On-line model predictive control of battery charging for a household with PV production

Frölke, Linde; Junker, Rune Grønborg; Bacher, Peder

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

ON-LINE MODEL PREDICTIVE CONTROL OF BATTERY CHARGING FOR A HOUSEHOLD WITH PV PRODUCTION

LINDE FRÖLKE
RUNE GRØNBORG JUNKER
PEDER BACHER

Technical University of Denmark

Series: DTU COMPUTE-TECHNICAL REPORT-2021
Number: 01
ISSN: 1601-2321

Contents

Chapter 1

Introduction

For the BIPVT-E projects[1] an on-line smart control of the battery charging was designed. This report describes the steps taken in order to make this control run in real time. In addition, we will evaluate the quality of the designed mechanism, and highlight future research areas.

The online model based control of the battery charging and discharging relies on the immediate and automatic availability of measurements. Section 2 outlines the setup and components used in this project for ensuring this continuous flow of data.

The measurements are used to predict the electricity production and consumption of the system, and update our model parameters. First, Chapter 3 discusses the method used to develop forecasting models, by introducing the general model type and model evaluation procedures. Thereafter, the fitted forecasting models are presented and analyzed in Section 4 and 5.

Finally, in Section 6, we discuss how the forecasts of electricity production and consumption are used in a model predictive control (MPC) setting. Combining our forecasts with externally obtained electricity pricing forecasts, we obtain an optimal charging and discharging schedule for a number of hours ahead.

[1] SolarSmartSystem Bornholm (EUDP, ref. 64017-05200) and Måleprogram og optimering af ny type energianlæg til enfamiliehuse med varmepumpe og batterilager (ELFORSK, ref. 350-022)

Chapter 2

Data collection and flow

This Chapter describes how the measurements needed for the MPC are collected and sent between different devices. The needed measurements are:

- electricity consumption
- electricity production
- power exchange with the grid
- power exchange with the battery
- the state of charge (SOC) of the battery.

2.1 Obtaining values from Fronius inverters

The Fronius inverters have a local API, that can be accessed by devices on the same network as the inverters. Data can also be retrieved from the Fronius online platform SolarWeb, but this can not be done automatically and real-time.

In order to run real-time, a Raspberry Pi (rPi) was set up to retrieve the needed values. To do so, the rPi has the software OpenHab 2 installed, 'an open source, technology agnostic home automation platform'.[1] OpenHab includes a Fronius binding that retrieves almost all needed values. We supplemented this with a own python-implemented script that collects the electricity production data. All values are retrieved every minute.

The OpenHab software also includes so-called persistence bindings, of which we use a local database InfluxDB[2] and the protocol MQTT, which will be described in the following Section.

[1] https://www.openhab.org/

[2] https://www.influxdata.com/products/influxdb-overview/

2.2 Data communication using MQTT

Message Queuing Telemetry Transport (MQTT) is a 'machine-to-machine (M2M)/Internet of Things connectivity protocol'.[3] Using this protocol, we send the measurements in the form of messages from the rPi to the server described in the next Section. In addition, we may use MQTT later to send control sequences from the server back to the rPi.

In this setup, the rPi and the server are *clients* to another server, which we refer to as the *broker*. The messages are labelled with a *topic*, such as 'LokesPlads', and sent to the broker. Any client that wants to receive messages on a certain topic can subscribe to it.

The messages are cent using the TCP protocol. The MQTT protocol is often chosen for its speed and ease, and messages are sent and received within a period of far less than a second.

2.3 ENFOR Server

The MPC model is run on a server provided by ENFOR.[4] The model itself is implemented in R [1].

The measurements are collected using a Python-implemented MQTT-listener that receives the raw measurements from the rPi, and stores them in a file. We have implemented several checks to make sure that the MQTT-listener is always 'listening', so that we will not loose any measurements.

The ENFOR setup requires that all measurements are collected in csv files of a fixed layout. Another Python script retrieves the collected raw measurements, re-samples them to 5-minute values, and re-formats them to meet the platform requirements. This script is scheduled to run every 5 minutes using CRON.

The ENFOR software is then able to read the csv values and provide the most recent values to the on-line running MPC model. In addition, the Numerical Weather Predictions (NWPs) are collected on the server as well, and provided to the model. The measurements and NWPs are used to update the model, forecast MPC inputs, and compute optimal control sequences for the battery charging. If the control was implemented on the battery, the control sequences computed by the MPC module would then be communicated back the the rPi, which would be able to communicate battery charging targets to the Fronius inverters.

[3] http://mqtt.org/
[4] https://enfor.dk/

Chapter 3

Method for forecasting model development

As mentioned earlier, the designed MPC for battery charging depends on the forecast of two variables: the power consumption and the power produced by the hybrid PV panels. The predictions are produced using a model for each of these variables, which will only rely on local measurements and weather forecasts provided by the ENFOR platform. Each model will be fitted separately for all forecast horizons $k \in \{1, 2, \ldots, 31\}$ hours, resulting in 2 times 31 fitted models with different parameters.

An 'administrational' challenge arises when implementing weather forecast-dependent models. In the current setup, it is necessary to keep track of hourly weather forecasts for horizons up to 31 hours ahead, and new forecasts only come in every 6 hours. As a consequence, it is difficult to set up a input data structure for the model that allows us to fit models easily and effectively.

To solve this problem, the data is organized in the following structure: PEDER have you introduced this data structure in research before?

3.1 General model introduction

We use tools from time series analysis to fit models that can capture the dynamics of power production and consumption from data.

In order to model non-linear relations between inputs and output, we use a modelling procedure that consists of two stages. In the first stage, (some) model inputs are transformed, for instance using a low-pass filter or Fourier basis. In the second stage, a linear regression model is applied to find a linear relation between the (transformed) inputs and the output variable. TODO cite peder. In the current work, we also applied a *recursive least squares* (RLS) model in the second stage. In this type of model, the regression parameters are updated every time new data is coming in, in our case every hour. However, the weight of the prediction at time $t-k$ in the least square optimization decreases exponentially with a factor λ^k.

This two-stage approach is a robust and fast way to fit relatively simple models that can capture complex, non-linear dynamics. See Chapter 5 in Hastie et al.(2009) [2] for more information.

While the linear regression parameters are updated every time step, the model parameters determining the transformation (low pass parameter for example) and the RLS forgetting factor λ need to be determined in advance. Optimal values of these parameters are determined in an off-line setting by minimizing the root mean squared error (RMSE) on a test data set, as also described in Bacher et al. [3].

Chapter 4

Forecasting Electricity Production

In order to apply model predictive control to the battery charging of the studied household, forecasts of the electricity production from its hybrid PV panels are needed. First, a short overview of the literature on solar power forecasting is presented in Section 4.1 In Section 4.2 we will introduce the different models that have been investigated. Next, the prediction performance of these models are evaluated in Section 4.3. Finally, conclusions and directions for further improvement are given in Section 4.4

4.1 Literature on solar power forecasting

It is known that the efficiency of photovoltaic panels depends on the solar angle of incidence on the panel. In a system with fixed PV panels, the incident angle varies over time as the sun changes position. The incidence angle affects the fractions of transmitted and reflected light, both on the glass surface of the panel and inside the panel. The internal incidence angle effect depends on module design and may therefore differ between different PV panels [**?**]. Jiang et al. found experimentally that the solar incident angle is a critical factor affecting the performance of fixed PV modules, i.e. their solar energy conversion efficiency [4].

This angle of incidence (AOI) effect applies mostly to the beam or direct component of the irradiation, while the diffuse solar energy conversion efficiency is almost independent of the AOI.

One can account for the incidence angle dependence of the direct radiation conversion efficiency by using an incidence angle modifier (IAM) function. A possible IAM is for example given in the IEC 61853-2:2016 standard, as

$$\tau(\theta) = \frac{1 - \exp(-\cos(\theta)/a_{\mathrm{r}})}{1 - \exp(-1/a_{\mathrm{r}})}, \tag{4.1}$$

with the total panel area A as a given constant. Note that the function depends on a single parameter a_{r}, which can be fitted to data.

Others fit a 5^{th} order polynomial to capture the AOI dependence.

Due to the fact that we measure total radiation on a horizontal plane, the share of absorbed energy depends on the solar position in a second way. The efficiency of the PV panels is different

for diffuse and beam radiation, and the share of diffuse/beam radiation is largely determined by the solar position.

Something about persistence models [5].

4.2 Model development

The predictions are based on a single input, the solar irradiation from the available numerical weather prediction (NWP). Fit basic model and extend with method to vary solar energy conversion efficiency over time (RLS), and dependent on time of day (spline functions).

4.2.1 Base model, linear regression without parameter updates

We start our model development with a very simple linear regression model without intercept, that fits an efficiency parameter η to the measured global irradiation $I_{g,t}$. The model will be referred to as PV_LS, as the efficiency parameter is fitted using Least Squares. The model is formulated as

$$P_{pv,t} = \eta A I_{g,t} . \tag{4.2}$$

4.2.2 Extension: linear regression with parameter updates

Even if this linear relation was correct, the parameter η would vary over time due to dirtiness on the panels. We can account for this by re-estimating the parameter η at each time new measurements become available, i.e. implementing the same model using RLS with a forgetting factor. This is the first extension have applied. In this case, the efficiency parameter would change slightly over time as

$$P_{pv,t} = \eta_t A I_{g,t} . \tag{4.3}$$

In this case, the efficiency parameter is implicitly dependent on time, as it is re-estimated each time new data comes in (on-line update).

