CARL VON OSSIETZKY universität OLDENBURG

Deutsches Zentrum für Luft und Raumfahrt - Institut für vernetzte Energiesysteme

Praxis-Seminar Modellierungsstudie

Comparison of Two Optimization Methods for Operating a Smart Home Power Grid

Author
Tjark Smalla

Professor
PD. Jan Freund

Supervisor
Dr. Peter Klement
Dr. Patrick Schönfeldt
Stefan Arens

June 2, 2020

Contents

List of Figures

[1] Mixed Integer Linear Programming
[2] Open Energy Modelling Framework
[3] Photovoltaic

Acronyms

DER	Distributed Energy Resource
PV	Photovoltaic
MILP	Mixed Integer Linear Programming
OEMOF	Open Energy Modelling Framework
DLR	Deutsches Institut für Luft und Raumfahrt
SD	Supply/Demand
SDM	Supply and Demand Matcher
COP	Coefficient of Performance

Chapter 1

Introduction

According to a report of the European Comission 26,7% of total energy consumption is caused by private households [7]. Due to the introduction of DER[1]s, private households are able to some of their demand by themselves. However, power generation from renewables and household demand usually do not occur at the same time. To account for this, energy can be stored in e.g. batteries or used to fulfill demands in different sectors like the heating sector. Scheduling power consumption and generation of connected devices (including batteries) in a home network is one of the requirements of the so called Demand Side Management.

Different methods to create these schedules already exist. One way to differ between them is by their scheduling horizon. The longer the scheduling horizon, the higher the uncertainties in the underlying predictions from e.g. the PV forecast[2].

In the following report, two methods for creating such a schedule will be analyzed and their performance will be compared. The first method is usually used for scheduling horizons greater than or equal to one day. In contrast to the first, the second method uses near real time scheduling. The simulations are performed by the OEMOF and the HAL python frameworks. Section 1.1 and 1.2 introduce both methods in detail. Chapter 2 explains in depth the tested home network (2.2), the used time series (2.3), the performance metrics (2.4) and the analyzed test setups (2.5). In the last two chapters 3 and 4, results are analyzed and conclusions about the properties of each method are made.

1.1 OEMOF in Detail

In the first method, an energy system is modeled as an MILP problem. This is then solved by one of the most efficient general purpose MILP solvers today, the branch-and-

[1]Distributed Energy Resource

cut algorithm[6]. An optimal energy system control schedule for the given initial conditions can be extracted from the result of the optimization.

Although iterative applications of this method exist, it is usually used for scheduling horizons greater or equal to one day[2][7].

The OEMOF framework provides the user with a set of components to model an energy system. Section 2.1 describes in more detail which subset of those components are used in the comparison. The online documentation provides a complete list of all available components. [1] The power flow is simulated along the edges of a bipartite graph which contains the before mentioned components as nodes.

Although the whole set of rules to generate the MILP problem from the component graph can not be covered by this report, the reader should get an intuition reading section 1.1.1. For further research, the official paper is recommended. [3]

1.1.1 Generating a MILP Problem from OEMOF Components

A MILP problem is made up of a set of linear inequalities and linear equalities A, and an objective function $c^T x$. The problem aims to find the solution vector x which minimizes the objective function while fulfilling $Ax \leq b$, $x \geq 0$ and $x_j \in \mathbb{Z}, \forall j \in J$. As a consequence of the last term, the solution vector x can be, but is not limited, to integers.

In the modeled energy system x holds the state of the system (e.g. power flow between two nodes or a battery charge) at a certain point in time t. The physical constraints are represented by A and b. An example for such a constraint would be the maximal power flow allowed between two components[2]. Besides user generated constraints, OEMOF introduces some static constraints inherent to an energy system. An example would be the requirement to have net power flow of 0 in the whole system (no energy flows out or into the system).

The objective function seen in equation 1.1 is made up of costs c which are generated by the (sometimes time dependent) states of the edges and nodes of the graph. An example node in the graph could be a power plant which generates costs according to the amount of power fed into the network at a certain time step of the simulation. Such an optimization state of a component is in the following referred to as degree of freedom of the component. Equation 1.2 provides an intuition how the simulation time has a significant impact on matrix and vector size of the problem and thus, on hardware requirements for the simulation. The cost function is given by

$$min : \sum_{t \in T} e(t) + \sum_{t \in T} n(t) + \sum_{(s,e) \in E} \sum_{i \in I_2} c^i_{(s,e)} \cdot w^i_{(s,e)} + \sum_{n \in N} \sum_{i \in I_4} c^i_n \cdot v^i_n \tag{1.1}$$

[2]Limited for example by the cable diameter

$$n(t) = \sum_{n \in N} \sum_{i \in I_1} c_n^i(t) \cdot v_n^i(t) \cdot \tau \tag{1.2}$$

$$e(t) = \sum_{(s,e) \in E} \sum_{i \in I_1} c_{(s,e)}^i(t) \cdot w_{(s,e)}^i(t) \cdot \tau \tag{1.3}$$

where w and v are the state of edges and nodes, N and E are the sets of nodes and edged, I_i are different cost sets, T is the simulation time and τ the time between to time steps.

1.2 HAL in Detail

The algorithm called HAL was developed at the DLR[3] Institute of Networked Energy Systems by Stefan Arens. It implements a market based control concept for supply and demand matching utilizing a continuous japanese auction. A similar concept was already proven to smoothen the net import profile of an energy system and is thus a promising method for controlling decentralized energy systems with high share of distributed generation[4]. Although, theoretically the auction can be performed in arbitrary intervals, it was designed and tested for short scheduling horizons. For this report, an interval of one minute is chosen.

