

Data-driven modeling to predict the consumption of energy in one or multiple sites

Dissertation presented by
Guillaume DACIER

for obtaining the Master's degree in
Mathematical Engineering

Supervisor(s)
Julien HENDRICKX, Georgios BIRPOUTSOUKIS

Reader(s)
Vincent WERTZ

Academic year 2018-2019

Abstract

This thesis deals with the extension of a modeling method in the field of time-series estimation, namely the Hannan-Rissanen(HR) method, and its application to a real life case. The method is used to construct models which are able to describe processes with non-seasonal, single seasonal or double seasonal behaviour. The purpose of modeling is to predict the output of these processes, which in this thesis represent energy consumption in one or multiple sites.

The HR method has been proposed before for the estimation of non-seasonal and single seasonal models. In this work, a modification of the method is proposed for the estimation of single seasonal models and the method is extended to cover the modeling case of double seasonal processes. Both at a simulation level as well as at a real process modeling case, results show that time-series estimates can be built in an efficient and systematic way using this method, in order to predict the output of processes with double seasonal characteristics. Last but not least, a way to automatize the modeling procedure is proposed in terms of this thesis.

Contents

Abbreviations	vii
List of Figures	ix
List of Tables	xiii
1 Introduction	1
1.1 The need for time-series modeling for prediction	1
1.2 Aim, methodology and contribution	2
1.3 Structure of the thesis	2
2 Literature review	5
2.1 Theoretical concepts	5
2.2 General methods for modeling time-series	7
2.3 Regular univariate models	8
2.3.1 Autoregressive (AR) model	9
Least squares estimation	10
The Yule-Walker estimates	11
2.3.2 Mixed Autoregressive-Moving Average (ARMA) model	11
The Hannan-Rissanen (HR) method	13
Maximum Likelihood (ML) methods	15
2.3.3 Mixed Autoregressive Integrated Moving Average (ARIMA) model	16
2.4 Seasonal univariate models	17
2.4.1 Seasonal Autoregressive (SAR) model	17
Relaxation to a linear problem	19
ML estimates	20
2.4.2 Mixed Seasonal Autoregressive-Moving Average (SARMA) model	20
2.4.3 Mixed Seasonal Autoregressive Integrated Moving Average (SARIMA) model	23
2.4.4 Double-seasonal models	23
2.5 Multivariate models	25
2.5.1 Multivariate Autoregressive models	26
2.5.2 Multivariate mixed Autoregressive-Moving Average models	28
2.5.3 Multivariate mixed Autoregressive Integrated Moving Average models	30
2.6 Choice of the estimation method	30
3 Simulation-based theoretical study	31
3.1 Extension of HR method for processes with seasonal behaviour	32
3.2 Properties of the proposed estimation method	33
3.2.1 Theoretical analysis	34
AR models	35
ARMA models	36
3.2.2 Autoregressive models	37

3.2.3	Mixed Autoregressive-Moving Average models	42
	Regular models	42
	Single seasonal processes	49
	Processes with double seasonality	55
3.3	Model structure selection	66
3.3.1	Regular models	68
3.3.2	Single seasonal models	70
3.3.3	Double seasonal models	71
3.4	Conclusion	74
4	Application of the methods to a real life process	75
4.1	Room 1	76
4.1.1	Data analysis	77
4.1.2	The model structure	78
4.1.3	Estimation method	80
4.1.4	Scanning order procedure	80
4.1.5	Diagnostics	83
4.1.6	Results given by the model	86
4.1.7	Robustness with respect to the number of data	88
4.1.8	Final model	89
4.1.9	Multi-step ahead predictions	94
4.2	Room 2	96
4.3	Room 3	97
4.4	Multivariate: Room 1 + another room	99
4.5	Multivariate: Room 2 + Room 3	100
4.6	Multivariate: Room 1 + two other rooms	102
4.7	Multivariate: Room 2 + Room 3 + another room	103
4.8	Comparison of the results	105
4.9	Conclusion	106
5	Conclusions and future work	107
	Bibliography	109

Abbreviations

2SAR	Double Seasonal Autoregressive
2SARIMA	Double Seasonal mixed Autoregressive Integrated Moving Average
2SARMA	Double Seasonal mixed Autoregressive-Moving Average
AR	Autoregressive
ARIMA	Mixed Autoregressive Integrated Moving Average
ARMA	Mixed Autoregressive-Moving Average
HR	Hannan-Rissanen
i.i.d.	Identically and independently distributed
LS	Least Square
MA	Moving Average
ML	Maximum Likelihood
MSE	Means Square Error
OLS	Ordinary Least Square
PE	Prediction Error
SAR	Single Seasonal Autoregressive
SARIMA	Single Seasonal mixed Autoregressive Integrated Moving Average
SARMA	Single Seasonal mixed Autoregressive-Moving Average
V2SAR	Multivariate double Seasonal Autoregressive
V2SARMA	Multivariate double Seasonal mixed Autoregressive-Moving Average
VAR	Multivariate Autoregressive
VARMA	Multivariate mixed Autoregressive-Moving Average
VSAR	Multivariate single Seasonal Autoregressive
VSARMA	Multivariate single Seasonal mixed Autoregressive-Moving Average

List of Figures

2.1	Illustration of the application of a neural network to forecast time-series	8
2.2	Plot of a process described by an ARIMA model and exposing a homogeneous non-stationary behaviour	16
3.1	Scatter plot of the parameter estimates of the $AR(8)$ of eq 3.6 coming from 100 Monte Carlo realizations	39
3.2	Scatter plot of the parameter estimates of the two variables $VAR(2)$ of eq 3.7 coming from 100 Monte Carlo realizations	39
3.3	Scatter plot of the parameter estimates of the $SAR(4)(4)_{24}$ of eq 3.8 coming from 100 Monte Carlo realizations	40
3.4	Scatter plot of the parameter estimates of the two variables $VSAR(2)(2)_{24}$ of eq 3.9 coming from 100 Monte Carlo realizations	40
3.5	Scatter plot of the parameter estimates of the $SAR(4)(2)_{24}(2)_{168}$ of eq 3.10 coming from 100 Monte Carlo realizations	41
3.6	Scatter plot of the parameter estimates of the two variables $VSAR(2)(1)_{24}(1)_{168}$ of eq 3.11 coming from 100 Monte Carlo realizations	41
3.7	Plots of the impulse response of the transfer function H^{-1} for two $ARMA(4, 4)$ models with parameters coming from Table 3.1. The estimation of H^{-1} (in red) is done using an $AR(\log(N)^{1.5})$ model with $N = 15000$	43
3.8	Scatter plot of the parameter estimates of two $ARMA(4, 4)$ coming from 50 Monte Carlo realizations. The model parameters are resumed in Table 3.1	45
3.9	Plot of the residuals (upper left graph) and histogram of residuals obtained after step one and step two of the HR method compared with the real innovation of the signal coming from the two $ARMA$ models of Table 3.1.	46
3.10	Impulse response of real transfer function H^{-1} and the estimated transfer function of the $VARMA(2, 2)$ model of eq 3.12 at step one of the HR estimation	47
3.11	Scatter plot of the parameter estimates of a two variables $VARMA(2, 2)$ of eq 3.12 coming from 50 Monte Carlo realizations	48
3.12	Plot of the residuals (upper left) and histogram of residuals obtained after step one and step two of the HR method compared with the real innovation of the signal coming from an $VARMA(2, 2)$ model of eq 3.12.	48
3.13	Impulse response of real transfer function H^{-1} and the estimated transfer function of a $SARMA(4, 4)(2, 2)_{168}$ model of eq 3.14 at step one in the HR estimation	50
3.14	Scatter plot of the parameter estimates of $SARMA(4, 4)(2, 2)_{168}$ of eq 3.14 coming from 50 Monte Carlo realizations with different values of N	51
3.15	Scatter plot of the parameter estimates of $SARMA(4, 4)(2, 2)_{168}$ of eq 3.14 coming from 50 Monte Carlo realizations with $p_{large} = \log(N)^{1.4}$ and different values of P_{large}	51
3.16	Scatter plot of the parameter estimates of $SARMA(4, 4)(2, 2)_{168}$ of eq 3.14 coming from 50 Monte Carlo realizations with $P_{large} = 2P$ and different values of p_{large}	52

3.17	Plot of the residuals (upper left) and histogram of residuals obtained after step one and step two of the HR method compared with the real innovation of the signal coming from a $SARMA(4, 4)(2, 2)_{168}$ model of eq 3.14	52
3.18	Scatter plot of the parameter estimates of $VSARMA(2, 2)(2, 2)_{168}$ of eq 3.15 coming from 25 Monte Carlo realizations with $P_{large} = 2(P + Q)$ and different values of p_{large}	54
3.19	Scatter plot of the parameter estimates of $VSARMA(2, 2)(2, 2)_{168}$ of eq 3.15 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.4}$ and different values of P_{large}	55
3.20	Scatter plot of the parameter estimates of $SARMA(4, 4)(2, 2)_{168}$ of eq 3.14 coming from 15 Monte Carlo realizations using the classical HR (in blue) and the extended HR (in green)	56
3.21	Impulse response of H^{-1} for a $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$ models (right sub-figure) of eq 3.16 and for a $2SARMA(2, 2)(1, 1)_{24}(2, 2)_{168}$ of same equation with last parameters of the first season removed	56
3.22	Impulse response of real transfer function H^{-1} and the estimated transfer function of the $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$ model of eq 3.16 at the first step of the extended HR estimation	57
3.23	Scatter plot of the parameter estimates of $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$ of eq 3.16 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.3}$ and different values of P_{2large}	58
3.24	Scatter plot of the parameter estimates of $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$ of eq 3.16 coming from 25 Monte Carlo realizations with $P_{2large} = 2P$ and different values of p_{large}	58
3.25	Scatter plot of the parameter estimates of a $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$ of eq 3.16 coming from 25 Monte Carlo realizations	59
3.26	Scatter plot of the parameter estimates of 2-variables $V2SARMA(2, 2)(1, 1)_{24}(1, 1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations with $P_{1large} = 3$ and $P_{1large} = 2$	60
3.27	Scatter plot of the parameter estimates of the 2-variables $V2SARMA(2, 2)(1, 1)_{24}(1, 1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.3}$ and $P_{2large} = 2$	61
3.28	Scatter plot of the parameter estimates of the 2-variables $V2SARMA(2, 2)(1, 1)_{24}(1, 1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.3}$ and $P_{1large} = 3$	62
3.29	Impulse response of real transfer function H^{-1} and the estimated transfer function of the $V2SARMA(2, 2)(1, 1)_{24}(1, 1)_{168}$ model of 2 variables of eq 3.17 at the first step of the extended HR estimation	63
3.30	Zoom of the upper left graph of Figure 3.29a	64
3.31	Scatter plot of the parameter estimates of a 2-variables $V2SARMA(2, 2)(1, 1)_{24}(1, 1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations	65
3.32	Plot of V_{val} obtained with different model orders used for the estimations of an $ARMA(4, 4)$	69
3.33	Plot of V_{val} obtained with different model orders used for the estimations of a $VARMA(3, 3)$ with 2 variables	69
3.34	Plot of V_{val} obtained with different model orders used for the estimations of a $SARMA$	70
3.35	Plot of V_{val} obtained with different model orders used for the estimations of a $VSARMA$ with 2 variables	71
3.36	Plot of V_{val} obtained with different seasonal model orders used for the estimations of a $2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$	71

3.37	Plot of V_{val} obtained with different regular and double seasonal model orders used for the estimations of a $2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$	72
3.38	Plot of V_{val} obtained with different regular and simple seasonal model orders used for the estimations of a $2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$	72
3.39	Plot of V_{val} obtained with different seasonal model orders used for the estimations of a $V2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$ with 2 variables	73
3.40	Plot of V_{val} obtained with different regular and double seasonal model orders used for the estimations of a $V2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$ with 2 variables	73
3.41	Plot of V_{val} obtained with different regular and simple seasonal model orders used for the estimations of a $V2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$ with 2 variables	74
4.1	Plot of the normalized electricity consumption of Room 1	77
4.2	Plot of the normalized electricity consumption of Room 1 for 2 random weeks	77
4.3	Plot of the autocorrelation function of the electricity consumption of Room 1	78
4.4	Plot of the deseasonalized electricity consumption $z(t) = (1 - B^{168})y(t)$ of Room 1	79
4.5	Plot of autocorrelation function of z until 2 week lags (336 hours) for Room 1	79
4.6	Plot of autocorrelation function until 2 week lags (336 hours) of Room 1 electricity consumption deseasonalized twice (with regard to the week and day)	80
4.7	Plot of the scanning order for Room 1 with $p = q = 15$ and with the seasonal AR and MA orders the same	81
4.8	Plot of the scanning order for Room 1 with $P_1 = Q_1 = 6$ and with the AR and MA orders the same for the regular and second seasonal orders	81
4.9	Plot of the scanning order for Room 1 with $p = q = 6$ and $P_1 = Q_1 = 6$	82
4.10	Plot of the scanning order for Room 1 with $P_2 = 3$ and $Q_2 = 5$ and with the AR and MA orders equal for the regular and first seasonal orders	83
4.11	Plot of the scanning order for Room 1 with $p = q = 5$, $P_2 = 3$ and $Q_2 = 5$	83
4.12	Plot of the scanning order for Room 1 with $P_1 = Q_1 = 6$, $P_2 = 3$ and $Q_2 = 5$	84
4.13	Plot of autocorrelation function of the residuals obtained using a $2SARMA$ model with Room 1	84
4.14	Plot of the scanning order for Room 1 with $P_1 = Q_1 = 6$, $P_2 = 3$ and $Q_2 = 5$	85
4.15	Plot of autocorrelation function of the residuals obtained using a $2SARMA$ model with Room 1	86
4.16	Plot of the real Room 1 normalized consumption for May 2018 and the estimation using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ when May 2018 is included in the training set or in the validation set	87
4.17	Plot of the cost function computed on the validation set (8% of the data) using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1, for different size of training set	88
4.18	For a fixed training set (92% of the data), plot of the cost function computed on validation sets of different size using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1	89
4.19	Plot of the cost function computed on the validation set (20% of the data) using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1, for different size of training set	90
4.20	For a fixed training set (80% of the data), plot of the cost function computed on validation sets of different size using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1	90
4.21	Plot of autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $2SARMA(22, 15, 0)(4, 6, 0)_{24}(3, 3, 1)_{168}$ of Room 1	91
4.22	Plot of the real and estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using a $2SARMA$ model	92
4.23	Plot of the real and rounded estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using a $2SARMA$ model	93