4.2.3 Extension: time-of-day dependent solar efficiency function

It is known the relation between measured solar irradiation and produced electricity is non-linear, first and foremost due to the AOI effects as described before. In addition, the irradiation is measured on a horizontal plane, while the PV panels reside on both sides of a tilted roof, which adds another non-linearity to the relation between $I_{g,t}$ and $P_{pv,t}$. If we would be able to make models for the production of both sides of the roof separately, we would be able to include physical knowledge about incidence angle effects into our model. We could do this by projecting the measured radiation on two planes of the roof, and then fit this radiation multiplied by a suitable IAM function to data. This is however not possible in the current project, as only the

total production of the two sides is measured. Separate models for the production of each side cannot be fitted, as the output variable is not measured. Something about not-working panels??

In order to still allow for a non-linear efficiency over the day, the efficiency is included in the model as a piece-wise polynomial function. This is a continuous and smooth function of the time of day, which is fit using a set of B-spline basis functions. By multiplying the base spline functions with the global irradiation, the measured global irradiation is divided into K components, which are nonzero at different times of day. An efficiency parameter η_i is fit for each $i \in \{1, 2, \ldots, K\}$ individually.

The B-spline basis functions are generated using the R-core package *splines* (version 3.5.1) [1]. The time-of-day acts as the predictor variable, and we choose a cubic spline with intercept (the latter ensures nonzero values at the boundaries). The boundary knots are placed at the earliest and latest non-zero irradiation measurement present in the data. In this setting, at least 4 basis spline functions are generated, and for each specified inner knot, the number of basis functions increases by one. The inner knots are placed uniformly within the interval specified by the boundary knots. Figure TODO illustrates the resulting model inputs after multiplying the basis splines with the global irradiation. Note that for all times t, the functions add up to 1.

The model including these splines is formulated as

$$P_{\mathrm{pv},t} = \sum_{i=1}^{K} \eta_i \, I_{\mathrm{g},t} \, S_{i,h} \tag{4.4}$$

where $S_{i,h}$ is the i^{th} spline function as defined above. It has the subscript h, because due to the way it is defined its value only depends on the hour of day.

4.2.4 Testing different models

We will evaluate the performance the base model, and of the models with all possible combinations of the two proposed extensions. An overview of the evaluated models is presented in Table 4.1.

	Least Squares	Recursive Least Squares
η constant	PV_LS_c	PV_RLS_c
η dependent on tday	PV_LS_tday	PV_RLS_tday

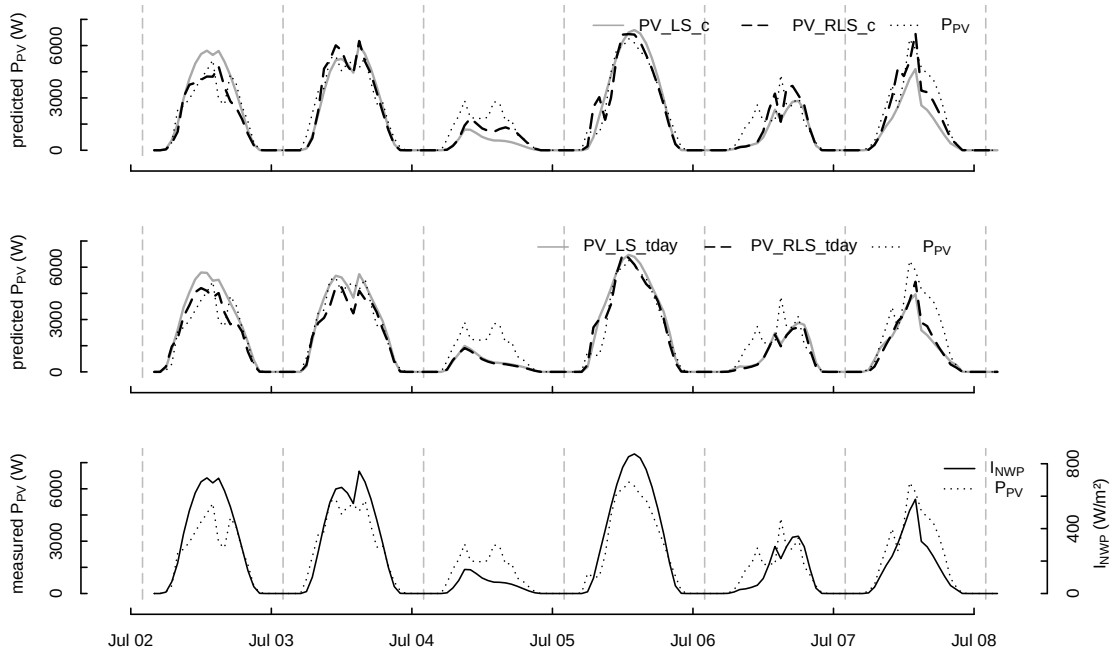
Table 4.1: Tested models with model names. The models are organized here by parameter fitting method and their representation of the panel efficiency η. The simplest model is PV_LS_c, whereas the most extended version of the model is PV_RLS_tday.

4.3 Results

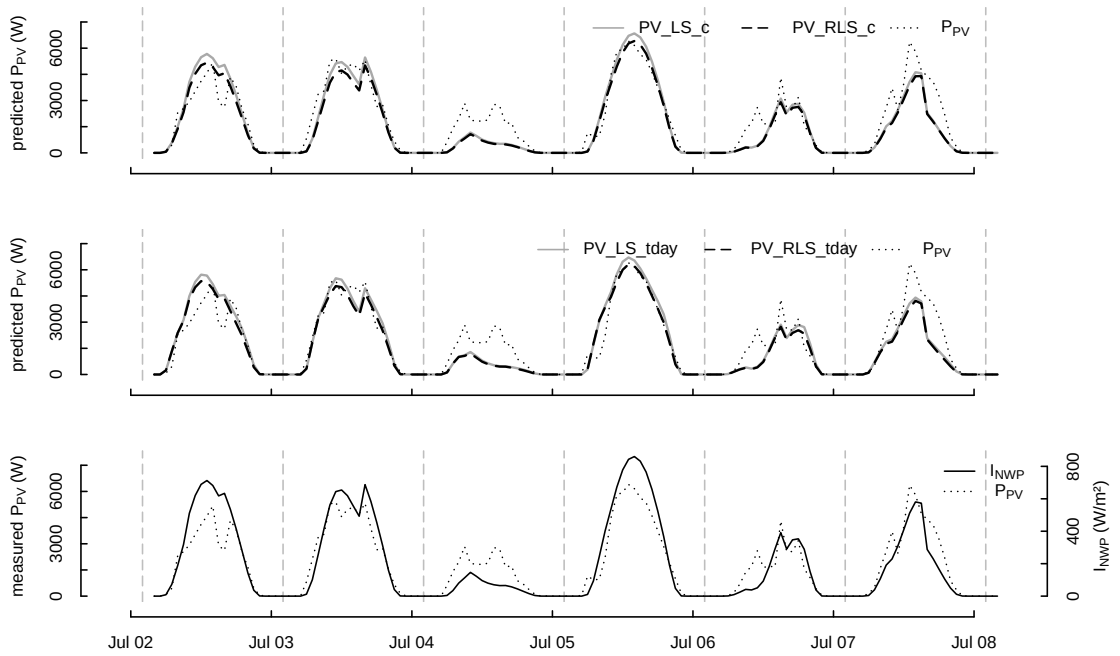
The 4 models as described in Section 4.2 are trained on a the training data set of 55 days. Thereafter, the performance of these fitted models test data set of days is compared.

To evaluate model performance, we will look closer at the predictions in some interesting periods for different prediction horizons k. In Figure 4.1, the predictions for all models are shown for $k = 1$ and $k = 2$, for a chosen period in the test data set containing days with different solar irradiation patterns. It can be seen that all models follow the measurements reasonably well.

For $k = 1$, it seems that the RLS models perform slightly better under cloudy conditions, whereas the LS and RLS models predict very similar values under clear conditions For $k = 2$, the RLS and LS models become almost identical. This is also the case for all models with higher prediction horizons, whose predictions are not plotted here.



(a) $k = 1$



(b) $k = 2$

Figure 4.1: Predicted power production for all models. The upper row contains the predictions for models with constant efficiency, the middle row the models with time of day dependent efficiency, the lower row the used input (k-ahead NWP of irradiation) and output (measured power production) of the models. The LS and RLS models perform very similar already for $k = 2$.

