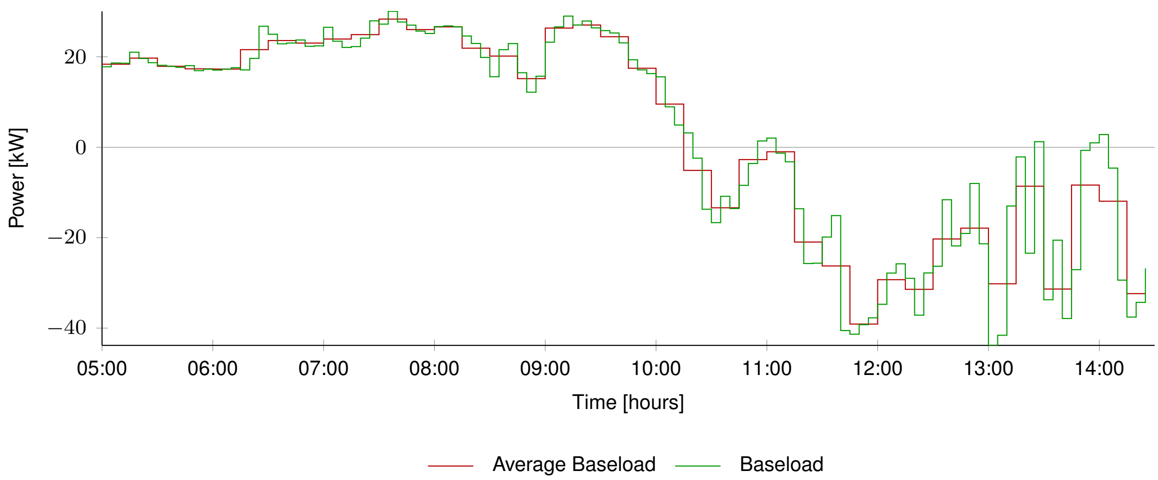


Figure 6.2: The baseload during the experiment, which is used by the online algorithm

6.2.1 Simulating the experiment

To first give an idea of the results that we may expect from the experiment, we created a scenario of the experiment and simulated it in DEMKit. We use the PV production and remaining load profiles of Figure 6.1 to recreate the same baseload as in the experiment. Moreover, two EV charging jobs are added that have the properties from Table 6.1. Figure 6.5 shows the baseload and measured total power consumption. The *EV charging powers* are plotted as a so-called stacked area plot, where the absolute charging power of the first EV is stacked upon the baseload and the absolute charging power of the second EV is stacked upon the charging power of EV1.

We see in the results that when EV1 arrives, a charging power is determined where the baseload is filled up to the fill-level. However, around 06:30, 08:30, and 09:15, we see that the EVs charge more, such that the total power consumption exceeds the fill-level. This happens because the online algorithm uses the average baseload of a fifteen minute interval in Figure 6.2 to determine the charge levels. For these times, the actual baseload exceeds the fill-level, but the average baseload does not. From around 10:00 we see that the charging powers of both EVs gradually increase as the PV production increases as well. The system tries to 'fill' the power consumption up to the fill-level, but is limited by the maximum charging powers of the EVs. The EVs charge at their maximum charging powers until their batteries are full. For this they make use of the energy produced locally and reduce the amount of energy that is fed back to the grid. Figure 6.3 and Figure 6.4 show the charging powers and SoC of both EVs in more detail.

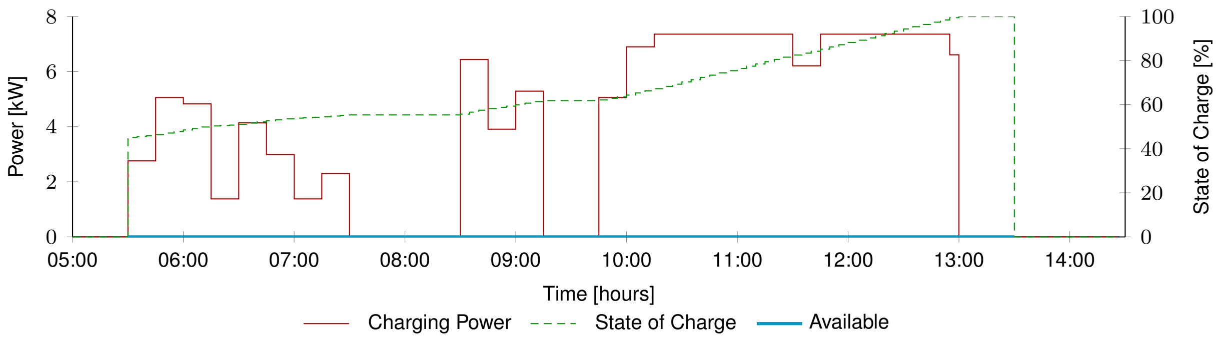


Figure 6.3: The charging power and the SoC of EV1 in the simulation

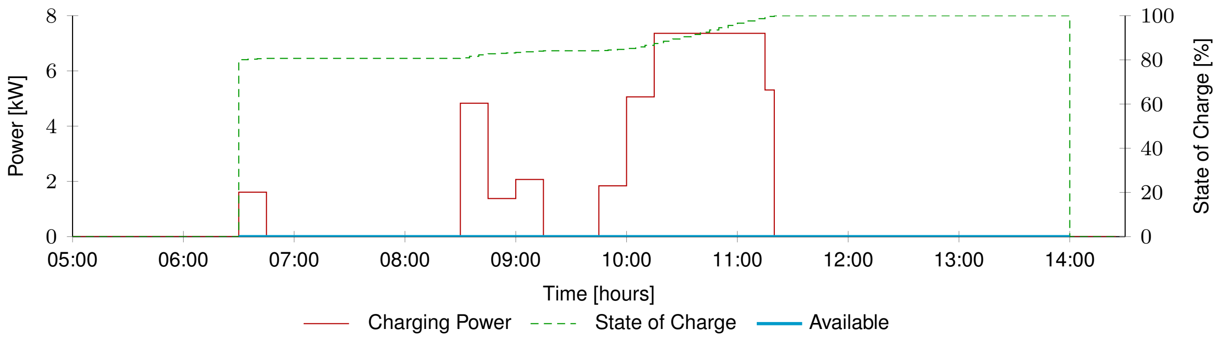


Figure 6.4: The charging power and the SoC of EV2 in the simulation

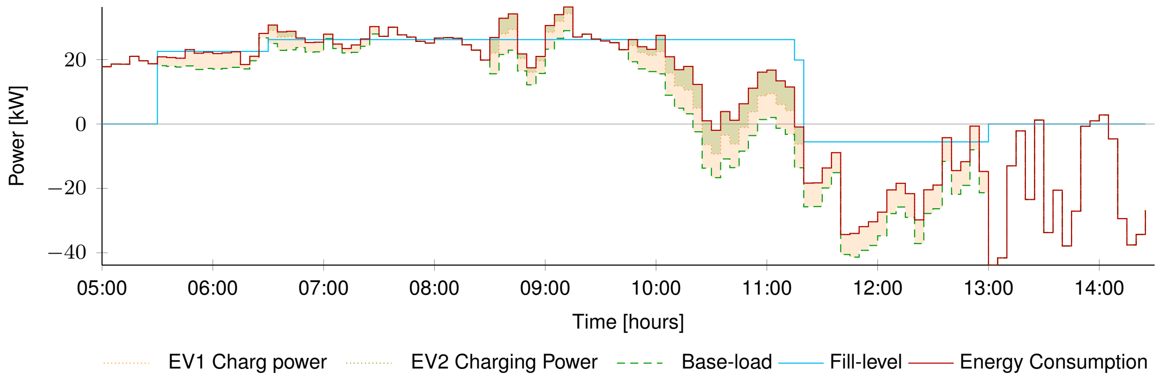


Figure 6.5: The results from the simulation created using the data from the experiment

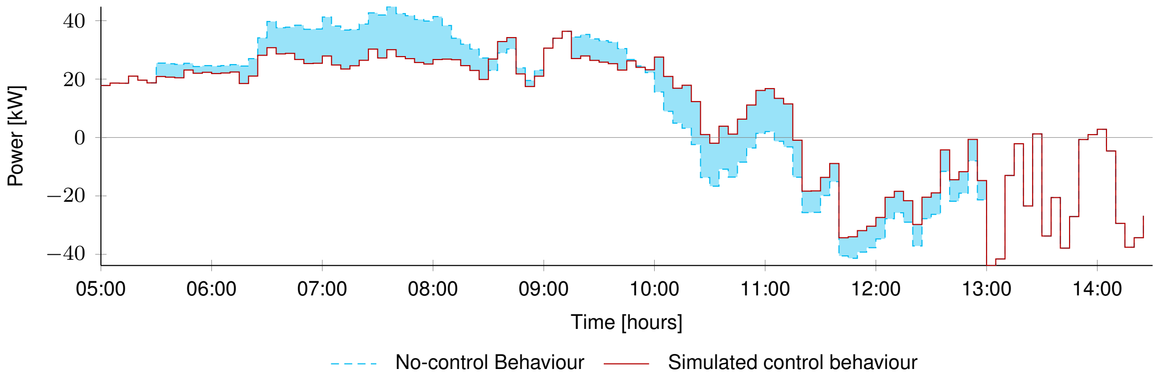


Figure 6.6: The difference between using DEM and the no-control behaviour in the simulation

Figure 6.6 compares the *no-control behaviour* to the power consumption of this simulation. The no-control behaviour shows the power consumption we would have seen when no DEM control is applied. Here we see even better that in a no-control situation, the consumed power is higher in the beginning, and the produced energy is not used locally but instead fed to the grid. We quantify the difference in Table 6.2. The column *Percentage* shows the relative improvement of the control behaviour compared to the no control behaviour. We see that when using DEM control, 18.95 kWh of energy is used to balance consumption and production, of the 42 kWh of flexibility that was available. The highest import peak is reduced by about 19%. Since the EVs are finished charging before the highest export peak, no improvement is seen here.

	Control behaviour	No-control	Improvement	Percentage [%]
Imported energy [kWh]	139.79	158.75	18.95	11.94
Exported energy [kWh]	64.34	83.29	18.95	22.76
Import peak [kW]	36.37	44.79	8.42	18.80
Export peak [kW]	43.82	43.82	0.00	0.00

Table 6.2: Quantification of the grid improvements for the simulation

6.2.2 Results of the experiment

The results obtained from the experiment are given in Figure 6.7, where the measured total consumption of the system (*Measured*) is given. The baseload is plotted as well, to show where the EVs are charged. The *EV charging powers* are the charging powers at which the EVs are charged during the experiment. Similar to Figure 6.5, they are stacked to visualize how they contribute to the measured profile. Lastly, *Fill-level* shows the fill-level as determined by the offline valley-filling algorithm. The fill-level is redetermined every fifteen minutes. Figure 6.8 and Figure 6.9 show the behaviour of the first and second EV respectively. Here, the expected charging powers are plotted, as well as the real charging power as measured by the charging stations. *State of Charge* shows the SoC of the batteries in the EVs and *Available* shows when the EVs are available to be charged. Some data of the experiment is missing, because of a fault during the experiment. Due to time constraints, there was no possibility to redo the experiment.

Missing data

The first thing we notice in these results is the greyed out area that is present from 09:30 until 11:45. Between these times, no reliable data is produced during the experiment. The reason for this lies in the acquisition of the SoC from the EVs. In the experiment, this is done every five minutes using the method as presented in Source code 5.5. However, it seems that the infrastructure to retrieve the SoC is not designed to be used this often. The request to retrieve the SoC from both EVs took over a minute to return at first, but starting at 09:30, this request returned an error instead of the vehicle diagnostics. Starting at 11:45, the code has been adapted to only acquire the SoC from the EVs every fifteen minutes. In time intervals where the real SoC is not retrieved, it is predicted using the expected charging powers of the EVs. This can be seen in the SoC in Figure 6.8, where the SoC fluctuates a bit at 13:00.

Charging power difference EV1

The second anomaly is seen in the difference between the expected and measured charging powers. Figure 6.8 shows that the real charging power does not exceed 3.7 kW, or 16 A. For all charging powers below this point, the real charging power follows the expected charging power reasonably. In Figure 6.7, we see that between 05:30 and 06:30, the measured energy consumption is below the fill-level. If the EV would have charged at the expected 7.4 kW instead of 3.7 kW, it would have followed the fill-level better. Also, Since the SoC of the EV rises slower than expected, the fill-level keeps getting a bit higher every time it is redetermined.