4.24	Plot of the of the metrics on the validation set for k -step ahead predictors ($k = 1, \dots, 23$) using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$ of Room 1	94
4.25	Plot of the of the metrics on the validation set for daily k -step ahead predictors ($k = 1, 24, 48, \dots, 192$) using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$ of Room 1	95
4.26	Plot of autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $2SARMA(23, 20, 0)(5, 5, 0)_{24}(2, 7, 1)_{168}$ of Room 2	96
4.27	Plot of the real and estimated normalized consumption of Room 2 for the training (80%) set and validation (20%) set using a 2SARMA model	97
4.28	Plot of autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $2SARMA(21, 23, 0)(6, 6, 0)_{24}(7, 7, 1)_{168}$ of Room 3	98
4.29	Plot of the real and estimated normalized consumption of Room 3 for the training (80%) set and validation (20%) set using a 2SARMA model	98
4.30	Plot of Room 1 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(22, 20, 0)(6, 6, 0)_{24}(4, 7, 1)_{168}$ with 2 variables	99
4.31	Plot of the real and estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using the model $V2SARMA(22, 20, 0)(6, 6, 0)_{24}(4, 7, 1)_{168}$ with 2 variables	100
4.32	Plot of Room 2 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 2 variables	101
4.33	Plot of the real and estimated normalized consumption of Room 2 for the training (80%) set and validation (20%) set using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 2 variables	101
4.34	Plot of Room 1 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 3 variables	102
4.35	Plot of the real and estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 3 variables	103
4.36	Plot of Room 2 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(23, 22, 0)(6, 4, 0)_{24}(3, 3, 1)_{168}$ with 3 variables	104
4.37	Plot of the real and estimated normalized consumption of Room 2 for the training (80%) set and validation (20%) set using the model $V2SARMA(23, 22, 0)(6, 4, 0)_{24}(3, 3, 1)_{168}$ with 3 variables	104

List of Tables

3.1	Model parameters corresponding at Figure 3.7	42
4.1	Values of the cost function and accuracy metric obtained for the 3 rooms studied, with the different estimated models of Chapter 4	105
4.2	Values of metrics with rounded predictions for the 3 rooms studied, with the different estimated models of Chapter 4	106

Chapter 1

Introduction

1.1 The need for time-series modeling for prediction

Nowadays the amount of applications where time-series modeling for prediction is required is growing quickly. More and more data become available in the form of a time-series regarding a large number of processes and scientific fields. Whether it is for industrial applications, prediction of renewable energy productions or energy consumption, large-scale integration of renewable energy sources in power systems [24], predictions in the field of tourism management [7, 8], etc, time-series estimation methods can be used to model the dynamics of these processes and predict their behaviour.

Focusing on energy applications, the increasing need of economic and environmentally friendly energy management renders more and more important the need to predict the future production as well as consumption of energy. Such processes are often governed by patterns that are repeated in time. The time period between two consecutive patterns of the same time is called season. Depending on the amount of types of patterns present, we are dealing with single seasonal processes (one seasonal pattern present), double seasonal processes (two seasonal patterns present) and so on so forth.

Applications of time-series modeling for prediction have already been studied for predictions of energy consumption and production, such as prediction of power output of photovoltaic system [19], global solar irradiance [18], load demand [22] or electricity load [5]. These energy predictions allow the adaptation of the electricity production and consumption of an electrical network, the optimization of its efficiency, as well as the prevention of some problems by implementing a system of prediction based alert emissions.

1.2 Aim, methodology and contribution

The aim of this thesis is to modify an existing time-series estimation method for the case of processes with single seasonal behaviour and to extend it to cover the case of processes with double seasonal behaviour. Once the extension of the method is proposed, the statistical properties of the model estimation method are studied. Afterwards, the proposed method is applied to model and predict the behaviour of a real life process. A model structure selection method is also proposed, tested on a number of simulations and applied to the real process.

The modeling method studied in this thesis is the Hannan-Rissanen (HR) developed by Hannan and Rissanen [14] and Hannan and Kavalieris [15]. This method belongs to the field of time-series estimations, and has been so far applied mostly to the modeling of processes which lack seasonal behaviour. Gómez [12] proposed an implementation to model processes with one seasonal pattern in their behaviour.

Energy consumption is often given in the form of a time-series, and so this thesis proposes a method to model energy consumption. The main contribution of the current work is then the adaptation of the HR method to model time-series in order to predict their (seasonal) behaviour. An automatic model structure selection procedure is also proposed, and extended simulations are provided to ensure the efficiency of the methods. Finally the extended HR and the model structure selection are used to model and predict the electricity consumption measured in an industrial site. The objective is to predict the consumption in one or multiple rooms in this site. Due to confidentiality issues, results based on the real data are only presented normalized.

1.3 Structure of the thesis

The current thesis is structured as follows:

- In Chapter 2, a summary of different time-series modeling methods available in the literature is provided. A number of model structures used to estimate processes with different characteristics are presented, and for each model, estimation methods are discussed. At the end of this chapter, the different methods are compared and arguments are provided with respect to the choice of the HR method as the estimation method to work with.
- In Chapter 3, an extended version of the HR method is developed and the theoretical properties of the estimates obtained using this method are studied. This study is followed by a verification provided based on a number of simulations. A method for brute force automatic model structure selection is also proposed in order to scan different candidate model structures for the same process.

- In Chapter 4, the proposed extended HR method as well as the model structure selection procedure are applied to a real life process. Results are provided for one and multiple energy consumptions which show that the model structure selection and the model estimation methods are able to deliver the desirable performance.
- Finally Chapter 5 includes the conclusion of this work, as well as a discussion of the possible future research directions.

Chapter 2

Literature review

This chapter presents different time-series modeling methods available in the literature. One specific method is chosen at the end of the chapter in order to propose an extension of it in Chapter 3. A study of the statistical properties of the extended method as well as an application on real processes will be performed later. Definitions and assumptions used throughout this work are provided in the following section.

2.1 Theoretical concepts

Definition 2.1 A *stochastic process* describes the probability structure of a sequence of successive observations.

In this work, we shall very often omit the word "stochastic" and talk about "process".

Property 2.1 A process is called *multivariate* when it describes several sequences that can depend on each other. Each sequence is presented by a variable. If the process contains only one variable, it is called *univariate*.

Definition 2.2 A discrete *signal* is a time-ordered sequence of numbers, represented by an array.

Given the measured output signal (i.e. the realizations) of an univariate process $Y = (y_1 \dots y_N)^T$ given in the form of a time-series, with N the number of successive measured outputs (or observations). It is assumed that the data-generating process is described by the

following equation

$$y_t = f(\mathbf{x}_t) + a_t \quad (2.1)$$

where $y_t \in \mathbb{R}$ is the observation at time t , f is a (linear or non-linear) function describing the relation between $\mathbf{x}_t$ and y_t , $\mathbf{x}_t \in \mathbb{R}^n$ is the vector of information available prior to time t from which y_t depends and $a_t \in \mathbb{R}$ is the innovation at time t .

In the multivariate case, the measured output signal is $Y = (\mathbf{y}_1 \dots \mathbf{y}_N)^T$ and the data-generating process is described by the following equation

$$\mathbf{y}_t = f(\mathbf{X}_t) + \mathbf{a}_t \quad (2.2)$$

where $\mathbf{y}_t = (y_{1t} \dots y_{Kt})^T \in \mathbb{R}^K$ is the observation at time t , f is a (linear or non-linear) function describing the relation between $\mathbf{X}_t$ and $\mathbf{y}_t$, $\mathbf{X}_t = (\mathbf{x}_{1t} \dots \mathbf{x}_{Kt})^T \in \mathbb{R}^{K \times n}$ is the matrix containing the vectors of information available prior to time t $\mathbf{x}_{it}$ from which y_{it} depends, $\mathbf{a}_t = (a_{1t} \dots a_{Kt})^T \in \mathbb{R}^K$ is the vector containing the innovations a_{it} for $i = 1, \dots, K$ which are linked to y_{it} and K is the number of variables included in the process.

Definition 2.3 We call *innovation* a_t the difference between the measured output at time t (y_t) coming from a process and the optimal prediction of that value based on information available prior to time t ($f(\mathbf{x}_t)$).

Definition 2.4 We call a *zero-mean Gaussian white noise* signal a_t , for $t = 1, \dots, N$, of variance σ_a^2 , a signal such that :

$$\mathbb{E}[a_t] = 0, \quad \forall t \quad (2.3)$$

$$\mathbb{E}[a_t a_{t-k}] = \begin{cases} \sigma_a^2 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases} \quad (2.4)$$

where $\mathbb{E}$ is the mathematical expectation operator.

Assumption 2.1 In this thesis, the innovation is supposed to be the realization of an independently and identically distributed (i.i.d) Gaussian white noise of zero mean and variance σ_a^2 .

This assumption comes from the fact that the innovation is seen as the unpredictable part of the measured signal. If the innovation was not i.i.d., it would imply that by knowing the previous innovations, we could guess the actual innovation (so it would be no more unpredictable). If the

innovation was not zero mean, we could find the mean and then subtract it from the innovation, resulting in a new innovation of zero mean.

Definition 2.5 We call **residual** the difference between a predicted value of a measured output of a process and the real measured output.

Property 2.2 A stochastic process is **stationary** if its statistical properties (moments of the distribution) do not change when shifted in time.

Assumption 2.2 In this thesis, the time-series process outputs are assumed to be generated by stationary processes.

The process stationarity is a necessary property to apply the methods described further. Nevertheless, for particular non-stationary processes it is possible to transform the output signal to fall back into the stationary assumption.

Definition 2.6 We call **model set** the set containing all the possible models having the same structure, i.e. the set of all models having the same parameters with different values.

The modeling of a time-series y refers to the estimations of $f(\mathbf{X}_t)$ in eq 2.1 with a certain model having in the general case a parameter matrix β . If the process is in the model set, it implies that there exist a parameter matrix β_0 with the same dimension as β such that the corresponding model describes exactly the process.

Definition 2.7 Given a measured time-series $y_1, \dots, y_t$, a k -step ahead **predictor** is an operator which maps $y_1, \dots, y_t$ to a prediction of a future time-series value y_{t+k} .

An **one-step ahead predictor** is a predictor for which $k = 1$ while a **multi-step ahead predictor** is a predictor with $k > 1$.

2.2 General methods for modeling time-series

There exist different ways to model a time-series in order to predict its output. A first machine learning technique is the Gaussian process regression and constitutes a type of Bayesian non-parametric modeling method. Its principle is to assume that the function f that describes the

time-series is in fact (in the general case) a function of a Gaussian process (and no more an explicit parameterized function). For applications of Gaussian process regression and further details, see [4, 10, 26].

Another machine learning strategy to predict time-series uses neural networks. These networks are composed of one or several layers and each layer includes one or more neurons. The first layer is an input layer and the last one an output layer. There might be some hidden layer between the input and the output and each successive layer are connected together to finally form a neural network. The model associates at each neuron a parameter and an activation function. It will learn from the

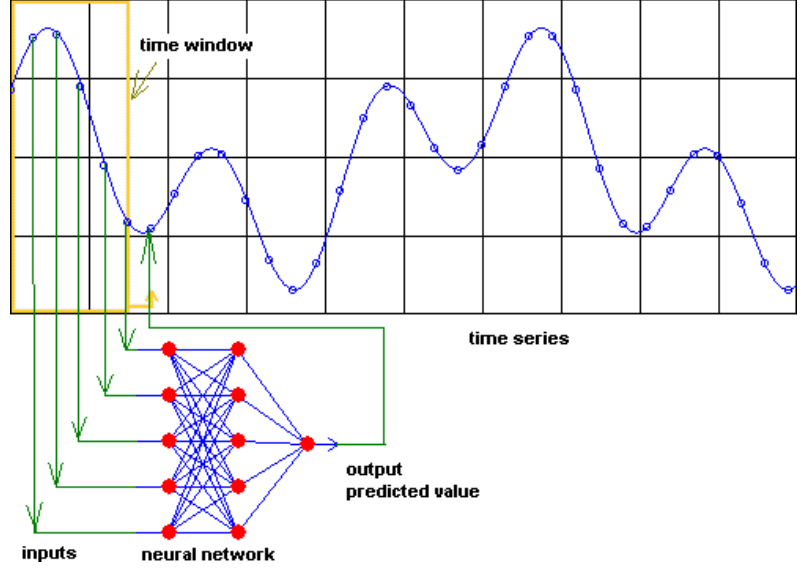


Figure 2.1: Illustration of the application of a neural network to forecast time-series

time-series how to calculate the best prediction by fixing the values of the parameters. Figure 2.1 shows a two-layer neural network used to compute an one-step ahead predictor. To have more information about modeling time-series using neural-networks, see [1, 17, 23].