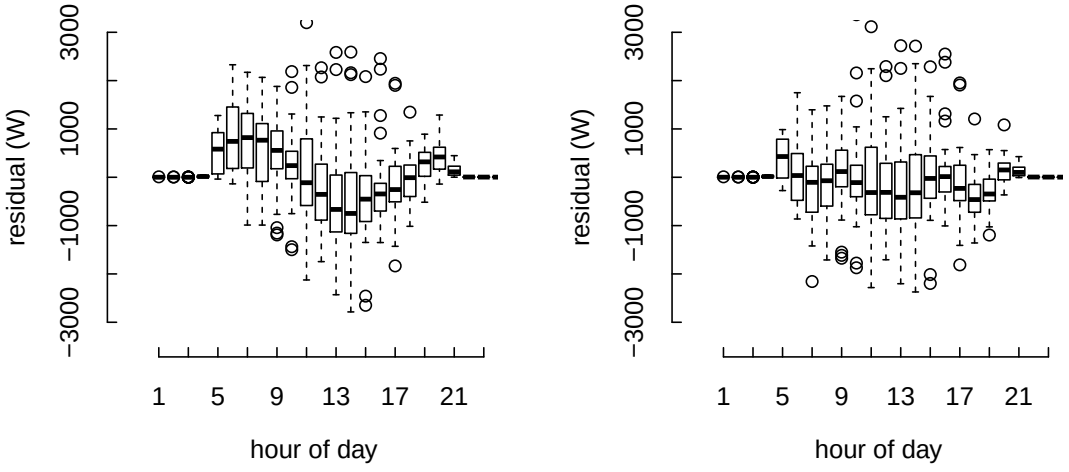
4.3.1 Residual distribution versus time-of-day

It is hard to evaluate and compare model performance using basic plots of the predictions such as in Figure 4.1a. To draw conclusions, we will consider the size and distribution of the residuals, which are defined as

$$\varepsilon_t = P_{\mathrm{pv},t} - \hat{P_{\mathrm{pv},t}} \tag{4.5}$$

where $\hat{P_{\mathrm{pv},t}}$ represents the predicted PV power. The distribution of the residuals for each time of day is shown for 1-hour-ahead predictions in Figure 4.2, and for 26-hour-ahead predictions Figure 4.3. It can be seen that the RLS improves the residual distribution for $k = 1$, but for $k = 26$ no improvement is visible in these plots. In fact, for all $k > x$, the predictions for the RLS and LS versions are similar. This is in agreement with the conclusion from visual inspection of the predictions: the corresponding RLS and LS models hardly differed from one another.

On the other hand, the time-of-day dependence of the efficiency improves the model visibly in all cases.



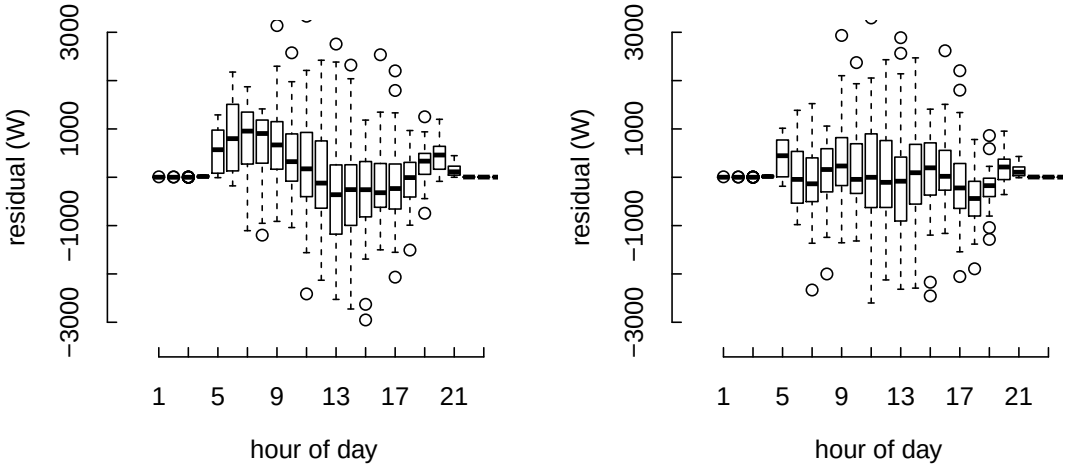
(a) PV_LS_c

(b) PV_LS_tday

(c) PV_RLS_c

(d) PV_RLS_tday

Figure 4.2: Residuals boxplot $k = 1$



(a) PV_LS_c

(b) PV_LS_tday

(c) PV_RLS_c

(d) PV_RLS_tday

Figure 4.3: Residuals boxplot $k = 26$

4.3.2 RMSE for power production models

To gain further insight on the performance of the models, we will look at various error measures and visualize how the error measures vary per model type and prediction horizon. First, it is evaluated how the *RMSE* for each model depends on the prediction horizon. The result is shown in Figure 4.4, which first of all shows that that the models including the efficiency as a function of the time of day perform best (PV_LS_tday and PV_RLS_tday). Furthermore, it is visible that the RLS models outperform the corresponding LS models for the shorter prediction horizons (1-4 hours ahead). After that, the models perform similarly, until the LS model outperforms the RLS for 17-31 hours ahead.

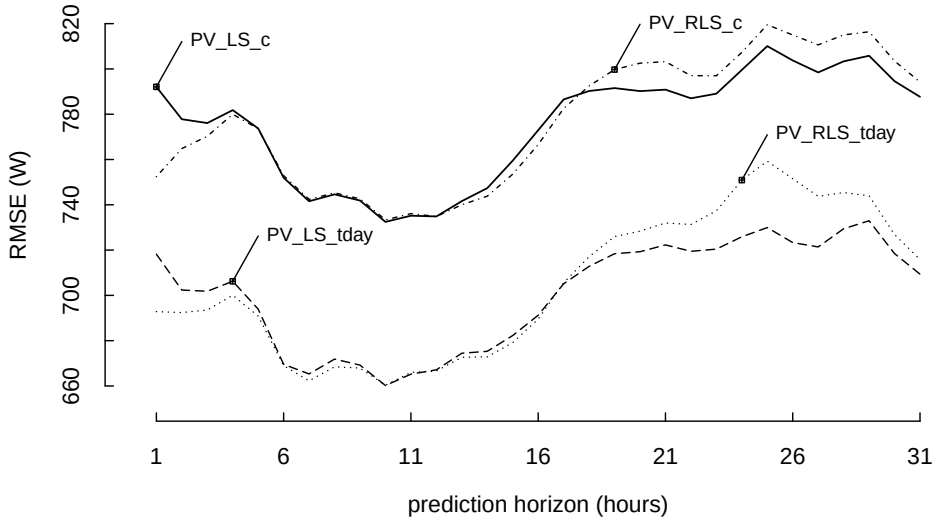


Figure 4.4: Root mean squared error (RMSE) as a function of prediction horizon k, for the 4 different tested models. A low RMSE is considered better. First of all, it is shown that the models including the efficiency as a function of the time of day perform best. Furthermore, it is visible that the RLS models outperform the corresponding LS models for short horizons from 1-4 hours ahead. After that, the models perform similarly, until the LS model outperforms the RLS for 17-31 hours ahead.

4.3.3 MARE for power production models

Next, we consider another error measure to compare the models, which scales the residual at each time by the size of the measured value at that time. This Mean Absolute Relative Error (MARE) as discussed in Section **??** is plotted as a function of prediction horizon in Figure 4.5. As with the RMSE, it is shown that the models including the efficiency as a function of the time of day perform best, as judged using this error measure. Their average error is around 19-21 percent of the true value for all horizons, whereas the models with constant efficiency are around 22-24 percent off. For the MARE however, the most complex model PV_RLS_tday performs best for

up to 22 hours ahead. For larger prediction horizons, model PV_LS_tday starts performing better. The two models with constant efficiency perform very similarly for prediction horizons over 4 hours ahead.

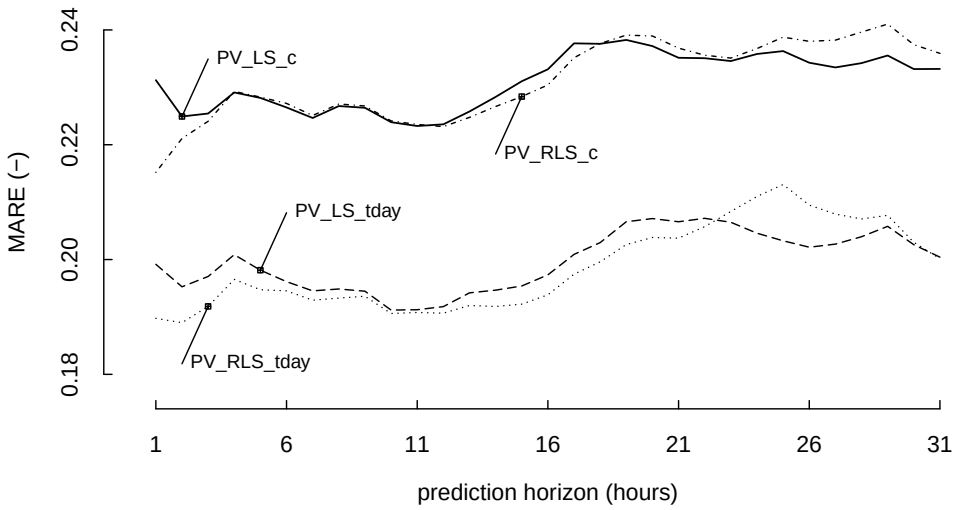


Figure 4.5: Mean Absolute Relative Error (MARE) as a function of prediction horizon k, for the 4 different tested models. A low MARE is considered better. Again, it is shown that the models including the efficiency as a function of the time of day perform best. In this error measure, the most complex model PV_RLS_tday performs best until 22 hours ahead. For larger prediction horizons, model PV_LS_tday starts performing better.

4.3.4 Estimated non-parametric function

Finally, the estimated efficiency functions are investigated. For the LS models, this function is constant over the entire test data set, and can therefore be plotted as a single line. This is the red line drawn in all plots of Figure 4.7. However, the RLS models' parameters are adapted over the time of the test data set, and therefore there are many fitted efficiency values to each hour of day. To show the trend in these values, they are visualized in boxplots in Figure 4.7. The prediction horizon in the rows of the plot are 1,2, and 31, respectively. The left column of the plot shows the models with efficiency constant efficiency, while the right column show the models with efficiency dependent on time of day.

Several observations can be made from Figure 4.7. First of all, the distribution of the RLS parameters becomes more narrow as the prediction horizon increases, which is explained by a larger value of λ for high k.