HAL users model their energy system by building a tree structure out of components. HAL distinguishes between two types of nodes: SDM[4] and devices. A SDM node performs a japanese auction on its child nodes. A device node implements a SD[5] function modeled after the physical features of the device it is supposed to represent. Section 1.2.1 explains the auction algorithm in more detail. The only requirement imposed on the implemented SD functions is monotonicity. This ensures that the SDM can control the market to either generate or consume more power, depending on the overall market situation. The children of an SDM can be devices but also other SDM. SDM at the bottom of the tree resolve sub markets first before communicating the resulting sub market demand to their parent SDM. Section 1.2.1 explains in more detail how the matchers try to achieve a target in their sub market. If each matcher achieves its goal, the energy system is in a favorable state in which the necessary net power import is zero. [7]

[3]Deutsches Institut für Luft und Raumfahrt
[4]Supply and Demand Matcher
[5]Supply/Demand

1.2.1 Auction Algorithm

In each time step, each SDM in the tree performs an auction in their sub market. The sub market is controlled by a virtual price signal $p \in [0, 100]$ on which the matchers children c react with a proposal of how much power $q \in \mathbb{R}$ they consume or generate according to their supply and demand function $q_c(p)$. A negative quantity indicates generation rather than consumption.

At the start of each auction in each time step t, the matcher sends a neutral price signal of $p_0 = 50$ to all of its children. They react with the quantity of power they would consume or generate for this price. The matcher then alters its price signal for the next iteration to get closer to its target of $q = 0$. Because of the monotonicity of all involved SD functions, it is safe to assume that consumption will increase when the price is lowered. For clarification, equation 1.4 displays the building rule applied in each iteration to create the next price signal where q_i is the sum of all supply/demand proposals received from the matchers children nodes c.

$$\begin{aligned} q_i &= \sum_{c \in C} q_c(p_i) \\ p_{i+1}^t &= p_i + s_i^t \\ s_i^t &= \begin{cases} 1, & \text{if } q_i \geq 0 \\ -1, & \text{else} \end{cases} \end{aligned} \tag{1.4}$$

The auction is closed as soon as

$$q_i = 0 \vee |q_i| > |q_{i-1}| \vee p \in \{0, 100\} \tag{1.5}$$

holds. When the auction is closed and $q_i \neq 0$, the last iteration increased the distance to the target and thus p_{i-1} is considered the optimal price for this timestep.

Chapter 2

Methodology

This chapter provides an overview about the test setup used to compare both algorithms explained in section 1.1 and 1.2. For both algorithms, an energy system has to be modeled. In order to compare them, these systems have to be as similar as possible. Section 2.1 discusses how both systems are modeled to ensure comparability despite the different approaches of the algorithms used. Based on these actions, section 2.2 gives an overview about the simulated energy system which uses time series briefly explained in section 2.3. The performance metrics used for the comparison are explained in section 2.4. Finally, section 2.5 explains the different test setups tested.

2.1 Ensuring Model Comparability

To ensure comparability of both systems, the components used in both frameworks were required to have a counterpart in the other framework. The only detail in which they are allowed to differ is the usage of the degree of freedom introduced in section 1.1.1. A list with all used components and their implementations in both frameworks is given in 2.1.1. To ensure that both simulations run with the same parameters at all time, the system under test is modeled in OEMOF and then converted into a HAL model with the same parameters. For implementation details please refer to the *hal_plugin* folder in the appended python code base.

The OEMOF framework optimizes costs whereas HAL optimizes on a net import/export of zero. In order to force OEMOF to optimize the net import as well, imported and exported power from the modeled system creates costs. Both components are further described in section 2.2.

2.1.1 Used components

This sections lists all used abstract components and their respective implementations in both frameworks.

Consumer and Generators

Consumer and Generators are components which either draw or feed power. Although in theory they could, in the context of this report they have no degree of freedom for the simulation. They are used to provide time dependent power consumption/generation.

- Name in OEMOF: Sink, Source
- Name in HAL: InelasticBehavior
- Parameter: Power consumption/generation(t)
- Degrees of Freedom: -

Storage

In one time step storages can consume energy which they later provide in another time step. In the test system, they are used as simple models for batteries as well as for thermal energy storages. However, a storage can only store energy up to its capacity c^{max}. The amount of energy stored in a storage is denoted by c. it is governed by

$$c_t = min(c^{max}, c_{t-1} + \eta\tau(P_t^{in} - P_t^{out})) \tag{2.1}$$

with the conversion factor η, the simulation interval τ and the power flow p. The amount of energy the storage can produce/consume in one time step is limited by p^{max}.

- Name in OEMOF: Generic Storage
- Name in HAL: Battery
- Parameters: η, c^{max}, P^{max}
- Degrees of Freedom: P_t^{in}, P_t^{out}

Transformer

A transformer is able to connect two energy sectors. It is used as a highly simplified model for sector coupling between power sector and thermal sector. Such a coupling is often achieved by a heat pump. A heat pump usually produces more thermal energy and it consumed of electric energy. This relation is given by its COP[1]. To further model the properties of a heat pump, energy flow is only allowed from electrical to thermal energy.

In HAL this is implemented by a SDM which optimizes the amount of power generated by itself instead of a net import/export of zero.

The transformed energy is calculated by $E_t^{transformed} = P_t^{in} \cdot COP \cdot \tau$.

- Name in OEMOF: Transformer
- Name in HAL: Parent Transformer
- Parameters: COP
- Degrees of Freedom: P_t^{in}

2.2 Tested Energy System

The tested energy system is supposed to represent an average household in the north of Germany. Power is generated by a PV system and consumed by common activities of the residents referred to as the household load. The power system is connected to a local thermal energy system by a heat pump. The thermal energy system itself also consists of a thermal storage and a thermal load.

