Bidding Strategy for the Swiss Secondary Control Reserve Market

by

B.S. van der Schaaf

Report submitted to the
Delft Institute of Applied Mathematics
to obtain the degree of

BACHELOR OF SCIENCE
in
APPLIED MATHEMATICS

Thesis Committee
Dr. M.E. Kootte, TU Delft, supervisor
Dr. R.J. Fokkink TU Delft

Delft, The Netherlands
July 2, 2024

TU Delft

Abstract

The rise in renewable energy sources causes more imbalances in the power grid. These imbalances are handled on the secondary control reserve (SCR) market. The prices on the market are currently predominately determined by hydropower plants, making the market unattractive to potential market players. This thesis explores the development of a bidding strategy for these new players to enter the Swiss secondary control reserve (SCR) market. This is a sensitive matter, since bidding too high would result in the bid not being accepted, and bidding too low would mean a player could have earned more money.

Two products are traded on the SCR market: negative control reserve (NCR), activated in case of an overbalance of the grid, and positive control reserve (PCR), activated in case of an underbalance of the grid. To develop a bidding strategy, the NCR and PCR bidding prices were modelled by using an ARIMA model to forecast the next week's maximum bid. The order of the model was selected by minimising the Akaike information criterion and the parameters were estimated by maximising the likelihood function. An ARIMA(1,2,1) model provided the lowest AIC score for both the NCR and PCR data. The accuracy of the models was tested by examining the mean absolute error (MAE), the root mean squared error (RMSE) and the bias. To test the performance of the model on the SCR market, two additional metrics were introduced: the percentage of the bids accepted (PAB), and the percentage of the total potential revenue earned (PMR). Although the MAE, RMSE and bias of the ARIMA(1,2,1) models were low, the PAB and PMR were low as well. This is because both models tended to estimate the forecasts higher than the actual maximum prices, resulting in the bids not being accepted. By shifting the model down to the lower bound of the 95% one-step confidence interval, the PAB and PMR were more than doubled. Therefore, the forecasts of the shifted ARIMA(1,2,1) models, generated the best bidding price for the upcoming week.

Layman Abstract

The rise in renewable energy sources causes more imbalances in the power grid. These imbalances are handled on the secondary control reserve (SCR) market. The prices on the market are currently predominately determined by hydropower plants, making the market unattractive to potential market players. This thesis explores the development of a bidding strategy for these new players to enter the Swiss secondary control reserve (SCR) market. This is a sensitive matter, since bidding too high would result in the bid not being accepted, and bidding too low would mean a player could have earned more money.

Two products are traded on the SCR market: negative control reserve (NCR), activated in case of an overbalance of the grid, and positive control reserve (PCR), activated in case of an underbalance of the grid. To develop a bidding strategy, the NCR and PCR bidding prices were modelled by using a model that could forecast the next week's bidding price. This model was then perfected such that its forecasts could be used to base bids on. That is, new players can bid the price that is predicted by the model. Doing so, the players would have a high chance for their bids to be accepted and would therefore create the most revenue.

Contents

Chapter 1

Introduction

The Renewable Energy Directive, published in 2018 by the EU, established a binding target to increase renewable energy generation by 30 percent by 2030. As of November 2023 this target has been raised to 42.5 percent [1]. Motivated by this binding target, more and more countries are transitioning away from fossil fuels and towards renewable energy sources for their energy mix. So too does Switzerland, showing an increase of 45 percent in the share of renewables in Switzerland's energy consumption, mainly being driven by hydropower [2].

As the share of renewables increases, energy production becomes less predictable, leading to more imbalances of the electricity grid. These imbalances are unfavourable, since the grid only functions properly at a specific frequency. The imbalances of the electricity grid are mitigated by Swissgrid, the Swiss transmission system operator, via a three-step market: the primary, secondary and tertiary control reserve market. The product traded on these markets is reserved electricity capacity which can be activated in case of an imbalance. Primary control reserves are automatically activated within seconds, transitioning to secondary control reserves after several minutes. If the imbalance persists beyond 15 minutes, tertiary control reserves are activated manually [3]. Each of these products is handled on its respective primary, secondary and tertiary control reserve market. Players on the markets are compensated for reserving capacity, and, in case of the secondary and tertiary reserve market, for activation of the reserved electricity [4]. Secondary control reserve (SCR) is generally activated for a longer period of time than primary control reserve (PCR) and is therefore tendered in greater volume than PCR [5]. Furthermore, SCR is compensated for both reservation and activation, whereas PCR is only compensated for reservation of electricity capacity. Tertiary control reserve (TCR) is less frequently activated, since it is only activated for persistent imbalances. The SCR market thus has the potential to generate the most revenue, and will therefore be the focus of this research.

Players on the SCR market make bids consisting of a price and volume. Swissgrid aggregates these bids, arranging volumes from lowest to highest price. It then accepts the lowest bids sequentially until the required volume to mitigate the imbalance is reached. Therefore it is important for players to not bid too high, as their bid might not get accepted. If their bid is accepted, the players are compensated on a pay-as-bid basis.

Secondary control reserve comes in two forms: control reserve to compensate an overbalance, called negative control reserve (NCR), and control reserve to compensate an underbalance, called positive control reserve (PCR). Prior to 2018, NCR and PCR were jointly tendered, meaning bidders had to be able to provide both NCR and PCR. As of 2018, secondary control reserve is tendered separately. This means that market players can now bid only on positive control reserve or negative control reserve [6]. This development allowed new players, who can only bid in one direction to enter the market. Players who can bid in both directions, are restricted to bid in one direction at a time.

Most active players are now hydropower plants, which is why the secondary control market prices are predominantly defined by the opportunity costs of hydropower [7]. Other types of players do not posses this knowledge on hydropower opportunity costs. Therefore, the SCR market is unattractive to enter as a non-hydropower plant. But with the increase of renewables in the Swiss energy mix comes an increase of imbalances of the grid, and thus a need for balancing parties. To encourage new players to enter the market, this paper aims to develop a bidding strategy that maximises the acceptance rate while also maximising the revenue, based purely on what is available to these non-hydropower plant potential players, namely historic data on the maximum accepted bid price. The research question is thus as follows: *what bidding strategy maximises the acceptance rate and potential revenue of players on the SCR market, based on the historic data of the maximum accepted bid prices?*

The question is answered by first implementing a model to forecast the maximum accepted bid price for the positive and negative control reserve, based on historic data provided by Swissgrid. The statistical model used in this research is an autoregressive integrated moving average (ARIMA) model. ARIMA is a classical time series model known for its precision and accuracy in forecasting economic time series [8]. After constructing the models for the NCR and PCR prices, the models are tested on their performance in terms of acceptance rate and potential revenue. Thirdly, the confidence intervals of the models are examined to optimise the model's acceptance rate and payout. Lastly a bidding strategy is formalised for players being able to only provide NCR and PCR, as well as a bidding strategy for players being able to provide both.

Chapter 2 elaborates on the mathematical theory behind the research. Chapter 3 offers a more thorough explanation of the Swiss' energy market structure. In Chapter 4, the methodology used for this research is described and substantiated. Chapter 5 analyses the maximum accepted bid price data to correctly build a model for its forecasts. In Chapter 6 and Chapter 7 the models for respectively the NCR prices and PCR prices are constructed and tested. Chapter 8 finally formalises a bidding strategy that can be used by new market players.

Chapter 2

Mathematical Theory

This chapter gives an overview of the mathematical theory that is used in this research paper. There is a great deal of literature available on theory of time series analysis. Different books and studies however, use different notation for this theory. The idea behind this chapter is to summarise the theory used in a precise and unambiguous way and to substantiate the mathematical context of the research. The reader may be so free to use this chapter as a cheat sheet throughout the rest of the paper. The theory described in this chapter is mainly based on the writings of Heij et al. [9], Shumway and Stoffer [10] and Hyndman and Athanasopoulos [11].

2.1 Basic notions in time series analysis

This section enumerates various basic notions in time series analysis. Section 2.1.1 explains some classic properties from probability theory, applied to time series analysis. Section 2.1.2 discusses the concept of stationarity, which will prove to be important for AR(I)MA models.

2.1.1 Notions from probability theory

A time series is a collection of $n > 0$ random variables observed at time points $t_1, t_2, ..., t_n$. A time series can be described by its joint distribution function:

$$F_{t_1,t_2,...,t_n}(c_1, c_2, ..., c_n) = P(x_{t_1} \leq c_1, x_{t_2} \leq c_2, ..., x_{t_n} \leq c_n)$$

In practice this joint distribution function is a cumbersome tool for analysing time series due to the large amount of variables (n) it takes in. Oftentimes the marginal distribution function

$$F_t(x) = P(x_t \leq x) \tag{2.1}$$

along with its corresponding marginal density function

$$f_t(x) = \frac{\mathrm{d}F_t(x)}{\mathrm{d}x}$$

is used to examine the marginal behaviour of a series. Many notions from classical probability theory can be translated to fit time series analysis. The most important notions for this research are listed below.

Mean function

The mean function μ_t is described by

$$\mu_t = \mathrm{E}(x_t) = \int_{-\infty}^{\infty} x f_t(x) dx \tag{2.2}$$

where E denotes the usual expected value operator.

Autocovariance and variance

The autocovariance function is defined as

$$\gamma(s,t) = \mathrm{cov}(x_s, x_t) = \mathrm{E}[(x_s - \mu_s)(x_t - \mu_t)] \tag{2.3}$$

The autocovariance is a measure for the linear dependence between the observations x_t and x_s of the same time series. If $\gamma(s,t) = 0$ then x_s and x_t are not linearly related, though they may be related in a different dependence structure. By setting $s = t$ the variance of a time series is obtained (Eq. 2.4).

$$\gamma(t,t) = \mathrm{E}[(x_t - \mu_t)^2] = \mathrm{var}(x_t) \tag{2.4}$$

Autocorrelation function

The autocorrelation function (ACF) of a time series is defined as

$$\rho(s,t) = \mathrm{corr}(x_s, x_t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} \tag{2.5}$$

and describes how well a time series at time s, x_s, can be predicted by a previous value of the time series, x_t, also known as a lag.

Partial autocorrelation function

The partial autocorrelation function (PACF) of a time series is defined as

$$\phi(s,t) = \mathrm{corr}(x_s, x_t | x_{s-1}, x_{s-2}, ..., x_{t+1}) \tag{2.6}$$

The partial autocorrelation measures how well x_s can be predicted by its lag x_t, given their intermediate values.

2.1.2 Stationarity

Many forecasting models only function properly if the time series shows regular behaviour [12]. A formal definition of this regular behaviour is given by the notion of weak stationarity. A time series is called weakly stationary if

(i) the mean function, μ_t (Eq. 2.2), is constant and does not depend on t,

(ii) the variance (Eq. 2.4) is constant and does not depend on time t, and

(iii) the autocovariance (Eq. 2.3) depends on k and t only through their difference $|k-t|$.

For the ease of reading, the term stationary will henceforth be used to mean weakly stationary. In case of a stationary time series the mean function will be denoted by $\mu = \mu_t$. For stationary time series the notation of the autocorrelation function can be simplified by setting $s = t + k$.

$$\gamma(t+k,t) = \text{cov}(x_{t+k}, x_t) = \text{cov}(x_k, x_0) = \gamma_k$$

Using the above expression, the autocorrelation function of a stationary time series can be written as

$$\rho_k = \text{corr}(x_{t+k}, x_t) = \frac{\gamma_k}{\gamma_0} \tag{2.7}$$

The partial autocorrelation function of a stationary time series can be written as

$$\phi(t+k,t) = \text{corr}(x_{t+k}, x_t | x_{t+k-1}, x_{t+k-2}, ..., x_{t+1}) = \phi_k \tag{2.8}$$

In case of unstationary time series, differencing can be useful to obtain stationarity. The differencing process, is defined as taking two consecutive points of a time series and computing their difference. A differenced time series, denoted by Δx_t, can be computed as follows.

$$\Delta x_t = x_t - x_{t-1} \tag{2.9}$$

Differencing a time series, x_t, of N data points yields a time series, Δx_t, of $n - 1$ data points since it is not possible to calculate a difference for the first observation. If the time series is not stationary after differencing once, it can be differenced multiple times to obtain stationarity. Differencing a time serie d time series is denoted by $\Delta^d x_t$. The time series is then called d-order differenced.

2.2 ARIMA models

Autoregressive integrated moving average (ARIMA) models are commonly used in time series forecasting. Section 2.2.1 elaborates on the structure of autoregressive, moving average and ARIMA models. Section 2.2.2 explains how the parameters are estimated for the model and section 2.2.3 describes how to choose the order of an ARIMA model and therefore the best suitable model. Section 2.2.4 describes the accuracy metrics used to evaluate the performance of the model.

2.2.1 ARIMA model structure

An ARIMA model is a combination of an autoregressive and moving average model. The "integrated" part of ARIMA refers to the differencing of the data to obtain stationarity.

Autoregressive models

In autoregressive models, the dependent variable x_t can be explained by a linear combination of its past values x_{t-i}.

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t \tag{2.10}$$

where x_t represents the value to be predicted at time t. x_t is determined by a linear combination of p of its previous values x_{t-i}, commonly known as lags. ϕ_i are the parameters determined during model fitting, which will be discussed in section 2.2.2. The white noise term ϵ_t is assumed to have a normal distribution with mean zero and variance σ_ϵ^2 and the property that $\mathrm{E}[\epsilon_t x_{t-k}] = 0$. Furthermore, x_t has mean zero and is assumed to be stationary.

If the mean μ is not zero, x_i can be replaced by $x_i - \mu$:

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + \cdots + \phi_p(x_{t-p} - \mu) + \epsilon_t$$

which can be rewritten as:

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t \tag{2.11}$$

where $\alpha = \mu(1 - \phi_1 - \phi_2 - \cdots - \phi_p)$. An autoregressive model using p lags is denoted by AR(p), where p is said to be the lag order.