Second, we can see that the efficiency function in all PV_(R)LS_tday models is asymmetric: it is for some reason higher in the first half of the day than the second. This does not necessarily match our knowledge of the building, which has the panels facing North and South. To make

sense of this result, we plot the mean solar irradiation and the mean PV production for each hour of day in Figure 4.6. Here it is visible that that the PV power production (solid line) is asymmetric with a peak to the left, whereas the NWP of the solar irradiation (dashed line) is symmetric. This explains the asymmetry in the estimated efficiency.

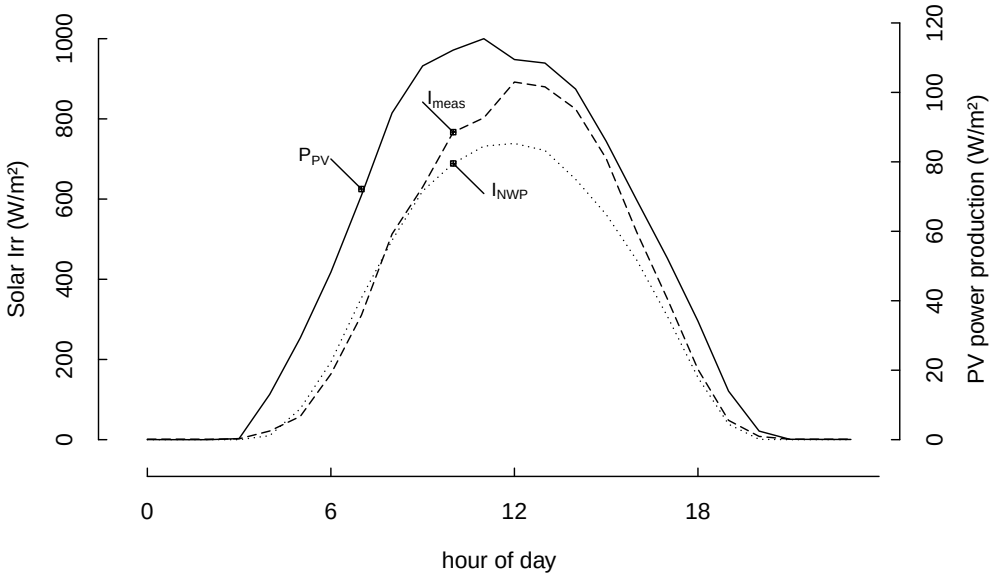


Figure 4.6: The mean of $P_{\mathrm{pv},t}$, I_{NWP}, and I_{meas} for each time of day over the test data set. Comparing the solid and dashed line shows that the production is asymmetric with a peak to the left, whereas the NWP of the solar irradiation symmetric. This results in the asymmetry of the estimated efficiency function.

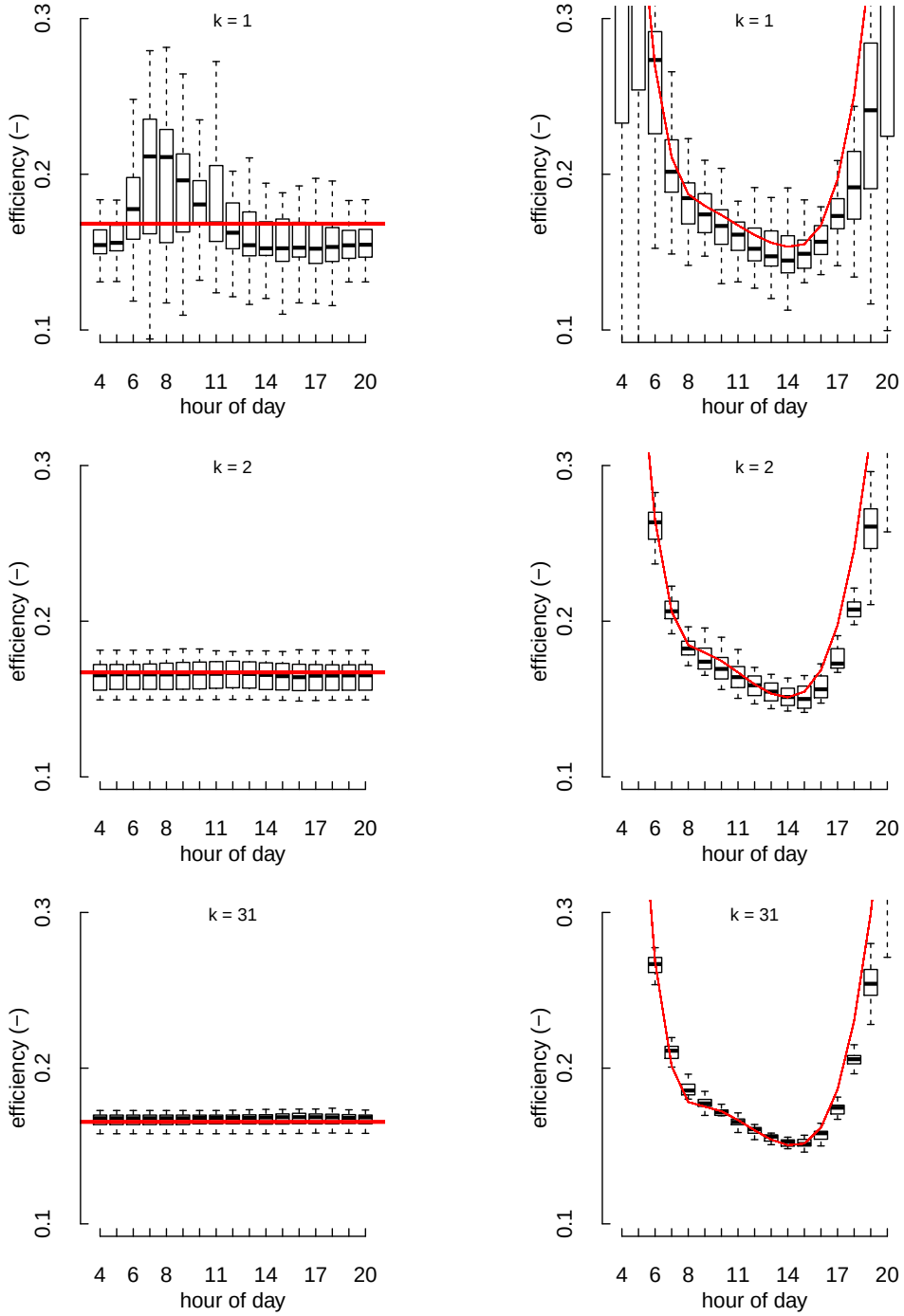


Figure 4.7: Fitted efficiency functions for different models and different prediction horizons. The prediction horizon is the same on each row of this plot. The left row shows the models with constant efficiency over the day, the right shows the models with efficiency included as a function of time of day. The red line in each plot is for the LS model, for which the efficiency function is not adapted over the test data set. The box-plots depict the distribution of the estimated efficiency versus time of day by RLS

4.4 Conclusion & Discussion PV prediction

From the analysis of predictions on a test data set of the 4 models, we can draw conclusions and point out directions for further improvement.

First of all, it has become clear that predictions greatly improve by including the conversion efficiency as a function of hour of day. Box-plots of the residuals versus time of day showed that this efficiency function can remove structural errors from the predictions, as shown in Figures 4.2 and 4.3. The better performance of the variable efficiency models was also shown using two error measures, the RMSE and MARE, in respective Figures 4.4 and 4.5.

The latter two figures also indicate that the RLS models perform better than the corresponding LS models for short prediction horizons. However, for predictions further than 5 hours ahead, the RLS and LS models perform similarly, and for more than 20 hours ahead the LS models started to have smaller RMSE and MARE than the RLS. We must note, however, that the available data set only covers about 3 months of data, which was to be used for both fitting and testing the models. Therefore, the data set does not contain all variation of solar positions as a function of time of day, as present over an entire year. It is likely that there would be more benefits to having a recursive updating of parameters, when the model would be fitted and tested on an entire year's measurements.

For this reason, the recursive model was chosen to implement in the on-line running MPC module, which should be able to run for the entire year. We identify two points for improvement in the implemented model PV_RLS_tday. First, a problem with the current setup occurs when the NWP structurally predicts the solar irradiation to start later than it is measured on site. In this case, the model cannot account for any *structural* production already there when the NWP is still zero, as there is no way to fit a constant to a irradiation of zero that will lead to nonzero production. This may be overcome by adding a fitted hour of day dependent function to the model, that is independent of any inputs. We will see an example of this in the next Chapter, as this is implemented in several of the power consumption models. Second, we should find a way to deal with the varying time of sunrise and sunset over the year. The solution to the previously explained problem could perhaps also be the solution to this one, but this remains to be tested.

Further future improvements of the model may be achieved by normalizing the solar irradiation with the clear sky irradiation. In addition, for 1 hour ahead prediction of PV power production using NWP of solar irradiation, the model can probably be improved by including information about production at the time of prediction, or the prediction error that was made for the production at the time of prediction. This can be done by including a moving average (MA) or auto-regressive (AR) term in the model. The latter can rather easily be implemented in the used linear regression model form, by lagging the output variable and adding it as an input to the model.

Chapter 5

Forecasting Electricity Consumption

In order to apply model predictive control to the battery charging of the studied household, forecasts of its electricity consumption are needed. In Section 5.1 we will introduce the different models that have been investigated. Next, the prediction performance of these models are evaluated in Section 5.2. Finally, conclusions and directions for further improvement are given in Section 5.2.5

5.1 Model development

For forecasting total electricity consumption, we will fit different models. We will fit a basic model, and extend it by allowing for transformation of the inputs, and adding inputs. Each model version will be fitted using both RLS and ordinary LS.