The methods described and studied further belong to a third class of techniques. It consists of simple model structure in which certain characteristics of the process, such as the seasonality, can be easily recognized and embedded to model the process. Once the model structure is chosen, different methods exist to compute the model parameter estimates.

The preference of the current study to this class is guided by the fact that there are already several applications of these techniques for the prediction of time-series which have shown good results (see [5, 7, 8, 18, 19, 22, 24]). Moreover, the simplicity of the model structures used to describe complex process dynamics offers the possibility to construct an automatic modeling procedure which is also part of the goal of this thesis.

2.3 Regular univariate models

The regular models are most of the time used to describe processes with no seasonal pattern. Before analyzing further, consider an univariate time-series described by y_t for which the values

for $t = 1, \dots, N$ are known. The objective is to find the model that best predicts (following a chosen criterion) the time-series. In this work, the model will be used to predict as well as possible the value of y_t knowing all information until time $t - 1$. Given a model of the process as well as data measured from the same, this prediction is given by the value of the one-step ahead predictor noted $\hat{y}_{t|t-1}$. To estimate a model of the process and build this predictor, the Prediction Error (PE) criterion, formalized in a Least Squares (LS) manner, is chosen. The latter considers the minimization of the squared distance between $\hat{y}_{t|t-1}$ and the real value y_t , i.e.

$$\min \frac{1}{N} \sum_{t=1}^N (e_t)^2 \text{ with } e_t = y_t - \hat{y}_{t|t-1} \quad (2.5)$$

The predictor depends on the model parameters, and the parameter estimates are the ones that minimize eq 2.5.

The choice of modeling criteria is dictated by the use of the model described further [2, 3]. It is important to note that different criterion have been proposed in the literature, [2, 11, 25].

2.3.1 Autoregressive (AR) model

The first model structure presented corresponds to processes with only AR characteristics.. In these models, y_t depends on previous y and on the innovation at time t , named a_t . Noting B the lag operator, i.e. $B^k y_t = y_{t-k}$, the AR process of order p , $AR(p)$ is described as follows :

$$\begin{aligned} y_t + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} &= a_t \\ \Leftrightarrow (1 + \phi_1 B + \dots + \phi_p B^p) y_t &= a_t \\ \Leftrightarrow \phi(B) y_t &= a_t \end{aligned} \quad (2.6)$$

In order to compute the one-step ahead predictor, eq 2.6 is written

$$y_t = H a_t \quad \text{with} \quad H = \phi^{-1}(B) \quad (2.7)$$

Suppose that everything is known until $t - 1$ (signals, innovations and model parameters), the only unknown for the calculation of $\hat{y}_{t|t-1}$ is a_t (which correspond to the unpredictable part

of the output measurement). So the best predictor will be such that

$$\begin{aligned} y_t - \hat{y}_{t|t-1} &= a_t \\ y_t - \hat{y}_{t|t-1} &= H^{-1}y_t \\ \hat{y}_{t|t-1} &= (1 - H^{-1})y_t \end{aligned} \tag{2.8}$$

The one-step ahead predictor of AR processes is then written as $\hat{y}_{t|t-1} = -\hat{\phi}_1 y_{t-1} - \dots - \hat{\phi}_p y_{t-p}$, with $\hat{\phi}_1, \dots, \hat{\phi}_p$ estimations of the real parameters $\phi_1, \dots, \phi_p$.

Assumption 2.3 *In this work, all the transfer functions are assumed to be monic.*

Recalling that the objective is to find the parameter vector (using the PE criterion 2.5) minimizing the error between the process output and the one-step ahead prediction, the parameter estimates are given by

$$\hat{\beta} = \arg \min_{\beta} \frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_{t|t-1})^2$$

where β contains $\phi_1, \dots, \phi_p$ in the case of AR processes.

Combining with eq 2.8, we finally have

$$\begin{aligned} \hat{\beta} &= \arg \min_{\beta} \frac{1}{N} \sum_{t=1}^N (y_t - (1 - H^{-1})y_t)^2 \\ \hat{\beta} &= \arg \min_{\beta} \frac{1}{N} \sum_{t=1}^N (H^{-1}y_t)^2 \end{aligned} \tag{2.9}$$

For the AR, eq 2.9 is written, for $i = 1, \dots, p$,

$$\hat{\phi}_i = \arg \min_{\phi_i} \frac{1}{N} \sum_{t=1}^N (y_t + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p})^2$$

There exist two main ways to compute these parameter estimates :

Least squares estimation

By isolating y_t in eq 2.6, we obtain $y_t = -\sum_{i=1}^p \phi_i y_{t-i} + a_t$. y_t is then a linear combination of p variables $y_{t-1}, \dots, y_{t-p}$ plus a white noise a_t . Finding the LS solution $\hat{\phi}_i$ is then equivalent to solving a linear LS problem. Calling the estimated parameter vector $\hat{\beta} = \begin{pmatrix} \hat{\phi}_1 & \hat{\phi}_2 & \dots & \hat{\phi}_p \end{pmatrix}^T \in$

$\mathbb{R}^p$, this can be done by computing the Ordinary Least Squares (OLS) solution

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

with

$$X = \begin{pmatrix} -y_p & -y_{p-1} & \dots & -y_1 \\ -y_{p+1} & -y_p & \dots & -y_2 \\ \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & -y_{N-2} & \dots & -y_{N-p} \end{pmatrix} \in \mathbb{R}^{N-p-1 \times p} \text{ and } Y = \begin{pmatrix} y_{p+1} \\ y_{p+2} \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^{N-p-1} \quad (2.10)$$

The Yule-Walker estimates

Box, Jenkins, and Reinsel [2] describe another equivalent way to obtain the LS estimation of the AR model parameters by using the autocorrelations of the system. Practically, an estimation of the process autocorrelations is used in the so-called Yule-Walker equations (see [27, 28]) to estimate the real parameters $\phi_1, \dots, \phi_p$. Both LS and Yule-Walker estimates admit the same values for the same data set.

2.3.2 Mixed Autoregressive-Moving Average (ARMA) model

This model is more complex because y_t depends not only on the innovation at time t , but also on previous innovations a_{t-1}, a_{t-2} , etc. An ARMA process of autoregressive order p and moving average order q , noted $ARMA(p, q)$, is expressed as follows :

$$y_t + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} = a_t + \theta_1 a_{t-1} + \dots + \theta_q a_{t-q} \quad (2.11)$$

with $\phi_1, \dots, \phi_p$ the parameters of the autoregressive part and $\theta_1, \dots, \theta_q$ the parameters of the moving average part.

It is possible to re-write eq 2.11 in the following form :

$$\phi(B)y_t = \theta(B)a_t \quad (2.12)$$

with

$$\begin{aligned} \phi(B) &= (1 + \phi_1 B + \phi_2 B^2 + \dots + \phi_p B^p) \\ \theta(B) &= (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) \end{aligned}$$

and where B is the lag operator ($B^k y_t = y_{t-k}$).

Property 2.3 *A stochastic process y_t described by eq 2.12 is called **stable** if the modulus of all the roots of the characteristic equation*

$$\Phi(\lambda) = 0$$

are greater than 1.

Property 2.4 *A stochastic process y_t described by eq 2.12 is called **invertible** if the modulus of all the roots of the characteristic equation*

$$\Theta(\lambda) = 0$$

are greater than 1.

Assumption 2.4 *In this work, the time-series process outputs are assumed to be generated by stable and invertible processes.*

The stability assumption allows to avoid processes that would diverge to infinity. The invertibility assumption is needed for the computation of the Hannan-Rissanen method explained further.

Using the same formulation as in eq 2.7 the model is written

$$y_t = H a_t \quad \text{with} \quad H = \phi(B)^{-1} \theta(B) \quad (2.13)$$

Taking eq 2.8, the one-step ahead predictor is $\hat{y}_{t|t-1} = -\phi_1 y_{t-1} - \dots - \phi_p y_{t-p} + \theta_1 a_{t-1} + \dots + \theta_q a_{t-q}$ and the parameter estimates are computed using eq 2.9 with β containing the parameters $\phi_1, \dots, \phi_p$ and $\theta_1, \dots, \theta_q$ and H coming from eq 2.13.

Unlike in the case of a purely AR process, H^{-1} for the general case of an ARMA process is nonlinear in the model parameters (H is a fraction of polynomials) and so the OLS estimates cannot be computed in the same way as for the AR. It is possible to solve directly the nonlinear optimization problem using a particular method (e.g. Newton's method), but good initial values of the parameter estimates are needed. There exist two main ways to compute an approximated solution of this non-linear LS problem.

The Hannan-Rissanen (HR) method

Also called the double-regression method, its based idea (see [14, 15]) is to use an estimation of the real innovation to compute an approximation of

$$\hat{\phi}_i, \hat{\theta}_j = \arg \min_{\phi_i, \theta_j} \frac{1}{N} \sum_{t=1}^N (H^{-1} y_t)^2 \quad (2.14)$$

for $i = 1, \dots, p$ and $j = 1, \dots, q$. To this end, let's call $r = \max(p, q)$ and $\hat{a}_t$ an approximation of the real innovation a_t for $t = 1, \dots, N$.

Equation 2.11 is transformed to isolate y_t and the real innovation is replaced by $\hat{a}_t$:

$$\begin{aligned} y_t &= -\phi_1 y_{t-1} - \dots - \phi_p y_{t-p} + \theta_1 a_{t-1} + \dots + \theta_q a_{t-q} + a_t \\ \Leftrightarrow y_t &= -\phi_1 y_{t-1} - \dots - \phi_p y_{t-p} + \theta_1 \hat{a}_{t-1} + \dots + \theta_q \hat{a}_{t-q} + \tilde{a}_t + \hat{a}_t \\ \Leftrightarrow y_t - \hat{a}_t &= -\phi_1 y_{t-1} - \dots - \phi_p y_{t-p} + \theta_1 \hat{a}_{t-1} + \dots + \theta_q \hat{a}_{t-q} + \tilde{a}_t \end{aligned} \quad (2.15)$$

We see that now y_t depends on a realization $\tilde{a}_t$ of a noise which depends on the real innovations and the estimated innovations $\hat{a}$. Because a_t is the realization of a Gaussian white noise of zero mean and $\hat{a}_t$ is an approximation of it, we do the assumption that $\tilde{a}_t$ is also a Gaussian white noise which allows to compute the OLS estimates of the eq 2.15. Indeed, this equation corresponds to a linear regression problem, where $y_t - \hat{a}_t$ depends on linear combination of y_{t-i} and $\hat{a}_{t-j}$ for $i = 1, \dots, p$ and $j = 1, \dots, q$. This is solved using the OLS estimator, with the approximation of parameter estimate vector $\hat{\beta} = \begin{pmatrix} \hat{\phi}_1 & \dots & \hat{\phi}_p & \hat{\theta}_1 & \dots & \hat{\theta}_q \end{pmatrix}^T \in \mathbb{R}^{p+q}$:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (2.16)$$

with

$$\begin{aligned} X &= \begin{pmatrix} -y_r & -y_{r-1} & \dots & -y_{r-p+1} & \hat{a}_r & \dots & \hat{a}_{r-q+1} \\ -y_{r+1} & -y_r & \dots & -y_{r-p+2} & \hat{a}_{r+1} & \dots & \hat{a}_{r-q+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & -y_{N-2} & \dots & -y_{N-p} & \hat{a}_{N-1} & \dots & \hat{a}_{N-q} \end{pmatrix} \in \mathbb{R}^{N-r-1 \times p+q} \quad \text{and} \\ Y &= \begin{pmatrix} y_{r+1} - \hat{a}_{r+1} \\ y_{r+2} - \hat{a}_{r+2} \\ \vdots \\ y_N - \hat{a}_N \end{pmatrix} \in \mathbb{R}^{N-r-1} \end{aligned} \quad (2.17)$$

For the computation of $\hat{a}_t$, recall that $H^{-1}(B)y_t = a_t$ and $H^{-1}(B) = \theta(B)^{-1}\phi(B)$ is a fraction

of polynomials in B . Instead of keeping a fraction of polynomials, $H^{-1}(B)$ can be expressed as a polynomial with infinite order (corresponding to an impulse response of infinite length). In fact, using the invertibility (Assumption 2.4) and monicity (Assumption 2.3) of the process, it is possible to write eq 2.13 (see [2]) as

$$\begin{aligned} H^{-1}y_t &= a_t \\ (1 + \sum_{k=1}^{\infty} \pi_k B^k)y_t &= a_t \end{aligned} \quad (2.18)$$

with

$$\sum_{k=1}^{\infty} |\pi_k| < \infty \quad (2.19)$$

Definition 2.8 We call *impulse response* of a dynamical system model (given by the transfer function H) the output signal that results when an impulse is applied to the system input.

Each coefficient π_k of eq 2.18 corresponds to the unit impulse response value of H^{-1} at lag k . Given eq 2.19, $\lim_{k \rightarrow \infty} \pi_k = 0$ and $H^{-1}(B)$ is stable, meaning that the impulse response of $H^{-1}(B)$ tends asymptotically to 0. It is then possible to approximate H^{-1} with a finite impulse response of sufficiently large length. This impulse response related to $H_{AR}^{-1}(B)$ corresponds to the impulse response of an AR model of sufficiently large order which is a sufficiently good approximation of $H^{-1}(B)$.