2.3 Time Series

The time series used for the simulation are either provided by the DLR or generated by the Load Profile Generator[2]. The Load Profile Generator simulates peoples activities triggered by their daily needs. Each activity can consume power or thermal energy as warm water. It was used to create the power load profile for the simulated household. The following is a comprehensive list of each used data source in the simulation:

- 2016_pred_res60s_7days_train_full.csv - Predicted output of a PV system in Oldenburg generated by the algorithm described in [5]

[1] Coefficient of Performance
[2] https://www.loadprofilegenerator.de/

- 2016_real_res60s_7days_train_full.csv - Actual PV output for the in 2016_pred_res60s_7days_ predicted system
- ol_data_pv_az180_inc36_mod23.csv - Actual PV output of an Oldenburg located PV system.
- sh_intensity=15_area=46.5.csv - Thermal load based on degree days. Section 2.3.1 describes this in more detail.
- SumProfiles.Electricity_processed.csv - Power load generated by the Load Profile Generator

Each time series can be found in the appended codebase in the folder *hal_plugin/data*.

In order to additionally analyze the influence of sector coupling on algorithm performance, both algorithms are tested in two different time periods. The simulation duration was set for both periods to one month. Figure 2.1 shows that available data limits the possible simulation periods to September 2016 to February 2017. In order to analyze the impact of sector coupling on the result, two periods were chosen. One containing no thermal load and high PV output and the other containing high thermal load and low PV output.

2.3.1 Thermal Load Model

The thermal load was calculated using the degree day method. A degree day dd is calculated by

$$max(0, T_b - T_t) * \tau = dd \tag{2.2}$$

where T_r is the base temperature of the considered building, T_t is the outside air temperature at time t and τ is the time interval in days.

The power consumption per degree day p^{dd} is then calculated by

$$p^{dd} = p^{year} / \sum_{t \in Y} dd_t \tag{2.3}$$

with Y being all degree day datapoints and p^{year} being the total power consumed for heating during the period of the year .

2.4 Relevant Performance Metrics

As explained in chapter 1, a good schedule reduces the net power import and peak loads. A net power import is related to costs for the household. Further, all power imported from

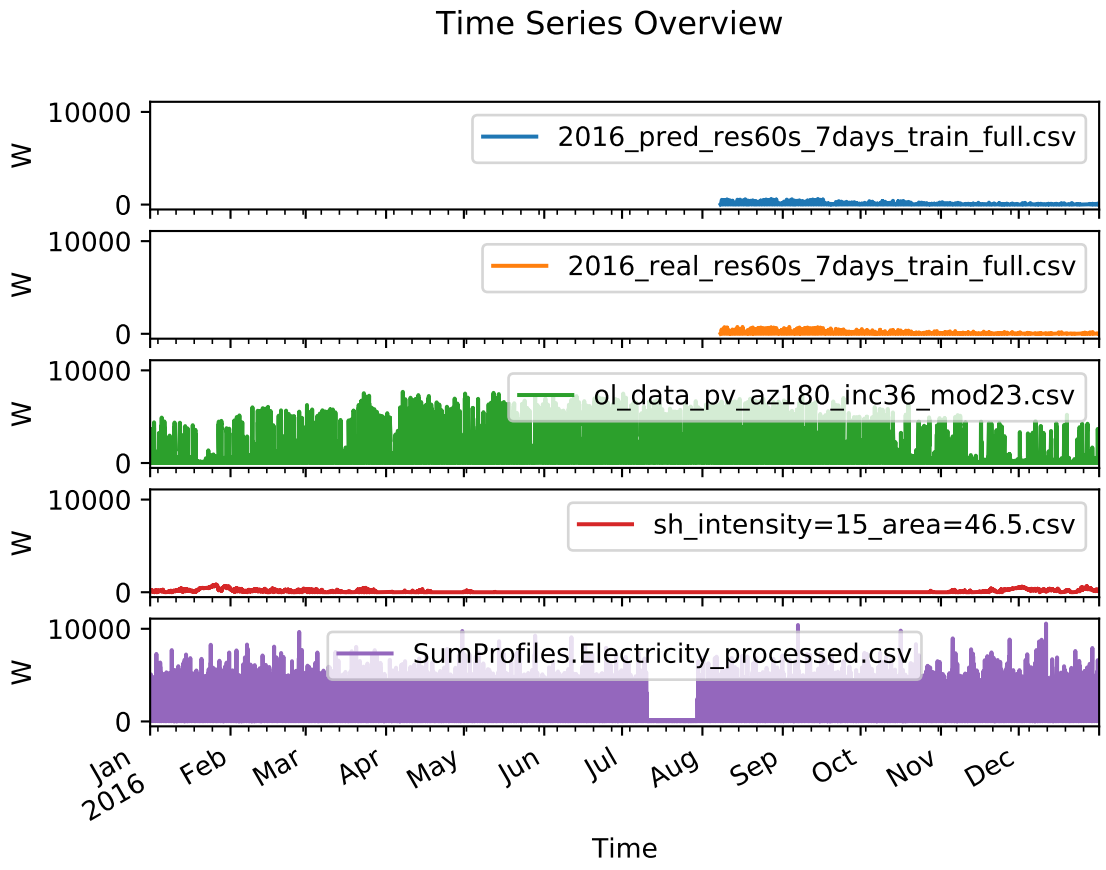


Figure 2.1: Overview of used time series with energy consumption/generation in Watts. Due to missing data, feasible simulation periods are only September to December. September allows for simulations without sector coupling effects, because the thermal load is constantly zero.

the grid leads to pressure on the infrastructure. A high net import should thus be avoided. Because of this, the main performance metric is the net power import over the whole simulation time. Besides the total power import, abundance and intensity are indicators for the pressure put onto the grids infrastructure while controlling the system according to the generated schedules.