The following theory can be useful to determine if a time series behaves like an AR(p) model.

Theorem 1. *For the partial autocorrelation of an AR(p) process holds the following property*

$$\phi_k = 0 \text{ for all } k > p$$

The proof of this theorem can be found in Appendix A.1.

Moving average models

An autoregressive model assumes that the forecasted value x_t is a product of a linear combination of p of its lags. A moving average process of order q, however, can be described by a linear combination of white noise terms:

$$x_t = \alpha + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \tag{2.12}$$

where $\epsilon_j \sim \mathcal{N}(0, \sigma_\epsilon^2)$, α a constant and $\theta_1, ..., \theta_q$ parameters.

A moving average model of order q is denoted by MA(q). A MA(q) process is always stationary. The following theorem can be useful to determine if a time series behaves accordingly a moving average model.

Theorem 2. *For the autocorrelation function of an MA(q) process holds the following property*

$$\rho_k = 0 \text{ for all } k > q$$

The proof of the theorem can be found in Appendix A.2.

Combining autoregressive and moving average models: AR(I)MA

Combining the autoregressive model (eq. 2.11) and the moving average model (eq. 2.12), an autoregressive moving average (ARMA) model is obtained. This model utilises the property of an AR model that future values can be determined, based on past values, and the property of an MA model, that uses previous errors to account for future errors. ARMA models are of the form

$$x_t = \alpha + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \tag{2.13}$$

where α, as in equation 2.11 accounts for a nonzero mean. As in the previous models, x_{t-i} are the lags, ϕ_i and θ_j are the parameters determined during the fitting of the model and $\epsilon_j \sim \mathcal{N}(0, \sigma_\epsilon^2)$ the white noise terms. An ARMA model considering p lags and q white noise terms is denoted as ARMA(p,q), where p and q are referred to as the order. By observing equation 2.13, it can be seen that autoregressive and moving average models are special cases of ARMA models. By setting q to zero, the moving average part of the ARMA model is eliminated, resulting in an autoregressive model of order p. Similarly, by setting p to zero, the autoregressive part is deleted, leading to an moving average model of order q.

An autoregressive integrated moving average model or ARIMA(p, d, q) model is similar to an ARMA model. They only differ in their requirement for stationary time series. An ARMA model requires a stationary time series as input, whereas an ARIMA model also allows non-stationary time series. An ARIMA(p, d, q) model takes a non-stationary time series as input and differences it d times to make it stationary. Based on the differenced, stationary time series, an ARMA model is constructed accordingly to equation 2.13.

2.2.2 Parameter estimation

If a time series x_t behaves like an ARIMA(p,d,q) process, where p, d, q are known, it can be written as

$$\Delta^d x_t = \alpha + \phi_1 \Delta^d x_{t-1} + \cdots + \phi_p \Delta^d x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \tag{2.14}$$

where $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$ a constant, $\phi_1, ..., \phi_p, \theta_1, ..., \theta_q$ parameters and $\{\epsilon_j\} \sim \mathcal{N}(0, \sigma_\epsilon^2)$.

To fit an ARIMA model on a time series, $\{\phi_1, ..., \phi_p, \mu, \theta_1, ..., \theta_q, \sigma_\epsilon^2\}$ need to be determined. This can be done by several methods. The method used in this paper is the method of maximum likelihood. The following notation is used.

$$\begin{aligned} \boldsymbol{\theta}^T &= (\phi_1, ..., \phi_p, \mu, \theta_1, ..., \theta_q) \\ \mathbf{X}_t^T &= (x_t, x_{t-1}, ..., x_1) \end{aligned}$$

The likelihood function of an ARIMA model [13] is given by

$$L(\mathbf{X}_N : \boldsymbol{\theta}, \sigma_\epsilon^2) = f(\mathbf{X}_N | \boldsymbol{\theta}, \sigma_\epsilon^2) \tag{2.15}$$

The likelihood function gives the probability that the observed data comes from the estimated model. The parameters that are obtained by maximising the likelihood function are denoted by $\{\hat{\phi_1}, ..., \hat{\phi_p}, \hat{\mu}, \hat{\theta_1}, ..., \hat{\theta_q}, \hat{\sigma_\epsilon}^2\}$. The predicted values resulting from an ARIMA model using the estimated parameters are denoted by $\hat{x_t}$, that is

$$\widehat{\Delta^d x_t} = \hat{\alpha} + \hat{\phi}_1 \Delta^d x_{t-1} + \cdots + \hat{\phi}_p \Delta^d x_{t-p} + \epsilon_t + \hat{\theta}_1 \epsilon_{t-1} + \cdots + \hat{\theta}_q \epsilon_{t-q} \tag{2.16}$$

where $\hat{\alpha} = \hat{\mu}(1 - \hat{\phi_1} - \cdots - \hat{\phi_p})$ and $x_{t-1}, ..., x_{t-p}$ observed values.

Deriving the likelihood function and maximising it to find the model parameters is a tedious task that is usually done by computers. Therefore the derivation and maximisation of the likelihood function will not be discussed in any further detail.

2.2.3 Akaike's information criterion

In section 2.2.2 the order p, d, q of the model was assumed to be known. In practice this is usually not the case. Akaike's information criterion (AIC) offers a measure to choose a model out of a set of possible models with varying order. The AIC is defined as

$$\text{AIC}(k) = -2\ \log(L) + 2k \tag{2.17}$$

where L denotes the maximum likelihood function and k the amount of parameters estimated. The larger the maximum likelihood and the lower the number of parameters, the lower the AIC. Good models are obtained by minimising the AIC. That is, finding a good balance between a high likelihood function and a low number of parameters. The AIC for an ARIMA(p, d, q) model is given by

$$\text{AIC}(k) = -2\ \log(L(\mathbf{X}_N : \hat{\boldsymbol{\theta}}, \hat{\sigma_\epsilon}^2) + 2(p + q + l + 1) \tag{2.18}$$

where $l = 0$ if $\mu = 0$, since μ then needs not to be estimated, and $l = 1$ if $\mu \neq 0$.

2.2.4 Accuracy metrics for model performance

Accuracy metrics are used to determine the accuracy of a models predictions. The three classical metrics used to evaluate the performance of the models treated in this research paper are the mean absolute error (Eq. 2.19), the root mean square error (Eq. 2.20) and the bias (Eq. 2.21).

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{x_i} - x_i| \tag{2.19}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{x_i} - x_i)^2} \tag{2.20}$$

$$Bias = \frac{1}{N} \sum_{i=1}^{N} \hat{x_i} - x_i \tag{2.21}$$

Here, N is the number of observations, x_i the observed value and $\hat{x_i}$ the predicted value.

Equation 2.19 shows the mean absolute error, as defined as the average value of all forecasting errors in absolute value. The mean absolute error gives a sense of the overall performance of the model. All outliers are weighted equally, therefore it is not sensitive to large outliers. Since the errors are averaged in absolute value it also does not disclose any information about whether or not the model has a bias in forecasting, meaning the forecasts are systematically too high or too low. The RMSE (Eq. 2.20) is more sensitive to outliers due to the squaring process. By summing over the difference between the observed and predicted values instead of the absolute difference like the MAE, the bias (Eq. 2.21) gives an indication of the model's tendency to systematically underestimate or overestimate.

2.3 Forecasting using ARIMA models

This section describes how one-step forecasting can be performed using an ARIMA model (sec. 2.3.1) and how the one-step confidence interval can be determined (sec. 2.23).

2.3.1 One-step forecasting

If a proper ARIMA model for time series data is established, the model can be used to forecast future values. If $x_t, x_{t-1}, ..., x_1$ are the observed values and one is interested in forecasting the future value after one time steps, by use of an ARIMA model, $\hat{x_{t+1}}$ can be written as

$$\widehat{\Delta^d x_{t+1}} = \hat{\alpha} + \hat{\phi}_1 \Delta^d x_t + \cdots + \hat{\phi}_p \Delta^d x_{t-p+1} + \epsilon_{t+1} + \hat{\theta}_1 \epsilon_t + \cdots + \hat{\theta}_q \epsilon_{t-q+1} \tag{2.22}$$

Since the observations $x_t, x_{t-1}, ..., x_1$ are known, the only values on the right hand side that are unknown are $\epsilon_{t+1}, \epsilon_t, ..., \epsilon_{t-q+1}$. The future error e_{t+1} is replaced by zero, since it cannot be determined at this point. The past error terms $\epsilon_t, ..., \epsilon_{t-1+1}$ are replaced by their corresponding residuals (difference in observed value and by model predicted value).

2.3.2 Confidence intervals

Recall that the white noise terms were assumed to be normally distributed with mean zero and standard deviation σ_ϵ^2, which was estimated by $\hat{\sigma}_\epsilon^2$. To determine the confidence interval of confidence level $1 - \alpha_0$, first the z-score $\xi_{1-\alpha_0/2}$ is derived from the standard normal distribution table. The z-score gives the number of standard deviations the forecasted value deviates from the actual observed value. The margin of error is then calculated by $E = \xi_{1-\alpha_0/2} \cdot \hat{\sigma}_\epsilon$.The confidence interval around the forecasted value $\hat{x_t}$ is then given by

$$[\hat{x_t} - \xi_{1-\alpha_0/2} \cdot \hat{\sigma}_\epsilon, \hat{x}_t + \xi_{1-\alpha_0/2} \cdot \hat{\sigma}_\epsilon] \tag{2.23}$$

This is the region in which the actual (to be) observed value x_t lies with a certainty of level $1 - \alpha_0$.

Chapter 3

Swiss Energy Market Background

Electricity is transported from producers to consumers via a transmission grid. In Europe this transmission grid functions at a frequency of 50 Hertz [6]. Power plants connected to the grid are designed to operate within a specific range of this frequency. If the supply surpasses the demand, the grid frequency increases. Significant increases in frequency can stress and potentially damage the equipment. That is why generators are automatically disconnected, which leads to partial black-outs. If the consumers demand surpasses the supply, power plants are not able to meet the consumers demand, resulting in power black-outs. Therefore, the frequency of the transmission grid must be kept stable at 50 Hertz at all times.

This leads to some particularities in the energy market structure. J. Abrell [4] provides an overview of the Swiss energy market's structure: electricity transmission grids are operated by a so-called Transmission System Operator (TSO). In Switzerland, the TSO is Swissgrid. Swissgrid is responsible for the transport of energy in the form of electricity and natural gas from suppliers to consumers through the transmission grid. Electricity generators and consumers are organised into so-called balancing groups (BGs). A BG is a group of various generators and consumers, represented by a single balancing responsible party (BRP). The BRP measures the consumption and generation of electricity at the BG level and communicates this to the TSO.

To account for the need for stability, the electricity market is organised into two different types of sub-markets: energy-only markets (EOM) and imbalance markets (IBM). An overview of the Swiss energy market's structure is depicted in Figure 3.1. Swissgrid uses electricity demand forecasts to determine the amount of energy that needs to be traded on the EOM to balance the grid. Most times, the expected demand and supply deviate from the actual demand and supply, resulting in an imbalance. These imbalances can be caused by, for example, a power plant shutting down. They can also be due to the intermittent character of renewable energy sources, making it harder to forecast their energy supply. The market actors responsible for causing an imbalance are fined in a process called imbalance pricing. Mitigating the imbalance is handled by the imbalance markets (IBM).

The EOM is further organised into sub-markets distinguished by their gate closure time, i.e. the point in time the market is cleared prior to delivery time. Amongst these markets are the day-ahead market (DAM), cleared a day before delivery, and the intraday market (IDM), cleared on the day of delivery. The product traded on the EOM is electricity itself. The product traded on the IBM is reserved electricity capacity, which can be activated in case of an imbalance. The IBM is organised in three markets: one for primary control reserve (PCR), secondary control reserve (SCR) and tertiary control reserve (TCR). In case of an imbalance, PCR is activated automatically. Within minutes, PCR is replaced by secondary control reserves, restoring the frequency of 50 Hertz. If the imbalance lasts longer than 15 minutes, TCR is activated manually. Providers are paid for reserving the electricity capacity. In addition, if secondary or tertiary reserve are activated, providers receive compensation.

There are about fifteen players active on the SCR market, of which 95 percent are hydropower plants. The bidding is done via a weekly auction, meaning power plants must ensure the availability of the power for one week. As of 2018, secondary control reserve is procured as asymmetric product [6], meaning that overbalances and underbalances are handled separately, by tendering negative control reserve (NCR), used to compensate for an overbalance, and positive control reserve (PCR), used to compensate for an underbalance. Swissgrid may require both positive and negative reserve at a time. Players, however, are restricted to bidding in one direction at a time. Market players submit bids in the form of price and capacity volume. Swissgrid aggregates the bids, arranging the volumes from lowest to highest price. Swissgrid then accepts the lowest bids sequentially until the required volume to mitigate the imbalance is reached. Compensation of the providers is done on a pay-as-bid basis, meaning providers get paid the amount they have bidden. If a bid is accepted, the plant must deliver. The volume demanded by Swissgrid is not known to the players.