To summarize, the Hannan-Rissanen method works in two steps :

First step A first estimation of the real model is done using an autoregressive model $H_{AR}^{-1}(B)y_t = a_t$ given in an impulse response form of sufficiently large order p_{large} (in the literature, it is agreed that $\log(N)^{1.5} \leq p_{large} \leq \log(N)^2$, see [11]). Using this autoregressive model, an estimation of the residuals is computed

$$\begin{aligned} \hat{a}_t &= y_t - \hat{y}_{t|t-1}^{AR} \\ &= y_t - \hat{\phi}_1^{AR} y_{t-1} - \dots - \hat{\phi}_{p_{large}}^{AR} y_{t-p_{large}} \end{aligned}$$

where $\hat{\phi}_i^{AR}$ are the LS parameter estimates of the AR.

Second step An approximation of the LS parameter estimates $\hat{\phi}_1, \dots, \hat{\phi}_p, \hat{\theta}_1, \dots, \hat{\theta}_q$ is done using the residuals obtained at step 1 and equations 2.16 and 2.17.

A third step can also be applied, using this time the residuals coming from the model estimated at step 2 in order to improve the estimated parameters [11]. For more details, a different formulation and a practical application, see [9].

Maximum Likelihood (ML) methods

The ML methods can also be used to compute parameter estimates. Their basic principle is to find the model parameters which maximize (an approximation of) the likelihood function of the process output.

To provide intuition in the notion of the likelihood function, suppose that we want to associate a N -dimensional random variable to an ARMA model. This random variable corresponds to the possible output measurements of the time-series. It has a probability function $p(Y|\beta, \sigma_a)$ which gives the probability to have the signal $Y = (y_1 \dots y_N)$ knowing the parameters of the model $\beta = (\phi_1 \dots \phi_p \theta_1 \dots \theta_q)$ and the standard deviation of the Gaussian white innovation σ_a . By fixing Y , we have the likelihood function $L(\beta, \sigma_a|Y)$ which is related to the probability that β and σ_a have generated Y .

From this definition, we understand that by maximizing L with respect to the parameters, we maximize the probability of finding the parameters which have the most chance to have generated the set of data Y . The parameters obtained by this maximization are called the Maximum Likelihood (ML) estimates. It is important to note that under the normality assumption of the error term a_t (as assumed in terms of this work), the ML estimates are equivalent to the LS estimates (from eq 2.14) in the case of ARMA processes (see [16]).

There exist different ways to compute the ML estimates. In fact, it is possible to transform the likelihood function into a simpler function that preserves the same optimum points. Typically, the log-likelihood function is used instead of the likelihood (due to its form).

A first technique to compute an approximation of the ML estimates is proposed by Box, Jenkins, and Reinsel [2]. They give a way to compute the unconditional sum of squares function for which the minimum is a good approximation of the optimum point of the likelihood function. A second technique is developed by Gómez [11]. He describes a way to compute an approximation of the log-likelihood function using the Kalman filter. This approximation uses a state space form of the model. For other applications concerning the use of the Kalman filter and state-space form to compute ML estimates, see [20, 21].

The most important drawback of the ML estimates is the fact that a non-linear optimization problem needs to be solved. It is thus necessary to use an efficient algorithm solving this type of problem. Moreover, non-linear optimization implies issues of local minima, convergence, etc that depend on the initial values. One possible way to compute these values would be to use the Hannan-Rissanen method explained before. So the HR would become part of the solution to compute the ML estimates.

2.3.3 Mixed Autoregressive Integrated Moving Average (ARIMA) model

This kind of model is used to describe specific time-series with homogeneous non-stationary behaviour. For these particular processes, we suppose that some difference version of the process output can render it stationary. As such we add an integrated part $(1 - B)^d$ of order d in the model structure.

For example, a time-series that seems to globally increase/decrease in time can be expressed by a Mixed Autoregressive Integrated Moving Average (ARIMA) with $d = 1$ (see an example of such time-series at Figure 2.2).

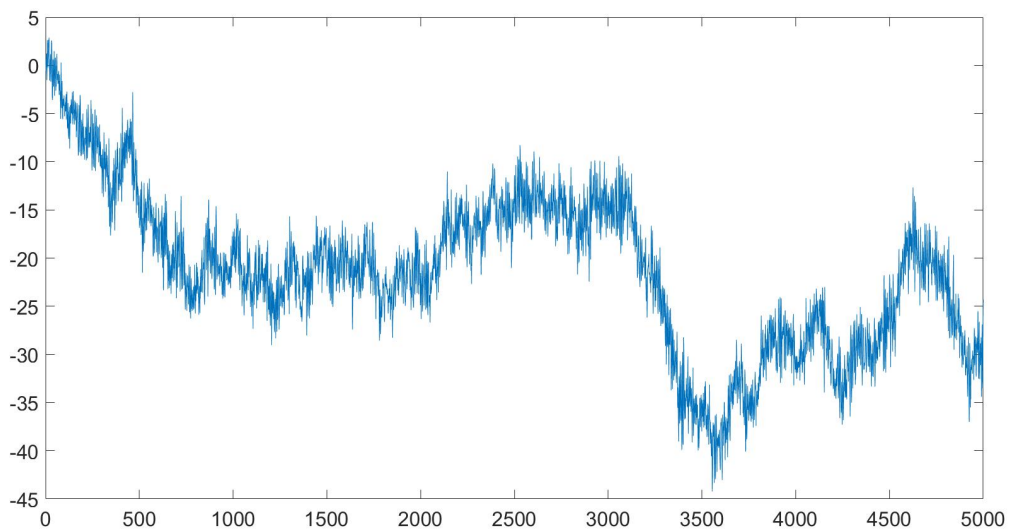


Figure 2.2: Plot of a process described by an ARIMA model and exposing a homogeneous non-stationary behaviour

An $ARIMA(p, q, d)$ model can be written as follows

$$(1 + \phi_1 B + \dots + \phi_p B^p)(1 - B)^d y_t = (1 + \theta_1 B + \dots + \theta_q B^q) a_t$$

$$\Leftrightarrow \phi(B) \nabla^d y_t = \theta(B) a_t \quad (2.20)$$

$$\Leftrightarrow \phi(B) z_t = \theta(B) a_t \quad (2.21)$$

with $\nabla = 1 - B$ the differencing operator and $z_t = \nabla^d y_t$. An $ARIMA$ model for y_t is in fact an $ARMA$ model for $z_t = \nabla^d y_t$. So to estimate such processes, a pre-processing of the signal y is done by transforming it into $z_t = \nabla^d y_t$. Afterwards, parameter estimates of the $ARMA$ model from eq 2.21 are computed.

Eq 2.20 can also be written

$$y_t = H a_t \quad \text{with} \quad H = \nabla^{-d} \phi(B)^{-1} \theta(B) \quad (2.22)$$

in order to compute the one-step ahead predictor using eq 2.8.

2.4 Seasonal univariate models

In many cases there exist time-series occurring from real phenomena which exhibit a seasonal pattern. For example, the consumption of energy, the outside temperature, etc are phenomena with seasonal behaviour.

Let's define the sampling time T_s corresponding to the time between two samples. In this thesis T_s is assumed to be constant. If T_s is expressed in hours, a season of 1 day corresponds to $\frac{24}{T_s}$ samples while a season of 1 week corresponds to $\frac{168}{T_s}$ samples. Knowing that in the application studied later the time-series have a sampling time of one hour ($T_s = 1$), it is said for the rest of the work that a daily seasonality corresponds to 24 samples and a weekly seasonality to 168 samples.

For this type of time-series, it is interesting to use models taking into account the seasonal pattern. And for each regular model described so far, a corresponding seasonal model exists and is presented in the following subsections.

2.4.1 Seasonal Autoregressive (SAR) model

First let's recall the equation of general non-seasonal autoregressive processes :

$$(1 + \phi_1 B + \dots + \phi_p B^p) y_t = a_t$$

At first sight, a simple way to add the seasonality could be simply to add terms with a seasonality lag. For example, noting s the season and P the seasonality order of the model, the model would be

$$(1 + \phi_1 B + \dots + \phi_p B^p + \Phi_1 B^s + \Phi_2 B^{2s} + \dots + \Phi_P B^{Ps}) y_t = a_t$$

This is not necessarily wrong but we can have a more complex model structure while keeping the same number of parameters.

In the literature, the most common seasonal model follows the same idea but multiplication of the regular and the seasonal part is considered. This is expressed in the following equation

$$\phi(B)\Phi(B^s)y_t = a_t \tag{2.23}$$

with

$$\begin{aligned}\phi(B) &= 1 + \phi_1 B + \dots + \phi_p B^p \\ \Phi(B^s) &= 1 + \Phi_1 B^s + \dots + \Phi_P B^{Ps}\end{aligned}$$

The ϕ_i 's correspond to the regular parameters and Φ_j 's to the seasonal ones. This is the multiplicative form of a seasonal Autoregressive model of regular order p , seasonal order P and season s noted $SAR(p)(P)_s$. Eq 2.23 can also be written

$$y_t = H a_t \quad \text{with} \quad H = \Phi(B^s)^{-1} \phi(B)^{-1} \theta(B) \quad (2.24)$$

The difference with the first sketch of model is illustrated with the following example. Consider a $SAR(2)(2)_s$ model. This model can be written

$$\begin{aligned}(1 + \phi_1 B + \phi_2 B^2)(1 + \Phi_1 B^s \Phi_2 B^{2s}) y_t &= a_t \\ \Leftrightarrow (1 + \phi_1 B + \phi_2 B^2 + \Phi_1 B^s + \phi_1 \Phi_1 B^{s+1} + \phi_2 \Phi_1 B^{s+1} \\ &\quad + \Phi_2 B^{2s} + \phi_1 \Phi_2 B^{2s+1} + \phi_2 \Phi_2 B^{2s+2}) y_t = a_t\end{aligned} \quad (2.25)$$

The last equation (2.25) is the non-multiplicative form of the model (having only addition between B terms). From this form, we see that the model takes into account $y_{t-1}, y_{t-2}, y_{t-s}$ and y_{t-2s} , but also $y_{t-s-1}, y_{t-s-2}, y_{t-2s-1}$ and y_{t-2s-2} . So there are more past information incorporated in the model (comparing to the first sketch of seasonal model) for the same number of parameters. These four last terms can be explained intuitively : it seems logical that if y_t depends on y_{t-1} and y_{t-2} , y_{t-s} also depends on y_{t-s-1} and y_{t-s-2} (the two previous realizations). Meaning that if y_t depends on y_{t-s}, y_{t-1} and y_{t-2} , it depends also on y_{t-s-1} and y_{t-s-2} . Doing the same reasoning for y_{t-2s} , we obtain 4 parameters which correlate y_t to 8 different lags.

The general non-multiplicative form of a $SAR(p)(P)_s$ of equation 2.23 is written as follows

$$(1 + \sum_{i=1}^p \phi_i B^i + \sum_{i=1}^P \Phi_i B^{is} + \sum_{i=1}^P \sum_{j=1}^p \phi_j \Phi_i B^{is+j}) y_t = a_t \quad (2.26)$$

Talking about the estimation of SAR processes, it becomes more complicated than for the regular case. Recall that we are still interested in the estimation of the best model in terms of the prediction error :

$$\hat{\phi}_i, \hat{\Phi}_j = \arg \min_{\phi_i, \Phi_j} \frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_{t|t-1})^2$$

for $i = 1, \dots, p, j = 1, \dots, P$ and where $\hat{y}_{t|t-1}$ is calculated using the non multiplicative model (eq

2.26) with $a_t = 0$. Due to the terms containing the multiplication of one regular and one seasonal parameters in eq 2.26, the LS problem becomes non-linear in the optimizing parameters.

To solve this non-linear problem, there exist two main methods proposed in the literature.

Relaxation to a linear problem

A first possibility is to relax the problem and make it linear. To this end, each lag present in the non-multiplicative model is associated with an independent parameter. The non-linear problem of $p + P$ parameters is thus transformed into a linear problem of $\tilde{p} = p + (p + 1)P$ parameters. The OLS estimator is then used to solve this new problem :

For a $SAR(p)(P)_s$ process, with $\hat{\alpha} = (\hat{\alpha}_1 \dots \hat{\alpha}_{\tilde{p}})^T \in \mathbb{R}^{\tilde{p}}$ the independent parameter vector (sorting by decreasing lag) of the relaxed non-multiplicative model

$$(1 + \sum_{i=1}^p \alpha_i B^i + \sum_{i=1}^P \sum_{j=0}^p \alpha_{(p+1)i+j} B^{is+j}) y_t = a_t \quad (2.27)$$

The parameter estimates are

$$\hat{\alpha} = (X^T X)^{-1} X^T Y \quad (2.28)$$

with

$$X = WM \in \mathbb{R}^{(N-\hat{p}-1) \times \tilde{p}} \\ W = \begin{pmatrix} -y_{\hat{p}} & -y_{\hat{p}-1} & \dots & -y_1 \\ -y_{\hat{p}+1} & -y_{\hat{p}} & \dots & -y_2 \\ \vdots & \vdots & \vdots & \ddots \\ -y_{N-1} & -y_{N-2} & \dots & -y_{N-\hat{p}} \end{pmatrix} \in \mathbb{R}^{N-\hat{p}-1 \times \hat{p}} \quad Y = \begin{pmatrix} y_{\hat{p}+1} \\ y_{\hat{p}+2} \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^{N-\hat{p}-1} \quad (2.29)$$

where $\hat{p} = Ps + p$ is the last lag in the $SAR(p)(P)_s$ model and $M \in \mathbb{R}^{\hat{p} \times \tilde{p}}$ is a selection matrix such that X contains only the column of W corresponding to the lag included in the $SAR(p)(P)_s$ model.