A possible reason why the charging power is capped at 16 A, lies in the design of the UPS and the battery cells, as explained in Section 3.1.1 and Section 5.2. Since the UPS has 25 A fuses, the charging powers in the EVSE were limited to 16 A before the experiment. Although the UPS is put in bypass mode for this experiment, eliminating the 25 A fuse limits, the limit of the charging powers is probably not removed from the EVSE. We were not able to change this during the experiment. The combination of the limited charging power and the time where the system did not work correctly results in the first EV not reaching a SoC of 100% before its deadline.

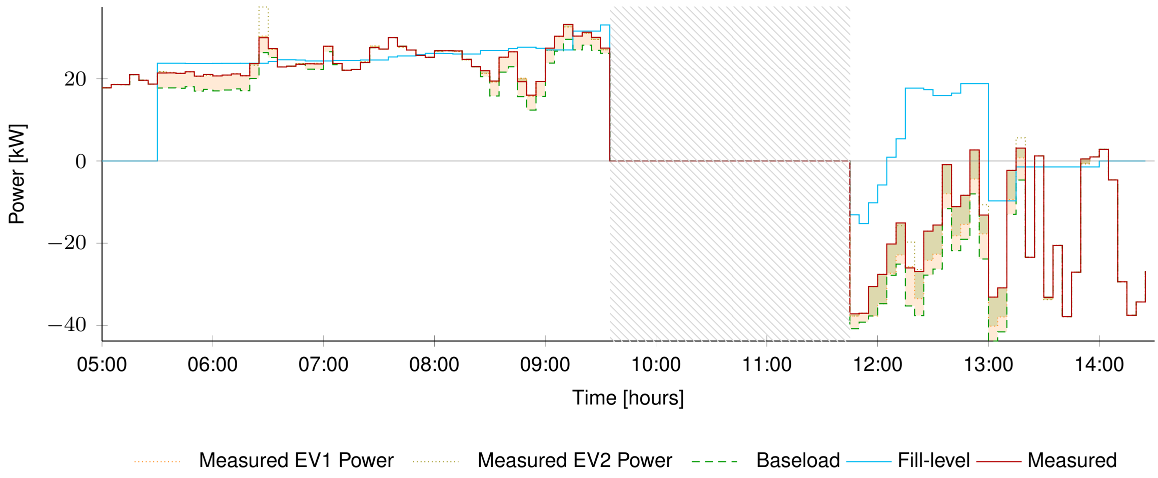


Figure 6.7: The results obtained from the experiment

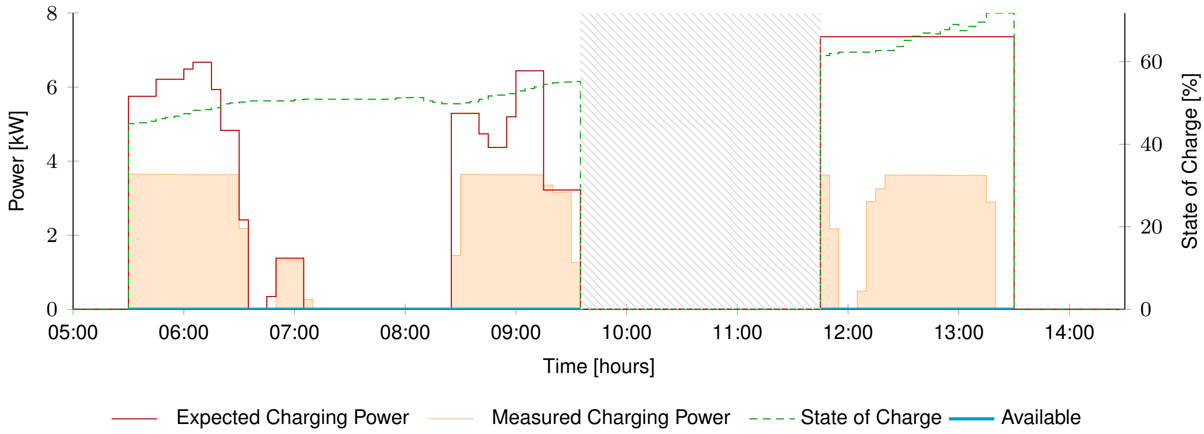


Figure 6.8: The expected and real charging powers, and the SoC of EV1 during the experiment

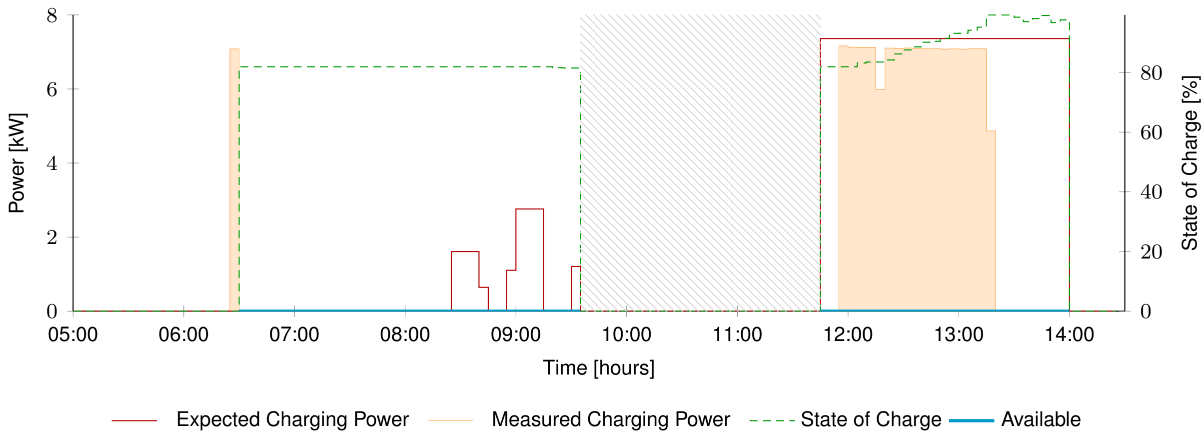


Figure 6.9: The expected and real charging powers, and the SoC of EV2 during the experiment

Charging power difference EV2

Lastly, an anomaly is observed in the difference between the expected and measured charging power of the second EV. Every minute, the DEM system checks if an EV has connected. During the anomaly, the EV connected and immediately started charging at its maximum charging power. At the next full minute, the system detected the EV and determined it should stop charging. For this reason, we see a very short peak in the real charging power just before the start of the job. However, we also observe that between 08:00 and 09:30, when the EV should charge, it does not. A possible explanation for this is that once the charging power is set to zero, the EV goes into some kind of error state where it will not charge during this job. From 11:45, the EV has been disconnected from and reconnected to the EVSE, to start a new job. As we can see, the EV does charge here, until the charging power goes to zero again. When this happens, the EV does not start charging again. At this point the SoC was close to but not entirely 100%. EV1 did not show this behaviour, but EV2 is the 2017 model, whereas EV1 is the 2018 model.

Grid improvements when using DEM

To give insight in the effect on the grid when using DEM control, we present a so-called *no-control behaviour* in Figure 6.10. For this, we also virtually charge the EVs as when no DEM control is applied. Without control, the EVs start charging at their maximum charging powers as soon as they are connected to the EVSE, to fully charge as fast as possible. We call this *greedy charging*. When EV1 starts this greedy charging immediately at 05:30 and EV2 at 06:30, the total power consumption results in the *No-control Behaviour* plot in Figure 6.10. In this figure, we also added the power consumption of the demo-site as given in Figure 6.7 in *Measured*. Because both the power consumption for a scenario with and without DEM control are plotted, it gives a good insight in the obtained improvement when using DEM control for this experiment. Between 05:30 and 09:30, the EVs charge more in the no-control behaviour, resulting in higher consumption from the grid. Moreover, between 11:45 and 13:30 in the no control case the EVs are fully charged already, so they cannot make use of the local PV production. Thus in the no-control behaviour, this results in feeding back more produced energy to the grid, instead of using it locally.

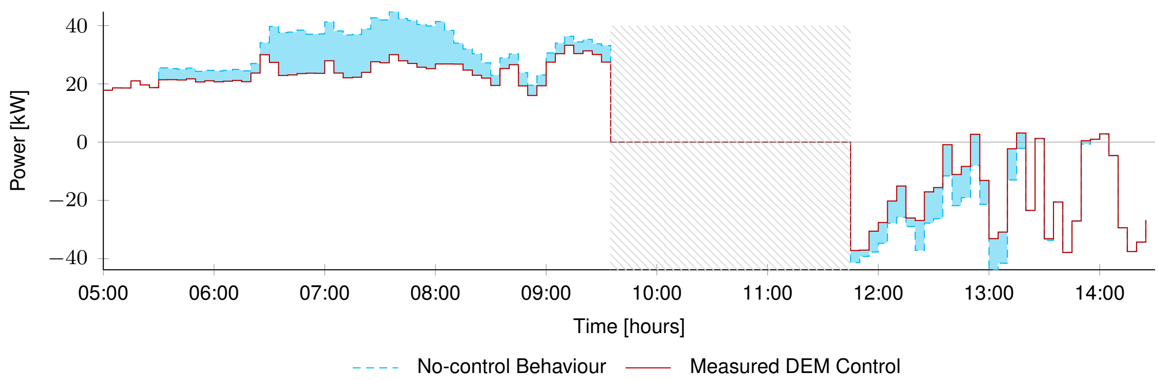


Figure 6.10: The difference between using DEM and the no-control behaviour in the experiment

The difference between the measured power consumption of the experiment and the no-control behaviour is quantified in Table 6.3. The column *Improvement* shows the absolute amount of energy/power that has been consumed or produced less in the experiment when compared to the no-control behaviour. Due to the missing data, we see a mismatch between the improvement in imported and exported energy. The results show a decrease of 25.95 kWh in imported energy compared to the no-control behaviour. The maximum possible improvement was 42 kWh, since this is the sum of the required charges of both charging jobs. From the table we can conclude that when using DEM control for this experiment, the system consumes about 18% less energy from the grid and about 32% less is fed to the grid. Moreover, the peak consumption and production powers are reduced by about 13% and 14% respectively.

	Control behaviour	No-control	Improvement	Percentage [%]
Imported energy [kWh]	118.42	144.37	25.95	17.98
Exported energy [kWh]	44.51	65.44	20.93	31.98
Import peak [kW]	38.81	44.79	5.98	13.36
Export peak [kW]	37.56	43.82	6.26	14.28

Table 6.3: Quantification of the grid improvements for the experiment

6.3 Simulating the battery

In Section 3.1.1 we discuss the battery that is available at the demo-site. This battery consists of multiple battery cells with a total capacity of 8 kWh, and a UPS to charge and discharge the battery cells. The battery cannot feed energy to the grid, but can only be discharged though charging EVs connected to the battery. The UPS can either be in a charging or discharging state. In the charging state, the battery cells are charged from the grid at a fixed power of 3 A or 1.2 kW. In this mode, the connected EVs are charged from the grid as well. In the discharging state, the UPS is decoupled from the grid, to prevent it is charged and it only discharges to the connected EVs. Table 6.4 shows an overview of the states.

	Input power from grid	Output power
Charging	EV powers + Fixed battery cells power (1.2 kW)	EV charging powers
Discharging	None	EV charging powers

Table 6.4: Battery states

As explained in Section 5.2, the battery is effectively removed from the demo-site during the pilot, because limitations in the fuses make it not possible to charge the EVs through the UPS. However, to study the potential improvement of this battery, a simulation is executed where the battery is included in the scenario. In this simulation, the output power of the battery is equal to the charging powers of the EVs of the simulation in Section 6.2.1. Figure 6.11 shows these charging powers stacked upon each other, such that the upper line shows the aggregated power consumption. Table 6.4 shows that the input power of the battery is either equal to zero or to its output power plus the battery cells charging power of 1.2 kW, depending on the state of the UPS.

Figure 6.12 shows the behaviour of the battery in the simulation, when starting with a full battery. Until 08:30, the battery is in the discharging state and is thus charges the connected EVs. At 08:30, the SoC of the battery is too low to charge the EVs, and therefore it switches to its charging state for two intervals. In these intervals, the input power is equal to the output power plus 1.2 kW. This gives the battery just enough energy to discharge from 09:00 till 09:15, resulting in an empty battery. Then the UPS switches to the charging state again, where it charges its battery cells until 13:00. Once the EVs are finished charging, the battery switches between its charging and discharging state based on the baseload.