5.1.1 Base model

We start our model development with a simple linear regression model, that takes the ambient temperature $T_{\mathrm{a},t}$ from the NWP as an input, and includes a constant. This is written as

$$P_{\mathrm{c},t} = c_1 T_{\mathrm{a},t} + c_2 \,, \tag{5.1}$$

where the parameters c_1 and c_2 may be fitted using LS or RLS. The models will be called PC_LS_$\mathrm{T_a}$ and PC_RLS_$\mathrm{T_a}$, respectively.

5.1.2 Extension: low-pass filter the ambient temperature

In order to allow for a delay of the effect of the ambient temperature on the indoor temperature, and therefore the electricity load, we introduce a low-pass filter transformation of this input. The resulting model is written

$$\mathrm{lp}(T_{\mathrm{a},t}) = a_{\mathrm{lp}} T_{\mathrm{a},t-1} + (1 - a_{\mathrm{lp}}) T_{\mathrm{a},t} \tag{5.2}$$

$$P_{\mathrm{c},t} = c_1 \,\mathrm{lp}(T_{\mathrm{a},t-1}) + c_2 \,, \tag{5.3}$$

where the parameters c_1 and c_2 may be fitted using LS or RLS. The models will be called PC_LS_lp(T_a) and PC_RLS_lp(T_a), respectively. The parameter a_{lp} is fitted off-line, as described in Section 3.1.

5.1.3 Extension: time-of-day dependent constant

In order to model periodicity in the electricity consumption, a linear combination of Fourier series is used. This is included to model the part of the constant fraction of the load, that is, the part that is independent of measured inputs. This is formulated as follows:

$$P_{c,t} = c_1 T_{a,t} + c_2 + c_3(h/24), \tag{5.4}$$

where c_3 is a function of the hour-of-day h. The function is fitted using Fourier series as

$$c_3(h/24) = \sum_{i=1}^{K} \phi_i fs_i(h/24), \tag{5.5}$$

where K is the number of functions in the Fourier series equal to double the number of harmonics n_{har}, and the i^{th} harmonic function fs_i is scaled by a factor ϕ_i. For example, the set of harmonics for $n_{har} = 2$ is

$$\left\{ \sin\left(\frac{2\pi h}{24}\right), \cos\left(\frac{2\pi h}{24}\right), \sin\left(\frac{2\cdot 2\pi h}{24}\right), \cos\left(\frac{2\cdot 2\pi h}{24}\right) \right\}. \tag{5.6}$$

These models will have tday in their model name. In this report, we have applied $n_{har} = 3$. Other numbers have been tested, with similar or worse results.

5.1.4 Testing different models

We will evaluate the performance the base model, and of the introduced model extensions. An overview of the evaluated models is presented in Table 5.1.

	Least Squares	Recursive Least Squares
base case	PC_LS_T_a	PC_RLS_T_a
add low-pass filtered T_a	PC_LS_lp(T_a)	PC_RLS_lp(T_a)
add $c_3(h)$	PC_LS_lp(T_a)_tday	PC_RLS_lp(T_a)_tday

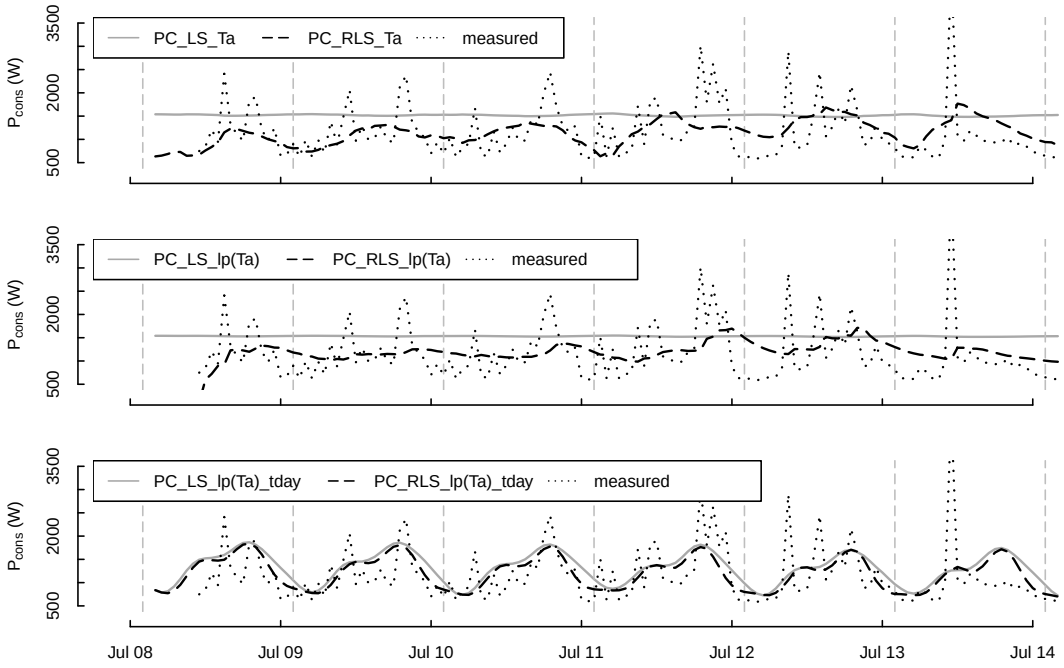
Table 5.1: Tested models with model names. The models are organized here by parameter fitting method and the extensions that are added. The simplest model is PC_LS_T_a, whereas the most extended version of the model is PC_RLS_lp(T_a)_tday.

5.2 Results

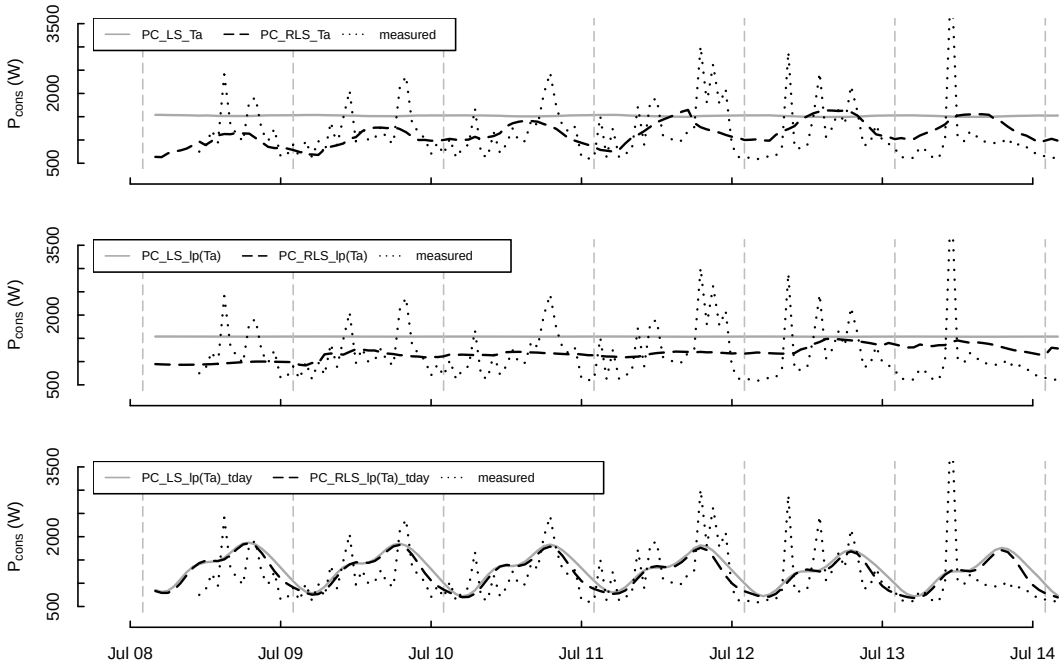
The 6 models as described in Section 5.1 are trained on the training data set of 55 days. Thereafter, their forecasting performance is evaluated and compared using several plots and error measures.

First, we will look closer at the predictions in some interesting periods for different prediction horizons k. In Figure 4.1, the predictions for all models are shown for $k = 1$ and $k = 2$, for a chosen period in the test data set containing days with different consumption patterns. For both $k = 1$ and $k = 16$, it is clear that the LS models without time of day dependent base load do not capture any dynamics of the system. The RLS models follow the measured consumption better in each case, however, the peaks seem to follow the true peaks rather than coincide with them. Finally, note that the LS and RLS models including Fourier series lead to much more similar predictions than the other model pairs. For this model, the forgetting factor is generally fitted to a value close to one, indicating that the model better captures the true dynamics of the system. Comparing the plots for $k = 1$ and $k = 16$, it seems that the predictions are very similar.

We will further test these observations using residual analysis in the next sections.



(a) $k = 1$



(b) $k = 16$

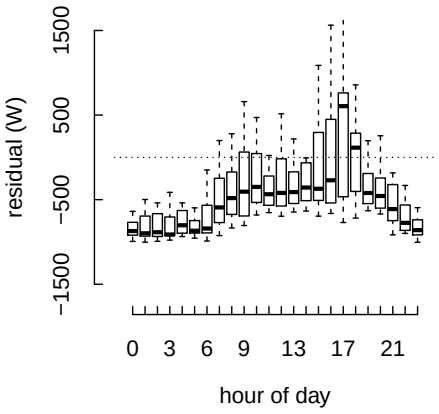
Figure 5.1: Measured versus predicted power consumption for all models, for selected values of k. The model is extended from top to bottom; each plot contains the LS and RLS versions of these extensions.