The model parameters can be obtained by multiplying $\hat{\alpha}$ by another selection matrix $M_{\alpha} \in \mathbb{R}^{(p+P) \times \tilde{p}}$ such that

$$\begin{pmatrix} \hat{\phi}_1 & \dots & \hat{\phi}_p & \hat{\Phi}_1 & \dots & \hat{\Phi}_P \end{pmatrix}^T = M_{\alpha} \hat{\alpha} \quad (2.30)$$

ML estimates

Box, Jenkins, and Reinsel [2] propose a method to compute an approximation of the unconditional sum of squares function of a *SAR* model. We can thus look for the minimum of this function to obtain the ML parameter estimates.

It is also possible to transform a *SAR* model into its state-space form. It is simply done using the non-multiplicative form of the model (eq 2.26). It is then possible to apply techniques using the Kalman filter to approximate the likelihood, log-likelihood or another function allowing to compute approximations of the ML estimates (see [11]).

These techniques present the same issues as for the *ARMA* models, due to the non-linear optimization problem. The estimated parameters obtained by relaxing the problem can be used as initial values to start the numerical optimization.

2.4.2 Mixed Seasonal Autoregressive-Moving Average (SARMA) model

The principle is exactly the same as for a *SAR* model, but a moving average part is added. A SARMA model, of regular AR order p , regular MA order q , seasonal order P and Q for the AR and MA, respectively, and a season s is expressed by the following equations

$$\phi(B)\Phi(B^s)y_t = \theta(B)\Theta(B^s)a_t \quad (2.31)$$

with

$$\begin{aligned} \phi(B) &= (1 + \phi_1 B + \phi_2 B^2 + \dots + \phi_p B^p) \\ \Phi(B^s) &= (1 + \Phi_1 B^s + \Phi_2 B^{2s} + \dots + \Phi_P B^{Ps}) \\ \theta(B) &= (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) \\ \Theta(B^s) &= (1 + \Theta_1 B^s + \Theta_2 B^{2s} + \dots + \Theta_Q B^{Qs}) \end{aligned}$$

This model is noted $SARMA(p, q)(P, Q)_s$ and can also be expressed by :

$$y_t = H a_t \quad \text{with} \quad H = \Phi(B^s)^{-1} \phi(B)^{-1} \theta(B) \Theta(B^s) \quad (2.32)$$

As in the case of a *SAR* model, the non-multiplicative form of a $SARMA(p, q)(P, Q)_s$ is

given by

$$(1 + \sum_{i=1}^p \phi_i B^i + \sum_{i=1}^P \Phi_i B^{is} + \sum_{i=1}^P \sum_{j=1}^p \phi_j \Phi_i B^{is+j}) y_t = (1 + \sum_{i=1}^q \theta_i B^i + \sum_{i=1}^Q \Theta_i B^{is} + \sum_{i=1}^Q \sum_{j=1}^q \theta_j \Theta_i B^{is+j}) a_t \quad (2.33)$$

Remark 2.1 *Generally the seasonality of the autoregressive and the moving average part are the same. This is explained by the fact that in reality, if there is a certain seasonality for y , the seasonality of the innovations should probably be the same. But we could imagine more specific applications for which this principle is not true (and even with, for example, a different seasonality for the AR and MA parts). In this work we deal with processes which in general have the same seasonality for the AR and MA parts.*

Concerning the computations of the parameter estimates, the methods used with the regular *ARMA* are adapted to take into account the seasonality, as follows :

The Hannan-Rissanen method

Gómez [12] proposed a method following the same steps as with the *ARMA* models :

First step An estimation of an *AR* model of sufficiently large order p_{large} is done (Gómez [12] proposed $p_{large} = \log(N)^2$). The resulting model is used to compute the residual $\hat{a}_t$ as an approximation of the real noise a_t

$$\begin{aligned} \hat{a}_t &= y_t - \hat{y}_{t|t-1}^{AR} \\ &= y_t - \sum_{i=1}^{p_{large}} \hat{\phi}_i^{AR} y_{t-i} \end{aligned}$$

where $\hat{\phi}_i^{AR}$'s are the parameter estimates of the *AR* of large orders. $\hat{a}_t$ is used in eq 2.33 to obtain :

$$\begin{aligned} y_t - \hat{a}_t &= -(\sum_{i=1}^p \phi_i B^i + \sum_{i=1}^P \Phi_i B^{is} + \sum_{i=1}^P \sum_{j=1}^p \phi_j \Phi_i B^{is+j}) y_t + \\ &\quad (\sum_{i=1}^q \theta_i B^i + \sum_{i=1}^Q \Theta_i B^{is} + \sum_{i=1}^Q \sum_{j=1}^q \theta_j \Theta_i B^{is+j}) \hat{a}_t + \tilde{a}_t \quad (2.34) \end{aligned}$$

where $\tilde{a}_t$ is the realization of a noise which depends on the real innovations and the estimated innovations $\hat{a}$, and is supposed to be a zero mean Gaussian white noise.

Second step The LS parameter estimates of eq 2.34 are computed. But this problem is not linear in the parameters yet, due to the product of the seasonal parameters with the regular ones. So, exactly as for the estimation of a *SAR* model, the non-multiplicative model (eq 2.34) is relaxed by adding independent parameters for each lag included in the model. It becomes then a linear LS problem :

For a *SARMA*(p, q)(P, Q)_s process, with $\hat{\alpha} = (\hat{\alpha}_{p,1} \dots \hat{\alpha}_{p,\tilde{p}} \hat{\alpha}_{q,1} \dots \hat{\alpha}_{q,\tilde{q}})^T \in \mathbb{R}^{\tilde{p}+\tilde{q}}$ the independent parameter vector of the non-multiplicative relaxed model

$$y_t - \hat{a}_t = -\left(\sum_{i=1}^p \alpha_i B^i + \sum_{i=1}^P \sum_{j=0}^p \alpha_{(p+1)i+j} B^{is+j}\right) y_t + \left(\sum_{i=1}^q \alpha_{p+(p+1)P+i} B^i + \sum_{i=1}^Q \sum_{j=0}^q \alpha_{p+(p+1)P+(q+1)i+j} B^{is+j}\right) \hat{a}_t + \tilde{a}_t$$

with $\tilde{p} = p + (p+1)P$ and $\tilde{q} = q + (q+1)Q$, the OLS solution is

$$\hat{\alpha} = (X'X)^{-1}X'Y \quad (2.35)$$

with

$$X = WM \in \mathbb{R}^{(N-r-1) \times (\tilde{p}+\tilde{q})}$$

$$W = \begin{pmatrix} -y_r & -y_{r-1} & \dots & -y_{r-\tilde{p}+1} & \hat{a}_r & \dots & \hat{a}_{r-\tilde{q}+1} \\ -y_{r+1} & -y_r & \dots & -y_{r-\tilde{p}+2} & \hat{a}_{r+1} & \dots & \hat{a}_{r-\tilde{q}+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y_{N-1} & -y_{N-2} & \dots & -y_{N-\tilde{p}} & \hat{a}_{N-1} & \dots & \hat{a}_{N-\tilde{q}} \end{pmatrix} \in \mathbb{R}^{(N-r-1) \times (\tilde{p}+\tilde{q})}$$

$$Y = \begin{pmatrix} y_{r+1} - \hat{a}_{r+1} \\ y_{r+2} - \hat{a}_{r+2} \\ \vdots \\ y_N - \hat{a}_N \end{pmatrix} \in \mathbb{R}^{(N-r-1)} \quad (2.36)$$

where $\hat{p} = Ps + p$, $\hat{q} = Qs + q$, $r = \max(\hat{p}, \hat{q})$ and $M \in \mathbb{R}^{(\hat{p}+\hat{q}) \times (\tilde{p}+\tilde{q})}$ is a selection matrix such that X contains only the columns of W corresponding to the lag included in the *SARMA*(p, q)(P, Q)_s model.

Once again, $\hat{\alpha}$ is multiplied by a selection matrix $M_\alpha \in \mathbb{R}^{(p+P+q+Q) \times (\tilde{p}+\tilde{q})}$ such that

$$(\hat{\phi}_1 \dots \hat{\phi}_p \hat{\Phi}_1 \dots \hat{\Phi}_P \hat{\theta}_1 \dots \hat{\theta}_p \hat{\Theta}_1 \dots \hat{\Theta}_P)^T = M_\alpha \hat{\alpha}$$

Note that the first step of this method does not take advantage of the seasonality of the model. In particular models for which s is large (more than 48 for example) have an impulse response that tend to zero between each season. This means that the AR at step 1 will estimate parameters that should be near 0. Furthermore, for even larger season, $p_{large} = \log(N)^2$ becomes

not large enough to take into account the seasonal lags (for $N = 15000$, $\log(N)^2 = 92.46$, or s can reach 168 or even 336). To counter that, an adaptation of the HR method for seasonal models is proposed at Section 3.1.

Maximum Likelihood methods

It is possible to use the techniques of Box et al. [2] and Gómez [11] with *SARMA* processes to approximate the ML estimates. The same issues are encountered regarding the optimization problem, and it is again possible to use the parameter estimates of the HR method as initial values of this problem.

2.4.3 Mixed Seasonal Autoregressive Integrated Moving Average (SARIMA) model

A *SARIMA*(p, q, d), (P, Q, D) $_s$ can be written as follows

$$\phi(B)\Phi(B^s)\nabla^d\nabla_s^D y_t = \theta(B)\Theta(B^s)a_t \quad (2.37)$$

$$\Leftrightarrow \quad \phi(B)\Phi(B^s)z_t = \theta(B)\Theta(B^s)a_t \quad (2.38)$$

with $\nabla = 1 - B$ and $\nabla_s = 1 - B^s$ the differencing operators and $z_t = \nabla^d\nabla_s^D y_t$. Like in the case of the regular models, the initial signal y is transformed by applying the differencing operators to have z and after the *SARMA* parameter estimates of eq 2.38 are computed.

Eq 2.37 can also be written

$$y_t = Ha_t \quad \text{with} \quad H = \nabla_s^{-D}\nabla^{-d}\Phi(B^s)^{-1}\phi(B)^{-1}\theta(B)\Theta(B^s) \quad (2.39)$$

which allows to compute the one-step ahead predictor using eq 2.8.

2.4.4 Double-seasonal models

The previous models describe processes with a single seasonal pattern in their behaviour. But in practise it's not rare to find processes with double seasonal behaviour (for example one daily and one weekly). For each of the single seasonality models seen previously, a double seasonal model can be constructed by multiplying the model by a third polynomial describing the second seasonal behaviour. The most general model structure of this kind is the Mixed double Seasonal Autoregressive Integrated Moving Average (*2SARIMA*). The *2SARIMA* of first season s_1 and second season s_2 such as $s_1 < s_2$ is described by the following equations

$$\phi(B)\Phi_{s_1}(B^{s_1})\Phi_{s_2}(B^{s_2})\nabla^d\nabla_{s_1}^{D_1}\nabla_{s_2}^{D_2}y_t = \theta(B)\Theta_{s_1}(B^{s_1})\Theta_{s_2}(B^{s_2})a_t \quad (2.40)$$

$$\phi(B)\Phi_{s_1}(B^{s_1})\Phi_{s_2}(B^{s_2})z_t = \theta(B)\Theta_{s_1}(B^{s_1})\Theta_{s_2}(B^{s_2})a_t \quad (2.41)$$

with

$$\begin{aligned} \phi(B) &= (1 + \phi_1 B + \phi_2 B^2 + \dots + \phi_p B^p) \\ \Phi_{s_1}(B^{s_1}) &= (1 + \Phi_1^{s_1} B^{s_1} + \Phi_2^{s_1} B^{2s_1} + \dots + \Phi_{P_1}^{s_1} B^{P_1 s_1}) \\ \Phi_{s_2}(B^{s_2}) &= (1 + \Phi_1^{s_2} B^{s_2} + \Phi_2^{s_2} B^{2s_2} + \dots + \Phi_{P_2}^{s_2} B^{P_2 s_2}) \\ \theta(B) &= (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) \\ \Theta_{s_1}(B^{s_1}) &= (1 + \Theta_1^{s_1} B^{s_1} + \Theta_2^{s_1} B^{2s_1} + \dots + \Theta_{Q_1}^{s_1} B^{Q_1 s_1}) \\ \Theta_{s_2}(B^{s_2}) &= (1 + \Theta_1^{s_2} B^{s_2} + \Theta_2^{s_2} B^{2s_2} + \dots + \Theta_{Q_2}^{s_2} B^{Q_2 s_2}) \\ \nabla &= 1 - B \\ \nabla_{s_1} &= 1 - B^{s_1} \\ \nabla_{s_2} &= 1 - B^{s_2} \end{aligned}$$

and is noted $2SARIMA(p, q, d)(P_1, Q_1, D_1)_{s_1}(P_2, Q_2, D_2)_{s_2}$. The Mixed double Seasonal Autoregressive-Moving Average $2SARMA(p, q)(P_1, Q_1)_{s_1}(P_2, Q_2)_{s_2}$ can be expressed as equation 2.41 by fixing d, D_1 and D_2 to zero and the double Seasonal Autoregressive $2SAR(p)(P_1)_{s_1}(P_2)_{s_2}$ by fixing also q, Q_1 and Q_2 to zero. In order to compute the one-step ahead predictor using eq 2.8, eq 2.40 can be written

$$y_t = H a_t \quad \text{with} \quad H = \nabla_{s_2}^{-D_2} \nabla_{s_1}^{-D_1} \nabla^{-d} \Phi_{s_2}(B^{s_2})^{-1} \Phi_{s_1}(B^{s_1})^{-1} \phi(B)^{-1} \theta(B) \Theta_{s_1}(B^{s_1}) \Theta_{s_2}(B^{s_2}) \quad (2.42)$$

Remark 2.2 For this work, the products $P_1 s_1$ and $Q_1 s_1$ are assumed to be strictly lower than s_2 , and p and q strictly lower than s_1 . If these conditions are not respected, the first seasonal parameters will overlap the second seasonal parameters (or regular parameters will overlap first seasonal ones) and same lags could be present in the model in different terms.