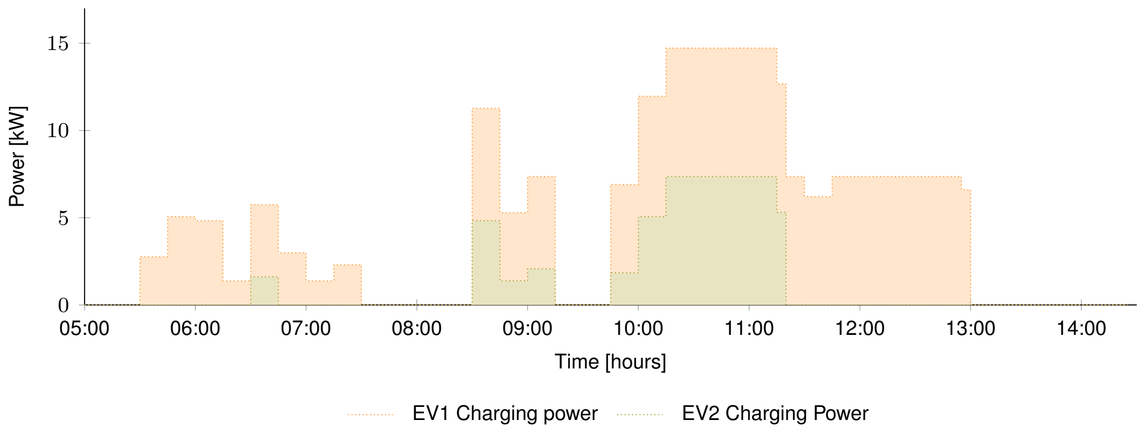


Figure 6.11: The output power of the battery consists of the charging powers of EV1 and EV2

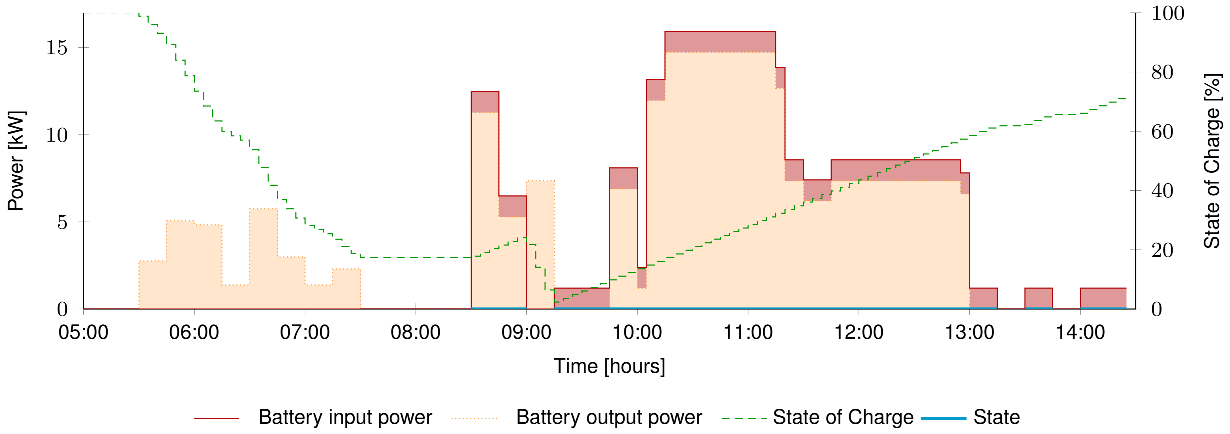


Figure 6.12: Behaviour of the battery during the simulation

Figure 6.13 shows the total power consumption for when the battery is used and when the battery is not used. Here we see that when the battery is in the discharging state (05:00 - 08:30 and 09:00-06:15), the total power consumption is lower, because the EVs are not charged from the grid. When the battery charges (08:30-09:00 and 09:15-13:00), the total power consumption is 1.2 kW higher. A large part of this higher power consumption comes from locally produced energy, reducing the amount of energy that is fed to the grid. However, since the batteries capacity and charging power are relatively low, the differences are not significant.

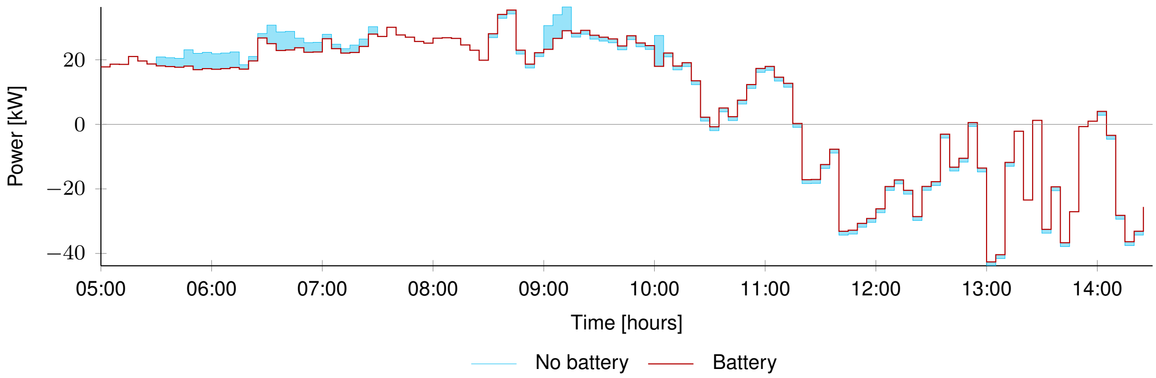


Figure 6.13: The difference between using and not using the battery to charge the EVs

The difference between using the battery and not using the battery is quantified in Table 6.5. The values for the no-battery situation are equal to the results of the simulation in Table 6.2, since this is the same simulation. Hence, note that for both situations, DEM control is used. This table shows the improvement when adding the battery to the DEM control. We see here that the absolute improvements for the consumed and produced energy are not the same any more, since the battery also stores energy. Since the capacity of the battery is relatively small (8 kWh), the improvement in the consumed and produced energy is only about 4% and 5% respectively.

When the battery discharges, the EVs charge entirely from the battery. This results in an improvement of 0.96 kW or about 3% in the highest consumption peak, since this power is consumed from the battery and not from the grid. The charging power of the UPS is only 1.2 kW, making that the highest production peak is reduced no more than this 1.2 kW, or about 3%.

	Battery	No battery	Improvement	Percentage [%]
Imported energy [kWh]	133.51	139.79	6.29	4.49
Exported energy [kWh]	61.11	64.34	3.23	5.02
Import peak [kW]	35.41	36.37	0.96	2.63
Export peak [kW]	42.62	43.82	1.20	2.74

Table 6.5: Quantification of the grid improvements when using the battery

Figure 6.14: The 2018 Opel Ampera-e being charged during the pilot

Chapter 7

Conclusions and recommendations

This thesis describes a smart grid that was realized at a demo-site located at Coteq in Almelo. A pilot is performed in this smart grid, where a decentralized energy management (DEM) system is validated. The valley-filling approach is used to control the distributed energy resources (DERs) in the smart grid. Section 7.1 states the conclusions that can be made from this research, followed by the recommendations in Section 7.2.

7.1 Conclusions

From the results of this thesis, we can answer the main research question as stated in the introduction:

> *How to design and implement a decentralized energy management (DEM) system at Coteq to control the flexibility provided by energy resources in a grid, with the goal of reducing stress on grid assets and balancing consumption and production, while providing insight on the effect of smart grid control to its stakeholders?*

To answer this question, we answer the following sub-questions:

> *How to use DEMKit as a basis to implement a DEM system?*

We extended DEMKit with multiple features to enable the transitioning from a simulation environment to implementing DEMKit for real-time DEM control. We developed an Influx database reader in DEMKit to acquire real-time and online input data. Moreover, we developed a feature that predicts the production of photovoltaic (PV) panels based on historic PV production data and a prediction of the global horizontal irradiation (GHI). This predicted PV production is used as input for DEM algorithms.

> *What energy resources at Coteq provide flexibility and to what extent do they provide this flexibility?*

Energy resources in the demo-site that provide control flexibility are two Opel Ampera-e EVs, and a battery that consists of an uninterruptible power supply (UPS) and multiple battery cells with a total capacity of 8 kWh. The EVs have a capacity of 60 kWh and a maximum charging power of 32 A, or 7.4 kW. The UPS either charges its battery cells from the grid at a fixed power of 1.2 kW, or is disconnected from the grid. When it is disconnected, the battery cells can only be discharged into charging EVs. When the UPS is charging, these EVs are charged from the grid instead of the battery cells.

To predict and control the behaviour of the UPS and the battery cells, a controller is developed in DEMKit that first optimizes the charging jobs of the EVs connected to the UPS. Then, the UPS is optimized and controlled accordingly.

What optimization algorithm should be used to control the flexibility?

The optimization algorithm used for DEM control is the valley-filling algorithm, which consists of an offline part and an online part. The first predicts the fill-level for a charging job based on historic base-load data and specifications of the charging job, and the latter determines the charge level for every charging EV based on the current base-load at every discrete time interval. For offline valley-filling, an algorithm is used that predicts the fill-level while taking into account a minimal charging threshold for the EVs. This predicted fill-level is then used in the online valley-filling algorithm, which is implemented using a double sided auction. Based on the amount of energy an EV still has to charge and the time when the job has to be finished, every EV defines a bid function that states the amount of power it is willing to consume. Moreover, a bid function is developed that takes into account the minimal charging threshold of the EV as well. The online algorithm divides the total available charge over all EVs using their bid functions, the fill-level, and the current base-load.

How can this algorithm be implemented in the demo-site at Coteq in Almelo?

To implement DEM control at Coteq, we added features to DEMKit that gather information about the EVs at the demo-site and their charging jobs. The reservation system Fleetster is used to acquire deadlines of new charging jobs and during a job, the SoC of the available EV is acquired through OnStar. A local charge point management system (CPMS) in the demo-site is also connected to DEMKit. This CPMS signals DEMKit when a new charging job commences, and sets the desired charging powers of the connected EVs.

We conducted a pilot, where we used the valley-filling algorithm in the demo-site to control the charging of two EVs. During the experiment, we acquired the SoCs of the EVs through OnStar every five minutes. However, the infrastructure to acquire the SoC through OnStar is not reliable enough to be used this often, which caused the experiment to fail halfway. To continue the experiment, the rate of using OnStar was reduced to once every fifteen minutes. This has given no problems for the remainder of the experiment. During the pilot, one of the charging stations was limited to a charging power of 3.7 kW instead of 7.4 kW. Due to this limit, the measured charging power of the EV differed from the expected charging power, making that the EV was not charged in time. Also, when the charging power of the 2017 Ampera-e was set to zero, it would not start charging again during the same charging job. This also caused a difference between the real and expected charging power. To start charging again, we had to reconnect the cable to the charging station.

> *How to provide stakeholders with easy insight on the effects from this DEM control system?*

By showing the difference between using, and not using DEM control to stakeholders, we give insight on the effect of DEM control. When we use DEM control to charge EVs in the demo-site, a simultaneous simulation is done where we charge the same EVs DEM control, using greedy charging. This simulation is called the no-control behaviour. The results of both the DEM control and the no-control behaviour are updated and shown in real-time, so that stakeholders can see the improvements that DEM control give.

> *To what extent does DEM control reduce stress on grid assets and to what extent does it balance consumption and production?*

We used the results of the experiment with the missing data to simulate in DEMKit how the experiment should have gone. In order to quantify how DEM control balances consumption and production for the experiment, these simulation results are compared to the no-control behaviour. This comparison shows that, when using DEM control, about 12% less energy is imported from the grid during the experiment, opposed to using no DEM control. About 23% more of the energy produced by the PV panels is used locally instead of fed to the grid. When the battery in the demo-site is used, the energy imported from the grid is reduced by another 4% and about 5% more of the locally produced energy is used instead of exported.

DEM control also reduces stress on grid assets, since it reduces both the peak import power and the peak power of the locally produced energy that is fed to the grid, called peak export power. When comparing the simulation of the experiment to the no-control behaviour, the peak import power is reduced by 19% in the situation with DEM control. The peak export power is not reduced, since the EVs finished their charging before this peak. Adding the battery reduces both the import and export peaks by another 3%.

7.2 Recommendations

From this research, recommendations for future research and improvements arose. First, we recommend to repeat the experiment from this thesis. For the next experiment, adaptations should be done in both the algorithms and the implementation. First we look at adaptations in the implementation. The acquisition of the SoC of the EVs should be made more robust. Also, the power limitations in the EVSE should be removed, to see whether the real charging powers match the expected charging powers better. When a feature to acquire the SoC from Volkswagen Golf GTE's is added to DEMKit, two of these EVs may be used in the experiment as well.