5.2.1 Residuals versus time-of-day

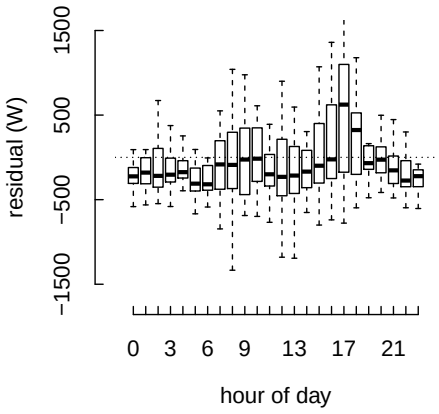
As in the previous chapter, we plot the distribution of the residuals for each time of day, to highlight any bias in the models and visualize improvements. These plots are shown for 1-hour-ahead predictions in Figure 5.2, and for 26-hour-ahead predictions Figure 5.3. Based on these plots, the predictions for $k = 1$ and $k = 26$ perform very similarly.

It can be seen that the RLS in all cases corrects a large negative bias for many hours of day. As the LS parameters were fitted for the period from April 13 to June 11, it is likely that the constant electricity consumption based on this period is larger than it should be for the testing period that runs from June 12 to July 16.

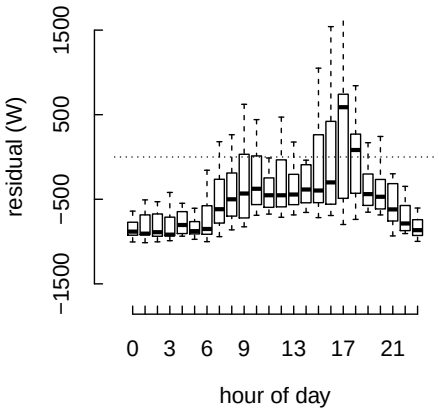
Furthermore, the addition of an hour-of-day dependent constant part of the electricity use successfully removes the bow-shaped pattern in the residuals as observed in for example Figure 5.2a, 5.2c, and 5.2d. This can be seen by comparing these to the lower row of Figure 5.2.



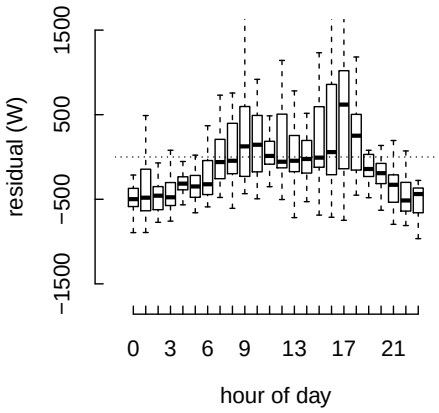
(a) PC_LS_T_a



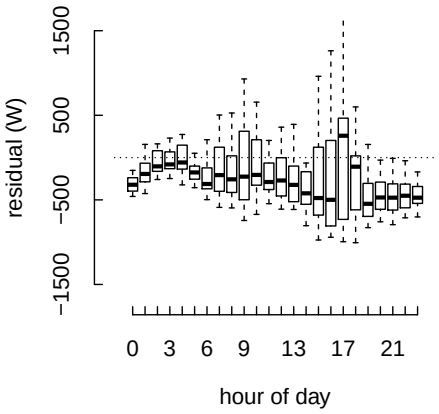
(b) PC_RLS_T_a



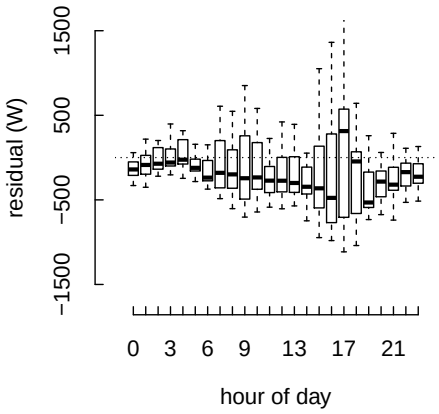
(c) PC_LS_lp(T_a)



(d) PC_RLS_lp(T_a)

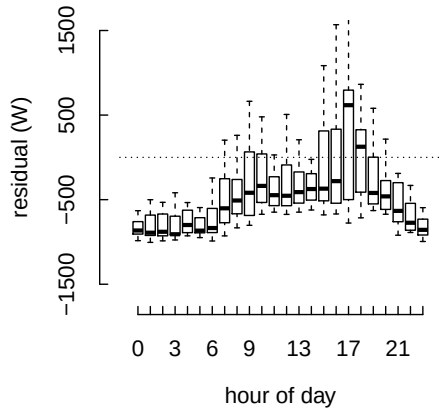


(e) PC_LS_lp(T_a)_tday

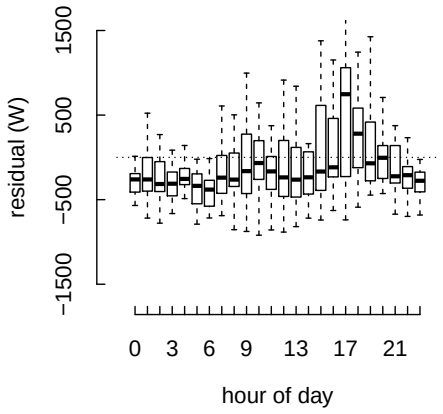


(f) PC_RLS_lp(T_a)_tday

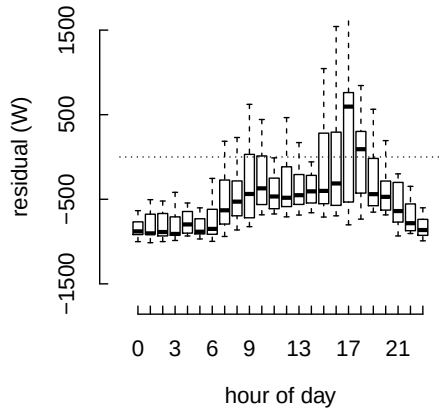
Figure 5.2: Residuals boxplot $k = 1$



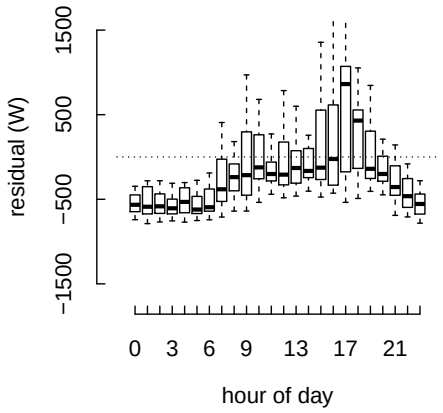
(a) PC_LS_T_a



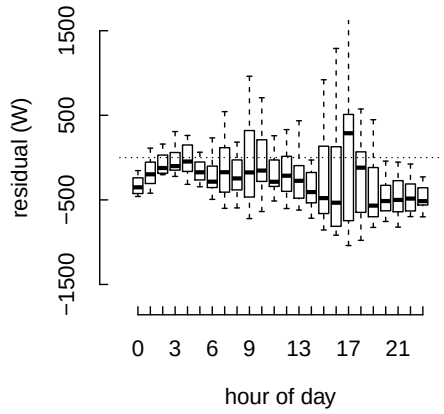
(b) PC_RLS_T_a



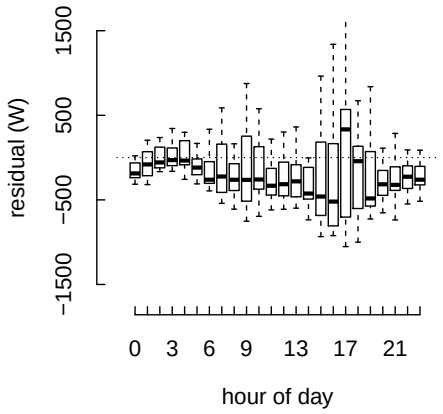
(c) PC_LS_lp(T_a)



(d) PC_RLS_lp(T_a)



(e) PC_LS_lp(T_a)_tday



(f) PC_RLS_lp(T_a)_tday

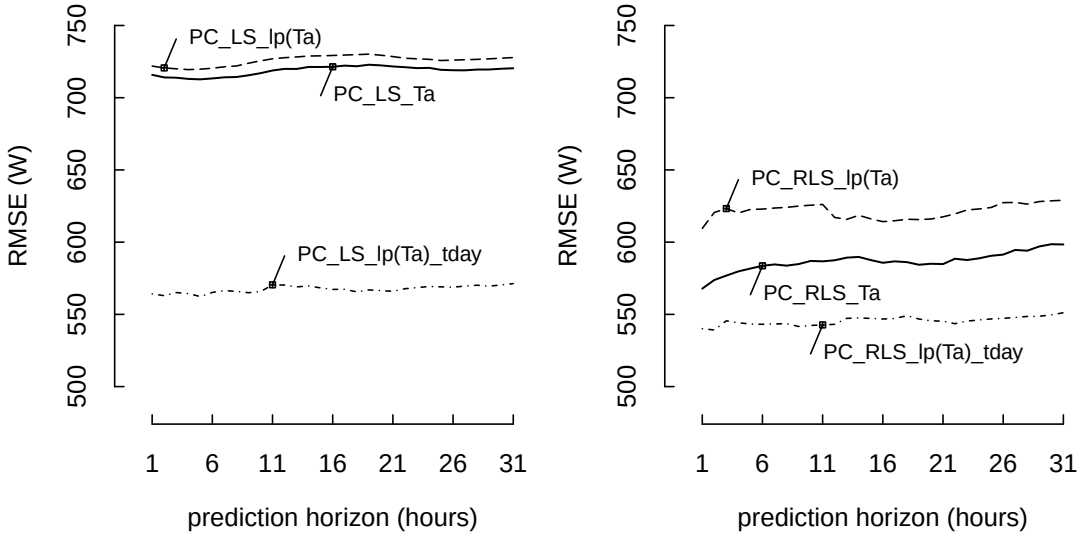
Figure 5.3: Residuals boxplot $k = 26$

5.2.2 RMSE for power consumption models

As done for the PV production models, we again compare the residuals of the different models using various error measures. First of all, Figure 5.4 shows the RMSE of all 6 tested models as a function of the prediction horizon k.