All the techniques explained previously in this chapter to compute the simple seasonal parameter estimates can be adapted to the double seasonal case. Nevertheless, there is no explicit version of the HR method in the literature to estimate process with double seasonal behaviour. At this purpose, the model is transformed in non-multiplicative form and then the same steps follow.

For the 2SAR case, the equations 2.28 and 2.29 with $\hat{p} = P_2 s_2 + P_1 s_1 + p$ and $\tilde{p} = p + (p + 1)P_1 + (p + (p + 1)P_1 + 1)P_2$ are used to compute the OLS solution of the relaxed non-multiplicative

problem. The selection matrices M and M_α are adapted to finally obtain the estimates $\hat{\phi}_i$, $\hat{\Phi}_j^{s_1}$ and $\hat{\Phi}_k^{s_2}$ for $i = 1, \dots, p$, $j = 1, \dots, P_1$ and $k = 1, \dots, P_2$.

For 2SARMA models, we can use the same first step of the HR method as for SARMA models. Concerning the second step, equations 2.35 and 2.36 with $\hat{p} = P_2 s_2 + P_1 s_1 + p$, $\hat{q} = Q_2 s_2 + Q_1 s_1 + q$, $\tilde{p} = p + (p + 1)P_1 + (p + (p + 1)P_1 + 1)P_2$, $\tilde{q} = q + (q + 1)Q_1 + (q + (q + 1)Q_1 + 1)Q_2$ and $r = \max(\hat{p}, \hat{q})$ are used to compute the OLS estimates of the relaxed non-multiplicative problem. The selection matrices M and M_α are also adapted to obtain finally the estimates $\hat{\phi}_i$, $\hat{\Phi}_j^{s_1}$, $\hat{\Phi}_k^{s_2}$, $\hat{\theta}_l$, $\hat{\Theta}_m^{s_1}$ and $\hat{\Theta}_n^{s_2}$ for $i = 1, \dots, p$, $j = 1, \dots, P_1$, $k = 1, \dots, P_2$, $l = 1, \dots, q$, $m = 1, \dots, Q_1$ and $n = 1, \dots, Q_2$.

The methods using the likelihood function can also be adapted to be applied n process with a double seasonal behaviour.

For the complete 2SARIMA model, y_t is transformed into $z_t = \nabla^d \nabla_{s_1}^{D_1} \nabla_{s_2}^{D_2} y_t$ and the 2SARMA parameter estimates of eq 2.41 are computed.

2.5 Multivariate models

A lot of different models to describe univariate processes were presented so far, but it's also possible to deal with the case of modeling several different process outputs simultaneously, when the latter are correlated with each other. In this section, adaptations of the univariate models to the multivariate case are presented.

When there are e.g. K different processes, y_t becomes a vector $\mathbf{y}_t = (y_{t,1} \ y_{t,2} \ \dots \ y_{t,K})^T \in \mathbb{R}^K$ containing the K variables $y_{t,i}$'s. With each variable $y_{t,i}$, a Gaussian white noise is associated. As such the innovation at time t is now a vector $\mathbf{a}_t = (a_{t,1} \ a_{t,2} \ \dots \ a_{t,K})^T \in \mathbb{R}^K$ containing the i.i.d. realizations of K zero mean Gaussian stationary processes.

Before analyzing the model structures, let's define a new parameter estimation criterion. Recall that the objective is to have PE as small as possible. The Euclidean distance defined as

$$\|\mathbf{w}_t\|_2 = \sqrt{w_{t,1}^2 + w_{t,2}^2 + \dots + w_{t,K}^2}$$

is the classical metric used to measure the distance between two points in several dimensions. Using this metric, the LS criterion is obtained for the multivariate case

$$\min \sum_{t=1}^N \frac{1}{N} \|\mathbf{e}_t\|_2^2 \text{ with } \mathbf{e}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$$

This choice is guided by the same reasons as for the univariate case (the estimator properties

are the same).

Remark 2.3 Note that Gómez [11] proposed a way to transform regular multivariate models into a state space form. This method can be adapted to transform multivariate seasonal models, using the non-multiplicative form. So it is possible for all the models presented further in this section to compute the ML estimators using the Kalman filter.

2.5.1 Multivariate Autoregressive models

The regular multivariate Autoregressive (VAR) model is written as follows

$$\mathbf{y}_t + \phi_1 \mathbf{y}_{t-1} + \cdots + \phi_p \mathbf{y}_{t-p} = \mathbf{a}_t \quad (2.43)$$

$$\Leftrightarrow (1 + \phi_1 B + \cdots + \phi_p B^p) \mathbf{y}_t = \mathbf{a}_t \quad (2.44)$$

$$\Leftrightarrow \phi(B) \mathbf{y}_t = \mathbf{a}_t \quad (2.45)$$

where $\phi_i \in \mathbb{R}^{K \times K}$ for $i = 1, \dots, p$ and B is the lag operator. In the ϕ_i 's, the parameters on the j th line influence the variable j (in red and orange in the following example) and the parameters on the k th column multiply the variable k (in green and blue). To illustrate this, a general AR of order 2 with 2 variables is developed :

$$\begin{aligned} & \mathbf{y}_t + \phi_1 \mathbf{y}_{t-1} + \phi_2 \mathbf{y}_{t-2} = \mathbf{a}_t \\ \Leftrightarrow & \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} + \begin{pmatrix} \phi_1(1,1) & \phi_1(1,2) \\ \phi_1(2,1) & \phi_1(2,2) \end{pmatrix} \begin{pmatrix} y_{t-1,1} \\ y_{t-1,2} \end{pmatrix} + \begin{pmatrix} \phi_2(1,1) & \phi_2(1,2) \\ \phi_2(2,1) & \phi_2(2,2) \end{pmatrix} \begin{pmatrix} y_{t-2,1} \\ y_{t-2,2} \end{pmatrix} = \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \\ \Leftrightarrow & \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} + \begin{pmatrix} \phi_1(1,1)y_{t-1,1} + \phi_1(1,2)y_{t-1,2} \\ \phi_1(2,1)y_{t-1,1} + \phi_1(2,2)y_{t-1,2} \end{pmatrix} + \begin{pmatrix} \phi_2(1,1)y_{t-2,1} + \phi_2(1,2)y_{t-2,2} \\ \phi_2(2,1)y_{t-2,1} + \phi_2(2,2)y_{t-2,2} \end{pmatrix} = \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \end{aligned}$$

which results in

$$\begin{cases} y_{t,1} + \phi_1(1,1)y_{t-1,1} + \phi_1(1,2)y_{t-1,2} + \phi_2(1,1)y_{t-2,1} + \phi_2(1,2)y_{t-2,2} = a_{t,1} \\ y_{t,2} + \phi_1(2,1)y_{t-1,1} + \phi_1(2,2)y_{t-1,2} + \phi_2(2,1)y_{t-2,1} + \phi_2(2,2)y_{t-2,2} = a_{t,2} \end{cases}$$

For the regular model, the OLS estimator is applied using the following equations :

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

with

$$X = \begin{pmatrix} -\mathbf{y}_p^T & -\mathbf{y}_{p-1}^T & \cdots & -\mathbf{y}_1^T \\ -\mathbf{y}_{p+1}^T & -\mathbf{y}_p^T & \cdots & -\mathbf{y}_2^T \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{y}_{N-1}^T & -\mathbf{y}_{N-2}^T & \cdots & -\mathbf{y}_{N-p}^T \end{pmatrix} \in \mathbb{R}^{(N-p-1) \times pK} \quad Y = \begin{pmatrix} \mathbf{y}_{p+1}^T \\ \mathbf{y}_{p+2}^T \\ \vdots \\ \mathbf{y}_N^T \end{pmatrix} \in \mathbb{R}^{(N-p-1) \times K}$$

where $\hat{\beta} = (\hat{\phi}_1 \quad \hat{\phi}_2 \quad \dots \quad \hat{\phi}_p)^T \in \mathbb{R}^{Kp \times K}$ is the parameter estimate matrix.

In case of seasonal behaviour present in the time-series, the multivariate Seasonal Autoregressive model (VSAR) is given by

$$\phi(B)\Phi(B^s)\mathbf{y}_t = \mathbf{a}_t$$

with

$$\begin{aligned} \phi(B) &= 1 + \phi_1 B + \cdots + \phi_p B^p \\ \Phi(B^s) &= 1 + \Phi_1 B^s + \cdots + \Phi_P B^{Ps} \end{aligned}$$

where ϕ_i and $\Phi_j \in \mathbb{R}^{K \times K}$ for $i = 1, \dots, p$ and $j = 1, \dots, P$. Exactly as for the univariate model, the LS problem can be relaxed to make it linear. The parameter estimate vector $\hat{\alpha} = (\hat{\alpha}_1 \quad \dots \quad \hat{\alpha}_{\tilde{p}})^T \in \mathbb{R}^{\tilde{p}K \times K}$ of the non-multiplicative model of $\tilde{p} = p + (p+1)P$ independent parameters is given by

$$\hat{\alpha} = (X^T X)^{-1} X^T Y$$

with

$$\begin{aligned} X &= WM \in \mathbb{R}^{(N-\hat{p}-1) \times \hat{p}K} \\ W &= \begin{pmatrix} -\mathbf{y}_{\hat{p}} & -\mathbf{y}_{\hat{p}-1} & \cdots & -\mathbf{y}_1 \\ -\mathbf{y}_{\hat{p}+1} & -\mathbf{y}_{\hat{p}} & \cdots & -\mathbf{y}_2 \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{y}_{N-1} & -\mathbf{y}_{N-2} & \cdots & -\mathbf{y}_{N-\hat{p}} \end{pmatrix} \in \mathbb{R}^{(N-\hat{p}-1) \times \hat{p}K} \\ Y &= \begin{pmatrix} \mathbf{y}_{\hat{p}+1} \\ \mathbf{y}_{\hat{p}+2} \\ \vdots \\ \mathbf{y}_N \end{pmatrix} \in \mathbb{R}^{(N-\hat{p}-1) \times K} \end{aligned}$$

where $\hat{p} = Ps + p$ and $M \in \mathbb{R}^{\hat{p}K \times \hat{p}K}$ is a selection matrix such that X contains only the columns of W corresponding to the lag included in the $VSAR(p)(P)_s$ model.

The last thing to do is to multiply $\hat{\alpha}$ by another selection matrix $M_\alpha \in \mathbb{R}^{(p+P) \times \tilde{p}K}$ such that

$$\begin{pmatrix} \hat{\phi}_1 & \dots & \hat{\phi}_p & \hat{\Phi}_1 & \dots & \hat{\Phi}_P \end{pmatrix}^T = M_\alpha \hat{\alpha}$$

Concerning the Multivariate double Seasonal Autoregressive (V2SAR) model, as for the univariate case it is possible to adapt the previous equations used for the *VSAR*. For a *V2SAR*(p)(P_1) $_{s_1}$ (P_2) $_{s_2}$, $\tilde{p} = p + (p+1)P_1 + (p + (p+1)P_1 + 1)P_2$ and $\hat{p} = P_2s_2 + P_1s_1 + p$ are used and the selection matrices M and M_α are adapted.

2.5.2 Multivariate mixed Autoregressive-Moving Average models

The double Seasonal multivariate *ARMA* of K variables, noted *V2SARMA*(p, q)(P_1, Q_1) $_{s_1}$ (P_2, Q_2) $_{s_2}$ is written

$$\phi(B)\Phi_{s_1}(B^{s_1})\Phi_{s_2}(B^{s_2})\mathbf{y}_t = \theta(B)\Theta_{s_1}(B^{s_1})\Theta_{s_2}(B^{s_2})\mathbf{a}_t$$

with

$$\begin{aligned} \phi(B) &= (1 + \phi_1 B + \phi_2 B^2 + \dots + \phi_p B^p) \\ \Phi_{s_1}(B^{s_1}) &= (1 + \Phi_1^{s_1} B^{s_1} + \Phi_2^{s_1} B^{2s_1} + \dots + \Phi_{P_1}^{s_1} B^{P_1 s_1}) \\ \Phi_{s_2}(B^{s_2}) &= (1 + \Phi_1^{s_2} B^{s_2} + \Phi_2^{s_2} B^{2s_2} + \dots + \Phi_{P_2}^{s_2} B^{P_2 s_2}) \\ \theta(B) &= (1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q) \\ \Theta_{s_1}(B^{s_1}) &= (1 + \Theta_1^{s_1} B^{s_1} + \Theta_2^{s_1} B^{2s_1} + \dots + \Theta_{Q_1}^{s_1} B^{Q_1 s_1}) \\ \Theta_{s_2}(B^{s_2}) &= (1 + \Theta_1^{s_2} B^{s_2} + \Theta_2^{s_2} B^{2s_2} + \dots + \Theta_{Q_2}^{s_2} B^{Q_2 s_2}) \end{aligned}$$

where $\phi_i, \Phi_j^{s_1}, \Phi_k^{s_2}, \theta_l, \Theta_m^{s_1}$ and $\Theta_n^{s_2} \in \mathbb{R}^{K \times K}$ for $i = 1, \dots, p, j = 1, \dots, P_1, k = 1, \dots, P_2, l = 1, \dots, q, m = 1, \dots, Q_1$ and $n = 1, \dots, Q_2$. For the multivariate mixed single Seasonal Autoregressive-Moving Average (VSARMA) and the multivariate mixed regular Autoregressive-Moving Average (VARMA) models, the parameters which not are part of the model are fixed to zero.