The following recommendations for the algorithms are given. The offline valley-filling algorithm for EV charging should be improved. Firstly, in order to make a good prediction of the fill-level, it is of importance to have a good prediction of the PV production. Using the linear regression method on historic PV production data and solar irradiation data already gives good results, but the predictions can be made more robust. Outliers in the points used for regression decrease the accuracy of the predictions. The prediction might be improved by only using days with clear skies as an input for the regression method.

Secondly, the way how we use the breakpoint search algorithm to determine the fill-level when multiple EV are charging should be adapted. Now, the algorithm first determines the fill-level for the first EV and applies the corresponding charging profile to the baseload. This is then used as input to define the fill-level for the following EV. However, this method returns an over-estimated fill-level, which is not optimal. An iterative solution can be considered, where the fill-level is iteratively determined for each EV, until it reaches an optimal fill-level.

Lastly, the offline valley-filling algorithm may be changed in such a way that it incorporates the arrival of new EVs when the fill-level is determined. In other words, that the algorithm looks ahead and predicts what charging jobs will have to be done on a day. This way, the EVs may charge more in the beginning, to reduce potential consumption peaks later on.

Moreover, we recommend to adapt the valley-filling algorithm in such a way that it can be used to control the charging and discharging of a battery. When the problems with the fuses of the battery in the demo-site are solved, an experiment may be performed where we use the flexibility of the battery as well. We can compare the results of such an experiment to the results obtained from the simulations with a battery in this thesis.

Concerning the online valley-filling algorithm, improvements may be made in determining the charge level. In the experiment, we determine the charge level every fifteen minutes for the following interval, using the average baseload for these fifteen minutes. We recommend to extend the online valley-filling algorithm to predict the baseload for the coming fifteen minutes to determine the charge level when the baseload is not known beforehand.

One of the goals of this research was to use DEMKit to transition from a simulated smart grid environment to a real DEM implementation. To better facilitate this transition, the addition of asynchronous processes should be considered. An asynchronous process may be used to detect when a new EV arrives. When using such an asynchronous event, the new EV is detected as soon as it connects, instead of at the next interval. Moreover, an asynchronous process may be used to acquire the SoC of an EV. In this thesis we observed that it can take quite some time to acquire the SoC directly from an EV, causing the entire platform to stall. If this is done asynchronously, the rest of the system can continue to operate using an estimated SoC.

One of the challenges that persists when controlling the charging of EVs, is the acquisition of data concerning the required charge and the deadline for the charging job. We recommend to look into EV charging using the CHAdeMO standard, since the communication protocol for this standard provides us with more information from the EV. Also, version 2 of the Open Charge Point Protocol (OCPP) is expected to provide a standard for retrieving the state of charge (SoC) of EVs.

List of acronyms

AC alternating current

ALPG Artificial Load Profile Generator

API application programming interface

CAES Computer Architecture for Embedded Systems

CAN controller area network

CCS combined charging system

CHP combined heat power

CPMS charge point management system

CP control pilot

DC direct current

DER distributed energy resource

DEM decentralized energy management

DMMP Discrete Mathematics and Mathematical Programming

DSO distribution system operator

EHV extra high voltage

EM energy management

EV electric vehicle

EVSE electric vehicle supply equipment

GHI global horizontal irradiation

HV high voltage

IEC International Electrotechnical Commission

LAN local area network

LV low voltage

MV medium voltage

OCPP Open Charge Point Protocol

PE protective earth

PLC power-line communication

PV photovoltaic

RES renewable energy source

SoC state of charge

TSO transmission system operator

UML unified modelling language

UPS uninterruptible power supply

Bibliography

[1] EnergiePortal, “Overzicht netbeheerders elektriciteit en gas.” [Online]. Available: https://www.energieportal.nl/onderwerpen/netbeheerders Last accessed on 1 July 2018. (Cited on pages 1 and 4).

[2] Netbeheer Nederland, “Kerngegevens energienetten.” [Online]. Available: http://energiecijfers.info/hoofdstuk-1/ Last accessed on 21 June 2018. (Cited on page 1).

[3] United Nations Treaty Collection, “Paris agreement.” [Online]. Available: https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en Last accessed on 21 June 2018. (Cited on page 2).

[4] Overheid.nl, “Besluit van 26 april 2018 tot vaststelling van het tijdstip van inwerkingtreding van de wijziging van de elektriciteitswet 1998 en van de gaswet (voortgang energietransitie).” [Online]. Available: https://zoek.officielebekendmakingen.nl/stb-2018-129.html Last accessed on 21 June 2018. (Cited on page 2).

[5] J. Gerdes, S. Marbus, and M. Boelhouwer, “Energietrends 2016,” Energieonderzoek Centrum Nederland, September 2016, Report. (Cited on pages 2 and 3).

[6] International Energy Agency (IEA), “Global EV Outlook 2018,” May 2018, Report, doi: 10.1787/9789264302365-en. (Cited on page 3).

[7] CBS, “Jaarmonitor wegvoertuigen: Aantallen.” [Online]. Available: https://www.cbs.nl/nl-nl/achtergrond/2017/21/jaarmonitor-wegvoertuigen-aantallen Last accessed on 1 July 2018. (Cited on page 3).

[8] M. H. H. Schoot Uiterkamp, “Robust planning of electric vehicle charging,” Master’s thesis, University of Twente, August 2016. (Cited on page 3).

[9] E. Loveday, “Tesla Ups Supercharger Charging Rate For Refreshed Model S 90D & P90D.” [Online]. Available: https://insideevs.com/tesla-ups-supercharger-charging-rate-refreshed-model-s-90d-p90d-video/ Last accessed on 26 June 2018. (Cited on page 3).

[10] Autoriteit Consument & Markt (ACM), “Netcode Elektriciteit,” 2015. [Online]. Available: https://www.acm.nl/sites/default/files/old_publication/publicaties/14381_netcode-elektriciteit-2015-12-18.pdf Last accessed on 6 August 2018. (Cited on page 4).

[11] G. Hoogsteen, A. Molderink, J. L. Hurink, G. J. M. Smit, B. Kootstra, and F. Schuring, "Charging electric vehicles, baking pizzas, and melting a fuse in lochem," *CIRED - Open Access Proceedings Journal*, vol. 2017, no. 1, pages 1629–1633, October 2017, doi: 10.1049/oap-cired.2017.0340. (Cited on page 4).

[12] University of Twente, "Energy in Twente | University of Twente." [Online]. Available: https://www.utwente.nl/ctit/energy/ Last accessed on 1 July 2018. (Cited on page 4).

[13] G. Hoogsteen, "DEMKit: a flexible smart grid simulation and demonstration platform written in Python," *International workshop Energy-Open*, pages 1–2, May 2017. (Cited on pages 4 and 10).

[14] M. E. T. Gerards, J. L. Hurink, and R. Hübner, "Demand side management in a field test: lessons learned," *CIRED - Open Access Proceedings Journal*, vol. 2017, pages 1678–1681(3), October 2017, doi: 10.1049/oap-cired.2017.1238. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/oap-cired.2017.1238 (Cited on page 4).

[15] P. van Oirsouw, *Netten voor distributie van elektriciteit*. Phase to Phase B.V., 2012, ISBN: 978-90-817-9831-0. (Cited on page 7).

[16] Icons made by Vectors Market from www.flaticon.com. (Cited on pages 8, 13, 21, and 24).

[17] G. Hoogsteen, "A cyber-physical systems perspective on decentralized energy management," Ph.D. dissertation, University of Twente, December 2017, ISBN 978-90-365-4432-0, doi: 10.3990/1.9789036544320. (Cited on pages 9, 10, 11, 13, and 32).

[18] A. Molderink, V. Bakker, J. L. Hurink, and G. J. M. Smit, "On indirect controlled cost functions based dsm strategies," *2013 IEEE Grenoble Conference*, June 2013, 10.1109/PTC.2013.6652464. (Cited on page 9).

[19] J. Kok, B. Roossien, P. Macdougall, O. V. Pruissen, G. Venekamp, R. Kamphuis, J. Laarakkers, and C. Warmer, "Dynamic pricing by scalable energy management systems - Field experiences and simulation results using PowerMatcher," *2012 IEEE Power and Energy Society General Meeting*, July 2012, doi: 10.1109/pesgm.2012.6345058. (Cited on page 9).

[20] J. Kok, "The PowerMatcher: Smart Coordination for the Smart Electricity Grid," Ph.D. dissertation, Vrije Universiteit Amsterdam, April 2013. (Cited on page 9).

[21] M. E. T. Gerards, H. A. Toersche, G. Hoogsteen, T. van der Klauw, J. L. Hurink, and G. J. M. Smit, "Demand side management using profile steering," in *2015 IEEE Eindhoven PowerTech*, June 2015, doi: 10.1109/PTC.2015.7232328. (Cited on page 9).

[22] T. van der Klauw, "Decentralized energy management with profile steering: Resource allocation problems in energy management," Ph.D. dissertation, University of Twente, May 2017, ISBN 978-90-365-4301-9, doi: 10.3990/1.9789036543019. (Cited on page 9).

[23] M. E. T. Gerards and J. L. Hurink, "Robust peak-shaving for a neighborhood with electric vehicles," *Energies*, vol. 9, no. 8, page 594, July 2016, doi: 10.3390/en9080594. (Cited on pages 9, 26, and 27).

[24] International Electrotechnical Commission (IEA), *Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 1: General requirements*, 2014, ISBN: 978-2-8322-1666-8. (Cited on page 14).

[25] International Electrotechnical Commission (IEA), *Electric vehicle conductive charging system - Part 1: General requirements*, 2017, ISBN: 978-28-322-3766-3. (Cited on page 14).

[26] JuicePoint, "EV Connectors." [Online]. Available: https://juicepoint.co.nz/knowledge-base/connectors/ Last accessed on 27 June 2018. (Cited on page 16).

[27] M. H. H. Schoot Uiterkamp, M. E. T. Gerards, and J. L. Hurink, "Fill-level prediction in online valley-filling algorithms for electric vehicle charging," 2018, accepted for 2018 IEEE PES ISGT Europe. (Cited on page 27).

[28] T. van der Klauw, M. E. T. Gerards, G. J. M. Smit, and J. L. Hurink, "Optimal scheduling of electrical vehicle charging under two types of steering signals," in *IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE Power & Energy Society, 10 2014, ISBN 978-1-4799-7720-8, doi: 10.1109/ISGTEurope.2014.7028746. (Cited on page 31).

[29] M. H. H. Schoot Uiterkamp, T. van der Klauw, M. E. T. Gerards, and J. L. Hurink, "Offline and online scheduling of electric vehicle charging with a minimum charging threshold," 2018, accepted for 2018 IEEE SmartGridComm. (Cited on page 32).

[30] G. Hoogsteen, A. Molderink, J. L. Hurink, and G. J. M. Smit, "Generation of flexible domestic load profiles to evaluate demand side management approaches," in *2016 IEEE International Energy Conference (ENERGYCON)*, April 2016, doi: 10.1109/ENERGYCON.2016.7513873. (Cited on page 35).

[31] InfluxData, "Influxdb - time series database monitoring & analytics." [Online]. Available: https://www.influxdata.com/ Last accessed on 5 July 2018. (Cited on page 47).

[32] K. Keller, "A day ahead electricity storage flexibility prediction for peak shaving," Master's thesis, University of Twente, September 2016. (Cited on page 49).

[33] S. Nykamp, T. Rott, K. Keller, and T. Knop, "Forecast the grid oriented battery operation to enable a multi-use approach and discussion of the regulatory framework," *CIRED - Open Access Proceedings Journal*, vol. 2017, no. 1, pages 2760–2763, 2017, doi: 10.1049/oap-cired.2017.0115. (Cited on pages 49 and 50).

[34] Solcast, "Solar radiation and estimated PV system power output data anywhere on earth." [Online]. Available: https://solcast.com.au/ Last accessed on 5 July 2018. (Cited on page 49).

[35] Fleetster, “Corporate car sharing software and fleet management.” [Online]. Available: https://www.fleetster.net/ Last accessed on 5 July 2018. (Cited on page 52).

[36] Opel, “Opel onstar | uw persoonlijke assistent.” [Online]. Available: http://www.opel.nl/onstar/index.html Last accessed on 5 July 2018. (Cited on page 54).

[37] Grzegorz Szostak, “Utility to get data from onstar service specially from myopel.” [Online]. Available: https://github.com/nyxnyx/onstar/tree/master/onstar Last accessed on 5 July 2018. (Cited on page 54).