It is shown that the models including a time of day dependent constant perform best. Also, generally the RLS models outperform the corresponding LS models. Surprisingly, the low-pass filter extension alone does not improve the RMSE in the LS models nor the RLS models. Note that the prediction horizon does not change the RMSE much in these models.

If we were to pick the best model based on this error measure, the most complex model PC_RLS_lp(T_a)_tday would be chosen.



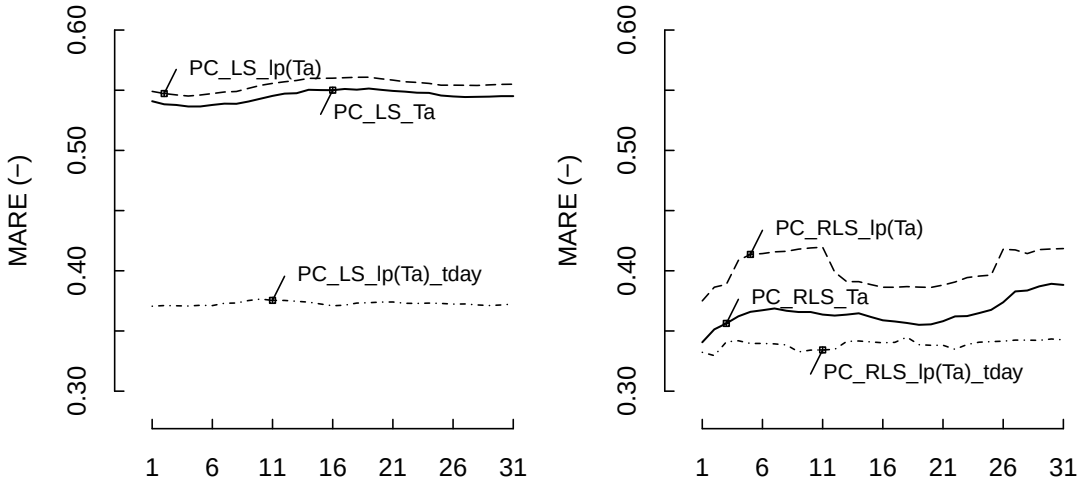
(a) LS models

(b) RLS models

Figure 5.4: Root mean squared error (RMSE) as a function of prediction horizon k, for the 6 different tested PC-models. A low RMSE indicates better predictions. The same line type is used for the corresponding models in the LS and RLS plot. It is shown that the models including a time of day dependent constant perform best, and that the RLS models outperform the corresponding LS models. Note that the prediction horizon does not affect the RMSE much in these models.

5.2.3 MARE for power consumption models

Now we consider the MARE as an error measure, and plot this measure as a function of prediction horizon in Figure 5.5. The hierarchy of model performance is similar to what was observed for the RMSE, apart from the result that the RLS models outperform the LS models more than before. For, example, model PC_RLS_lp(T_a) has a lower MARE that model PC_LS_lp(T_a)_tday.



(a) LS models

(b) RLS models

Figure 5.5: Mean Absolute Relative Error (MARE) as a function of prediction horizon k, for the 6 different tested consumption models. A low MARE indicates better predictions. The same line type is used for the corresponding models in the LS and RLS plot. The hierarchy of model performance is similar to what was observed for the RMSE.

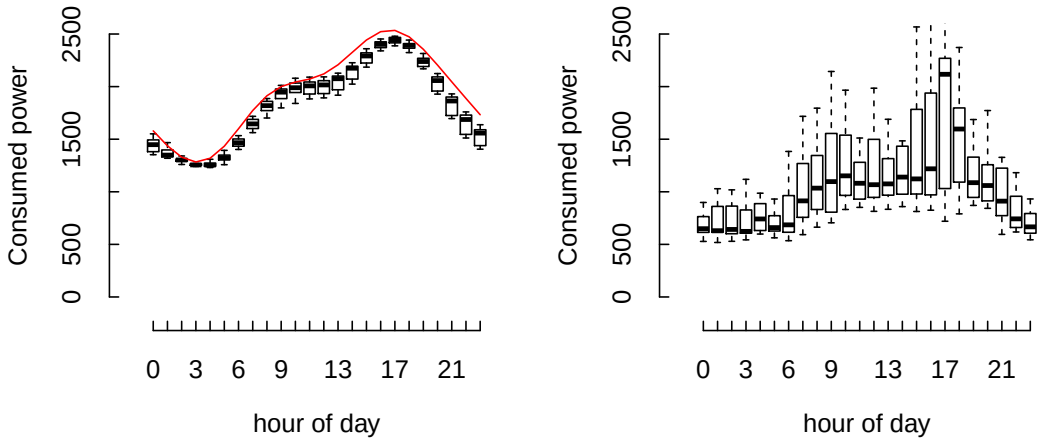
5.2.4 Estimated non-parametric function

Finally, the shape of the fitted function $c_3(h/24)$ as defined in Equation 5.5 will be investigated. More specifically, we will compare the total consumption that is independent of the input $T_{\mathrm{a},t}$, for models PC_LS_lp($\mathrm{T_a}$)_tday and PC_RLS_lp($\mathrm{T_a}$)_tday. For the LS model, this function is constant over the entire test data set, plotted in red in Figure 5.6. For the RLS model, the function varies over the test data set, as the parameters are updated. This function is depicted in a box plot in Figure 5.6a. To compare the shape, the same is done for the power consumption as a function of hour of day, in Figure 5.6b.

First, note that he constant part is generally larger than the measured. As the coefficient fitted to the NWP of the ambient temperature is negative, the final prediction of the power consumption will be closer to the measured power consumption. The constant part actually also contains a factor $-c_1 \cdot T_{\mathrm{i}}$, because the power consumption is actually linearly related to the difference between the indoor and ambient temperature.

It can be seen that the fitted hour-of-day function of the input-independent part of the consumption follows the trend seen in the measured power consumption. In addition, the shape is very similar to typical Danish consumption curves, as for example found in [6].

Note that the constant part of power consumption of the LS model is consistently slightly higher than for the RLS model. This can be expected, as the function fitted on the training data is unchanged in the LS model, so that the base load fitted here is still adapted to data from April



(a) Fitted constant part of power consumption for $k = 1$. Red line for PC_LS_lp(T_a)_tday and boxplots for PC_RLS_lp(T_a)_tday.

(b) Measured power consumption

Figure 5.6: Time of day dependence of the fitted and measured power consumption on the testing data set. The constant part is generally larger than the measured, as the influence of a larger $T_{a,t}$ on the power consumption is negative (i.e. the coefficient fitted to the NWP of the ambient temperature is negative).

and May, whereas the training data set contains dates in June and July. This shows the advantage of using a recursive model in this setting.

5.2.5 Conclusion and Discussion Electricity Load Model

From the analysis of predictions on a test data set of the 6 load forecasting models, we can now draw conclusions and point out directions for further improvement. First of all, plots of the predictions already revealed the improvements achieved both by updating parameters recursively, and by including a time of day dependent constant load function. These observations were confirmed by box-plots of the residuals versus time of day in Figures 5.2 and 5.3, which showed that structural prediction errors were reduces. Finally, the error measures RMSE and MARE were also reduced by these model extensions, as depicted in Figures 5.4 and 5.5.

For both applied error measures, the RLS models perform better than the corresponding LS models for all prediction horizons. Using the MARE, model PC_RLS_T_a scores better than PC_LS_lp(T_a)_tday for some k, whereas it scores worse on the RMSE. As the RMSE punishes large errors, this indicates that the prediction errors of PC_RLS_T_a have larger peaks than those of PC_LS_lp(T_a)_tday, whereas the average percentage deviation from the true value is lower. Here it is up to the modeller to decide whether it is needed to avoid large errors.

Note that the models PC_(R)LS_lp(T_a) did not preform better than the preceding models without the low-pass filter PC_(R)LS_T_a. This is unexpected, as the low-pass filter makes the model more flexible, i.e. by fitting parameter $a_{lp} = 0$ it becomes equal to the non-filtered version. However, it may be that the low-pass parameter a_{lp} is over-fitted to the training period, as this off-line parameter is *not* updated in the recursive procedure.

It was found in all plots and tests that the most extended model PC_RLS_lp(T_a)_tday performs best. As opposed to case of power *production* forecasting, the time of day dependent electricity consumption already changed significantly over the relatively short data set availabe to us, so that the benefits of RLS were already clear. This model was chosen to implement in the on-line MPC module.

We identify three points for improvement in the implemented model PC_RLS_lp(T_a)_tday. Firstly, the fact that we obtains very similar results for different k indicates that the models could be grouped, so that they will share their off-line parameters λ and a_{lp}. Second, the model may be improved by fitting different parameters for weekend and weekdays as in [6], which can be implemented rather easily using binary variables. However, it may be that more data is needed to fit these models. Finally, it should be investigated whether including the effect of solar irradiation on the electricity demand can improve the predictions.

Chapter 6

Model Predictive Control

6.1 ControlModel.R setup

Make block diagram, 3 blocks, each represents 1 of the functions *updateData*, *estimate*, and *predict*.