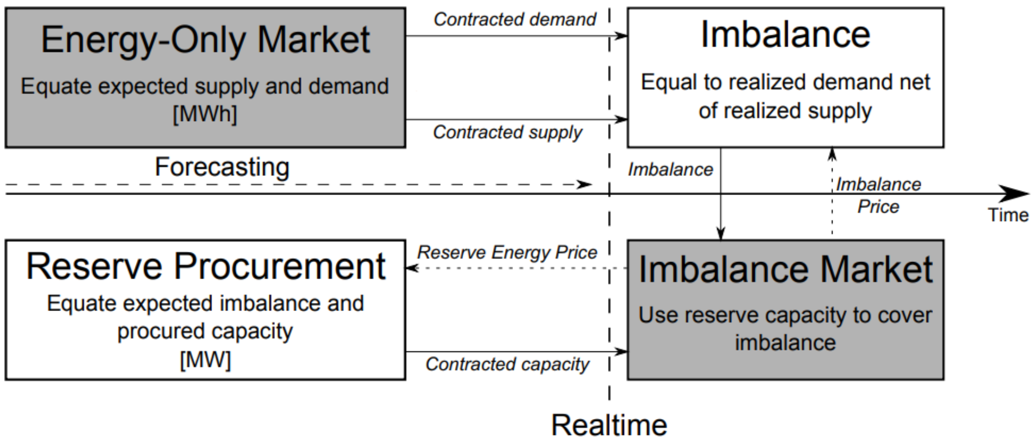


Figure 3.1: Overview of the Swiss energy market structure [4]

Chapter 4

Methodology

This chapter explains and substantiates the research methodology. Section 4.1 discusses the model that is used to forecast the data, namely an autoregressive integrated moving average (ARIMA) model. Section 4.2 discusses the method of analysing and preparing the data prior to modelling. There are several ways to determine the orders of an ARIMA model. Section 4.3 elaborates on the method that is handled in this paper. Section 4.4 discusses how the parameters are estimated for the model. Section 4.5 gives the step-by-step instruction of how a series of one-step forecasts is generated to test the accuracy of the model. The accuracy metrics used to test the accuracy are illustrated in section 4.6. The 95% confidence interval of the model will be examined. Section 4.7 explains why this is done, and why in particular the 95% confidence interval is examined and not intervals of lower certainty levels.

4.1 ARIMA

An autoregressive integrated moving average model forecasts future outcomes purely based on historical time series. It uses the assumption of autoregression, where future values can be described as linear combination of its lags. In addition, the moving average part utilises previous errors to account for future errors. The integrated part allows the modelling of non-stationary time series [14]. ARIMA models do not make any assumptions about the distribution of the data, making it applicable to data with any kind of distribution. ARIMA models are more suitable for short-term forecasting, since their accuracy tends to reduce for longer forecast horizons. Another downside is the model's sensitivity to outliers and its shortcomings in predicting turning points in time series [15].

ARIMA models have a wide range of applications, many of which find its place in price forecasting. Gao et al. [16] compare the performance of an ARIMA model and an artificial neural network (ANN) on the forecasting of electricity prices for power markets. The results show that the ARIMA model gives greater results than the ANN. Zhou et al. [17] use an ARIMA model to forecast the electricity spot market price. They furthermore extend their ARIMA model with error correction to increase the model's accuracy.

Due to its proven success in electricity price forecasting, its usefulness for short-term forecasting – as is the case in this research – and its requirement for only historical data, the ARIMA model is used to model and forecast the maximum bidding price of the SCR market in this research.

4.2 Data analysis and preprocessing

Prior to modelling, the data is analysed first to spot any notable behaviour of the time series. For instance, if the time series shows seasonal behaviour, the ARIMA model has to be extended to a seasonal ARIMA model [11]. The data analysis is done by plotting the data and visually spotting trends, seasonal components, and outliers.

After analysing the data, it must be preprocessed to properly fit an ARIMA model. Outliers are removed to ensure smoother forecasting and transformations of the data are made to stabilise the variance. In addition, the data is differenced, to examine if doing so makes the data stationary.

The last step in preparing the data for modelling, is splitting it into a training and test set. The training set is used to "train" the model, that is, determining the order and parameters of the model. The test set is used to test the predictive accuracy of the model. The training set should be sufficiently large to train the model but leave enough data for the test set to assess the performance of the model. A common split assigns 80 percent of the data to the training set and 20 percent of the data to the test set [18]. This split offers sufficient data to train the model. The forecast of the maximum bidding price is only needed for one week. Therefore, the test set should also be large enough to evaluate the model's performance, given the short forecasting horizon.

4.3 Order selection

Selecting the right order p and q of an ARIMA(p, d, q) model is a well-studied topic in statistics. In this research, the order is selected by means of the Akaike information criteria (AIC) [9]. A collection of ARIMA models with different orders, is tested. This selection is composed by setting a maximum order and, subsequently, testing all models with order less than or equal to the maximum order. From the model collection, the model with the lowest AIC score is then selected. Recall from section 2.2.3 that the AIC is a trade-off of the fit and complexity of the model. Therefore choosing the model with the lowest AIC score provides a good fit for the model while avoiding overfitting. The AIC score is chosen as criterion over other model selection information criteria, such as the Bayesian information criterion (BIC), since it allows more complex models while preventing overfitting [19].

As for the order d of differencing, this is determined by examining how many times the time series needs to be differenced to become stationary. This examination is done by iteratively differencing the time series until stationarity is obtained.

4.4 Parameter estimation

Shumway and Stoffer [10] present different methods of parameter estimation. A frequently used method, is the one of maximum likelihood, i.e. maximising the likelihood function (Eq. 2.15). The ARIMA function in Python estimates the parameters by the method of maximum likelihood as well [20]. Therefore, this method is handled in this research. Deriving and maximising the likelihood function by hand is beyond the scope of this research, and will therefore not be discussed any further.

4.5 Forecasting

To test the accuracy of the model, a series of forecasts is generated over the domain of the test data. The forecasts are then compared to the actual data to determine the model's accuracy (see sec. 4.6). Since the forecasting horizon of the model is one week, the following is done to create the forecasts for the test part of the data:

1. The model is trained on the training data.
2. Using the trained model, a one-step forecast is calculated (Eq. 2.22).
3. The maximum accepted bid of the current week is made public after market clearance. To calculate the forecast of the next step, the forecasted value of the current week is replaced by the actual maximum accepted bid price.
4. Using the original model, – that is, not re-training the model using the new data point of the current week – the next one-step forecast is generated.
5. Step 3 and 4 are iterated over the domain of the test data to create a series of one-step forecasts.

4.6 Accuracy metrics

To assess the accuracy of the model, five accuracy metrics are used. Three of these are classical accuracy methods and have already been mentioned briefly in section 2.2.4, namely the mean absolute error (MAE), root mean squared error (RMSE) and bias.

Equation 2.19 shows the mean absolute error, as the average value of all forecasting errors in absolute value. The mean absolute error provides a sense of the overall performance of the model. All outliers are weighted equally, therefore it is not sensitive to large outliers. Since the errors are averaged in absolute value, it also does not disclose any information about whether or not the model has a bias in forecasting, meaning the forecasts are systematically too high or too low.

Large outliers in this forecasting would mean a bid is either excessively high or low. A bid excessively high would not be accepted and one much too low would mean the BRP could have earned more money for their service. Both cases are undesirable. Therefore, another accuracy metric more sensitive to outliers is used, namely the root mean square error (Eq. 2.20). This accuracy metric is more sensitive to outliers and has proven useful to models whose errors are normally distributed [21], which is an assumption for ARIMA models.

In case of a systematic error, i.e. the model is systematically too high or too low, the forecasted bid would either never be accepted due to the systematically high forecast or the BRP would be underpaid by bidding the structurally under forecasted price. The bias 2.21 is used to detect this phenomenon in the models [22]. As shown in Equation 2.21, the bias averages all forecasting errors. In the case of systematic over- or underforecasting, the bias would respectively take on a strong positive or negative value.

The MAE, RMSE and bias are classic accuracy metrics often used in literature. Since the objective of finding a good fitted model on the data is for a BRP to bid as high as a price as possible while not surpassing the maximum price, two extra accuracy metrics for the purpose of this research are composed. The first metric is called the "percentage accepted bids" (PAB) and returns the percentage of the bids being accepted if one were to bid according to the model. The second additional metric is called the "percentage maximum revenue" (PMR) and returns the percentage of the maximum income a BRP can earn using the model's forecasts. These metrics are introduced to test the model's usefulness on the SCR market. The PAB and PMR are defined as

$$PAB = \frac{\sum_{i=1}^{N} \mathbb{1}_{\hat{x_i} \leq x_i}}{N} \tag{4.1}$$

$$PMR = \frac{\sum_{i=1}^{N} \hat{x_i} \mathbb{1}_{\hat{x_i} \leq x_i}}{\sum_{i=1}^{N} x_i} \times 100 \tag{4.2}$$

where $\hat{x_i}$ denotes the forecasted value, x_i the observed value and N the number of observations. The indicator function, $\mathbb{1}_{\hat{x_i} \leq x_i}$ ensures that only the bids below the observed maximum price x_i are included in the summation.

Using these five accuracy metrics provides a thorough overview of the model's performance.

4.7 Confidence intervals

The aim of the research is to formalise a bidding strategy by utilising one-step forecasts of the maximum price. If one were to bid according to the forecast, the bid might not be accepted if the forecast were even slightly higher than the actual maximum price. To increase the certainty of one's bid getting accepted, the 95% confidence interval is inspected. The 95% confidence interval gives a range around the forecasted value in which the actual value lies with a certainty of 95%. If favourable for the PMR, the model is shifted down to the lower bound of the confidence interval. In that case, there is a 95% chance that the actual value is greater than or equal to the shifted forecasted value and, therefore, for the bid to be accepted. Confidence intervals of lower certainty are not considered. This is because using lower-level confidence intervals, could lead to the bid not to get accepted because the actual value lies below the lower bound. Confidence intervals of higher levels are not considered, since the 95% confidence interval is assumed to provide sufficient certainty of bid acceptance.

Chapter 5

Data

This chapter elaborates on the data used in this research. Section 5.1 offers details on the entire dataset available for this research. Section 5.2 provides a statistical analysis of the data used for the model. Section 5.3 explains the data preparations and modifications applied prior to modelling.

5.1 Definitions of available data

The data used in this research is provided by Swissgrid and consists of two datasets. Both datasets concern biddings on the secondary control reserve market. A bid consists of a price per megawatt and a volume of electricity (in megawatts). As stated in Chapter 3, the secondary control reserve market is enabled in case of an underbalance or an overbalance of the electricity grid. Therefore, biddings can be done in two directions: downbidding, to compensate for an overbalance of the grid, and upbidding, to compensate for an underbalance of the grid. One of the datasets provides statistics on negative control reserve (NCR), used in case of an overbalance, the other one on positive control reserve (PCR), used in case of an underbalance.

The data from both datasets span from 01-01-2018 until 29-03-2020 and provide weekly data on the biddings. The datasets provide the maximum price of the accepted bids, the minimum price of all accepted bids and the average price per megawatt electricity. In addition, they provide information on the capacity of the accepted bids, namely the largest and smallest bid accepted in terms of megawatt, and the average, median and standard deviation of the bids in terms of megawatt. Lastly, both provide the total under- or overbalance of the grid, the total amount paid to the BRP's for compensating the disbalance and the number of bids made. The object of this research is to forecast the maximum price of all accepted bids. Therefore, the data on the maximum accepted bid prices is discussed in more detail in the next sections.

5.2 Statistical analysis

This section is divided in two subsections. Section 5.2.1 gives a statistical analysis of the NCR data and section 5.2.2 covers the PCR data.

5.2.1 Analysis of NCR data

The maximum accepted bid prices of NCR is depicted in figure 5.1, where week 0 corresponds to the first week of 2018.

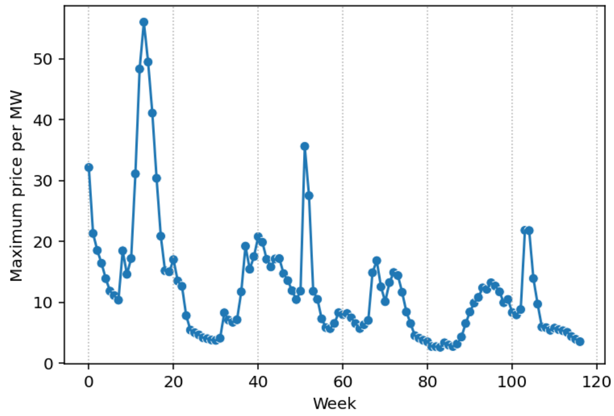


Figure 5.1: Weekly maximum bidding price of NCR

Figure 5.1 shows that the time series shows a downward trend and cyclic behaviour. In case of seasonal data, fluctuations occur in a fixed frequency, that is, in a monthly, quarterly or yearly pattern. In Figure 5.2, the peaks in the data are highlighted with grey. The first peak occurs from around week 8 (end of February 2018) until weak 23 (end of May 2018). The second peak occurs from around week 31 (end July 2018) until week 54 (begin January 2019). The third peak occurs around 66 (begin April 2019) until week 75 (begin June 2019). The last peak occurs around week 89 (mid-September 2019) and lasts until week 107 (mid-January 2020). These peaks do not occur in a monthly, quarterly or yearly fixed frequency.

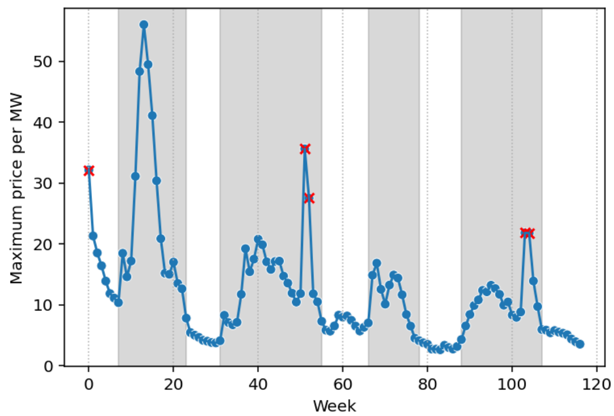


Figure 5.2: Weekly maximum bidding price of NCR, seasonality assessment

The peaks occurring around week 50 and week 100, however, hint at a seasonal pattern. These peaks coincide with the last week of the year and the first week of the new year and are marked with a red 'x' in Figure 5.2. The data point at week 0 (first week of 2018) seems to follow this pattern as well. It is, however, difficult to say with full certainty that these points follow a seasonal pattern due to the small size of the dataset. The peaks could also be outliers.