Appendix A: Literature study

UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

On the realization of a smart grid demo-site at Coteq in Almelo

a literature study

Klaas Hoekstra

M.Sc. Thesis
Embedded Systems
April 2018

Supervisors:
dr.ir. M.E.T. Gerards (Marco)
prof. dr. ir. G.J.M. Smit (Gerard)
prof.dr. J.L. Hurink (Johann)
dr. ir. G. Hoogsteen (Gerwin)

Computer Architecture for Embedded Systems
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

The ongoing energy transition in the Dutch electricity grid may lead to grid capacity problems and unbalance between the production and consumption of electric energy. Renewable energy sources like photovoltaic panels are introduced in the grid in residential areas and cannot be controlled. On the other hand, the increasing amount of electric vehicles that have to be charged and the introduction of heat pumps to replace boilers, increase the amount of energy that is consumed significantly. But these new devices provide a certain amount of flexibility in when to consume energy. We need smart ways to control these devices. When controlling devices to tackle the mentioned problems, we speak of decentralized energy management in a smart grid.

One of the stakeholders interested in controlling devices in a smart grid through decentralized energy management is Coteq, a distribution service operator in the Netherlands. Coteq is responsible for the availability of the grid in parts of the Netherlands. To stimulate the development of smart grids, Coteq provides a demo-site containing PV panels, charging stations for electric vehicles and a stationary battery. This research focuses on the development of a smart grid implementation for the demo-site at Coteq.

In order to successfully realize a smart grid demo-site, we conducted a literature research that surveys different decentralized energy management approaches. Two classes of approaches are defined: control-based- and planning-based approaches. The first class only takes the current state of the grid into account to define its control actions, while the latter also looks at a prediction of future energy consumption and production. Besides that, we have also reviewed smart grid pilots that are completed successfully. Different lessons are learned from these smart grid pilots.

From the literature research, we conclude that it is attractive to use a planning-based decentralized energy management approach in the smart grid demo-site, which should decrease peaks in the grid and maximize the consumption of locally generated electricity. A promising approach we can use to control the charging of electric vehicles is the planning-based robust peak-shaving approach.

Contents

Chapter 1

Introduction

This literature research serves as an introduction to the final project of realizing a smart grid at the demo-site located at Coteq in Almelo. This introduction gives a motivation of why to realize such a demo-site, states the problems to overcome in order to realize the demo-site, by formulating the research questions, and gives an outline for the rest of the report.

1.1 Motivation

Multiple stakeholders are interested in the realization of the smart grid demo-site and all have their own motivation. Here, the social importance of this research, the motivation for Coteq, and the motivation for the University of Twente are discussed.

1.1.1 Energy transition

The energy distribution networks in the Netherlands consist of nationwide electricity- and gas distribution [1]. These networks play an important role in the energy supply chain. The electricity distribution network, simply called electricity grid or grid from now on, can be divided in a high voltage (HV), a medium voltage (MV), and a low voltage (LV) part, where the HV part of the grid is used to transport electricity over large distances. In the MV network, the voltage is stepped down, to connect local industry and large office buildings to the grid. Lastly, the LV network is used to provide availability of electricity in the residential area.

Within the electricity grid, a clear distinction between producers and consumers existed in the past. Producers are power generators that insert energy in the grid and are historically situated in the HV grid. Consumers draw energy from the grid and range from large industries on the HV grid to the residential users in the area of the LV grid. This supply chain was designed using a paradigm based upon centralized energy production using fossil fuels. Since the grid has no means of storing energy, the production and consumption of electricity in the grid must be balanced at all times.

In this design, assets in residential areas consume energy from the grid. This energy has to be transported to the residential area via the HV and MV grid. However, the grid has limited transportation capacity to transport the energy to the residential areas.

If we look at the total energy usage, we note that a large part of the total energy demand in the residential area comes from heating households. This heating demand is currently fulfilled mostly by boilers powered by natural gas and not by electric energy. This implies that the heating demand at this time does not put a large burden on the electricity grid.

However, the situation in the energy supply chain is quickly changing, on both the production and consumption side of the grid. This change in the energy supply chain is called the *energy transition* and consists of the introduction of renewable energy production and the electrification of the energy demand.

Renewables

In order to decrease the amount of fossil fuels used for electric energy production, energy produced by alternative renewable energy sources (RESs) is introduced in the grid. An important aspect of the energy transition is the introduction of this renewable energy. A large part of renewable energy comes from wind parks, directly connected to the HV grid. But next to the generation of electricity through wind, an increase is seen in the amount of renewable energy produced by distributed producers (RESs) in the LV grid. These RESs are mainly photovoltaic (PV) panels that produce electricity from solar energy. Various facts and figures of the energy usage in the Netherlands are given in [2]. For example, in 2016, 5.6% of the produced energy in the Netherlands originates from a renewable source. In 2016, all households with solar panels could deliver a total power of 1407 MW, which is significantly more than the 651 MW in 2014.

A characteristic of renewable energy is that most of its production is uncontrollable. This makes it challenging to balance the production and consumption of energy in the future electricity grid. When a household produces more energy than it consumes, it becomes a temporary producer in the electricity grid, leading to electricity flowing in reverse direction, while the current electricity grid was not designed for this purpose.

Electrification of the energy demand

Another part of the energy transition is the replacement of consuming assets that used non-renewable sources of energy (e.g., natural gas), by assets that use electricity as their source of energy. In this way they can consume energy generated by RESs, as most RESs produce electricity. This trend is called the electrification of the energy demand in the energy supply chain and results in a shift from the distribution of energy using other carriers, such as natural gas, towards electricity as energy carrier. Because of this, the electricity grid must be able to handle this extra load.

Two main assets are responsible for this shift in demand. First, modern heating systems for buildings in the residential area, such as heat pumps, use electricity to replace natural gas used by boilers. In 2016, 85% of households in the residential area were heated with gas boilers, and only 1.5% used electric heat pumps. But the amount of installed heat pumps increased with 30% compared to the previous year [2].

Secondly, the introduction of electric vehicles (EVs) causes a shift in demand. Although EVs are not very common yet, the amount of EVs in the Netherlands is growing significantly. In the beginning of 2017, 3% of the passenger cars in the Netherlands are (partly) powered by electricity, and 95% still use fossil fueled combustion engines for propulsion. Thus, the amount of EVs in the Netherlands is not significant yet, but when comparing to the year before, a relative increase of EVs in the Netherlands of 17% is observed [3]. It is expected that this increase continues in the coming years.

The introduction of EVs causes an increase of electrical energy demand in residential areas, because these vehicles have to be charged. Therefore, charging stations are distributed throughout the country. On average, charging a depleted EV battery on a daily basis requires two times more electricity than the daily consumption of an average Dutch household [4]. As a consequence, placing charging stations in the low voltage grid in the residential area results in a rapid increase of the energy consumption in this grid. This gets even worse when charging stations are used with a high output power, called fast chargers. They make that the peak load is expected to be even higher.

The simultaneity of EVs charging in the residential area causes another potential threat for the electricity grid, because most people start charging their EV at the same time. In a regular residential area this is after work. Furthermore, charging an EV also takes relatively long compared to an electric kettle for example.

To prevent the mentioned threats from happening, the load on the grid can be controlled. The new assets provide a certain amount of flexibility on the demand side of the grid. For example, the charging of an EV can be delayed to some extent. Since the production of renewable energy is uncontrollable, this demand side flexibility can be used to keep production and consumption in the electricity grid in balance. Resources that have a certain flexibility on the demand side are called distributed energy resources (DERs).

1.1.2 Interest Coteq

The problems caused by the energy transition that is happening in the Dutch electricity grid aks for a proper treatment by the involved stakeholders. The networking company Coteq is one such stakeholder and is interested in finding proper solutions. Networking companies in The Netherlands own and maintain electricity grid assets to facilitate the distribution of electric energy to consumers. There are two types of these networking companies: transmission system operators (TSOs), who are responsible for the transmission network (the HV grid), and distribution service operators (DSOs), who are responsible for the MV- and LV grids [1].

Coteq is one of the DSOs in the Netherlands and is thus responsible for connecting producers and consumers to MV- and LV grids in some parts of the Netherlands. To make sure this connection is not interrupted, constraints on minimum and maximum voltage in the grid, the deviation on the frequency and the power quality are defined. Also, the load in the LV grid is constrained by the maximum transformer rating, the current that can flow through conductors, and the rating of protective fuses in the grid.

The introduction of PV in the residential area causes households in the low voltage grid to produce electricity during peak sun hours, instead of consuming it. The load peaks in the grid caused by this production may even be higher than the maximum load constraints in the low voltage distribution grid, potentially causing transformers to overload and cables to overheat. Also, the increasing amount of EVs that are charged in the low voltage grid may cause stress and overloading, resulting in faster degradation of assets in the distribution grid, or even a blackout. Since the DSOs are responsible to keep the connections in the low voltage grid uninterrupted, it is of their interest to find methods and solutions to avoid severe ageing of grid assets, in order to provide a reliable grid to their customers.

1.1.3 Interest University of Twente

Part of the Computer Architecture for Embedded Systems (CAES) chair and the Discrete Mathematics and Mathematical Programming (DMMP) chair of the University of Twente have formed the Energy group [5]. The active research field of this group is on providing intelligent smart grid control concepts and energy management methodologies with a focus on residential (micro) grids. Previous research is in creating demand side planning algorithms for assets in the low voltage grid, and the development of a flexible smart grid simulation and demonstration platform, called DEMKit [6], [7].

The Energy group already has knowledge concerning smart grid technologies and optimization algorithms, and the DEMKit simulator allows for detailed simulations of grid assets in a smart grid environment. Implementations of smart grids have already been realized in a real-life environment by the Energy group, and this provides a good basis to take the development of smart grid sites to a next level.

This research is in the interest of the University of Twente, to gain knowledge in how to move from a simulated environment to a real-life demo site. Moreover, there is a vision to merge the controlling of a real smart grid into the DEMKit simulation platform, by using hardware in the loop techniques. This way, the development time to go from a simulated smart grid to a real and working smart grid decreases significantly.

1.2 Problem statement

The goal of this research is to realize a smart grid demo-site on the location of Coteq in Almelo. Multiple grid assets are already in place, like PV panels on the rooftop of the buildings, EV charging stations to charge both EVs from Coteq's pool of vehicles and EVs of employees, and a battery to store (locally) generated electricity. However, no logic is available to make these assets cooperate to create better grid conditions. Coteq wants to implement a smart grid to control these DERs and to provide a demo for other stakeholders to see the potential of smart grid systems. Besides that, the University of Twente wants to use this opportunity to extend their DEMKit simulator to enable a robust transition from a simulation environment to a real-life smart grid system, for many different cases and grid assets.

In order to realize this demo-site, a literature research is conducted to acquire knowledge on current research in the field of smart grids, decentralized energy management (DEM) approaches, and electric vehicle (EV) charging. Also, pilots that are already up and running are reviewed to gain information in setting up a smart grid demo-site. This leads to the following research question:

> *What can current literature on decentralized energy management (DEM) approaches and coordinated EV charging teach us about developing a smart grid demo-site for a distribution service operator (DSO), where the objectives for the DSO have to be met through controlling flexibility in EV charging?*

To answer this research question, the following sub-questions are formulated:

1. *What objectives can and should a smart grid demo-site realize for the DSO?*
2. *What decentralized energy management (DEM) algorithms have already been developed and how can they be implemented in the demo-site?*
3. *How can EV charging be used to control the flexibility in the system, to realize the objectives of the DSO?*
4. *What lessons can be learned from smart-grid pilots that are already running for some time and which of these lessons should be taken into account when developing the demo-site?*

1.3 Outline of the literature study

The remainder of this literature study is organized as follows. In Chapter 2, related work on DEM is described. Multiple DEM implementations are discussed, the objectives of a DSO are identified, and coordinated EV charging proposals are discussed. In Chapter 3, different smart grid and DEM pilots are presented. Lastly, conclusions and recommendations are given in Chapter 4.

Chapter 2

Decentralized energy management

In the old grid, supply would follow demand and thus only a few large producers had to be controlled in the grid to maintain the balance between production and consumption. Now, many small producers and flexible consumers are spread throughout the grid. The goal of decentralized energy management (DEM) is to control the increasing amount of DERs in a smart grid, in order to balance the consumption with the production by RESs [8]. DEM takes into account both energy supplying assets (e.g. PV) and energy demanding assets (e.g. EVs) in the grid. With the use of DEM, objectives set by the DSO can be realized. Hereby, a DSO may have different objectives which it tries to realize by the introduction of a smart grid. When reviewing different DEM approaches, the objectives for a smart grid from a DSO point-of-view are identified as well.