2.5 Simulation Setup

To allow algorithm comparability, a python plugin which converts an OEMOF energy system into a HAL configuration was developed. Starting with a simple power network consisting only of consumers/generators and a storage, the plugin was further extended to allow for more complex simulations like the simulated sector coupling.

To create a baseline for the following simulations, a simple simulation is done using the energy system described in 2.2 and the data described in 2.3.

The file ol_data_pv_az180_inc36_mod23.csv is used as input for the PV system if not mentioned otherwise. Both algorithms performances are compared according to the metrics defined in section 2.4.

One of the main research question is the dependence of OEMOF generated schedules on its scheduling horizon. By slicing OEMOFs optimization problem into multiple smaller problems, the schedule horizon can be shortened. In the following, this setup is referred to as incremental setup.

To analyze both algorithms dependence on the correctness of the models input data, three additional setups are simulated. In each setup, the baseline simulation is performed with both algorithms using a certain set of parameters. Section 2.6 explains how an energy system control schedule is generated from this first baseline simulation. This schedule is then used to control the OEMOF energy system in a second simulation. Except for the PV systems output data and the schedule controlled component behavior, the second simulation equals the first. Because of HALs short schedule horizon of one minute, it is not used to create a control schedule, but as an online system controller in the first and the second simulation.

For the second simulation setup, noise is induced into the PV data of the second simulation step. The noises standard deviation is then increased to analyze a potential correlation between schedule performance and input disparity during schedule creation and application. In the following this simulation setup is referred to as noise setup. For each standard deviation, an ensemble of 10 simulation runs is performed to mitigate random effects.

In the third setup, the baseline simulation from which the schedule is created, is based

on PV prediction data. The schedule is then used to control a system which is fed by the actual PV data for the same time period the prediction was made for. HAL is used both times as an online controller. In the following, this setup is referred to as real data setup.

The fourth setup equals the third, but HAL is not used as an online controller. Instead, the system in the second simulation step is controlled by a schedule. This way both algorithm's ability to create a performant schedule for offline system controlling is analyzed. In the following this setup is referred to as schedule vs. schedule setup.

2.6 Creating a Schedule from Simulation Output

Output of both algorithms is the power consumption/generation of each device at each time step of the simulation. In a second simulation step the devices can be controlled by this information to mimic their behavior from the previous run.

The second simulation step is also performed by using the OEMOF framework. Offline system control is achieved by replacing the degrees of freedom of the simulation graph by the fixed behavior from the previous simulation run. A storage for example is replaced by a source node and a sink node which are configured to feed or draw power according to the storage behavior of the first simulation run.

2.6.1 Noise Induced PV Data

Equation 2.4 shows how the noise induced PV power output $p_i^{noise}(t)$ consists of the sum of the time dependent PV data from $ol\_data\_pv\_az180\_inc36\_mod23.csv$ $p(t)$ and the noise, generated by an Ohrnstein Uhlenbeck process ou with a mean of zero, a fixed relaxation parameter τ and a given standard deviation σ_i. The output power is clipped at zero and the maximal possible power output of the PV system p^{max}.

$$p_i^{noise}(t) = min(p^{max}, max(0, p(t) + ou(t, 0, \sigma_i, \tau))) \tag{2.4}$$

In comparison with OEMOFs long term schedule, HALs performance is expected to increase with increasing difference between PV data for schedule creation $p(t)$ and PV data during system controlling $p_i^{noise}(t)$.

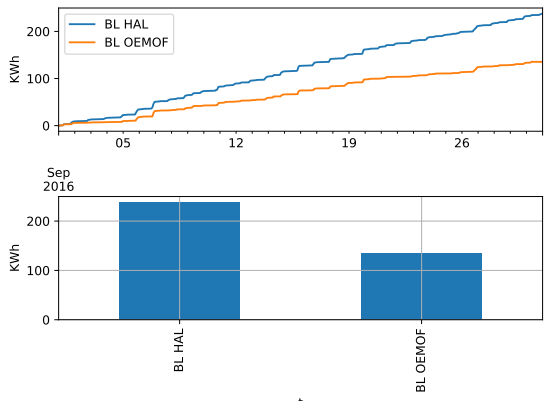
Chapter 3

Results

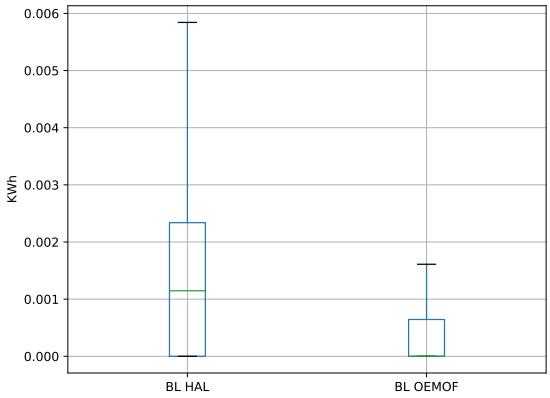
All results discussed in this chapter are based on the simulation output which can be found in the appended codebase in the folder *result_data*.

3.1 Baseline Results

The results of the baseline setup discussed in 2.5 are displayed in 3.1 a). As expected, OEMOFs overall net import is with only 69% of HALs total net import much lower. Although, the maximum net import peak of 0.174018kW caused by HAL and 0.173485kW caused by OEMOF is similar, analyzing figure 3.1 *b*) reveals that peaks in OEMOFs simulation results are not as frequent as in HALs results.



(a) Above: Cummulated Net Import in kWh. Below: Total Net Import in kWh.