The downward trend implies that the mean function of the time series is not constant over time. The fluctuations in the time series are dampened over time, implying the variance is not constant as well. Therefore, it needs to be examined if just differencing the data is enough to make it stationary.

5.2.2 Analysis of PCR data

The maximum accepted bid prices of PCR are depicted in figure 5.3, where week 0 is the first week of 2018.

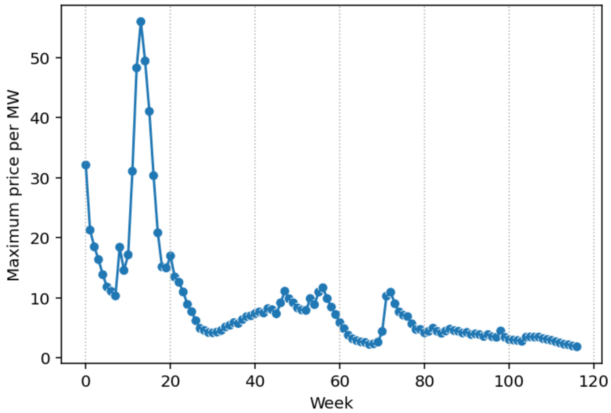


Figure 5.3: Weekly maximum bidding price of PCR

This data too shows a downward trend, implying an inconstant mean function. To assess seasonality the peaks are highlighted with grey in Figure 5.4. The peaks in the data occur around week 12-14 (end of March 2018, beginning of April 2018), around week 47 (half November 2018), in week 55 and 56 (half of January 2019) and from week 71 until week 73 (half of May 2019). Since these peaks do not occur in a seasonal pattern – that is in a monthly, quarterly or yearly pattern –, the data is interpreted to be non-seasonal. Since the fluctuations are not of equal size, the variance is inconstant over time. Therefore, just as for the NCR data, it needs to be examined if just differencing the data is enough to make it stationary.

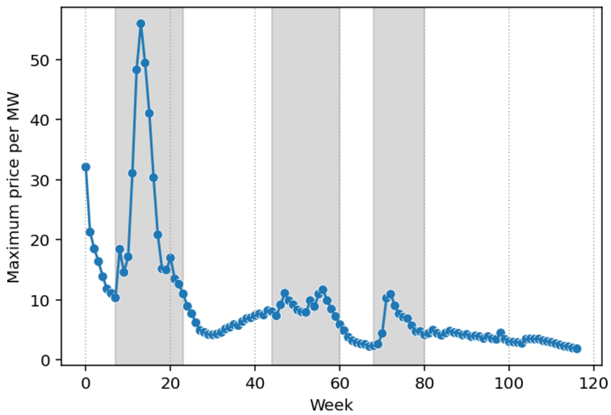


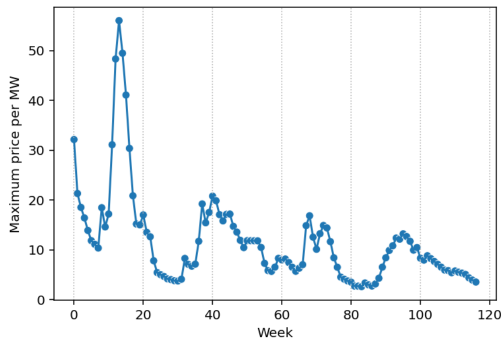
Figure 5.4: Weekly maximum bidding price of PCR, seasonality assessment

5.3 Data preparation

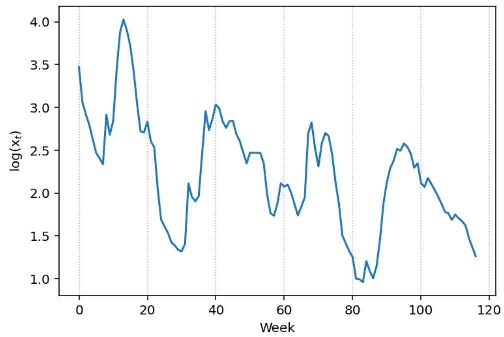
This section discusses the data modifications and preparations necessary for the NCR (sec. 5.3.1) and PCR (sec. 5.3.2) data to be properly modelled.

5.3.1 NCR data preparation

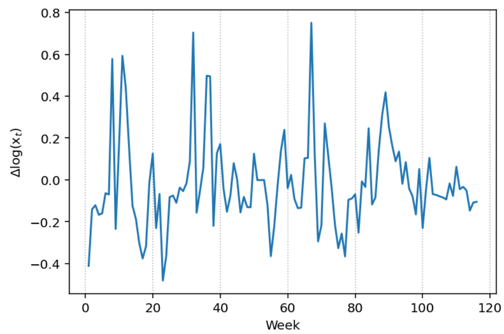
Since it cannot be said with certainty that the peaks around week 52 and week 104 (Fig. 5.1) are due to seasonality, they are treated as outliers. Outliers do not follow the pattern of the data and therefore create noise that is unwanted for proper modelling [23]. To solve this issue, the outliers are deleted from the dataset. The gaps in the dataset are filled in by linear interpolation [24], resulting in Figure 5.5a. To stabilise the variance, a logarithmic transformation [25] is applied to the data, resulting in Figure 5.5b with a more stable variance. To see if the differencing done by the ARIMA model is enough to make the data stationary, the data is differenced (Eq. 2.9) to eliminate trend (Fig. 5.5c). The data, however, still does not appear to be stationary. Differencing the data a second time results in Figure 5.5d. Figure 5.5d shows the time series to have a stable mean and roughly constant variance. Therefore, using the logarithmic transformed data and letting the ARIMA model difference it twice, should result in a proper model.



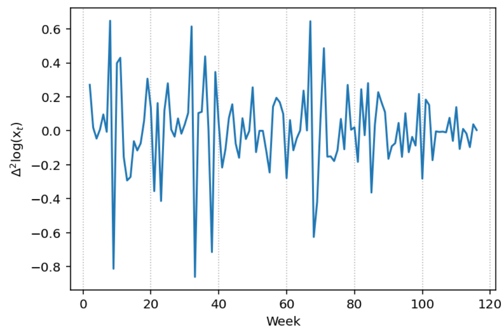
(a) Removal of outliers from NCR data



(b) Logarithmic transformation of data



(c) Differencing data once

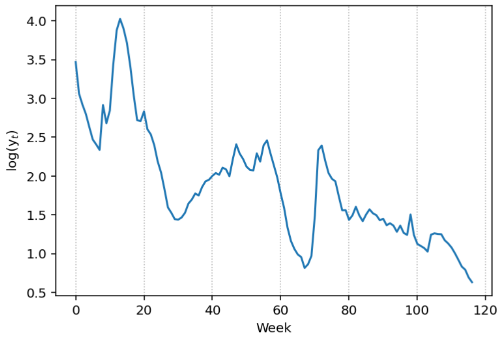


(d) Differencing data a second time

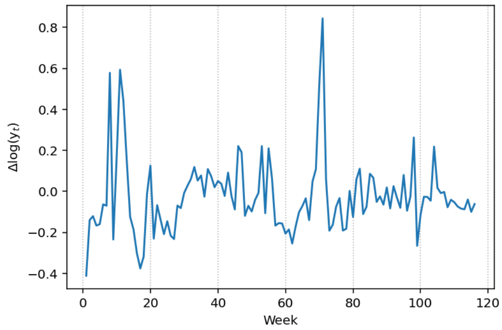
Figure 5.5: NCR data preparation

5.3.2 PCR data preparation

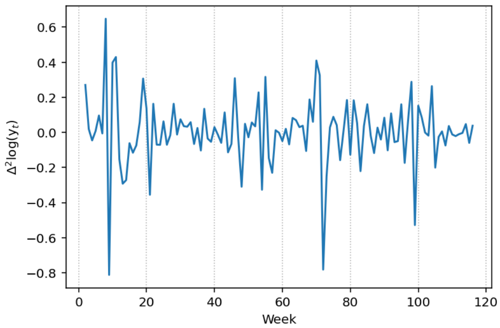
No clear outliers are spotted in Figure 5.3. Therefore the whole dataset will be used for modelling. Similarly to the NCR data, a logarithmic transformation is applied to the PCR data to stabilise the variance (Fig. 5.6a). After which the data is differenced to remove the trend (Fig. 5.6b). Again, the data does not appear to be stationary. Therefore the data is differenced a second time, resulting in Figure 5.6c, showing a time series with constant mean function and roughly constant variance. Therefore, using the ARIMA model on the logarithmic transformed data should generate accurate predictions.



(a) Logarithmic transformation of PCR data



(b) Differencing data once



(c) Differencing data a second time

Figure 5.6: PCR data preparation

Chapter 6

Modelling Negative Control Reserve Bidding Strategy

In this chapter, a bidding strategy is composed for the negative control reserve, using an ARIMA model. Section 6.1 describes the modelling of the maximum accepted bid prices of negative control reserve (NCR). Section 6.2 adapts the model to develop a version whose forecasts can be used for bidding. The Python code used for this chapter can be found in Appendix B.1.

6.1 Modelling maximum accepted NCR bidding prices

This section discusses the modelling of the NCR data, using an ARIMA model. Section 6.1.1 determines the order settings. Section 6.1.2 discusses the implementation of the tested models and selects the best model. The model's performance on forecasting is tested in section 6.1.3.

6.1.1 Order setting

Prior to implementing the ARIMA(p, d, q) model, the orders of the tested models need to be determined. To clarify, using an ARIMA model for the logarithmic transformed data, amounts to the same as using an ARMA model for the logarithmic, twice-differenced data. The differencing order is encapsulated in the integration part of ARIMA, that is, in the order of d. The logarithmic transformation is not encapsulated in the ARIMA model but is used to stabilise the variance. Figure 5.5d shows that the logarithmic transformed, twice differenced data is stationary. Therefore the ARIMA model is used for the logarithmic transformed data, and the order of d is set at 2.

Setting the order of p or q to zero, leads to a moving average or autoregressive model for the logarithmic transformed, twice-differenced data. Recall from Theorem 1 and Theorem 2 that the autocorrelation function of an MA(q) process drops to zero after q lags and that the partial autocorrelation function of an AR(p) process drops to zero after p lags. Examining the ACF and PACF plots of the modified time series, $\Delta^2 \log(x_t)$, could thus simplify the modelling process. If the ACF and PACF show clear signs of being a moving average or autoregressive process, the number of models tested can be reduced. The respective ACF and PACF plots are depicted in Figure 6.1 and Figure 6.2.

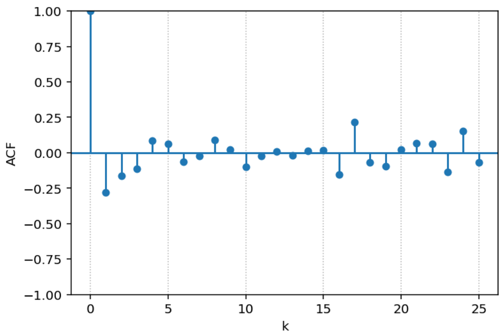


Figure 6.1: ACF of $\Delta^2\log(x_t)$

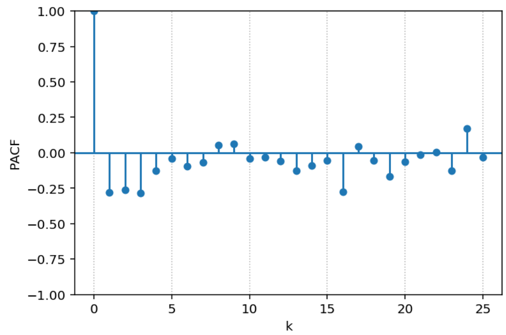


Figure 6.2: PACF of $\Delta^2\log(x_t)$

Both figures show fluctuating behaviour around zero and thus no clear signs of $\Delta^2\log(x_t)$ behaving like an autoregressive or moving average process. Based on the plots, no orders can be excluded. Additionally, since ACF and PACF plots can be ambiguous for real data, it is not excluded that $\Delta^2\log(x_t)$ behaves like an autoregressive or moving average process.

The Akaike information criteria (AIC, Eq. 2.17) gives an indication of the quality of a model. By increasing the orders p and q, chances are that the AIC increases as well, in case the likelihood function cannot compensate for the increasing amount of parameters. Setting the orders of an ARIMA function too high could lead to overfitting the data, in which case the white noise is captured rather than the underlying trend of the time series. Therefore the orders are restricted by a maximum order. The maximum orders for the first modelling cycle are set at $p_{max} = 10$ and $q_{max} = 10$, assuming that a higher order will cause overfitting, which is represented by a higher AIC. If upon evaluation of the AIC scores, the AIC scores corresponding to high orders are relatively low, it can be decided to increase the maximum orders and test the corresponding models.

Summarising, the models tested on $\log(x_t)$ (note that differencing is included in the order setting) are of the form

$$\widehat{\Delta^2\log(x_t)} = \hat{\alpha} + \hat{\phi}_1\Delta^2\log(x_{t-1}) + \cdots + \hat{\phi}_p\Delta^2\log(x_{t-p}) + \epsilon_t + \hat{\theta}_1\epsilon_{t-1} + \cdots + \hat{\theta}_q\epsilon_{t-q} \quad (6.1)$$

where $p \in \{0, 1, ..., p_{max}\}$, $q \in \{0, 1, ..., q_{max}\}$, and $\hat{\phi}_1, ..., \hat{\phi}_p, \hat{\mu}, \hat{\theta}_1, ..., \hat{\theta}_q, \hat{\sigma_\epsilon}^2$ result from maximising the likelihood function (Eq. 2.15). To obtain a result that is of the same form as the original time series, $\widehat{\Delta^2\log(x_t)}$ is "differenced back" and its exponential is taken.