Two main approaches for DEM can be distinguished. Firstly, control-based DEM steer the system based on the current state and do not look ahead. The second class of approaches is planning-based DEM, which use predictions of the future to optimize the energy profile throughout the day. Some popular DEM implementations are discussed here.

2.1 Control-based DEM

A widely used control-based DEM approach is the auction-based powermatcher, as presented by Kok in [9], [10]. A powermatcher cluster is organised as a logical tree, where the root is the auctioneer agent and the leaves are formed by device agents. In such a cluster, local device agents, representing DERs, communicate with the auctioneer agent through optional concentrator agents. An example powermatcher cluster is shown in Figure 2.1.

In the first step of the powermatcher algorithm, all device agents define a bid function that states the amount of power a device is willing to buy or sell, dependent on the energy price. This can be the only actual market price for energy, but it can also be an arbitrary specified range. These bid functions are communicated to the concentrator agents, that aggregates the different bid functions to make the system scalable. Then, all aggregated bid functions are sent to the auctioneer agent, which selects an equilibrium price, or market clearing price. Lastly, this clearing price is communicated back to all device agents, that start consuming or producing power in accordance with the clearing price in their bid functions.

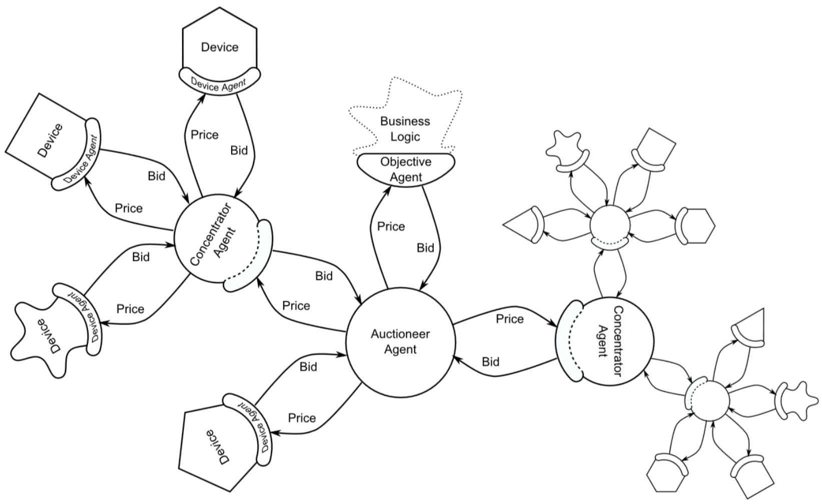


Figure 2.1: Example of a cluster of agents in the powermatcher [10]

Evaluation of DSO goals: Peaks in the grid

Since a control-based DEM approach does not take any predictions for future time periods into account, it may use the flexibility in the system too soon. As an example, if a battery in a house is controlled in combination with PV by a control-based DEM system, the battery may start charging as soon as more energy is produced by the PV than the house consumes. This may lead to the battery being full before the highest peak in PV production occurs. Thus, the high peak still has to be fed back through the grid, which may lead to grid capacity problems. This is already a problem in Germany [11].

Minimizing these peaks in the network is an objective of the DSO. Grid assets that are responsible for the connectivity of households to the grid, such as transformers or underground cables, can only sustain a certain amount of power and high peaks can exceed the maximum power rating of the assets. Also, these assets have a certain lifespan. This lifespan decreases when the amount of high peaks and large deviations in the load increase. The DSOs are responsible to keep these assets online, and therefore an objective of the DSOs is to minimize high peaks in the network and to keep the load on the assets as flat as possible over time. This is also called peak-shaving.

2.2 Planning-based DEM

The second class of DEM approaches are the planning based approaches. These approaches use predictions of the future to optimize the control. Five different planning-based DEM approaches are introduced here. Also, a comparative analysis of research on coordinated EV charging using different management strategies, including DEM, is discussed.

2.2.1 Profile steering

A planning-based DEM approach is the profile steering approach as presented in [6], [12]. In the initial phase of profile steering, a central controller asks all connected devices to create a profile of their expected energy production or consumption for an upcoming period. Such a profile is a vector of discrete levels of power consumption (or production) in time. Different methods to construct such an initial profile can be used. For example, a device can decide to start consuming energy as soon as possible, or it can try to flatten out its profile as much as possible.

After the initial phase, the iterative phase starts, in which the central controller aggregates the profiles of all devices to create a total power profile. The central controller has a certain system objective, for example load flattening. Based on the difference between the system objective and the received aggregated power profile, the controller asks each device to create an (updated) candidate profile.

When the central controller has received all these updated candidate profiles, it decides which device gives the best improvement towards the system objective when the new candidate profile is used instead of the original profile. Only this device is asked to update its profile to the new candidate profile. The choice that only one device updates its profile is to prevent oscillations in the aggregated profile. The process is iterated until the total power profile is close enough to the system objective of the central controller.

The profile steering approach is made scalable by the introduction of intermediate layers between the devices and central controller. An intermediate controller aggregates the profiles of its devices and send this to the central controller. It uses the profile steering approach to ask its devices for profiles as the central controller would do. Therefore, each device only has to predict its own profile and schedule its flexibility, and the aggregation of energy profiles can be done decentralized.

Evaluation of DSO goals: Consumption of locally generated energy

The system objective of the central controller can be load flattening, or using energy when it is cheapest. Another system objective is to maximize the consumption of locally generated energy. In order to do this, the controller creates an objective profile where it schedules energy consumption on moments that a lot of local energy generation is expected. Consuming locally generated energy is beneficial because one does not have to buy energy from the grid.

The latter system objective is also an objective for the DSO in a smart grid. If this energy is not used by the producer, it has to be transported through the grid. If there is a lot of production, the transport of this energy can stress or reduce the lifespan of grid assets for which the DSO is responsible. Also, when energy is needed later, it has to be transported back again, meaning that the assets are stressed once more. Summarizing, when the generated energy is consumed locally, the transport of energy can be reduced significantly.

2.2.2 Robust peak-shaving

A planning-based DEM implementation that does not require detailed predictions is presented in [13], where it is proven that a robust, peak-shaving DEM approach can be designed, using only a few, easy to predict characteristics. In the approach, an algorithm controls a group of EVs, both to avoid transformer peaks and to keep house profiles as flat as possible.

The algorithm uses a so-called valley-filling approach, which fills the "valleys" in a profile to a fill level Z, so that the amount to be charged is reached exactly at the deadline. Figure 2.2 demonstrates the algorithm for an EV that arrives at 18:00 and has to be completely filled at 07:00.

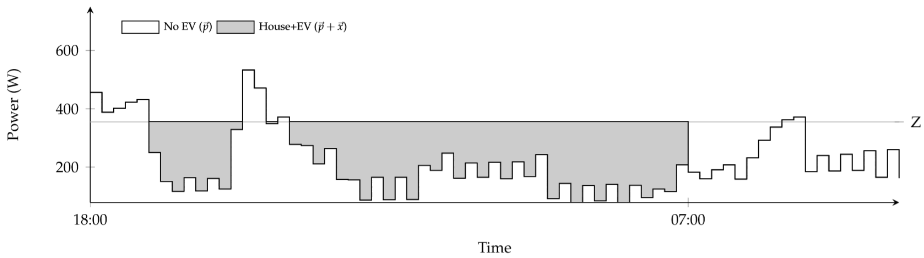


Figure 2.2: Optimal EV planning for peak shaving using a fill-level approach [13]

The fill level Z is dependent on the shapes of the so-called "valleys" in the energy profile, the amount of energy to be charged, and when the charging should be finished (the deadline). In the solution presented in [13], an approach to find the fill level Z is proposed which does not know the exact power consumption of the entire day. An algorithm is presented that predicts the optimal Z based on historic data, and also evenly spreads out the prediction error over all charging intervals, to avoid peaks in the energy profiles. It uses only a prediction of the fill level characteristic and the load of the next time interval as an input.

The approach is also extended to a neighbourhood level, where all houses first make a flat profile using the aforementioned algorithm. When the total power profile exceeds a certain threshold, the houses reduce their charging rate while locally maintaining an as flat as possible profile. Only a prediction on the number of intervals where the EV are charged is needed for this approach.

2.2.3 TRIANA

A third planning-based DEM approach called TRIANA is presented in [14]. The authors present a DEM methodology that makes use of three steps: prediction, planning, and real-time control. In the TRIANA concept, the first step is that all devices make a prediction of their power consumption or generation, to identify the basic behaviour of a cluster. To optimize the power consumption, all devices receive steering signals from a central controller in the next step. These steering signals are vectors of discrete values, which represent prices that vary over time. The devices adapt their power consumption profiles with the goal of minimizing the local cost, based on the steering signals. The newly predicted profile is communicated back to the central controller and this process is iterated until a desired global profile is achieved.

In the last step, TRIANA controls the appliances based on the planned profile. This approach depends on local predictions on a device level, which implies that accurate prediction have to be made and that the system has to deal with prediction errors.

2.2.4 Intelligator

DEM is used to control DERs in a smart grid. An example of a DER is an EV charging station, also called electric vehicle supply equipment (EVSE). When multiple EVSEs are present in a smart grid, coordinated charging strategies can be used to optimize the charging schedule of all EVSEs on an aggregated level.

An approach for coordinated EV charging using the Intelligator algorithm is proposed by Leemput et al. [15], where a case study of coordinated EV charging for peak shaving in a low voltage grid is considered. In the research, a real grid topology is simulated using 15-minute intervals, with realistic cable lengths as shown in Figure 2.3. Real household electricity consumption profiles of 62 households in Flanders are used, that were measured on a fifteen-minute basis in the LINEAR project [16]. Also, artificial PV has been added to 30 of the simulated houses, where the PV profiles are measured at an installation of KU Leuven and scaled to match the PV generation to the load of the selected household. Lastly, 30 households received an EV, with a charging profile based on the Flemish mobility behaviour [17].

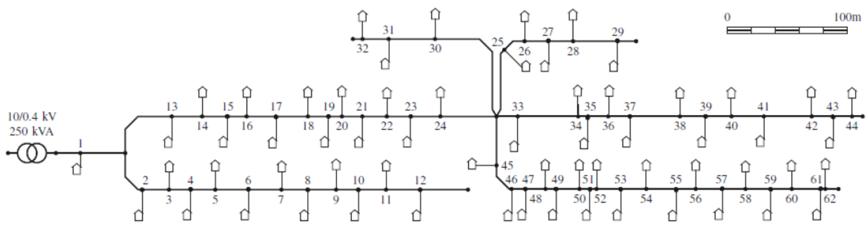


Figure 2.3: Grid topology used for simulating coordinated EV charging [15]

The authors first simulated the setup without any form of coordinated EV charging. Here, the EVs charge as soon as they arrive home, until the battery is completely recharged. To gain knowledge from the simulation, a grid impact analysis is done, where it is studied what percentage of time the voltage at each household stays within the allowed band of 10% over or under the nominal voltage. The results show that for the households furthest away from the transformer, less than 5% of the simulations result in voltage deviations and an unbalance between the phases that exceed the allowed limits in Flanders.

A coordination algorithm for EV charging, called Intelligator [18] is used to address this.

This algorithm is a multi-agent market-based algorithm, based on the PowerMatcher [9]. Every DER in Intelligator has an agent that places priority bid functions to a higher agent. A priority bid function tells the amount of power a DER is willing to consume for different priorities. Note that in this implementation, not the real market electricity price is used, but an arbitrary priority scale. At the top level all bids are aggregated, and a clearing priority is selected based on the balance between consumption and production, while keeping in mind local restraints.

The grid impact analysis of the coordinated simulation shows that unbalance and deviations in the voltage at the households furthest away from the transformer are reduced compared to the uncoordinated charging simulation. As there is almost no correlation between PV power production and EV availability, the coordination algorithm does not avoid negative power demand due to a high amount of PV production. The research concludes that their online algorithm does a successful job in peak shaving, using only limited data. Moreover, it has a positive impact on the voltage deviations at the households furthest from the transformer.

2.2.5 EV charging control mechanism respecting transformer limits

A market based multi-agent control algorithm to control the charging of a fleet of EVs that respects the transformer's power and voltage limits is presented in [19]. Similar to the Intelligator [18], this algorithm is based on the PowerMatcher, where device agents communicate with substation agents, which in their turn communicate with a single auctioneer agent. Moreover, the same grid topology and EV arrival and departure times from the LINEAR project in Flanders, as used in [15], are used for the simulations presented by the authors. Figure 2.4 shows the hierarchy of the proposed algorithm.