To compute the LS parameter estimates, the HR method is adapted to the multivariate case. The first step still involves the estimation $\hat{\mathbf{a}}_t$ of the real innovation $\mathbf{a}_t$ using a corresponding multivariate AR model of sufficiently large (p_{large} has the same values as in the univariate cases).

For the second step, the OLS solution of the relaxed non-multiplicative problem (using $\hat{a}_t$

instead of a_t), $\hat{\alpha} = \left(\hat{\alpha}_{p,1} \quad \dots \quad \hat{\alpha}_{p,\hat{p}} \quad \hat{\alpha}_{q,1} \quad \dots \quad \hat{\alpha}_{q,\hat{q}} \right)^T \in \mathbb{R}^{(\hat{p}+\hat{q})K \times K}$, is computed as follows

$$\hat{\alpha} = (X'X)^{-1}X'Y$$

with

$$X = WM \in \mathbb{R}^{(N-r-1) \times (\hat{p}+\hat{q})K}$$

$$W = \begin{pmatrix} -\mathbf{y}_r^T & -\mathbf{y}_{r-1}^T & \dots & -\mathbf{y}_{r-\hat{p}+1}^T & \hat{\mathbf{a}}_r^T & \dots & \hat{\mathbf{a}}_{r-\hat{q}+1}^T \\ -\mathbf{y}_{r+1}^T & -\mathbf{y}_r^T & \dots & -\mathbf{y}_{r-\hat{p}+2}^T & \hat{\mathbf{a}}_{r+1}^T & \dots & \hat{\mathbf{a}}_{r-\hat{q}+2}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{y}_{N-1}^T & -\mathbf{y}_{N-2}^T & \dots & -\mathbf{y}_{N-\hat{p}}^T & \hat{\mathbf{a}}_{N-1}^T & \dots & \hat{\mathbf{a}}_{N-\hat{q}}^T \end{pmatrix} \in \mathbb{R}^{(N-r-1) \times (\hat{p}+\hat{q})K}$$

$$Y = \begin{pmatrix} \mathbf{y}_{r+1}^T - \hat{\mathbf{a}}_{r+1}^T \\ \mathbf{y}_{r+2}^T - \hat{\mathbf{a}}_{r+2}^T \\ \vdots \\ \mathbf{y}_N^T - \hat{\mathbf{a}}_N^T \end{pmatrix} \in \mathbb{R}^{(N-r-1) \times K}$$

where $r = \max(\hat{p}, \hat{q})$ and $M \in \mathbb{R}^{(\hat{p}+\hat{q})K \times (\hat{p}+\hat{q})K}$ is a selection matrix such that X contains only the columns of W corresponding to the lag included in the model. $\hat{p}$ and $\hat{q}$ are the lag of the last parameter and $\tilde{p}$ and $\tilde{q}$ are the number of relaxed problem parameters, respectively for the AR and MA part.

For the regular model $VARMA(p, q)$, it holds

$$\tilde{p} = p \quad \tilde{q} = q \quad \hat{p} = p \quad \hat{q} = q$$

and α contains the estimates of the model (W is the identity matrix).

For the single seasonal model $VSARMA(p, q)(P, Q)_s$, it holds

$$\tilde{p} = p + (p+1)P \quad \tilde{q} = q + (q+1)Q \quad \hat{p} = Ps + p \quad \hat{q} = Qs + q$$

and the model parameter estimate vector is obtained by multiplying $\hat{\alpha}$ by a selection matrix $M_\alpha \in \mathbb{R}^{(p+P+q+Q)K \times (\tilde{p}+\tilde{q})K}$.

Finally for the double seasonal model $V2SARMA(p, q)(P_1, Q_1)_{s_1}(P_2, Q_2)_{s_2}$, it holds

$$\begin{aligned} \tilde{p} &= p + (p+1)P_1 + (p + (p+1)P_1 + 1)P_2 \\ \tilde{q} &= q + (q+1)Q_1 + (q + (q+1)Q_1 + 1)Q_2 \\ \hat{p} &= P_2s_2 + P_1s_1 + p \\ \hat{q} &= Q_2s_2 + Q_1s_1 + q \end{aligned}$$

and the model parameter estimate vector is obtained by multiplying $\hat{\alpha}$ by a selection matrix $M_\alpha \in \mathbb{R}^{(p+P_1+P_2+q+Q_1+Q_2)K \times (\tilde{p}+\tilde{p})K}$.

2.5.3 Multivariate mixed Autoregressive Integrated Moving Average models

The principle is exactly the same as for the univariate models: first the data are transformed with the differencing operators and then the estimates of the resulting *ARMA* model are computed. The only thing that changes is the fact that the differencing operators are matrices of dimension $K \times K$ in order to make the output signal stationary. E.g., in case of $K = 2$, the matrix of the regular differencing operator is given by $\begin{pmatrix} (1-B)^d & 0 \\ 0 & (1-B)^d \end{pmatrix}$.

2.6 Choice of the estimation method

So far different methods to compute estimation of the models presented in this chapter have been explored. For the rest of the work we focus on the HR method, which consists of a first estimation of the innovations using an AR model of sufficiently large order, followed by the computation of the LS parameter estimates using the estimated innovations. As explained before, for processes with seasonal behaviour, the actual formulation of the HR method does not take advantage of the seasonal behaviour at the first step. So an extension of the method is done at Section 3.1 to improve the method for the seasonal case.

This choice can be explained as follows : the two main methods for model estimation are the Hannan-Rissanen and the Maximum Likelihood. But for the computation of the ML estimates, a non-linear (and often non-convex) optimization problem needs to be numerically solved. This means that good initial values need to be found before solving the problem.

So in fact it is really interesting to focus on the Hannan-Rissanen method. Using this method, there is no need to solve a non-linear problem, even for the seasonal models for which the estimates are computed using the relaxed problem. Furthermore, these estimates could be used as initial values for the likelihood methods. This means that the HR method potentially solves the problem of finding good initial values for the computation of ML estimates (the methods could be then combined). Nevertheless the combination of the methods will not take place in this thesis.

Chapter 3

Simulation-based theoretical study

As explained in the previous chapter, the model parameters to describe seasonal processes will be estimated based on the HR method. For the latter it will be shown at Section 3.2.3 that issues occur with models having large seasons, and we have seen that its first step does not take advantage of the seasonal behaviour of the model. So first of all an extension of the current HR method is proposed, followed by verification of the efficiency of the implementation. This is done by comparing the theory with the practice with multiple simulation examples (i.e. for which the real "unknown process" is known). The precision of the method will be verified in order to conclude on the expectation of obtaining good results on real cases.

Two types of simulations are presented. The first one uses a Monte Carlo method to study the values of the estimated parameters as well as their statistical properties. The second one is a study of the model structure selection. By scanning the PE obtained using different model orders for the same signal, a model structure is derived which predicts the process output in an optimal way. Investigation of the values of the cost function obtained during the scanning indicates whether the proposed model structure selection step can be expected to work properly in a real case.

It is important to note that every method implemented in order to model a process (estimations of each kind of model presented before, computation of the predictions, ...) have been tested using the real innovations to verify that they give the exact real value. Indeed, if the method knows the innovations, it can predict perfectly the model.

Before going further, let's recall the estimation methods. For regular AR models, the LS parameter estimates are computed using the previous lag of y_t present in the model. For the single and double seasonal models, first the LS parameter estimates of the relaxed non-multiplicative problem (see eq 2.27) are calculated. Then the estimates corresponding to the ϕ and Φ model parameters are used to give the final estimated model (using the selection matrix of eq 2.30).

For ARMA with seasonal behaviour, an extension of the HR method is used, and is presented in the following section.

3.1 Extension of HR method for processes with seasonal behaviour

As explained previously, certain issues arise when using the classical HR method (presented in the previous chapter) with seasonal models. The first step involves the computation of an AR model with a large impulse response. In the literature, Gómez [12] proposes to fix this order to $\log(N)^{1.5}$ for regular models, $\log(N)^2$ for single seasonal orders and the double seasonal case is not discussed.

The biggest problem occurs when the model contains a large season. In this case, the order of the AR model of step 1 becomes not sufficiently large to approximate well the innovations. So the seasonality behaviour is not modelled properly at step 1, and it results in poor seasonal parameter estimates at step 2. This can be seen at Figure 3.20 in blue, where the seasonal MA parameter estimates, obtained using the classical HR method to model a SARMA process, are almost zero.

A first solution could be simply to increase the order of the intermediate AR model. However, this could result in the estimation of a lot of parameters, with a certain number of them equals to zero. The idea here is to use seasonal intermediate AR models instead of regular ones, to take advantage of the seasonal behaviour of the process at step 1.

To this aim, the first step of the extended HR method to model processes with single seasonal behaviour consists in the estimation of a $SAR(p_{large})(P_{large})_s$ of sufficiently large orders. Once the estimated parameters are computed using the OLS estimator on the relaxed non-multiplicative problem (see equations 2.27, 2.28, 2.29 and 2.30), approximation of the real innovation a_t is computed as follows :

$$\begin{aligned}\hat{a}_t &= y_t - \hat{y}_{t|t-1}^{AR} \\ &= y_t - \sum_{i=1}^{p_{large}} \hat{\phi}_i^{AR} y_{t-i} - \sum_{i=1}^{P_{large}} \hat{\Phi}_i^{AR} y_{t-is} - \sum_{i=1}^{P_{large}} \sum_{j=1}^{p_{large}} \hat{\phi}_j^{AR} \hat{\Phi}_i^{AR} y_{t-is-j}\end{aligned}$$

where $\hat{\phi}_i^{AR}$'s and $\hat{\Phi}_j^{AR}$'s are the regular and seasonal LS parameter estimates of the $SAR(p_{large})(P_{large})_s$ of large orders, respectively.

For the second step, the estimated residuals $\hat{a}_t$ are used in the non-multiplicative form of the SARMA model (eq 2.34) to compute the final LS parameter estimates (in the same way as in

the classical HR method).

Concerning the estimation of $2SARMA$ models, the first step of the extended HR method estimates a $2SAR(p_{large})(P_{1large})_{s_1}(P_{2large})_{s_2}$ model. The parameter estimates are computed in the same way as for SAR models using the OLS estimator of the relaxed non-multiplicative model. Then an approximation of the real innovation a_t is computed in the following way :

$$\begin{aligned}\hat{a}_t &= y_t - \hat{y}_{t|t-1}^{AR} \\ \Leftrightarrow \hat{a}_t &= y_t - \sum_{i=1}^{p_{large}} \hat{\phi}_i^{AR} y_{t-i} - \sum_{i=1}^{P_{1large}} \hat{\Phi}_i^{s_1, AR} y_{t-is} - \sum_{i=1}^{P_{2large}} \hat{\Phi}_i^{s_2, AR} y_{t-is_2} - \sum_{i=1}^{P_{1large}} \sum_{j=1}^{p_{large}} \hat{\phi}_j^{AR} \hat{\Phi}_i^{s_1, AR} y_{t-is_1-j} \\ &\quad - \sum_{i=1}^{P_{2large}} \sum_{j=1}^{p_{large}} \hat{\phi}_j^{AR} \hat{\Phi}_i^{s_2, AR} y_{t-is_2-j} - \sum_{i=1}^{P_{2large}} \sum_{j=1}^{P_{1large}} \hat{\Phi}_j^{s_1, AR} \hat{\Phi}_i^{s_2, AR} y_{t-is_2-j s_1} \\ &\quad - \sum_{i=1}^{P_{2large}} \sum_{j=1}^{P_{1large}} \sum_{k=1}^{p_{large}} \hat{\phi}_k^{AR} \hat{\Phi}_j^{s_1, AR} \hat{\Phi}_i^{s_2, AR} y_{t-is_2-j s_1-k}\end{aligned}$$

where $\hat{\phi}_i^{AR}$'s, $\hat{\Phi}_j^{s_1, AR}$'s and $\hat{\Phi}_k^{s_2, AR}$, are the regular, first seasonal and second seasonal LS parameter estimates of the $2SAR$ model of large orders, respectively.

Then for the second step, the estimated residuals are used with the non-multiplicative form of the $2SARMA$ model to compute the final LS parameter estimates.

Finally, to describe completely this extended method, the orders of the seasonal AR models of the first step need to be chosen. But the choice of p_{large} and P_{large} for single seasonal and of p_{large} , P_{1large} and P_{2large} for double seasonal models will depend on the model structure used to approximate the process. This choice is done and discussed in Section 3.2.3.