6.1.2 Implementation and model selection

The data is loaded into Python and split into the training and test set, containing 80 and 20 percent of the datapoints, respectively. The models as described in the previous section are implemented, using the `ARIMA` function from the `statsmodels` module. The parameters are estimated by maximising the likelihood function (Eq. 2.15). The AIC (Eq. 2.17) score is calculated for each of the models. The AIC scores of all models tested can be found in Appendix B.2. Recall that the lower the AIC scores, the better the model. The model with the lowest AIC is the ARIMA(1,2,1) model with a score of -6.224. By printing the model's summary (App. B.3) in Python, the estimated parameters can be extracted: $\hat{\phi}_1 = 0.3923$, $\hat{\theta}_1 = -0.9986$, $\hat{\sigma_\epsilon}^2 = 0.0494$. The model summary does not return any value for $\hat{\mu}$. This can be explained by the fact that, by applying a logarithmic transformation and differencing the data twice, the mean has become constant and equal to zero, which can be seen in Figure 5.5d.

The accuracy of this model is further evaluated in the next section. The models with high orders show high AIC scores, therefore the maximum order will not be increased to test more models.

6.1.3 Forecasting performance

As described in section 4.5, to test the model's accuracy, a series of one-step forecasts is generated over the domain of the test data. Figure 6.3 shows the original, predicted and forecasted time series of the ARIMA(1,2,1) model.

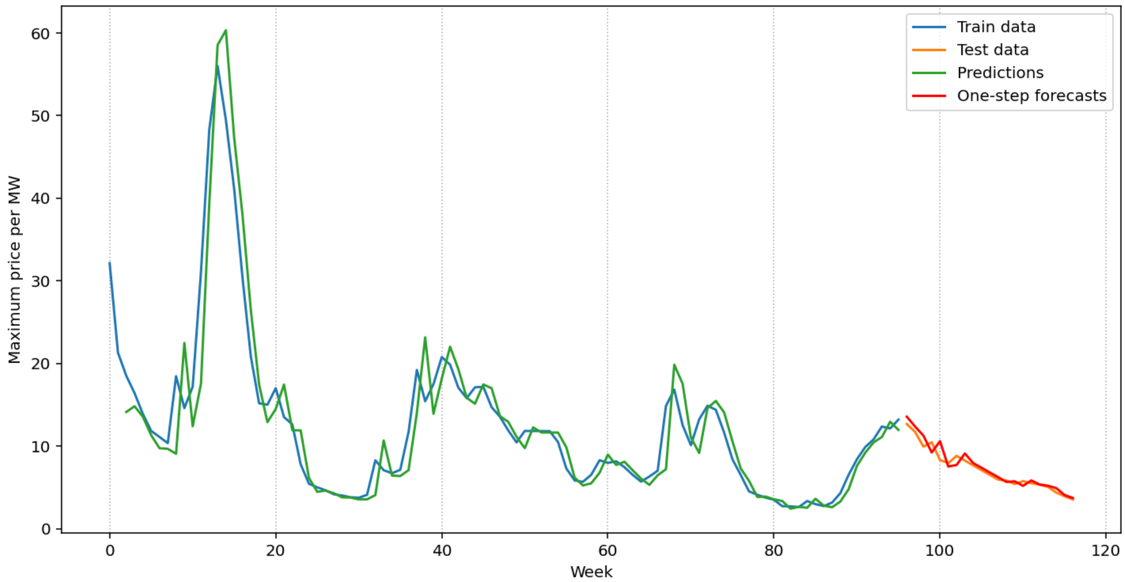


Figure 6.3: ARIMA(1,2,1) predictions and forecasts on NCR data

As can be seen from the plot, the model does not show large deviations from the original time series. One of ARIMA's weaknesses, however, is visible from the plot, namely her ability to predict turning points. As can be seen around week 18, while the peak in the original time series has already occurred, the peak of the prediction occurs one time step later, as is the case for all local maxima. This can be explained by the structure of the model. The model takes only one lag into account. Futhermore, the data is differenced twice to obtain stationarity. If the previous lag of the differenced time series were to be positive, it would indicate a rising trend. Since the model only looks at the previous lag, it suspects that this trend will persevere so it estimates that the next data point is higher than the previous one, when in fact it is lower. For the next time step the model again looks at one lag. This lag is a result of differencing too. So the lag is negative since the actual data is now following a downward trend. The model then estimates a decrease as well. But all of this will happen exactly one time step later than the original data. The result is a model whose local maxima and minima occur exactly one step later than the actual data.

Table 6.1 shows the mean absolute error (MAE), root mean squared error (RMSE), bias, percentage accepted bids (PAB) and the percentage maximum revenue (PMR) of the model. For the calculation of the PAB and PMR, the forecasted value has been rounded down to two decimal points. This is to prevent the forecast from surpassing the actual value by rounding up. The rounding is done to two decimal points since bids are in euro, which only use two decimal points. The accuracy metrics from the table have been rounded to three decimal points. The PAB and PMR have been rounded to one decimal point and are given in percentage.

	ARIMA(1,2,1)
MAE	0.589
RMSE	0.791
Bias	0.255
PAB	23.8
PMR	23.5

Table 6.1: Accuracy metrics for ARIMA(1,2,1) on NCR data

The MAE and RMSE are both small. Still the PAB and PMR remains low with only 23.8% and 23.5%. This can be caused by the model's tendency to slightly overestimate the forecast for each time step. The bias of 0.255 hints at this problem. The next section attempts to solve this issue of overestimation by inspecting the confidence interval

6.2 Adapting ARIMA(1,2,1) for the SCR market

The problem with the ARIMA(1,2,1) model is that it slightly overestimates the forecasted value. Even with a slight overestimation the bid will not be accepted. Therefore, bidding based on the ARIMA(1,2,1) model would not be very beneficial in the SCR market.

By shifting the model slightly down, this issue could be solved. The forecasts would then lie below the actual maximum prices, resulting in the bids getting accepted. This shift in the model will be based on the one-step confidence interval of the model. If the model were to be shifted to the lower bound of the 95% confidence interval, then there would be a 95% chance of the bid being accepted. Recall from section 2.23 that the confidence interval of level $1 - \alpha_0$ for an ARIMA model is given by

$$[\hat{x_t} - \xi_{1-\alpha_0/2} \cdot \hat{\sigma_\epsilon}, \hat{x_t} + \xi_{1-\alpha_0/2} \cdot \hat{\sigma_\epsilon}]$$

As mentioned in section 6.1.2, $\hat{\sigma_\epsilon}^2 = 0.0494$. $\xi_{1-\alpha_0/2}$ can be obtained from the standard normal distribution table [26]. For $\alpha_0 = 0.05$, $\xi_{1-\alpha_0/2} = 1.96$. By shifting the model down with $1.96\hat{\sigma_\epsilon}$, there is a 95% chance that the forecasted value is less than or equal to the actual value and, therefore, for the bid to be accepted. The shifted model returns the following accuracy metrics

	shifted ARIMA(1,2,1)
MAE	0.556
RMSE	0.770
Bias	-0.180
PAB	71.4
PMR	58.0

Table 6.2: Accuracy metrics for shifted ARIMA(1,2,1) on NCR data

Both the MAE and the RMSE are lower compared to the unshifted model. The bias, too, is lower. Furthermore it has become negative, which is, in the context of this research, a good thing since this means that the model tends to slightly underestimate the forecast, which results in the bid being accepted. This is apparent in the PAB and PMR which are 47.6 and 34.5 percent points higher for the shifted model than for the unshifted model.

It can be concluded that shifting the model down is beneficial for both the PAB and PMR. However, shifting the model even further down is not guaranteed to be beneficial over the long run. As described before, local minima and maxima of the model occur exactly one time step later than they occur in the original data. This property of the model is not just restricted to local minima and maxima. The model mimics the entire behaviour of the original data, only one time step ahead in time. That is why, when there is a rising trend in the price, the model falls below the actual maximum accepted bid prices. Shifting the model even further down would then decrease the PMR. Since the test data shows a predominately falling trend, this would not reflect in the accuracy metrics calculated over the test data's domain. However, since the historical data shows a pattern of rising and falling prices, it is decided to not shift the model further down. The final model, which forecasts form the bidding strategy, is then the shifted ARIMA(1,2,1) model with a shift of $1.96\hat{\sigma_\epsilon}$.

Chapter 7

Modelling Positive Control Reserve Bidding Strategy

The steps taken to compose a bidding strategy for the positive control reserve (PCR) are similar to the steps taken in the previous chapter. That is why some explanations might be a bit briefer. The reader is referred to the previous chapter if any clarifications are needed. The structure of this chapter is the same as in the previous chapter. That is, in section 7.1, the maximum accepted bid prices of PCR are modelled. In section 7.2, the model is modified, resulting in a version whose forecasts serve as the basis for bids. The Python code used for this chapter can be found in Appendix C.1.

7.1 Modelling maximum accepted PCR bidding prices

The modelling of the maximum accepted bid prices using an ARIMA model is discussed in this section. In section 7.1.1 the order settings are determined. Section 7.1.2 elaborates on the implementation of the models and selects the best model. Section 7.1.3 discusses the performance of the model on the SCR market.

7.1.1 Order setting

Again, the orders of the models to be tested need to be determined. Figure 5.6c shows that the logarithmic transformed, twice-differenced PCR data is stationary. Therefore, d is set at 2. Figure 7.1 and Figure 7.2 show the respective autocorrelation and partial autocorrelation function of the logarithmic transformed, twice differenced data.

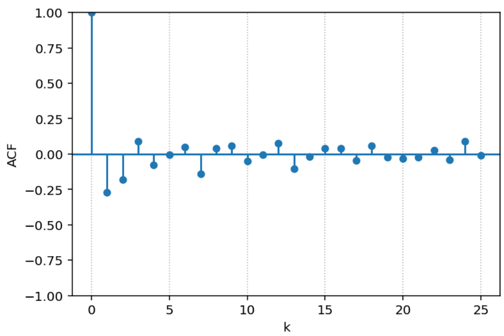


Figure 7.1: ACF of $\Delta^2\log(y_t)$

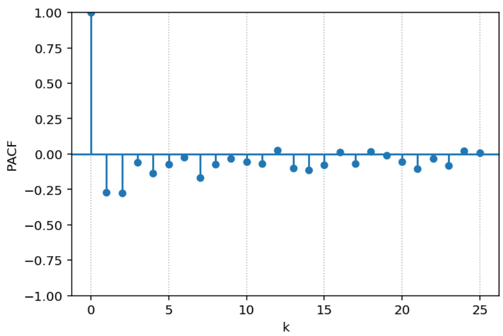


Figure 7.2: PACF of $\Delta^2\log(y_t)$

Both the ACF and the PACF show fluctuating behaviour around zero, so neither of them drops to zero after a certain amount of lags. Therefore no order is ruled out based on the ACF and PACF plots. Following the same arguments made in section 6.1.1 the maximum order for p and q is set at 10. The models that will be tested on the PCR data are thus of the form

$$\widehat{\Delta^2\log(y_t)} = \hat{\alpha} + \hat{\phi}_1\Delta^2\log(y_{t-1}) + \cdots + \hat{\phi}_p\Delta^2\log(y_{t-p}) + \epsilon_t + \hat{\theta}_1\epsilon_{t-1} + \cdots + \hat{\theta}_q\epsilon_{t-q} \quad (7.1)$$

where $p \in \{0, 1, ..., p_{max}\}$, $q \in \{0, 1, ..., q_{max}\}$, and $\hat{\phi}_1, ..., \hat{\phi}_p, \hat{\mu}, \hat{\theta}_1, ..., \hat{\theta}_q, \hat{\sigma_\epsilon}^2$ result from maximising the likelihood function (Eq. 2.15). The maximum bidding price of the PCR is denoted by y_t.

7.1.2 Implementation and model selection

After loading the data into Python and splitting it into a training and test set, the models, as described in the previous, section are implemented using again the `ARIMA` function from the `statsmodels` module. The parameters are estimated by maximising the likelihood function. The AIC is calculated for each model and the scores can be found in Appendix C.2. The model with the lowest AIC score is the ARIMA(1,2,1) model with an AIC of -49.982. The model summary (App. C.3) in Python returns the estimated parameters, namely $\hat{\phi}_1 = 0.4698$, $\hat{\theta}_1 = -0.9974$ and $\hat{\sigma_\epsilon}^2 = 0.0312$. Again, the model summary does not return a value for $\hat{\mu}$ since the logarithmic transformed, twice-differenced data is stationary with mean equal to zero. The accuracy of the ARIMA(1,2,1) model is further examined in the next section. Models of higher orders show higher AIC scores as well. Therefore, the maximum order will not be increased to test more models.

7.1.3 Forecasting performance

A series of one-step forecasts is generated over the domain of the test data. Figure 7.3 shows the original time series, the predictions and forecasts based on the ARIMA(1,2,1) model.