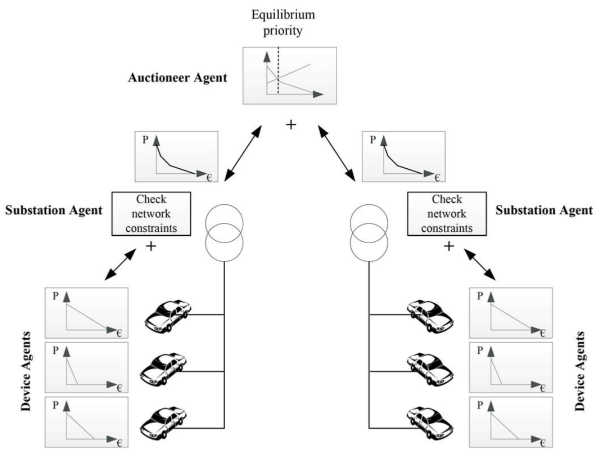


Figure 2.4: Information hierarchy of the algorithm proposed in [19]

In the first part of their research, the authors prove that the PowerMatcher heuristic can be interpreted as a utility maximization problem. It is assumed that each substation agent coordinates a subset of all the EV device agents connected to the system. Every device agent communicates its bid function to its substation agent, which has to solve the utility maximization problem. The price that is chosen by the auctioneer agent to charge to the customers is called the equilibrium price. For every possible equilibrium price, the substation agent aggregates all the bids of its device agents. Then, each substation agent sends the consumption of its cluster for every possible equilibrium price to the auctioneer agent. The auctioneer agent then determines the market clearing price using the aggregated bids of the substation agents. This equilibrium price is then sent back to the device agents, so that they know the amount of power to consume.

In the second part of the research, the authors extend the heuristic with transformer and voltage constraints. The constraints are added to the utility maximization problem. Moreover, the authors present a method to include these constraints by only changing the substation agent. The device agents and auctioneer agent are not changed compared to the existing market based control algorithm. For every possible equilibrium price, the substation agents solve the utility optimization problem with constraints. To reduce computation complexity, the authors suggest to first determine the prices that possibly cause network constraint problems with a load flow analysis. The optimization problem with constraints is then only solved for the problematic prices.

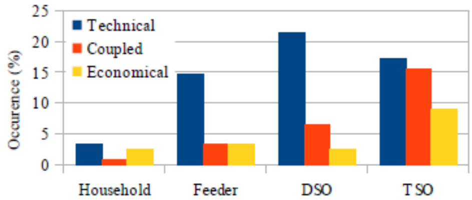
The authors compare their implementation to other utility maximization algorithms for smart grids in the literature. They state that most other implementations use an iterative exchange of messages to find the optimal solution. In every iteration, the central agent broadcasts a price and the device agents answer with their planned consumption for the price. The iterations continue until convergence of the price takes place. An advantage of the proposed method is a significant reduction in communication requirements, because no iteration is required. This method results in an immediate convergence. However, a disadvantage of the method proposed by the authors is the computational requirement. Since the substation agents solve the utility optimization problem with constraints for every possible equilibrium price, the problem is also solved for equilibrium prices that are far from optimal and will never be used in practice.

In the simulation, the authors assume that each household has an EV with a maximum charge capacity of 3.3kW. Three-day simulations are done for a regular transformer, and a transformer with a reduced capacity of 125 kVA. It shows that the peaks due to simultaneous charging of EVs are limited because of the transformer constraints. The EVs are charged at minimal cost, and do not harm the grid.

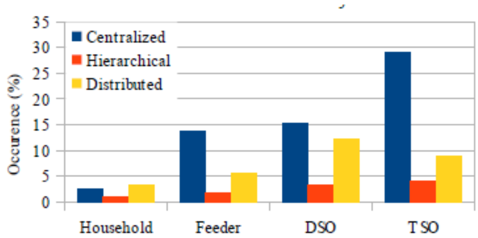
2.2.6 Comparative analysis of research on coordinated EV charging

In [20], Leemput et al. present a comparative analysis of coordination strategies for charging EVs. Three parameters are used to characterize and compare different researches on coordinated EV charging: the research objective, the place in the grid that is controlled, and the method of controlling. The research objective is categorized in impact analysis, scenario analysis, benchmark, and a complete system. The place in the grid is categorized in a household, a feeder, DSO level, and TSO level. Lastly, the method of controlling is categorized in a centralized approach, decentralized approach, and a hierarchical approach. Also, the field of research is distinguished. This is categorized in technical, economical, or a combination of the two. Correlation mappings between the parameters and the research category are presented. This gives a clear insight in what research is done in the field of coordinated EV charging.

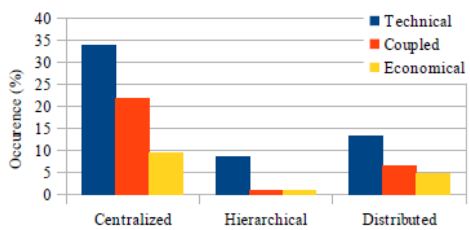
In the results, six correlation mappings are presented. In the mappings, the percentage of researches that focuses on what parameter category is presented. For example, as Figure 2.5a shows that most researches focus on the technical aspect of coordinated charging, and the least on a purely economical research category. Also, most of the technical researches are on a feeder or DSO scale, and most of the researches that focus on both the technical and economical aspect, on a TSO scale. Figure 2.5b shows that most researches make use of a centralized approach on all scales except for the household scale. Here, a distributed approach is most popular. Lastly, Figure 2.5c shows that the technical objective is used most, followed by the coupled objective and lastly the economical objective.



(a) Correlation mapping a



(b) Correlation mapping b



(c) Correlation mapping c

Figure 2.5: Correlation mappings [20]

Chapter 3

Smart grid Pilots

Multiple DEM approaches have been discussed in Chapter 2. In the DEM approaches, algorithms are provided or simulations show the working of the approach. However, pilots and field tests are needed to gain knowledge on how to implement DEM in a real-life situation. In the past, different pilots already have been successfully finished, and some of the results are discussed here.

3.1 Demand response in Yokohama

Honda et al. present the experiences of demand response (DR) in a demonstration project in Yokohama in [21]. It involves residential households, buildings, factories, and EVs in Yokohama. In this project, the load reduced through DR, compared to the expected power consumption, is called *negawatt*.

In the demonstration project, the community energy management system (CEMS) monitors the local power grid, and sends DR requests to the Integrated building energy management system (BEMS) when it is needed. The request is then sent from the Integrated BEMS to the customers, to ask them to reduce the power load. After the customers reduced their load, the Integrated-BEMS receives information about the amount of reduced load through the DR action, called *negawatt*. An overview of the setup can be seen in Figure 3.1.

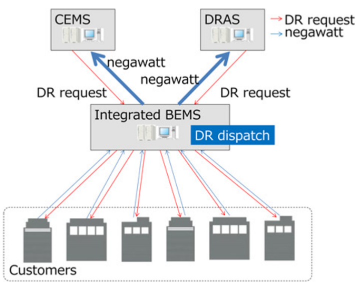


Figure 3.1: Overview of the DR setup used in Yokohama [21]

In the research, the authors focus on the last stage of a three-stage experiment, where the capability of fast-DR is evaluated. In fast-DR, customers respond to DR requests within fifteen minutes or one hour. In the first stage of the experiment, the capability of shaving or shifting customer's peak load was evaluated. The results showed significantly different DR capabilities for each customer. However, in the first stage it was not possible to verify and control whether the amount of adjustment was in accordance with the target of each customer. Therefore, in the second stage, each customer's amount of adjustment was evaluated. The results show that six of ten customers reduced more electricity demand than targeted, but two customers could not successfully reduce their demand. To overcome this, the capability of Fast-DR in the form of one hour ahead DR and fifteen minute ahead DR was evaluated in the third stage.

For the third stage, all fourteen customer were contracted with a DR programme concerning not only the amount of load reduction, but also the response time and duration. Different types of customers were contracted, such as factories, facilities and more. Not all customers were capable of doing Fast-DR; some could only do day-ahead DR. In the period of July 2014 up to January 2015, in total 51 DR events were sent to the contracted customers. In the results of the research it is shown that the success rate for Fast-DR in response to these events was 79.4% and for Day-ahead-DR 73.7%. The success rate represents the percentage of successful reductions in response to DR events. From the results it can be concluded that the success rate of Fast-DR is higher than for Day-ahead-DR.

3.2 Merygrid mircogrid in Belgium

In [22], the authors describe a microgrid pilot in Belgium. Three companies are connected in the Merygrid microgrid, which can operate in an islanded mode, or be connected to the main grid. In total 60 kWp of PV and 200 kVA of run-of-the-river turbines in the microgrid provide generation. For the pilot, a smart energy management system (EMS) is designed that optimizes the microgrid operation and its interaction with the main grid. A goal of the project is to design a storage system in the microgrid. In the research, data from the microgrid pilot is used and the proposed storage system is tested in a simulation environment.

This smart EMS contains a controller, as shown in Figure 3.2. In the controller, the microgrid management is split into two parts: an operational planning controller and a real-time controller. The first, the operational planning controller, consists of three modules. Module one builds forecasting models to estimate the generation and consumption of assets in the microgrid, using historical data and external information. Moreover, it generates a prediction of different scenarios, representing probable evolutions in the microgrid.

The second module uses these scenarios to determine a schedule to control the assets in the microgrid for the coming hours. For every 15 minutes, an optimization problem is solved for the different scenarios. The goal is to optimise a schedule by finding a trade-off between multiple goals in the microgrid. Receding horizon control is used to solve this optimization problem [23]. In receding horizon control, the module receives a sequence

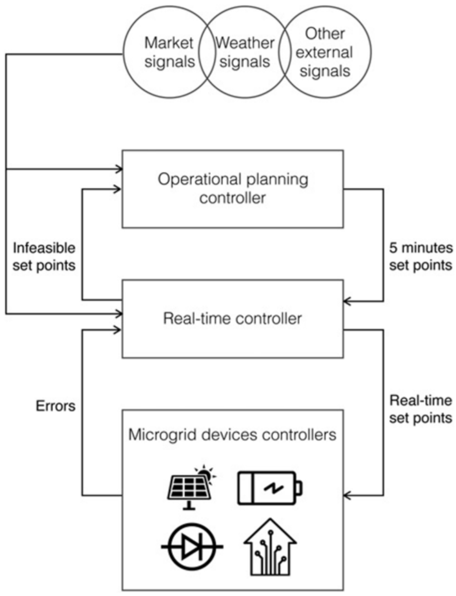


Figure 3.2: Overview of the smart EMS design [22]

of actions and determines the corresponding evolution of the state of the microgrid. Every scenario represents a different sequence of actions and is called a 'horizon'. Actions that can be taken are using flexibility in the battery, consumer devices, or the provisioning of reserve. The module determines the horizon with the highest reward; where it best represents its goal. The last module uses feedback from the real-time controller to estimate the state of the microgrid (i.e., the state of charge (SoC) of the battery), so it can determine if the schedule is correct.

The real-time controller solves a problem similar to the receding horizon optimisation in the second module of the operational planning controller, but at a higher frequency, and using a more detailed model of the system. The real-time controller can control DERs in the microgrid. A model predictive control (MPC) framework is used to make sure that the difference between the set points from the operational planning controller and the actual state of the microgrid is minimal, while still satisfying the constraints in the microgrid.

In the results of the research a simulation of the actual microgrid is done, instead of implementing the smart EMS in the microgrid. In this simulation, some flexibility is declared in the grid in the start time of motors, a fridge, and an oven. Also, a battery is connected, which can inject energy into the main grid when the price is high. The simulations show that the designed EMS is capable of injecting energy into the main grid when the price is relatively high.

3.3 TRIANA pilot in Germany

In [24], the lessons learned from the SmartOperator demand-side management (DSM) field test project of RWE Deutschland AG are presented. The DSM approach in this field test uses the three steps of the TRIANA concept: prediction, planning and real-time control [25]. A home energy controller (HEC) based on Innogy SmartHome [26] is installed in every house, and a centralised controller coordinates the decisions on a neighbourhood level. The TRIANA software is added to the HEC and is used to steer DERs based on the local PV production, other measurements, and signals from the centralised controller.