(b) Net Import Peak Distribution. Whiskers are 1.5 of the inter quantile range.

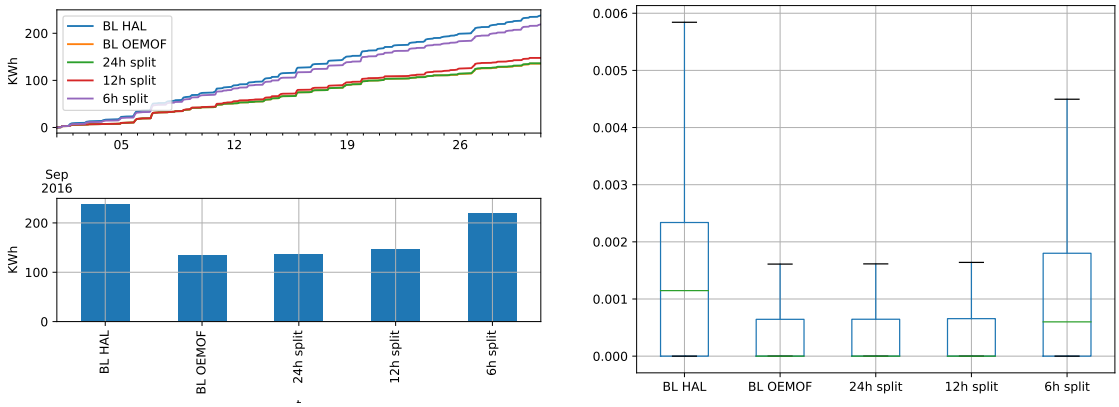
Figure 3.1: The MILP optimization with the OEMOF framework caused less power import. If power is imported, the amount of power is usually small compared to HALs power imports.

3.2 Incremental Setup

In the first simulation step a baseline simulation for the simulation period of September was performed. In a second, third and fourth step, OEMOFs scheduling horizon was reduced to 24, 12 and 6 hours. In the following, the net power import of all runs was compared.

Figure 3.2 shows that splitting the optimization problems increases the imported power amount and the occurrence of net power peaks. However, even the 6 hour OEMOF optimization split performs better than controlling the energy system with the HAL algorithm.

The 1 day split with a net import of 136.26kWh performs almost identical as the simulation without split and a net import of 135.44kWh. Analyzing the behavior of the optimization problem in detail, this is likely because the only degree of freedom in these setup is the battery. In the evening, the household load increases and the PV system generates no power. As a result the battery is used as energy producer during the evening until depleted. Thus each day the system starts with an empty battery.



(a) Above: Cummulated Net Import in kWh. Below: Total Net Import in kWh.

(b) Net Import Peak Distribution. Whiskers are 1.5 of the inter quantile range.

Figure 3.2: Performance of OEMOF optimized simulations decreases with short scheduling horizons. The biggest performance difference is between the 12h and 6h split. However, the 6h split performs better than the baseline simulation with the HAL algorithm.

3.3 Noise Setup

In order to analyze the dependence of a schedule controlled energy system on the correctness of the input data used for schedule creation, the performance of a schedule on differently strong noised PV input data was tested. Figure 3.3 shows how the net import of the schedule controlled systems increases with increasing noise variance. The stronger the deviation between input data used for schedule creation and input data during system control the bigger the necessary net import.