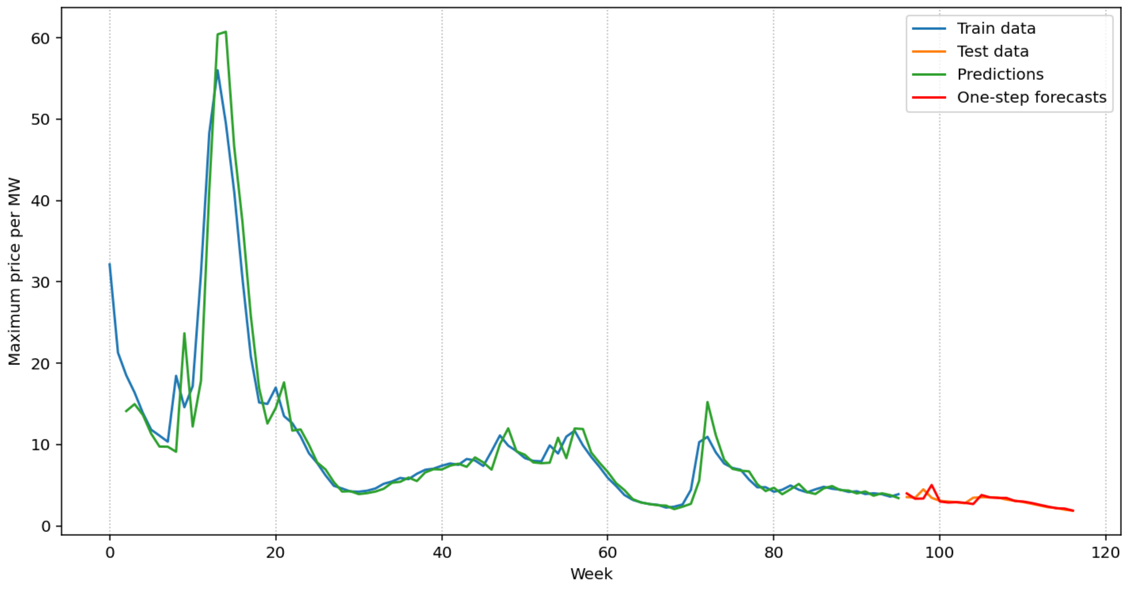


Figure 7.3: ARIMA(1,2,1) predictions and forecasts on PCR data

Again, the ARIMA model struggles with turning points.

Table 7.1 shows the MAE, RMSE, bias, PAB and PMR of the model. The rounding is the same as in the previous section.

	ARIMA(1,2,1)
MAE	0.256
RMSE	0.478
Bias	0.035
PAB	38.1
PMR	37.6

Table 7.1: Accuracy metrics for ARIMA(1,2,1) on PCR data

Again the MAE, RMSE are relatively small. The bias however shows a small tendency to overestimate, which is why the PMR is only 37.5%. Therefore the confidence interval will be examined in the next section in an attempt to solve this issue.

7.2 Adapting ARIMA(1,2,1) for the SCR market

Due to the low PAB and PMR scores of the ARIMA(1,2,1) model, the forecasts of the model would not be very useful to base bids on. Similarly to section 6.2, the idea is to shift the model down to the lower bound of the 95% confidence interval to increase the chances of the bid being accepted (PAB) and therefore to increase the PMR. The 95% one-step confidence interval of the PCR data is given by

$$[\hat{y}_t - \xi_{1-\alpha_0/2} \cdot \hat{\sigma_\epsilon}, \hat{y}_t + \xi_{1-\alpha_0/2} \cdot \hat{\sigma_\epsilon}]$$

The model estimated the following value for the standard deviation of the white noise terms, $\hat{\sigma_\epsilon}^2 = 0.0312$. Using the value of $\xi_{1-\alpha_0/2}$ in section 6.2, the following confidence interval is derived.

$$[\hat{y}_t - 1.96 \cdot \hat{\sigma_\epsilon}, \hat{y}_t + 1.96 \cdot \hat{\sigma_\epsilon}]$$

Therefore, by shifting the ARIMA(1,2,1) model down with a shift of $1.96\hat{\sigma_\epsilon}$, the forecasts lie on the lower bound of the 95% confidence interval, meaning there is a 95% chance that the actual value is higher than the forecasted, and thus for the bid to be accepted. The shifted model returns the accuracy metrics shown in Table 7.2.

	ARIMA(1,2,1)
MAE	0.439
RMSE	0.570
Bias	-0.312
PAB	90.5
PMR	76.5

Table 7.2: Accuracy metrics for shifted ARIMA(1,2,1) on PCR data

While the MAE, RMSE and the bias have increased compared to the original model, the PMR has doubled in percent points. Therefore it can be concluded that bidding, according to the shifted model, would be more profitable.

Chapter 8

Formalised Bidding Strategy

Chapter 6 and Chapter 7 both concluded that a shifted ARIMA(1,2,1) would be most beneficial to base bids on. This chapter formalises a bidding strategy based on these shifted ARIMA models. Section 8.1 defines the strategy for players who can only provide negative control reserve (NCR) or positive control reserve (PCR). Section 8.2 defines the strategy for players who can provide both.

8.1 Bidding strategy for players bidding on NCR or PCR only

Players who can only provide NCR, can use the shifted ARIMA(1,2,1) model from Chapter 6 as follows: generate a one-step forecast for every week and bid the price from the one-step forecast. When the market is cleared, feed the actual maximum accepted bid price of that week to the model without retraining it. To determine the price to bid for the next week, generate a one-step forecast again. The forecasted value is then the price to bid for the next week.

Generating a bid for PCR is done in the exact same way. Expect for one thing, instead of using the model based on the NCR data, use the shifted ARIMA(1,2,1) model based on the PCR data, described in Chapter 7.

8.2 Bidding strategy for players bidding on both NCR and PCR

Recall that players being able to provide both NCR and PCR are restricted to bidding on one of the two at a time. Therefore the player should bid on the one that has the highest potential revenue.

To determine which one has the highest potential revenue, the expected value of the bids on NCR and PCR is inspected. If one were to bid on NCR, following the strategy proposed in the previous section, one would bid the price generated by the one-step forecast of the shifted ARIMA model, trained on the NCR data, $\widehat{x_{t+1}}$. If one bids on PCR, one would bid the price generated by the one-step forecast of the shifted ARIMA model, trained on the PCR data, $\widehat{y_{t+1}}$. The expected value of both bids can be calculated by multiplying the forecasted price with the probability that the bid is accepted, that is

$$\text{E(revenue from NCR)} = \widehat{x_{t+1}} \cdot \text{P}(\widehat{x_{t+1}} \leq x_{t+1})$$

and for PCR,

$$\text{E(revenue from PCR)} = \widehat{y_{t+1}} \cdot \text{P}(\widehat{y_{t+1}} \leq y_{t+1})$$

Since both models were shifted to the lower bound of the 95% confidence interval, the theoretical probability of the bid being accepted is 0.95. The percentage accepted bids (PAB), however, says something different. For the shifted ARIMA model for NCR, it has a value of 0.580, for PCR it has a value of 0.905.

Although the PAB deviates from the theoretical probability of the bids being accepted, the theoretical probability is used to calculate the expected value. That is, because the PAB was only calculated over a short time interval and it is assumed that the PAB will converge to 0.95 over a longer time period.

Returning to the bidding strategy, players should first determine which of the two bids has the highest expected revenue. Since the probability of the bids being accepted is 0.95 for both, it would suffice to compare the forecasted NCR price to the forecasted PCR price. The player should bid on the control reserve that has the highest forecasted price. The price to bid is then determined by the forecast of the shifted ARIMA(1,2,1) model, belonging to the control reserve with the highest price.

Chapter 9

Conclusion

The research question was: what bidding strategy maximises the acceptance rate and potential revenue of players on the SCR market, based on the historic data of the maximum accepted bid prices?

To answer the question, an ARIMA model was created for both the negative control reserve (NCR) and the positive control reserve (PCR) data. It was found that for both the data, an ARIMA(1,2,1) provided the best balance in fit and model complexity.

For the NCR data, the original ARIMA(1,2,1) model resulted in an acceptance rate (PAB) of 23.8% while earning 23.5% of the potential revenue (PMR). Both of these numbers were on the lower side due to the tendency to slightly overestimate the forecasts of the model. In an attempt to improve the PAB and PMR, it was shifted down to the lower bound of the 95% confidence interval. Theoretically, there would then be a 95% chance of a bid getting accepted, based on the forecasts of this model. Comparing a series of forecasts to the test data showed that this was actually 71.4%, and the new PMR was 58.0%. Both these metrics showed an improvement compared to the original model. Therefore, it was concluded that bidding according to the shifted ARIMA(1,2,1) model would generate the most revenue.

The bidding strategy for the PCR data was developed in a similar way. For the PCR data, the ARIMA(1,2,1) model had a PAB of 38.1% and a PMR of 37.6%. This model too was shifted down to the lower bound of its one-step 95% confidence interval. The PAB and PMR resulting from the shifted model were 90.5% and 76.5% respectively. Similarly to the NCR data, the shifted model for the PCR data showed improvement over the original model. Therefore, using the forecasts of this model to form a bid would produce more revenue.

Players, able to provide only NCR or PCR, can use the forecasts for the next week of the models mentioned above to form a bid. Players, able to provide both NCR and PCR, should first compare the forecasted prices of both models. They should then bid on the control reserve with the highest forecasted price. The bid they then make, consists of the highest forecasted price, as generated by the corresponding model.

Chapter 10

Discussion

This chapter discusses the shortcomings of this research paper and proposes topics to investigate in future research.

Firstly, the orders of the models were determined by first setting a maximum order and subsequently comparing the AIC scores of all models with order less than or equal to this maximum order. By doing this, the complexity of the model was limited. However, it did exclude a wide range of models from testing. It can only be said that the ARIMA(1,2,1) provided the best model for the NCR and PCR data from all models tested. It cannot, however, be said that it was in fact the best model. For example, an ARIMA(21,2,15) could have had a lower AIC score, and would therefore have been selected, had it been tested.

Secondly, from both models was apparent that ARIMA does not do well with turning points (local maxima or minima). Peaks in the ARIMA(1,2,1) models occurred exactly one time step later than they occurred in the observed data. This could be an explanation for why the ARIMA(1,2,1) model for the PCR data had a better performance in terms of PAB and PMR than the ARIMA(1,2,1) model for the NCR data. Namely, because the performance was tested on the test data. The test data of the PCR shows less fluctuating behaviour than the test data of the NCR, making it easier for the ARIMA model to generate accurate forecasts.

For further research, more accurate order selection methods can be used, as to not exclude any good models from the testing. In section 5.3.1, outliers were removed from the original dataset. This was done to create a smoother model for the data. For further research, the model can be extended to include the outliers as well since they seemed to occur in a yearly pattern.

The models in this research were selected based on the Akaike information criterion (AIC). The lowest AIC, however, does not guarantee that the accuracy metrics have the best scores as well. To improve the models, they could be selected based on a good balance between a low AIC score and good accuracy metrics scores.

To ensure a good long-term performance of the models, the models can be retrained every year. If a different model provides a better fit, then bids can be based on the new model's forecasts.

Since the goal of this research was to determine a bidding strategy purely based on the historic data, no external factors that could influence the maximum bidding price have been taken into consideration. For further research, one can look into the factors that influence the price, starting with the opportunity costs of hydropower plants.

Bids on the secondary control reserve market consist of a price per megawatt and a capacity. In this paper, only the price was considered. Swissgrid aggregates secondary control reserve by accepting the bids from lowest to highest until the the needed capacity is obtained. So bidding slightly under the maximum price could imply that, although the price is high, the capacity is not. Therefore it could be more advantageous to bid a low price against a high volume than a high price against a lower volume. This could be an interesting topic to research.

Lastly, other models to forecast the price can also be considered. Time series forecasting methods that do not struggle as much with turning points, such as exponential smoothing, can be examined to construct a model. Forecasting by use of neural networks could also give more accurate results. Research from Gao et al. [16] stated that the ARIMA model performed better on electricity price forecasting than a neural network. However, neural networks have improved since this paper was published. Therefore, a neural network might outperform ARIMA models in price forecasting in this day and age, and is therefore worth examining.

Bibliography

[1] European Union. (n.d.). *Renewable energy targets*. European Commission. https://ap.lc/fUwSK

[2] *Switzerland - Countries & Regions* - IEA. (n.d.). IEA. https://www.iea.org/countries/switzerland/renewables

[3] *What is aFRR (automatic frequency restoration reserve)?* (n.d.). https://www.next-kraftwerke.com/knowledge/afrr

[4] Abrell, J. (2016). *The Swiss Wholesale Electricity Market*. SCCER CREST Working Paper.

[5] *Grid stability*. (n.d.). https://www.swissgrid.ch/en/home/operation/regulation/grid-stability.html

[6] *Control power markets*. (n.d.). https://www.swissgrid.ch/en/home/operation/market/control-energy.html

[7] Schillinger, M. (2020). Balancing-market design and opportunity cost: The Swiss case. *Utilities Policy, 64*, 101045. https://doi.org/10.1016/j.jup.2020.101045

[8] Siami-Namini, S., & Namin, A. S. (2018, 16 March). *Forecasting Economics and Financial Time Series: ARIMA vs. LSTM*. arXiv.org. https://arxiv.org/abs/1803.06386

[9] Heij, C., Heij, C., De Boer, P., Franses, P. H., Kloek, T., Van Dijk, H. K., & Rotterdam, A. a. T. E. U. I. (2004). *Econometric Methods with Applications in Business and Economics*. Oxford University Press.

[10] Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and its applications: With R Examples*. Springer.