The centralised controller starts by sending five steering signals (profiles) to the HECs. One of this steering signals instructs the HEC to maximise the self-consumption of the house, and the other four are artificial energy price profiles that influence the house load profile to prevent issues on a neighbourhood level. How the HEC uses the flexibility is determined using the three steps of TRIANA. First, a prediction of the loads and available flexibility in the house is made for a 24h timespan. Based on this, five energy profiles for this timespan are generated in the second step, one for every steering signal from the centralised controller. These energy profiles are sent back to this centralised controller, which chooses the profile that has the best effect on the power grid. In the third and last step, TRIANA controls the appliances to realize the selected plan. A main challenge of this method is to make good predictions in the HEC.

In the two years this field test has been running, seven lessons can be learned for how to successfully implement a DSM field test. Lesson one focuses on the fact that peaks from time-based pricing can be mitigated by keeping them in mind in the planning step. The second lesson states that compensation should be introduced, since some stakeholder always have less undesirable results than others due to prediction errors. In lesson three, it is learned that controlling a house can be simplified if a buffered DER can control its operation to a desired load, or based on a fill-level approach. The authors conclude in the fourth lesson that near-optimal decisions can be made when the prediction for the first time interval and for the overall energy volume is accurate. Lesson five states that ”good, frequent and synchronised measurements and control are needed”. Also, in lesson six, it is learned that collecting data from all connected devices on a central location provides early evaluation and improvements in a field test. Lastly, lesson seven focuses on the fact that a DSM should be designed in such a way that it can handle missing data, changing circumstances and possible downtime of a part of the system.

Chapter 4

Conclusions and recommendations

For this literature study, related work on different subjects was studied. Different decentralized energy management (DEM) approaches were reviewed, and the use of DEM for coordinated EV charging was studied. Also, active and completed smart grid pilots were reviewed. Conclusions for all three subjects are stated in Section 4.1, followed by recommendations in Section 4.2.

4.1 Conclusions

In the literature on DEM approaches, two main classes of approaches are distinguished: control-based DEM approaches and a planning-based DEM approaches. The first class of approaches do not look ahead and only base their control decisions on the current state of the system. It has been discussed that such a system can still lead to grid-stressing situations, since it uses the flexibility of the system in a non optimal way. Therefore, planning-based DEM approaches are more attractive to use in smart grid demo-sites. Depending on what information is available to use as inputs for the algorithm, the most suitable planning-based DEM approach has to be chosen.

Next to the discussed DEM approaches, two objectives that a smart grid should realize for a DSO were identified. Firstly, the peak-shaving objective is identified to prevent a decrease of lifespan of grid assets that the DSO is responsible for. Also peaks in the grid can be higher than the maximum power rating of the grid assets, so by spreading these peaks through peak-shaving, the grid does not have to be reinforced. Secondly, an objective for the DSO is to maximize the consumption of locally generated electricity. If energy is consumed where and when it is produced, it does not have to be transported through the grid.

Since at the demo-site at Coteq EV charging stations are present, multiple coordinated EV charging algorithms have been discussed. As can be seen from the comparative analysis, many research focuses on centralized approaches and most research is done from a technical point of view. Since a DSO is not allowed to influence energy prices in the energy market, economical approaches are out of scope for the demo-site at Coteq. Therefore, the development of this demo-site focuses on the technological aspect. Since the University of Twente is developing hierarchical DEM approaches, this research focuses on a hierarchical approach as well. Because the Intelligator approach makes use of priority bid functions instead of market prices, it may be possible to use it in a DSO environment. However, the robust peak-shaving method seems promising to control the charging of EVs, for little information is needed as an input and peak shaving is a main objective for the DSO.

When looking into smart grid pilots, many pilots work in collaboration between different stakeholders, as the Yokohama DR pilot in Japan. However, since this pilot relies on its customers to respond to DR requests themselves, it is not a viable solution to use in the demo-site that is to be developed. The research on the connected microgrid in Belgium shows that splitting the global daily planning and the real-time control of a system can help to maintain a balance in a microgrid. However, since the demo-site at Coteq is always connected to the main grid and does not operate in an islanded mode, it is less suitable for this research. Some of the lessons learned in the TRIANA pilot are good to take into account on beforehand, but not all lessons are suitable for the smart grid demo-site at Coteq. An example of a suitable lesson is ”lesson seven”, which shows that a robust design has to be developed that can cope with downtime of (part of) the system.

4.2 Recommendations

From the conclusions above, several recommendations are defined. Firstly, in order to find the best DEM algorithm for the smart grid demo-site at Coteq, the available information that can be used as input for the algorithm has to be defined. Secondly, for the development of the DEM system, it should be determined whether the robust peak-shaving algorithm can easily be implemented for control of the charging of the EVs connected to the electric vehicle supply equipment (EVSE). Also, an important step in the realization of the demo-site is to develop a way to move from the simulation environment to a real demo-site. A final recommendation is to look into using DEMKit [7] as a way to make this transition from the simulation environment.

List of acronyms

BEMS	building energy management system
CAES	Computer Architecture for Embedded Systems
CEMS	community energy management system
DER	distributed energy resource
DEM	decentralized energy management
DMMP	Discrete Mathematics and Mathematical Programming
DR	demand response
DSM	demand-side management
DSO	distribution service operator
EMS	energy management system
EV	electric vehicle
EVSE	electric vehicle supply equipment
HEC	home energy controller
HV	high voltage
MV	medium voltage
MPC	model predictive control
LV	low voltage
PV	photovoltaic
RES	renewable energy source
SoC	state of charge
TSO	transmission system operator

Bibliography

[1] EnergiePortal, “Overzicht netbeheerders elektriciteit en gas.” [Online]. Available: https://www.energieportal.nl/onderwerpen/netbeheerders Last accessed on 13 March 2018. (Cited on pages 1 and 3).

[2] J. Gerdes, S. Marbus, and M. Boelhouwer, “Energietrends 2016,” Energieonderzoek Centrum Nederland, September 2016, last accessed on 13 March 2018. (Cited on page 2).

[3] CBS, “Jaarmonitor wegvoertuigen: Aantallen.” [Online]. Available: https://www.cbs.nl/nl-nl/achtergrond/2017/21/jaarmonitor-wegvoertuigen-aantallen Last accessed on 13 March 2018. (Cited on page 3).

[4] M. H. H. Schoot Uiterkamp, “Robust planning of electric vehicle charging,” Master’s thesis, University of Twente, August 2016. (Cited on page 3).

[5] University of Twente, “Energy in Twente — University of Twente.” [Online]. Available: https://www.utwente.nl/ctit/energy/ Last accessed on 13 March 2018. (Cited on page 4).

[6] M. E. T. Gerards, H. A. Toersche, G. Hoogsteen, T. van der Klauw, J. L. Hurink, and G. J. M. Smit, “Demand side management using profile steering,” in *2015 IEEE Eindhoven PowerTech*, June 2015, doi: 10.1109/PTC.2015.7232328. (Cited on pages 4 and 8).

[7] G. Hoogsteen, “Demkit: a flexible smart grid simulation and demonstration platform written in python,” *International workshop Energy-Open*, pages 1–2, May 2017. (Cited on pages 4 and 19).

[8] G. Hoogsteen, “A cyber-physical systems perspective on decentralized energy management,” Ph.D. dissertation, University of Twente, December 2017, ISBN 978-90-365-4432-0, doi: 10.3990/1.9789036544320. (Cited on page 6).

[9] J. Kok, B. Roossien, P. Macdougall, O. V. Pruissen, G. Venekamp, R. Kamphuis, J. Laarakkers, and C. Warmer, “Dynamic pricing by scalable energy management systems - Field experiences and simulation results using PowerMatcher,” *2012 IEEE Power and Energy Society General Meeting*, July 2012, doi: 10.1109/pesgm.2012.6345058. (Cited on pages 6 and 11).

[10] J. Kok, “The PowerMatcher: Smart Coordination for the Smart Electricity Grid,” Ph.D. dissertation, Vrije Universiteit Amsterdam, April 2013. (Cited on pages 6 and 7).

[11] S. Nykamp, T. Rott, N. Dettke, and S. Kueppers, “The project ”ElChe” Wettringen: storage as an alternative to grid reinforcements - experiences, benefits and challenges from a DSO point of,” in *International ETG Congress 2015; Die Energiewende - Blueprints for the new energy age*. (Cited on page 7).

[12] T. van der Klauw, “Decentralized energy management with profile steering: Resource allocation problems in energy management,” Ph.D. dissertation, University of Twente, May 2017, ISBN 978-90-365-4301-9, doi: 10.3990/1.9789036543019. (Cited on page 8).

[13] M. E. T. Gerards and J. L. Hurink, “Robust peak-shaving for a neighborhood with electric vehicles,” *Energies*, vol. 9, no. 8, page 594, July 2016, doi: 10.3390/en9080594. (Cited on page 9).

[14] A. Molderink, V. Bakker, M. Bosman, J. L. Hurink, and G. J. M. Smit, “Management and control of domestic smart grid technology,” *IEEE transactions on smart grid*, vol. 1, no. 2, pages 109–119, September 2010, doi: 10.1109/TSG.2010.2055904. (Cited on page 9).

[15] N. Leemput, F. Geth, B. Claessens, J. V. Roy, R. Ponnette, and J. Driesen, “A case study of coordinated electric vehicle charging for peak shaving on a low voltage grid,” in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, October 2012, doi: 10.1109/ISGTEurope.2012.6465656. (Cited on pages 10 and 11).

[16] E. Peeters, C. Develder, J. Das, J. Driesen, and R. Belmans, “LINEAR: towards a breakthrough of smart grids in Flanders,” in *i-SUP 2010 : Innovation for Sustainable Production, Proceedings*, 2010, pages 3–6. (Cited on page 10).

[17] J. Roy, N. Leemput, B. S, F. Geth, P. Tant, and J. Driesen, “An availability analysis and energy consumption model for a flemish fleet of electric vehicles,” in *European Electric Vehicle Congress (EEVC), Brussels, Belgium*, October 2011. (Cited on page 10).

[18] M. Ghijsen and R. D’hulst, “Market-based coordinated charging of electric vehicles on the low-voltage distribution grid,” in *2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, October 2011, doi: 10.1109/SGMS.2011.6089021. (Cited on pages 10 and 11).

[19] S. Weckx, R. Dhulst, B. Claessens, and J. Driesensam, “Multiagent Charging of Electric Vehicles Respecting Distribution Transformer Loading and Voltage Limits,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pages 2857–2867(9), November 2014, doi: 10.1109/tsg.2014.2345886. (Cited on page 11).

[20] N. Leemput, J. V. Roy, F. Geth, P. Tant, B. Claessens, and J. Driesen, "Comparative analysis of coordination strategies for electric vehicles," in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*, December 2011, doi: 10.1109/ISGTEurope.2011.6162778. (Cited on pages 12 and 13).

[21] K. Honda, K. Kusakiyo, S. Matsuzawa, M. Kosakada, and Y. Miyazaki, "Experiences of demand response in yokohama demonstration project," *CIRED - Open Access Proceedings Journal*, vol. 2017, pages 1759–1762(3), October 2017, doi: 10.1049/oap-cired.2017.0789. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/oap-cired.2017.0789 (Cited on page 14).

[22] B. Cornélusse, D. Ernst, L. Warichet, and W. Legros, "Efficient management of a connected microgrid in Belgium," *CIRED - Open Access Proceedings Journal*, vol. 2017, pages 1729–1732(3), October 2017, doi: 10.1049/oap-cired.2017.0211. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/oap-cired.2017.0211 (Cited on pages 15 and 16).

[23] H. S. Bidgoli, M. Glavic, and T. V. Cutsem, "Receding-horizon control of distributed generation to correct voltage or thermal violations and track desired schedules," in *2016 Power Systems Computation Conference (PSCC)*, June 2016, doi: 10.1109/PSCC.2016.7540818. (Cited on page 15).

[24] M. E. T. Gerards, J. L. Hurink, and R. Hbner, "Demand side management in a field test: lessons learned," *CIRED - Open Access Proceedings Journal*, vol. 2017, pages 1678–1681(3), October 2017, doi: 10.1049/oap-cired.2017.1238. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/oap-cired.2017.1238 (Cited on page 17).

[25] V. Bakker, A. Molderink, J. Hurink, G. Smit, S. Nykamp, and J. Reinelt, "Controlling and optimizing of energy streams in local buildings in a field test," in *22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013)*, June 2013. (Cited on page 17).

[26] Innogy SmartHome, "Ein perfekter Tag in Ihrem Leben mit innogy SmartHome." [Online]. Available: https://www.innogy.com/web/cms/de/3759188/home/ Last accessed on 13 March 2018. (Cited on page 17).