Updatedata takes measurements and weather forecasts coming in at time t_k and the SOC setpoint from the MPC that was computed at time t_{k-1}. Then using the measured load and PV production, it can compute how much electricity has to be bought/sold from the grid, and how much the battery is charged/discharged. From these flows and the real electricity price at this time, we can compute the real costs of this time step.

Next, *Updatedata* forecasts the PV production and load of the next N hours, given the newest measurements and weather predictions. This forecast step also updates the RLS parameters of both models.

The function *estimate is not used*, as the model is updated in updatedata as well. The function could be of use in a later version of the setup, to re-estimate the transformation parameters and the forgetting factor.

Then the *predict* function uses the forecasts to compute optimal battery charging/discharging and corresponding optimal grid imports/exports for the next N time steps.

Time series of computed variables of interest are stored to allow for analysis after the simulation. These variables are:

- Predicted and realized import/export from the grid
- Predicted and realized SOC of the battery (these may differ due to constraints on the power import/export)
- Predicted and realized costs
- deal with missing values. As it is implemented now, the MPC will only run if for all forecasts (PV production, load, spotprice) are available for all

- battery simulation: now the MPC is followed by fixing the set battery volume and adapting grid imports to that. could do it otherwise

6.2 Introduction

Consider a system consisting of a photovoltaic (PV) and a battery, which is also connected to the electrical grid. Utilizing the PV, battery and grid we are interested in fulfilling the demand for electricity in a household in the cheapest possible way. The prices for buying from the electrical grid varies with time to reflect the varying cost of producing electricity or possibly as a result of time-varying environmental impacts. It is also possible to sell electricity to the grid, but for a lower price than buying from it. This problem can be formulated in the following way

$$\text{Minimize} \quad \sum_{k=1}^{N} \left(\lambda_k g_k^- - \tau\lambda_k g_k^+\right) \tag{6.1a}$$

$$\text{subject to}_{1\leq k\leq N} \quad d_k = p_k + b_k^- - b_k^+ + g_k^- - g_k^+, \tag{6.1b}$$

$$b_k = b_{k-1} + c_B b_k^+ - b_k^-, \tag{6.1c}$$

$$0 \leq b_k \leq b_{\max}, \tag{6.1d}$$

$$0 \leq b_k^+ \leq b_{\max}^+, \tag{6.1e}$$

$$0 \leq b_k^- \leq b_{\max}^-, \tag{6.1f}$$

$$g_k^-, g_k^+ \geq 0. \tag{6.1g}$$

Here k is a time index, and so 6.1b to 6.1g has to be valid for $k \in \{1,2,...,N\}$. Equation 6.1a gives the total cost of fulfilling the demand for the next N time steps, in time step k it is possible to buy electricity for λ_k and sell electricity for $\tau\lambda_k$. g_k^- and g_k^+ are the amount of electricity bought and sold from/to the grid. Equation 6.1b expresses how the demand, d_k, has to be fulfilled by a combination of PV production, p_k, battery charge and discharge, b_k^- and b_k^+, and the grid. Equations 6.1c to 6.1f describes the physical constrains of the battery. b_k is the amount of electricity stored in the battery, which according to 6.1d can at most be $b_{\max}$, in particular b_0 is the amount of electricity in the beginning of the time horizon. c_B is the coefficient of efficiency of the battery. It is limited how fast the battery can be charged and discharged with a maximum of $b_{\max}^+$ and $b_{\max}^-$ respectively as described by 6.1e and 6.1f respectively.

Since this formulation only considers a time horizon of N time steps it sees no value in having stored electricity in the battery at time N. Subsequently the solution is always going to empty the battery, which might not be the best thing to do in the long run. To avoid this we can add an extra term in 6.1a that describes the value of the electricity stored in the battery in the end of the time horizon, $-\lambda_B b_N$. λ_B should be chosen so that it matches the value of one unit of stored electricity.

We want to rewrite the problem in the standard form, which does not include any equalities. We have two of these, namely 6.1b and 6.1c. Fortunately we can change 6.1b to an inequality where we require the right-hand side to be larger than the left-hand side. This is possible since (if we assume positive electricity prices) it is never profitable to make the right-hand side larger

than the left-hand side, and so the solution to the problem is going to yield an equality anyways. When it comes to 6.1c we can write b_k without using b_{k-1}:

$$b_k = b_0 + \sum_{i=1}^{k} \left(c_B b_i^+ - b_i^- \right).$$

This means that we do not have to include the equation governing the evolution of the electricity in the battery, but can insert the above expression directly in 6.1d. This rids us of another equality.

We add the value of stored electricity to the objective function and rewrite the equalities to get

$$\text{Minimize} \quad \sum_{k=1}^{N} \left(\lambda_k g_k^- - \tau \lambda_k g_k^+ \right) - \lambda_B \left(b_0 + \sum_{i=1}^{N} \left(c_B b_i^+ - b_i^- \right) \right) \tag{6.2a}$$

$$\text{subject to}_{1 \le k \le N} \quad d_k \le p_k + b_k^- - b_k^+ + g_k^- - g_k^+, \tag{6.2b}$$

$$0 \le b_0 + \sum_{i=1}^{k} \left(c_B b_i^+ - b_i^- \right) \le b_{\max}, \tag{6.2c}$$

$$0 \le b_k^+ \le b_{\max}^+, \tag{6.2d}$$

$$0 \le b_k^- \le b_{\max}^-, \tag{6.2e}$$

$$g_k^-, g_k^+ \ge 0. \tag{6.2f}$$

Notice that we can remove b_0 from the objective function since it is just a constant.

The standard form of linear optimization programs is given by

$$\underset{x}{\text{Minimize}} \quad f^\top x \tag{6.3a}$$

$$\text{subject to} \quad Ax \le b, \tag{6.3b}$$

$$x_{\min} \le x \le x_{\max}. \tag{6.3c}$$

Here x is all the decisions to be made in a time horizon, which in our case is the electricity bought/sold to the grid and the charging/discharging of the battery. Each of these decisions have to be made for all time steps. This can be vectorized by:

$$G^- = \begin{bmatrix} g_1^- & g_2^- & \dots & g_N^- \end{bmatrix}, \quad B^- = \begin{bmatrix} b_1^- & b_2^- & \dots & b_N^- \end{bmatrix},$$
$$G^+ = \begin{bmatrix} g_1^+ & g_2^+ & \dots & g_N^+ \end{bmatrix}, \quad B^+ = \begin{bmatrix} b_1^+ & b_2^+ & \dots & b_N^+ \end{bmatrix}.$$

We gather all these decisions in

$$x = \begin{bmatrix} G^- & G^+ & B^- & B^+ \end{bmatrix}^\top.$$

Similarly we can gather the electricity prices, demand and PV:

$$\Lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_N \end{bmatrix}, \quad D = \begin{bmatrix} d_1 & d_2 & \dots & d_N \end{bmatrix},$$
$$P = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}.$$

It is now clear that in the standard formulation we have to let

$$f^\top = \begin{bmatrix} \Lambda & -\tau\Lambda & \lambda_B 1^\top & -\lambda_B c_B 1^\top \end{bmatrix},$$

where 1 is a column vector of 1's with length N. For the constraints we get

$$A = \begin{bmatrix} -I & I & -I & I \\ \underline{0} & \underline{0} & -U & c_B U \\ \underline{0} & \underline{0} & U & -c_B U \end{bmatrix},$$

$$b = \begin{bmatrix} P - D & (b_{\max} - b_0)1^\top & b_0 1^\top \end{bmatrix}^\top.$$

Here 0 is similar to 1 but consists of 0's instead of 1's. On the other hand $\underline{0}$ is an $N \times N$ matrix of 0's. I is the $N \times N$ dimensional identity matrix. U is an upper triangular matrix with zeros below the diagonal and ones in and above the diagonal.

Finally

$$x_{\min} = \begin{bmatrix} 0^\top & 0^\top & 0^\top & 0^\top \end{bmatrix}^\top,$$
$$x_{\max} = \begin{bmatrix} +\infty 1 & +\infty 1 & b^-_{\max} 1^\top & b^+_{\max} 1^\top \end{bmatrix}^\top.$$

This linear problem can be solved with standard methods, but obviously the demand and produced PV can not be known in advance. The prices might not be known in advance either. This means that in reality we would have to use forecasts of these, and possibly consider the uncertainty directly in the optimization instead of pretending that the problem is deterministic.

To use this in practice one would have to solve the problem for each time-step, which puts some limitations on the size of N. Seeing that the accuracy of the forecasts goes down with the size of N, the marginal improvements of increasing N vanishes anyways.

Bibliography

[1] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[2] Trevor Hastie, Tibshirani Robert, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY, 2009.

[3] Peder Bacher, Henrik Madsen, and Henrik Aalborg Nielsen. Online short-term heat load forecasting for single family houses. In *IECON Proceedings (Industrial Electronics Conference)*, 2013.

[4] Joe-Air Jiang, Jen-Cheng Wang, Kun-Chang Kuo, Yu-Li Su, and Jyh-Cherng Shieh. On evaluating the effects of the incident angle on the energy harvesting performance and MPP estimation of PV modules. *International Journal of Energy Research*, 38:1304–1317, 2014.

[5] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de Pison, and F. Antonanzas-Torres. Review of photovoltaic power forecasting. *Solar Energy*, 136:78–111, 2016.

[6] Frits Møller Andersen, Mattia Baldini, Lars Gårn Hansen, and Carsten Lynge Jensen. Households' hourly electricity consumption and peak demand in Denmark. *Applied Energy*, 208:607–619, 2017.