[11] Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

[12] *Understanding the Importance of Stationarity in Time Series* — Hex. (2023, 1 December). https://hex.tech/blog/stationarity-in-time-series/

[13] Christiansen, L. E. (2017). 02417 Lecture 7 Part A: Estimating Parameters in ARMA Models [Video]. https://www.youtube.com/watch?v=IcxMywGiWUc

[14] Science, N. B. P. A. D. (n.d.). *Understanding ARIMA Models for Machine Learning — Capital One*. Capital One. https://www.capitalone.com/tech/machine-learning/understanding-arima-models/

[15] Hayes, A. (2024, 6 April). *Autoregressive Integrated Moving Average (ARIMA) Prediction Model*. Investopedia. https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

[16] Gao, G., Lo, K., & Fan, F. (2017). Comparison of ARIMA and ANN Models Used in Electricity Price Forecasting for Power Market. Energy And Power Engineering, 09(04), 120–126. https://doi.org/10.4236/epe.2017.94b015

[17] Zhou, M., Yan, Z., Ni, Y., Li, G., & Nie, Y. (2006). Electricity price forecasting with confidence-interval estimation through an extended ARIMA approach. *IEE Proceedings. Generation, Transmission And Distribution, 153(2)*, 187. https://doi.org/10.1049/ip-gtd:20045131

[18] Brownlee, J. (2020, 26 August). Train-Test Split for Evaluating Machine Learning Algorithms. Machine Learning Mastery. https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

[19] Smith, E. (2023, 25 November). *AIC vs BIC: Difference and Comparison*. Ask Any Difference. https://askanydifference.com/aic-vs-bic/

[20] *statsmodels.tsa.arima.model.ARIMA - statsmodels 0.14.1*. (n.d.). https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html

[21] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

[22] González-Sopeña, J., Pakrashi, V., & Ghosh, B. (2021). An overview of performance evaluation metrics for short-term statistical wind power forecasting. *Renewable & Sustainable Energy Reviews*, 138, 110515. https://doi.org/10.1016/j.rser.2020.110515

[23] *Outlier Removal: Everything You Need to Know When Assessing Outlier Removal Skills*. (n.d.). https://rb.gy/19lo4i

[24] Agrawal, R. (2024, 18 March). *Interpolation Techniques Guide & Benefits — Data Analysis (Updated 2024)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/power-of-interpolation-in-python-to-fill-missing-values/

[25] West, R. M. (2021). Best practice in statistics: The use of log transformation. *Annals Of Clinical Biochemistry*, 59(3), 162–165. https://doi.org/10.1177/00045632211050531

[26] Bijma, F., Jonker, M., & Van Der Vaart, A. W. (2017). *An Introduction to Mathematical Statistics*.

Appendix A

Proofs

A.1 Proof Theorem 1

Theorem 1. *For the partial autocorrelation of an AR(p) process holds the following property*

$$\phi_k = 0 \text{ for all } k > p$$

Proof. Let x_t ben an AR(p) process. Thus x_t can be written as

$$x_t = \alpha + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \epsilon_t$$

where ϵ_t is white noise with mean zero, variance σ_ϵ^2, and property $\mathrm{E}[\epsilon_t x_{t-k}] = 0$. $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$ and $\phi_1, ..., \phi_p$ are parameters. x_t is assumed to be stationary.

Recall that stationarity implies a constant mean, i.e. $\mu_t = \mu$. To simplify notation, $x_{t+1}, ..., x_{t+k-1}$ is denoted by X_k. It suffices to show that $\mathrm{cov}(x_t, x_{t+k}|x_{t+1}, ..., x_{t+k-1}) = 0$ for all $k > p$ (see Eq. 2.3, Eq. 2.5).

$$\mathrm{cov}(x_t, x_{t+k}|x_{t+1}, ..., x_{t+k-1}) = \mathrm{E}[(x_t - \mu)(x_{t+k} - \mu)|x_{t+1}, ..., x_{t+k-1}]$$

$$= \mathrm{E}[(x_t - \mu)(\mu(1 - \sum_{i=1}^{p} \phi_i) + \sum_{i=1}^{p} \phi_i x_{t+k-i} + \epsilon_{t+k} - \mu|X_k]$$

$$= \mathrm{E}[(x_t - \mu)(-\mu \sum_{i=1}^{p} \phi_i + \sum_{i=1}^{p} \phi_i x_{t+k-i} + \epsilon_{t+k}|X_k]$$

$$= \mathrm{E}[-\mu x_t \sum_{i=1}^{p} \phi_i + x_t \sum_{i=1}^{p} \phi_i x_{t+k-i} + x_t \epsilon_{t+k} + \mu^2 \sum_{i=1}^{p} \phi_i - \mu \sum_{i=1}^{p} \phi_i x_{t+k-i} - \mu \epsilon_{t+k}|X_k]$$

$$= \mathrm{E}[-\mu x_t \sum_{i=1}^{p} \phi_i|X_k] + \mathrm{E}[x_t \sum_{i=1}^{p} \phi_i x_{t+k-i}|X_k] + \mathrm{E}[x_t \epsilon_{t+k}|X_k] + \mathrm{E}[\mu^2 \sum_{i=1}^{p} \phi_i|X_k]$$

$$- \mathrm{E}[\mu \sum_{i=1}^{p} \phi_i x_{t+k-i}|X_k] - \mathrm{E}[\mu \epsilon_{t+k}|X_k] \text{ (**)}$$

It will be shown that either the terms cancel out or are equal to zero. First of all, $\mu^2 \sum_{i=1}^{p} \phi_i$ (the fourth term), is a constant and the expected value of a constant is just the constant itself. Secondly, since ϵ_{t+k} is a future term independent of $x_t, ..., x_{t+k-1}$, the third and last term amount to zero. Thirdly, $k > p$ implies that $t + k - p > t$. Therefore all terms $x_{t+k-i} : i \in \{1, ..., p\}$ appear in the conditional part of the expectation. Therefore, since x_{t+k-i} is known, it becomes constant and can be brought out of the conditional expectation, i.e.

$$\mathrm{E}[x_t \sum_{i=1}^{p} \phi_i x_{t+k-i} | X_k] = \sum_{i=1}^{p} \phi_i x_{t+k-i} \mathrm{E}[x_t | X_k]$$

Using all of this, (**) can be written as

$$-\mu^2 \sum_{i=1}^{p} \phi_i + \mu \sum_{i=1}^{p} \phi_i x_{t+k-i} + \mu^2 \sum_{i=1}^{p} \phi_i - \mu \sum_{i=1}^{p} \phi_i x_{t+k-i} = 0$$

□

A.2 Proof Theorem 2

Theorem 2. *For the autocorrelation function of an MA(q) process holds the following property*

$$\rho_k = 0 \text{ for all } k > q$$

Proof. Let x_t be a moving average process of order q. Thus x_t can be written as

$$x_t = \alpha + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

where ϵ_j are i.i.d. with mean zero and variance σ_ϵ^2, α a constant and $\theta_1, ..., \theta_q$ parameters.

Since MA(q) is a stationary process, it has a constant mean function given by

$$\begin{aligned}\mu_t = \mathrm{E}[x_t] &= \mathrm{E}[\alpha + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}] \\ &= \mathrm{E}[\alpha] + \mathrm{E}[\epsilon_t] + \mathrm{E}[\theta_1 \epsilon_{t-1}] + \mathrm{E}[\theta_2 \epsilon_{t-2}] + \cdots + \mathrm{E}[\theta_q \epsilon_{t-q}] \\ &= \alpha\end{aligned}$$

Let $k > q$. It suffices to show that the autocovariance is zero for all $k > q$. Since the process is stationary, the autocovariance be written as

$$\begin{aligned}\gamma_k &= \mathrm{cov}(x_{t+k}, x_t) = \mathrm{cov}(x_t, x_{t-k}) = \mathrm{E}[(x_t - \mu_t)(x_{t-k} - \mu_{t-k}] \\ &= \mathrm{E}[(\alpha + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} - \alpha)(\alpha + \epsilon_{t-k} + \theta_1 \epsilon_{t-k-1} + ... + \theta_q \epsilon_{t-k-q} - \alpha)] \\ &= \mathrm{E}[(\epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q})(\epsilon_{t-k} + \theta_1 \epsilon_{t-k-1} + ... + \theta_q \epsilon_{t-k-q})] \; (*)\end{aligned}$$

Since $k > q$, x_t and x_{t-k} have no overlapping white noise terms, i.e. $\epsilon_{t-i} \neq \epsilon_{t-k-j}$ $\forall i, j \in \{1, ..., q\}$. Recall that ϵ_j are i.i.d. with mean zero and therefore $\mathrm{E}[\epsilon_i \epsilon_j] = \mathrm{E}[\epsilon_i]\mathrm{E}[\epsilon_j] = 0$ for $i \neq j$. Thus $\mathrm{E}[\epsilon_{t-i} \epsilon_{t-k-j}] = 0 \; \forall i, j \in \{1, ..., q\}$. Therefore, and by the linearity of the expectation operator, (*) can be written as

$$\mathrm{E}\left[\epsilon_t \epsilon_{t-k} + \epsilon_t \sum_{i=1}^{q} \theta_i \epsilon_{t-k-i} + \epsilon_{t-k} \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{q} \theta_i^2 \epsilon_{t-i} \epsilon_{t-k-i}\right] = 0$$

□

Appendix B

Modelling NCR Data

B.1 Python code

```
import math
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error

### Python script for NCR data ###

## Importing data from excel file
minweekly=pd.read_excel('C:/Users/berbe/OneDrive/Documenten
        /Studie/Technische Wiskunde 3/BEP/Bewerkte data/minweekly.xlsx')
maxprice=pd.DataFrame(minweekly.loc[:,'Max Price'])

# Plot max price + seasonality check
plt.axvspan(7, 23, color='grey', alpha=0.3)
plt.axvspan(31, 55, color='grey', alpha=0.3)
plt.axvspan(66,78, color='grey', alpha=0.3)
plt.axvspan(88,107, color='grey', alpha=0.3)
sns.lineplot(maxprice)
sns.scatterplot(maxprice)
plt.xlabel('Week')
plt.ylabel('Maximum price per MW')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.scatter([0, 51, 52, 103, 104], [32.13, 35.60, 27.50, 21.81, 21.77],
        color='red', marker='x')
plt.show()
```

```
## Defining accuracy metrics
# Defining PMR
def pmr(x: pd.DataFrame, x_hat):
    pmr_num=0
    for i in range(len(x)):
        if x.iloc[i][0] >= x_hat[i]:
            pmr_num+=math.floor(x_hat[i]*100)/100
        else :
            pmr_num+=0
    pmr_den=x.sum()
    pmr=(pmr_num/pmr_den) *100
    return pmr

# Defining bias
def bias(x: pd.DataFrame, x_hat):
    x_mean=x.mean()
    x_hat_mean=np.mean(x_hat)
    bias=x_hat_mean-x_mean
    return bias

# Defining PAB
def pab(x: pd.DataFrame, x_hat):
    pab_num=0
    for i in range(len(x)):
        if x.iloc[i][0] >= x_hat[i]:
            pab_num+=1
        else :
            pab_num+=0
    pab_den=len(x)
    pab=(pab_num/pab_den) *100
    return pab

## Data preparation
# Removing outliers and substituting with lin. interpolated values
maxprice.iloc[51]=11.82
maxprice.iloc[52]=11.81
maxprice.iloc[103]=8.23
maxprice.iloc[104]=7.66
maxprice.iloc[105]=7.08
maxprice.iloc[106]=6.51

# Plot of modified data
sns.lineplot(maxprice)
sns.scatterplot(maxprice)
plt.xlabel('Week')
plt.ylabel('Maximum price per MW')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()
```

```
# Log transformation
maxprice_log=np.log(maxprice)
sns.lineplot(maxprice_log)
plt.xlabel('Week')
plt.ylabel('log(x$_t$)')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

# Differencing log data
maxprice_logd=maxprice_log.diff()
sns.lineplot(maxprice_logd)
plt.xlabel('Week')
plt.ylabel('log(x$_t$)')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

# Differencing log data second time
maxprice_logd2=maxprice_logd.diff()
sns.lineplot(maxprice_logd2)
plt.xlabel('Week')
plt.ylabel('$^2$log(x$_t$)')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

## Inspection ACF and PACF
# ACF plot log, 2 times differenced data
plot_acf(maxprice_logd2[2:118], title=None, alpha=None, lags=25)
plt.xlabel('k')
plt.ylabel('ACF')
plt.grid(axis='x', linestyle=':')
plt.show()

# PACF plot log, 2 times differenced data
plot_pacf(maxprice_logd2[2:118], title=None, alpha=None, lags=25)
plt.xlabel('k')
plt.ylabel('PACF')
plt.grid(axis='x', linestyle=':')
plt.show()

## Modelling NCR data
# Splitting data
maxprice_train=maxprice_log.iloc[0:96]
maxprice_test=maxprice_log.iloc[96:118]
```

```
# Testing models on AIC
max_p=10
max_q=10
d=0
ARMA_table=np.zeros(((max_p+1)*(max_q+1),3))
for i in range(0,max_p+1):
    for j in range(0,max_q+1):
        ARMA=ARIMA(maxprice_train, order=(i,2,j))
        ARMA_fit=ARMA.fit()
        ARMA_aic=ARMA_fit.aic
        ARMA_table[d][0]=i
        ARMA_table[d][1]=j
        ARMA_table[d][2]=ARMA_aic
        d+=1
print(ARMA_table)

## One-step forecasting using ARIMA(1,2,1)
# Fit the ARIMA model on the training data
model = ARIMA(maxprice_train['Max Price'], order=(1, 2, 1))
fitted_model = model.fit()
model_pred=np.exp(fitted_model.predict(start=2,
        end=len(maxprice_train)-1, dynamic=False))

# Extracting parameters
print(fitted_model.summary())

# One-step forecasts on the test data
history = list(maxprice_train['Max Price'])
forecasts = []

for t in range(len(maxprice_test)):
    output = fitted_model.forecast(steps=1)
    a=output.tolist()
    yhat = a
    forecasts.append(yhat)
    obs = maxprice_test['Max Price'].iloc[t]
    history.append(obs)
    fitted_model = fitted_model.append([obs], refit=False)
    print(f'predicted={yhat}, expected={obs}')

# Plotting the results
plt.figure(figsize=(12,6))
plt.xlabel('Week')
plt.ylabel('Maximum price per MW')
plt.plot(np.exp(maxprice_train), label='Train data')
plt.plot(np.exp(maxprice_test), label='Test data')
plt.plot(model_pred, label='Predictions' )
plt.plot(maxprice_test.index, (np.exp(forecasts)),
        label='One-step forecasts', color='red')
```

```
plt.grid(axis='x', linestyle=':')
plt.legend()
plt.show()

# Computing accuracy metrics for ARIMA(1,2,1)
model_MAE = mean_absolute_error(maxprice[96:118],np.exp(forecasts))
model_RMSE=math.sqrt(mean_squared_error(maxprice[96:118],
        np.exp(forecasts)))
model_bias=bias(maxprice[96:118],np.exp(forecasts))
model_PMR=pmr(maxprice[96:118],np.exp(forecasts))
model_PAB=pab(maxprice[96:118],np.exp(forecasts))

# Computing accuracy metrics for shifted ARIMA(1,2,1)
shift=math.sqrt(0.0494)*1.96
shift_MAE = mean_absolute_error(maxprice[96:118],np.exp(forecasts)-shift)
shift_RMSE=math.sqrt(mean_squared_error(maxprice[96:118],
        np.exp(forecasts)-shift))
shift_bias=bias(maxprice[96:118],np.exp(forecasts)-shift)
shift_PMR=pmr(maxprice[96:118],np.exp(forecasts)-shift)
shift_PAB=pab(maxprice[96:118],np.exp(forecasts)-shift)
```