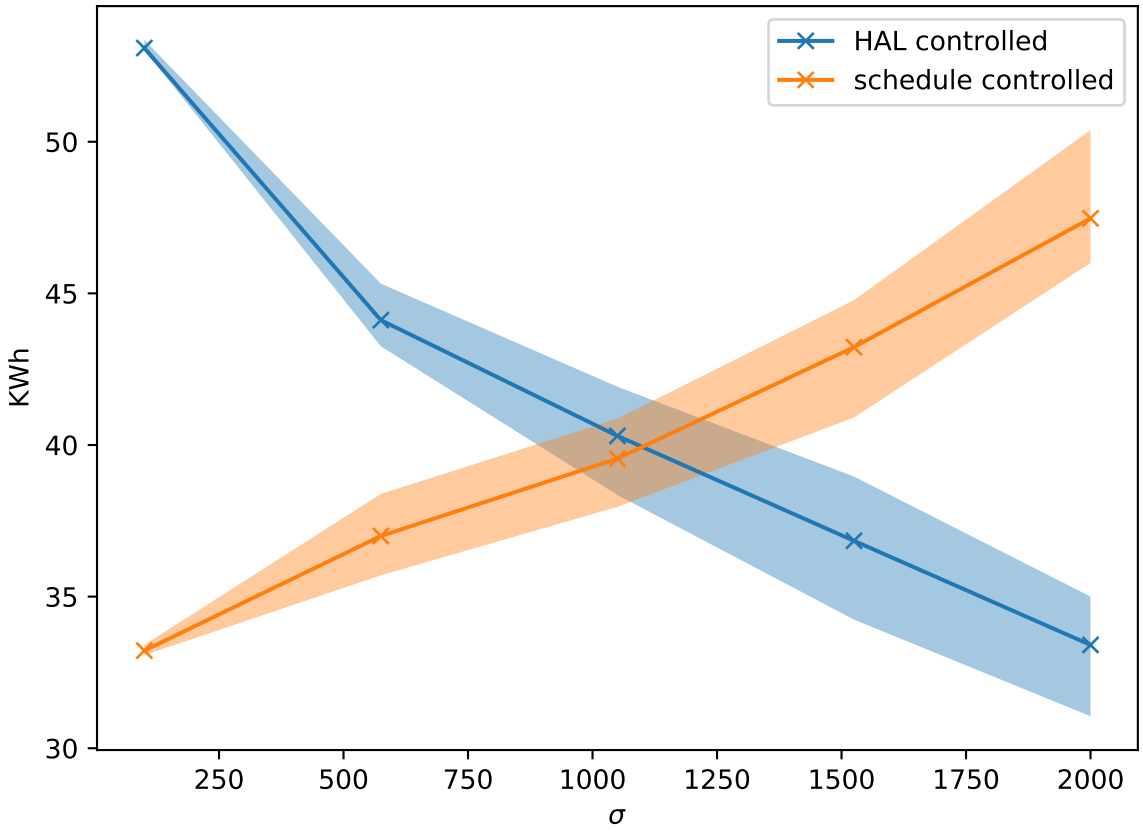


Figure 3.3: Total net import for different noise standart deviations σ. The areas mark the 25 and 75 percentile of the ensemble runs. The solid lines mark the median of the ensemble runs. HALs import seems to be negatively correlated to σ while OEMOFs import rises with increasing *sigma*.

Another interesting observation is the negative correlation between noise variance and the net import of the energy system controlled by HAL. This is likely caused by the additive noise. The resulting PV output consists of the sum of the generated noise and real PV data. As a result the PV system produces power even during the night when in reality

no sun would shine. The almost real time control behavior of HAL allows to use this additional power while the schedule controlled system does not expect and consume power in these times. Figure 3.4 shows how the PV output in the morning and evening is used by the HAL algorithm to load the storage.

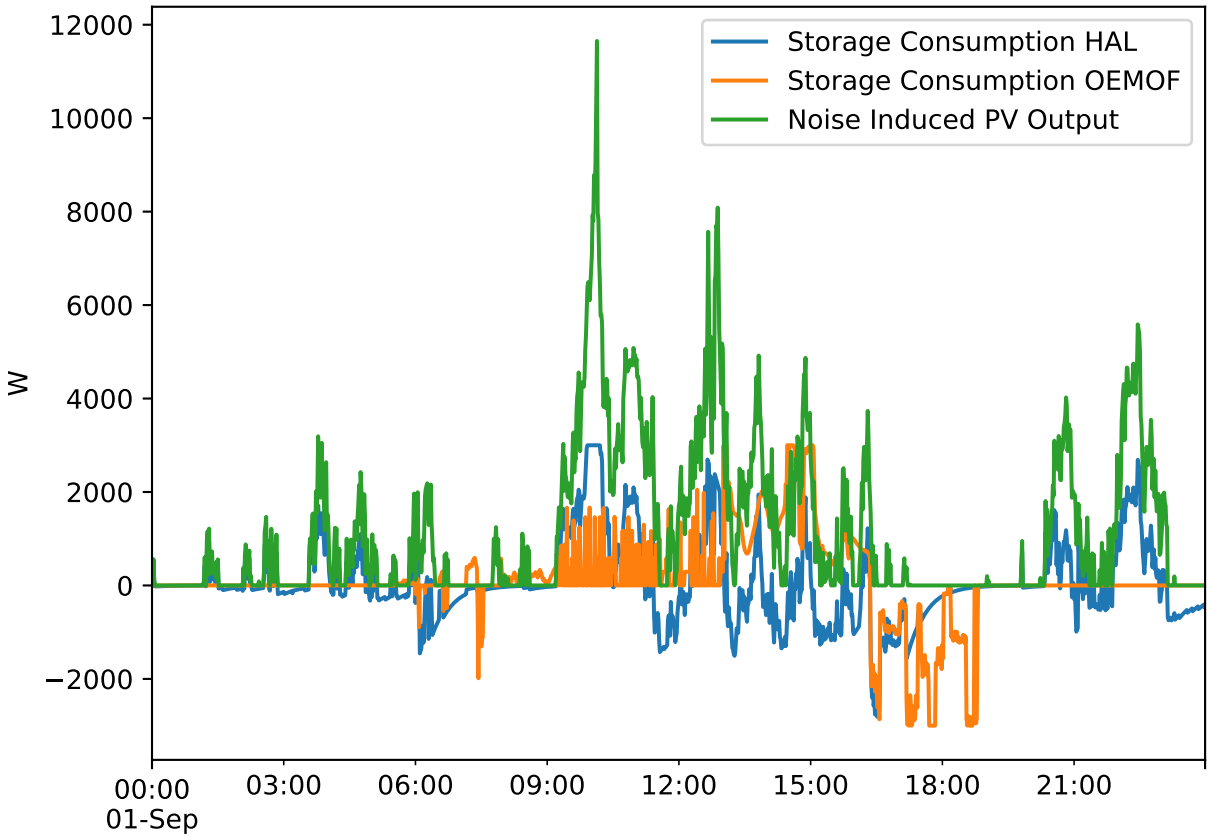
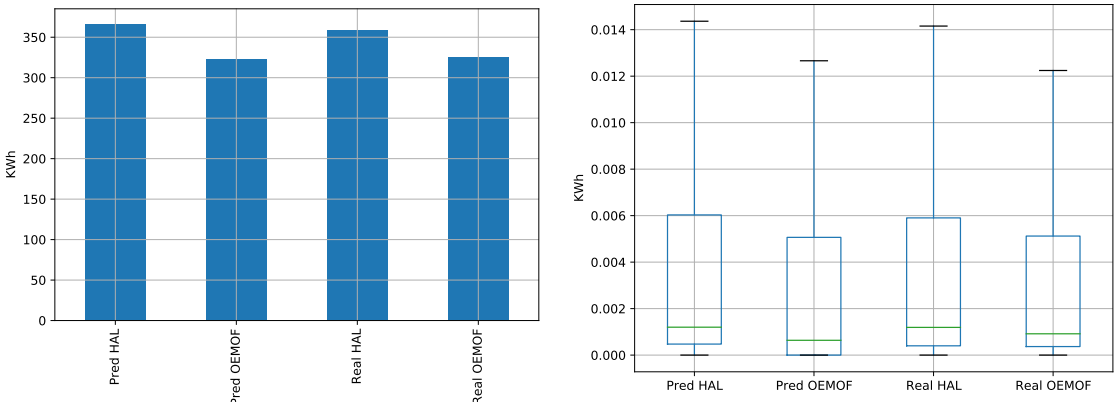


Figure 3.4: Power Flow during the first day of one ensemble run with $\tau = 30$ and $\sigma = 2000$.