B.2 AIC scores

p	q	AIC
0	1	3.410
0	2	-5.105
0	3	-4.660
0	4	-2.780
0	5	-0.926
0	6	-0.537
0	7	1.012
0	8	2.752
0	9	-0.652
0	10	0.899
1	0	14.817
1	**1**	**-6.224**
1	2	-4.434
1	3	-2.789
1	4	-0.791
1	5	1.208
1	6	1.065
1	7	2.587
1	8	2.142
1	9	0.997
1	10	2.981
2	0	10.084
2	1	-4.432
2	2	-2.228
2	3	-1.942
2	4	-5.656
2	5	-3.652
2	6	-1.799
2	7	-1.112
2	8	0.196

p	q	AIC
2	9	3.024
2	10	3.325
3	0	3.664
3	1	-2.473
3	2	-0.492
3	3	-5.219
3	4	-3.643
3	5	-0.664
3	6	-0.569
3	7	0.317
3	8	1.796
3	9	4.794
3	10	6.733
4	0	3.940
4	1	-2.003
4	2	1.042
4	3	-3.305
4	4	-1.411
4	5	3.898
4	6	1.052
4	7	-2.383
4	8	3.992
4	9	6.670
4	10	8.656
5	0	5.759
5	1	-0.011
5	2	1.971
5	3	-3.006
5	4	0.382
5	5	1.926

p	q	AIC
5	6	5.204
5	7	2.851
5	8	6.467
5	9	9.014
5	10	10.856
6	0	6.398
6	1	7.921
6	2	3.669
6	3	9.705
6	4	8.211
6	5	5.123
6	6	4.795
6	7	3.939
6	8	6.680
6	9	-2.125
6	10	7.462
7	0	7.373
7	1	9.227
7	2	5.418
7	3	12.737
7	4	10.190
7	5	3.999
7	6	6.674
7	7	4.784
7	8	0.299
7	9	9.074
7	10	7.837
8	0	9.068
8	1	11.023
8	2	7.273

p	q	AIC
8	3	14.712
8	4	4.224
8	5	9.406
8	6	7.868
8	7	6.662
8	8	8.378
8	9	6.971
8	10	5.680
9	0	10.906
9	1	12.801
9	2	-0.956
9	3	4.063
9	4	6.032
9	5	10.961
9	6	9.400
9	7	7.934
9	8	13.080
9	9	8.317
9	10	6.086
10	0	12.558
10	1	3.608
10	2	4.009
10	3	5.136
10	4	7.992
10	5	2.571
10	6	11.878
10	7	13.655
10	8	15.476
10	9	10.575
10	10	11.950

B.3 Model Summary

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Max Price   No. Observations:                  117
Model:                 ARIMA(1, 2, 1)   Log Likelihood                  16.570
Sample                              0   AIC                            -27.140
                                - 117   BIC                            -18.906
Covariance Type                   opg   HQIC                           -23.798
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3923      0.093      4.197      0.000       0.209       0.575
ma.L1         -0.9986      1.346     -0.742      0.458      -3.637       1.640
sigma2         0.0494      0.064      0.773      0.440      -0.076       0.175
==============================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                38.14
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):               0.21   Skew:                             0.89
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.19
==============================================================================
```

Appendix C

Modelling PCR Data

C.1 Python code

```
    import seaborn as sns
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error

### Python script for upbidding data ###

## Importing data from excel file
plusweekly=pd.read_excel('C:/Users/berbe/OneDrive/Documenten
        /Studie/Technische Wiskunde 3/BEP/Bewerkte data/plusweekly.xlsx')
maxprice=pd.DataFrame(plusweekly.loc[:,'Max Price'])

# Plot max price + seasonality check
plt.axvspan(7, 23, color='grey', alpha=0.3)
plt.axvspan(44, 60, color='grey', alpha=0.3)
plt.axvspan(68,80, color='grey', alpha=0.3)
sns.lineplot(maxprice)
sns.scatterplot(maxprice)
plt.xlabel('Week')
plt.ylabel('Maximum price per MW')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

## Defining accuracy metrics
# Defining PMR
def pmr(x: pd.DataFrame, x_hat):
    pmr_num=0
    for i in range(len(x)):
        if x.iloc[i][0] >= x_hat[i]:
```

```
            pmr_num+=math.floor(x_hat[i]*100)/100
        else :
            pmr_num+=0
    pmr_den=x.sum()
    pmr=(pmr_num/pmr_den) *100
    return pmr

# Defining bias
def bias(x: pd.DataFrame, x_hat):
    x_mean=x.mean()
    x_hat_mean=np.mean(x_hat)
    bias=x_hat_mean-x_mean
    return bias

# Defining PAB
def pab(x: pd.DataFrame, x_hat):
    pab_num=0
    for i in range(len(x)):
        if x.iloc[i][0] >= x_hat[i]:
            pab_num+=1
        else :
            pab_num+=0
    pab_den=len(x)
    pab=(pab_num/pab_den) *100
    return pab

## Data preparation
# Log transformation
maxprice_log=np.log(maxprice)
sns.lineplot(maxprice_log)
plt.xlabel('Week')
plt.ylabel('log(x$_t$)')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

# Differencing log data
maxprice_logd=maxprice_log.diff()
sns.lineplot(maxprice_logd)
plt.xlabel('Week')
plt.ylabel('log(x$_t$)')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

# Differencing log data second time
maxprice_logd2=maxprice_logd.diff()
sns.lineplot(maxprice_logd2)
plt.xlabel('Week')
```

```
plt.ylabel('$^2$log(x$_t$)')
plt.legend([],[],frameon=False)
plt.grid(axis='x', linestyle=':')
plt.show()

## Inspection ACF and PACF
# ACF plot log, 2 times differenced data
plot_acf(maxprice_logd2[2:118], title=None, alpha=None, lags=25)
plt.xlabel('k')
plt.ylabel('ACF')
plt.grid(axis='x', linestyle=':')
plt.show()

# PACF plot log, 2 times differenced data
plot_pacf(maxprice_logd2[2:118], title=None, alpha=None, lags=25)
plt.xlabel('k')
plt.ylabel('PACF')
plt.grid(axis='x', linestyle=':')
plt.show()

## Modelling PCR data
# Splitting data
maxprice_train=maxprice_log.iloc[0:96]
maxprice_test=maxprice_log.iloc[96:118]

# Testing models on AIC
max_p=10
max_q=10
d=0
ARMA_table=np.zeros(((max_p+1)*(max_q+1),3))
for i in range(0,max_p+1):
    for j in range(0,max_q+1):
        ARMA=ARIMA(maxprice_train, order=(i,2,j))
        ARMA_fit=ARMA.fit()
        ARMA_aic=ARMA_fit.aic
        ARMA_table[d][0]=i
        ARMA_table[d][1]=j
        ARMA_table[d][2]=ARMA_aic
        d+=1
print(ARMA_table)

## One-step forecasting using ARIMA(1,2,1)
# Fit the ARIMA model on the training data
model = ARIMA(maxprice_train['Max Price'], order=(1, 2, 1))
fitted_model = model.fit()
model_pred=np.exp(fitted_model.predict(start=2,
        end=len(maxprice_train)-1, dynamic=False))
```

```
# Extracting parameters
print(fitted_model.summary())

# One-step forecasts on the test data
history = list(maxprice_train['Max Price'])
forecasts = []

for t in range(len(maxprice_test)):
    output = fitted_model.forecast(steps=1)
    a=output.tolist()
    yhat = a
    forecasts.append(yhat)
    obs = maxprice_test['Max Price'].iloc[t]
    history.append(obs)
    fitted_model = fitted_model.append([obs], refit=False)
    print(f'predicted={yhat}, expected={obs}')

# Plotting the results
plt.figure(figsize=(12,6))
plt.xlabel('Week')
plt.ylabel('Maximum price per MW')
plt.plot(np.exp(maxprice_train), label='Train data')
plt.plot(np.exp(maxprice_test), label='Test data')
plt.plot(model_pred, label='Predictions')
plt.plot(maxprice_test.index, (np.exp(forecasts)),
        label='One-step forecasts', color='red')
plt.grid(axis='x', linestyle=':')
plt.legend()
plt.show()

# Computing accuracy metrics for ARIMA(1,2,1)
model_MAE = mean_absolute_error(maxprice[96:118],np.exp(forecasts))
model_RMSE=math.sqrt(mean_squared_error(maxprice[96:118],
        np.exp(forecasts)))
model_bias=bias(maxprice[96:118],np.exp(forecasts))
model_PMR=pmr(maxprice[96:118],np.exp(forecasts))
model_PAB=pab(maxprice[96:118],np.exp(forecasts))

# Computing accuracy metrics for shifted ARIMA(1,2,1)
shift=math.sqrt(0.0312)*1.96
shift_MAE = mean_absolute_error(maxprice[96:118],np.exp(forecasts)-shift)
shift_RMSE=math.sqrt(mean_squared_error(maxprice[96:118],
        np.exp(forecasts)-shift))
shift_bias=bias(maxprice[96:118],np.exp(forecasts)-shift)
shift_PMR=pmr(maxprice[96:118],np.exp(forecasts)-shift)
shift_PAB=pab(maxprice[96:118],np.exp(forecasts)-shift)
```

C.2 AIC scores

p	q	AIC
0	1	-37.662
0	2	-48.126
0	3	-47.198
0	4	-46.719
0	5	-44.986
0	6	-43.021
0	7	-41.263
0	8	-43.491
0	9	-42.325
0	10	-40.522
1	0	-32.671
1	**1**	**-49.982**
1	2	-48.120
1	3	-44.656
1	4	-44.967
1	5	-43.664
1	6	-41.664
1	7	-39.914
1	8	-43.692
1	9	-40.190
1	10	-41.195
2	0	-37.188
2	1	-48.086
2	2	-46.581
2	3	-44.635
2	4	-43.845
2	5	-39.877
2	6	-39.630
2	7	-42.798
2	8	-41.321
2	9	-39.358
2	10	-38.927
3	0	-35.811
3	1	-46.380
3	2	-44.307
3	3	-45.137
3	4	-44.649
3	5	-41.774
3	6	-40.715
3	7	-41.146
3	8	-39.224
3	9	-37.304
3	10	-31.772
4	0	-35.398
4	1	-45.242
4	2	-42.389
4	3	-45.161
4	4	-42.213
4	5	-40.022
4	6	-37.519
4	7	-38.869
4	8	-36.591
4	9	-34.933
4	10	-33.245
5	0	-33.577
5	1	-43.414
5	2	-41.618
5	3	-40.582
5	4	-41.227
5	5	-40.592

p	q	AIC
5	6	-37.735
5	7	-37.644
5	8	-33.399
5	9	-33.219
5	10	-32.095
6	0	-31.820
6	1	-42.966
6	2	-39.631
6	3	-40.408
6	4	-39.347
6	5	-38.933
6	6	-35.804
6	7	-35.315
6	8	-33.542
6	9	-29.900
6	10	-32.088
7	0	-33.951
7	1	-42.834
7	2	-40.979
7	3	-41.901
7	4	-39.425
7	5	-41.165
7	6	-34.815
7	7	-34.394
7	8	-31.487
7	9	-28.106
7	10	-29.622
8	0	-32.303
8	1	-41.024
8	2	-39.339

p	q	AIC
8	3	-40.005
8	4	-38.383
8	5	-37.330
8	6	-34.925
8	7	-31.450
8	8	-30.083
8	9	-28.033
8	10	-27.414
9	0	-30.700
9	1	-39.873
9	2	-37.337
9	3	-38.122
9	4	-36.821
9	5	-34.107
9	6	-31.871
9	7	-31.496
9	8	-28.058
9	9	-26.650
9	10	-27.537
10	0	-30.409
10	1	-38.580
10	2	-35.899
10	3	-35.935
10	4	-34.662
10	5	-34.794
10	6	-32.141
10	7	-30.739
10	8	-31.169
10	9	-23.141
10	10	-30.055

C.3 Model summary

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Max Price   No. Observations:                  117
Model:                 ARIMA(1, 2, 1)   Log Likelihood                  39.827
Sample                              0   AIC                            -73.655
                                - 117   BIC                            -65.420
Covariance Type:                  opg   HQIC                           -70.313
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.4698      0.080      5.850      0.000       0.312       0.627
ma.L1         -0.9974      0.572     -1.743      0.081      -2.119       0.124
sigma2         0.0312      0.017      1.873      0.061      -0.001       0.064
==============================================================================
Ljung-Box (L1) (Q):                   0.09   Jarque-Bera (JB):                74.40
Prob(Q):                              0.76   Prob(JB):                         0.00
Heteroskedasticity (H):               0.30   Skew:                             1.00
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.40
==============================================================================
```