3.4 Real Data Setup

The Noise Setup analyzed in 3.3 suggested that the HAL algorithm can adapt better to uncertainties in the input PV data and thus achieve better performance than a schedule controlled approach. However, the model used for creating these uncertainties was far from a realistic situation. For a more realistic approach, a data pair of actual PV data and its prediction was used in the real data setup. Although not as strong as in the noise setup, the same effects can be seen. Figure 3.5 shows that the HAL algorithm improves its performance against the schedule controlled simulation during the September period. In the HAL controlled approach, the simulation performed with real PV data achieved 2.16% less net power import. In contrast, the schedule controlled approach requires 0.76% more net import. The peak analysis shows also a slight improvement in the real data run with the HAL algorithm and a slight decline with the OEMOF created schedule.



(a) Total Net Import.

(b) Net Import Peak Distribution. Whiskers are 1.5 of the inter quantile range.

Figure 3.5: Simulation results with real PV and prediction data for September. OEMOFs first simulation run performs slightly better than the second which is controlled by a schedule. HAL is improving in the second run. HAL is used in both runs as a near realtime controller.

As in the noise setup, the HAL algorithm can react to unforeseen changes and load the battery when PV output is unexpectedly high. Besides the optical indication in figure 3.6, the correlation between PV output and battery power consumption and generation is listed in table 3.1. The HAL controlled storage consumption correlates more with the actual PV data than with the predicted one. In the schedule controlled simulation the situation is flipped.

Table 3.1: Correlation between PV input and storage consumption

	HAL	Schedule	Pred PV	Actual PV
HAL controlled Storage Balance	1.00	0.38	0.21	0.38
Schedule controlled Storage Balance	0.38	1.00	0.34	0.29
Pred PV Output	0.21	0.34	1.00	0.80
Actual PV Output	0.38	0.29	0.80	1.00

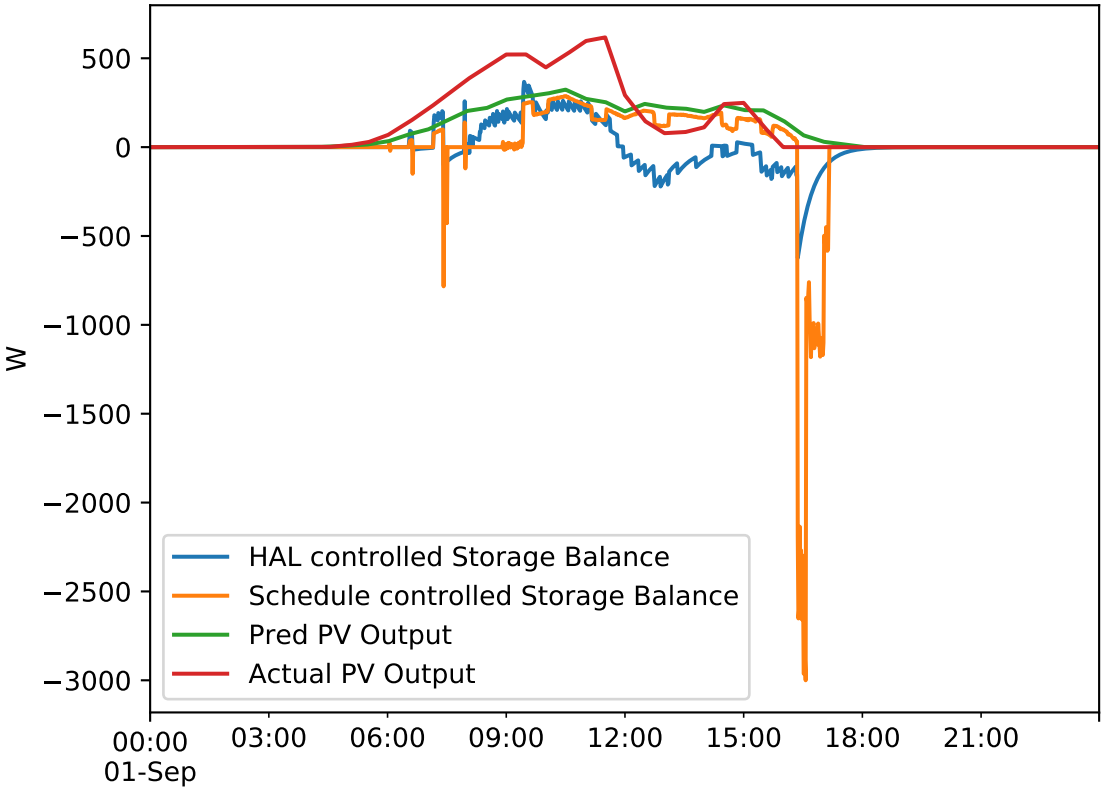


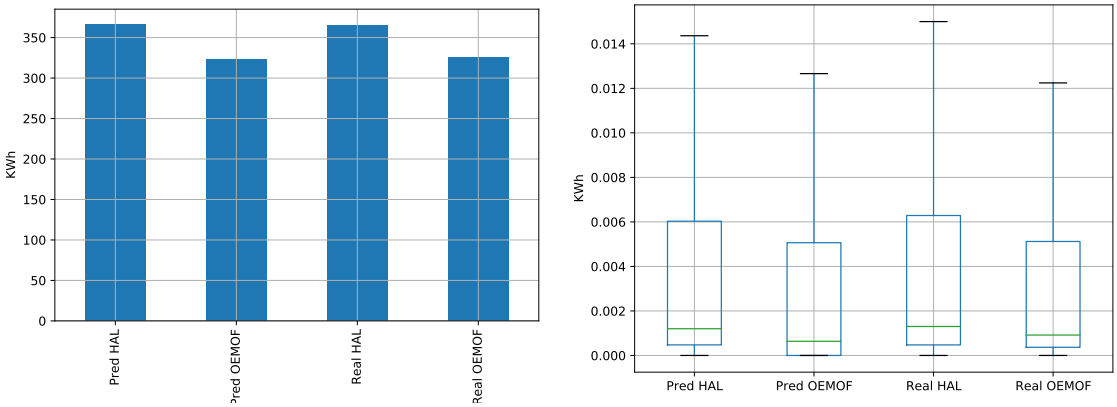
Figure 3.6: Storage consumption and PV output during the first day of the September simulation. Around 9am, the HAL algorithm uses the peak in PV production.

The simulation results of the second period (December) turned out to behave very similar to the simulations performed during September. However, in September the relative difference of the net import in the first (prediction data) and the second (real data) simulation run is 2.591% whereas it is 0.046% in December. Equation 3.1 explains how these values were calculated with n being the net import. This is likely due to the fact that there is a high thermal load and almost no PV output. As a result, the PV generated power is often instantly consumed by the heat pump which limits the algorithms ability to use the battery. A bigger PV system or additional generators might create more interesting results.

$$\frac{n^{RealOEMOF}}{n^{RealHAL}} - \frac{n^{PredOEMOF}}{n^{PredHAL}} \tag{3.1}$$

3.5 Schedule vs. Schedule Setup

In the last setup HAL was not used as an online, near real time controller for the simulation with real PV input data. Using the same method as before on the simulation results of OEMOF, a schedule was created from HALs result on the predicted PV data. As can be seen in figure 3.7, this run results in slightly higher net import peak. However, Table 3.2 shows that the total net import in the scheduled controlled run is less than in HALs online run on the prediction data while OEMOFs schedule run has a higher total net import than in the prediction run. This could be an indicator that HAL created schedules can handle uncertainties better.



(a) Total Net Import in .

(b) Net Import Peak Distribution. Whiskers are 1.5 of the inter quantile range.

Figure 3.7: Simulation results with real PV and prediction data controlled by schedules. The second simulation runs (Real HAL, Real OEMOF) have a slightly higher total net import and more and higher peaks than the first simulation runs on the PV prediction.

Table 3.2: Sum of all simulation runs net imports.

	sum
Pred HAL	366.77
Pred OEMOF	322.48
Real HAL	365.20
Real OEMOF	324.96

Chapter 4

Conclusions

This report should enable the reader to get an intuition about the properties of the two presented algorithms for controlling energy systems with sector coupling. These properties should be qualified to inform the reader about the application possibilities of a MILP based optimization algorithm and an agent based supply and demand matching algorithm. To compare complex energy systems, the HAL algorithm was extended to be able to model sector coupled devices. A smart grid for a household was modeled, simulated and analyzed after performance metrics which were introduced. Different test setups showed the properties of both algorithms with different scheduling horizons ranging from one month to one minute.

The incremental setup and the noise setup showed, that the performance of a schedule controlled energy system depends on the uncertainty of the input data. By reducing the scheduling horizon to 6 hours, the uncertainty could be reduced with little impact on performance if the system resets during the night. The greater the uncertainty in the predictions used for schedule creation, the better performs the HAL algorithm in comparison to long term schedules. This is supported by the simulations performed with real PV output data in the real data setup. This report only covers uncertainty in PV data. In a real situation, the household demand is another major uncertainty factor in predictions.

The schedule vs. schedule setup showed that the HAL algorithm is not suitable to be used for long term scheduling horizons.

The noise setup shows how near realtime control strategies help to use unexpected short term changes in PV output optimally. Although short term changes in pv power during the night won't happen in the real application, further research could show that clouds can likely cause similar effects during the day. Another interesting research question is to find out how and if the optimal scheduling horizon correlates with the uncertainty in the input data used for schedule creation.

Bibliography

[1] Oemof documentation. `https://oemof.readthedocs.io/en/stable/` . Accessed 2 June 2020.

[2] Andrew Keane Alireza Nouri, Alireza Soroudi. Strategic scheduling in smart grids.

[3] S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann. The open energy modelling framework (oemof) - a new approach to facilitate open science in energy system modelling. *Energy Strategy Reviews*, 22:16–25, 2018.

[4] Koen Kok, C. Warmer, and Rene Kamphuis. Powermatcher: Multiagent control in the electricity infrastructure. pages 75–82, 01 2005.

[5] Nailya Maitanova, Jan-Simon Telle, Benedikt Hanke, Matthias Grottke, Thomas Schmidt, Karsten von Maydell, and Carsten Agert. A machine learning approach to low-cost photovoltaic power prediction based on publicly available weather reports. *Energies*, 13:735, 02 2020.

[6] NTNU Trondheim. Milp algorithms: branch-and-bound and branch-and-cut. `https://www.itk.ntnu.no/_media/fag/fordypning/tk16/milpalgorithms.pdf`. Accessed 2 June 2020.

[7] Petr Fiedler Zdanek Bradac, Vaclav Kaczmarczyk. Optimal scheduling of domestic appliances via milp.