3.2 Properties of the proposed estimation method

The first simulations consist of an arbitrary choice of the true "unknown" process followed by corresponded signal generations. For each signal generation, different random white noises are used, keeping the same variance. Then an estimate of the parameters for each generated signal is done. The so-called Monte Carlo method considers repeatedly random sampling to obtain numerical results. In this way the statistical properties (bias and variance) of the estimates can be systematically studied.

Nonetheless, an analytical analysis of the parameter estimates is done to see if the statistical

properties of the estimates can be described analytically.

3.2.1 Theoretical analysis

Let's first express some definitions and concepts which will be useful in this chapter. For a vector of data $Y \in \mathbb{R}^N$ ($Y \in \mathbb{R}^{N \times K}$ for the multivariate case), a model structure is chosen. This model structure is included in a model set $\mathcal{M}$ of all the possible models having the same characteristics (seasonality, orders, etc). The model structure is also parametrized by a vector $\beta \in \mathbb{R}^{n_\beta}$ containing the $n_\beta = (p + P1 + P2 + q + Q1 + Q2)K^2$ parameters of the model (ϕ_i , θ_j , etc). For univariate models, $K = 1$ and n_β is thus the sum of the orders of the model. For multivariate models, K^2 comes from the fact each parameter is a matrix of dimension $K \times K$. If the model is not a $2SARMA$, the orders not involved in the model are fixed to zero (e.g. for a $SAR(p)(P_1)_{s_1}$, $n_\beta = (p + P_1)K^2$).

An estimated vector $\hat{\beta} \in \mathbb{R}^{n_\beta}$ is mapped from Y via an estimator function, which is described by the estimation method. Since Y is a realization of a stochastic process and $\hat{\beta}$ depends on Y , $\hat{\beta}$ can be seen as a realization of a random variable with specific statistical properties. These properties are affected both by the estimation method and the distribution of the noise.

A good way to quantify the statistical properties of $\hat{\beta}$ is the Mean Square Error (MSE) (equivalent to the Mean squared error states in the previous chapter, see [27]).

Definition 3.1 *The **Mean Square Error (MSE)** of the parameter estimate vector $\hat{\beta}$ is given by*

$$MSE(\hat{\beta}) = b(\hat{\beta})^T b(\hat{\beta}) + Cov(\hat{\beta}) \quad (3.1)$$

where $MSE(\hat{\beta}) \in \mathbb{R}^{n_\beta \times n_\beta}$, $b(\hat{\beta}) = \mathbb{E}[\hat{\beta}] - \beta_0 \in \mathbb{R}^{n_\beta}$ is a vector called bias, $Cov(\hat{\beta}) = \mathbb{E}[(\hat{\beta} - \mathbb{E}[\hat{\beta}])(\hat{\beta} - \mathbb{E}[\hat{\beta}])^T] \in \mathbb{R}^{n_\beta \times n_\beta}$ is the covariance matrix of the estimated parameter vector $\hat{\beta}$ and $\beta_0 \in \mathbb{R}^{n_\beta}$ is the true parameter vector (describing exactly the real process).

Property 3.1 *We call **unbiased** estimator, an estimator for which the bias of the estimates is zero, i.e.*

$$\begin{aligned} b(\hat{\beta}) &= \mathbb{E}[\hat{\beta}] - \beta_0 = 0 \\ &\Leftrightarrow \mathbb{E}[\hat{\beta}] = \beta_0 \end{aligned}$$

Now that the concepts have been defined, let's begin the analytical analysis.

AR models

A process of N realizations described by an univariate AR model can be written as

$$Y = X\beta_0 + a \quad (3.2)$$

where $Y \in \mathbb{R}^{N-\hat{p}-1}$ and $X \in \mathbb{R}^{N-\hat{p}-1 \times n_\beta}$ result from eq 2.10 or 2.29 for the regular or the seasonal model respectively, $\hat{p}$ is equal to p , $Ps + p$ or $P_2s_2 + P_1s_1 + p$ for $AR(p)$, $SAR(p)(P)_s$ or $SAR(p)(P_1)_{s_1}(P_2)_{s_2}$ models, respectively, $\beta_0 \in \mathbb{R}^{n_\beta}$ is the true parameter vector with n_β equals to the sum of the model orders and $a \in \mathbb{R}^{N-\hat{p}-1}$ is the realization of a Gaussian white noise of zero-mean and variance σ_a^2 .

The aim is to compute the LS estimates of β_0 , i.e. $\hat{\beta} = \arg \min_{\beta} \frac{1}{N} \sum_{t=1}^N (H^{-1}y_t)^2$. As $H^{-1} = (1 + B\phi_1 + \dots + B^p\phi_p)(1 + B^{s_1}\Phi_1^{s_1} + \dots + B^{P_1s_1}\Phi_{P_1}^{s_1})(1 + B^{s_2}\Phi_1^{s_2} + \dots + B^{P_2s_2}\Phi_{P_2}^{s_2})$ for the more complex AR models (with double seasonal behaviour), we can say that H^{-1} is linear for all AR models. The estimates are computed using the OLS estimator :

$$\begin{aligned} \hat{\beta} &= (X^T X)^{-1} X^T Y \\ &= (X^T X)^{-1} X^T (X\beta_0 + a) \\ &= (X^T X)^{-1} X^T X\beta_0 + (X^T X)^{-1} X^T a \\ &= \beta_0 + (X^T X)^{-1} X^T a \\ \Leftrightarrow \mathbb{E}[\hat{\beta}] &= \beta_0 + E[(X^T X)^{-1} X^T a] \end{aligned}$$

Since the noise is independent from Y

$$\begin{aligned} \Leftrightarrow \mathbb{E}[\hat{\beta}] &= \beta_0 + E[(X^T X)^{-1} X^T] E[a] \\ &= \beta_0 \quad \text{because } E[a] = 0 \end{aligned}$$

The estimates are thus unbiased i.e. $b(\hat{\beta}) = 0$. Concerning the covariance of the estimated parameter vector $\hat{\beta}$:

$$\begin{aligned} (\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E[\hat{\beta}])^T &= (\hat{\beta} - \beta_0)(\hat{\beta} - \beta_0)^T \\ &= ((X^T X)^{-1} X^T a)((X^T X)^{-1} X^T a)^T \\ &= (X^T X)^{-1} X^T a a^T X (X^T X)^{-1} \\ \Leftrightarrow \mathbb{E}[(\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E[\hat{\beta}])^T] &= \mathbb{E}[(X^T X)^{-1} X^T a a^T X (X^T X)^{-1}] \end{aligned}$$

Once again, like the noise is i.i.d. :

$$\begin{aligned}
\Leftrightarrow \mathbb{E}[(\hat{\beta} - E[\hat{\beta}])(\hat{\beta} - E[\hat{\beta}])^T] &= \mathbb{E}[(X^T X)^{-1} X^T \mathbb{E}[aa^T] X (X^T X)^{-1}] \\
&= \mathbb{E}[(X^T X)^{-1} X^T \sigma_a^2 I X (X^T X)^{-1}] \\
&= \sigma_a^2 \mathbb{E}[(X^T X)^{-1} X^T X (X^T X)^{-1}] \\
&= \sigma_a^2 \mathbb{E}[(X^T X)^{-1}]
\end{aligned}$$

So the MSE of the estimates is $MSE(\hat{\beta}) = Cov(\hat{\beta}) = \sigma_a^2 \mathbb{E}[(X X^T)^{-1}]$. The term $\mathbb{E}[(X X^T)^{-1}]$ is difficult to analyse analytically, due to the fact that the y inside are correlated and depend on previous realizations of the noise a . Nevertheless $\mathbb{E}[(X X^T)^{-1}]$ is probably correlated to the size of Y , N (because the size of X depends on N), and to the white noise, so to σ_a . This will be verified in the simulations.

For the multivariate case, the same analysis can be conducted and the same conclusions are obtained.

ARMA models

This kind of processes is described in general by the following equation :

$$Y = X_a \beta_0 + a \quad (3.3)$$

where Y is the process output, X_a refers to the equation 2.17 using the real innovation instead of $\hat{a}$ (X_a contains previous value of the process output Y and of the innovation), β_0 is the true vector of model parameters and a is a Gaussian white noise of zero-mean and variance σ_a^2 .

The first step of the (extended) HR consists in an estimation $\hat{a}$ of the innovation, using an AR model of sufficiently large orders. This is written as follows

$$Y = X_{AR} \beta_{AR} + \hat{a}$$

with X_{AR} the previous lag appearing in the AR model and β_{AR} its parameters.

In the second step the residuals $\hat{a}$ are used to compute an estimation of the real parameters of the $ARMA$ model. As seen in the section 2.3.2, the process can be written

$$Y - \hat{a} = X_{\hat{a}} \beta_0 + \tilde{a} \quad (3.4)$$

where $X_{\hat{a}}$ is the same as in equation 2.17 ($X_{\hat{a}}$ contains previous Y and estimated noise $\hat{a}$), β_0 is

the true parameter vector of the model and $\tilde{a}$ is a new noise appearing due to the estimation of the innovation. By combining the last equation with equation 3.3, the expression of the new noise is

$$\tilde{a} = (X_a - X_{\hat{a}})\beta_0 + a - \hat{a}$$

The OLS solution of equation 3.4 is then computed :

$$\begin{aligned} \hat{\beta} &= (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T (Y - \hat{a}) \\ &= (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T (X_{\hat{a}}\beta_0 + \tilde{a}) \\ &= (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T X_{\hat{a}}\beta_0 + (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T \tilde{a} \\ &= \beta_0 + (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T \tilde{a} \\ &= \beta_0 + (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T ((X_a - X_{\hat{a}})\beta_0 + a - \hat{a}) \\ &= (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T X_a \beta_0 + (X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T (a - \hat{a}) \\ \Leftrightarrow \mathbb{E}[\hat{\beta}] &= \mathbb{E}[(X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T X_a \beta_0] + \mathbb{E}[(X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T (a - \hat{a})] \\ &= \mathbb{E}[(X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T X_a] \beta_0 + \mathbb{E}[(X_{\hat{a}}^T X_{\hat{a}})^{-1} X_{\hat{a}}^T (a - \hat{a})] \end{aligned} \quad (3.5)$$

We see that in this case, the analytical expression of $\mathbb{E}[\hat{\beta}]$ is more complex. X_a , $X_{\hat{a}}$, a and $\hat{a}$ are correlated, and so it is difficult to simplify eq 3.5. The parameter estimates are then probably biased, and the expression of $Cov(\hat{\beta})$ is not written explicitly because it is too complex to deduce some statistical properties from it. So the bias and the covariance of the estimates are analyzed during the Monte-Carlo simulations in the following sections.

3.2.2 Autoregressive models

In order to study the statistical properties of $\hat{\beta}$, simulations with univariate and multivariate AR, SAR and 2SAR models are presented in this section. For each AR process, 100 parameter estimates coming from a different random white noise realization have been computed. Scatter plots of the results are shown on the figures that follow. 3 different groups of points can be observed in these plots: in red it is the real value of the parameters (β_0), in blue the estimated parameters coming from a signal of size $N = 10000$ and in green the estimated parameters coming from a signal 5 times longer ($N = 50000$). Parameter values are depicted on both axis for each plot.

The real parameter values have been fixed arbitrarily, and such that the assumptions established in Chapter 2 are respected. The multivariate models have 2 variables, and the seasons are 24 for single seasonal and 24 and 168 for double seasonal models (this corresponds to the daily and weekly seasonal behaviour of the real process that will be analyzed in the next chapter).

The models used to generate the simulations are :

AR(8) :

$$(1 - 0.7B + 0.4B^2 - 0.5B^3 + 0.2B^4 + 0.1B^5 + 0.5B^6 - 0.25B^7 - 0.2B^8)y_t = a_t \quad (3.6)$$

VAR(2) :

$$\left[I_2 + \begin{pmatrix} -0.7 & 0.1 \\ -0.4 & -0.4 \end{pmatrix} B + \begin{pmatrix} 0.4 & 0.3 \\ -0.6 & 0.2 \end{pmatrix} B^2 \right] \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} = \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \quad (3.7)$$

SAR(4)(4)₂₄ :

$$(1 + 0.7B + 0.5B^2 - 0.2B^3 - 0.3B^4)(1 - 0.5B^{24} + 0.3B^{48} + 0.3B^{72} - 0.6B^{96})y_t = a_t \quad (3.8)$$

VSAR(2)(2)₂₄ :

$$\begin{aligned} & \left[I_2 + \begin{pmatrix} -0.4 & 0.1 \\ -0.6 & 0.1 \end{pmatrix} B + \begin{pmatrix} -0.4 & -0.3 \\ 0.1 & 0.2 \end{pmatrix} B^2 \right] \\ & \left[I_2 + \begin{pmatrix} -0.2 & -0.2 \\ -0.4 & 0.3 \end{pmatrix} B^{24} + \begin{pmatrix} 0.3 & -0.3 \\ 0.8 & -0.2 \end{pmatrix} B^{48} \right] \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} = \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \end{aligned} \quad (3.9)$$

2SAR(4)(2)₂₄(2)₁₆₈ :

$$(1 + 0.5B + 0.6B^2 - 0.3B^3 - 0.3B^4)(1 - 0.4B^{24} + 0.3B^{48})(1 + 0.25B^{168} - 0.3B^{336})y_t = a_t \quad (3.10)$$

V2SAR(2)(1)₂₄(1)₁₆₈ :

$$\begin{aligned} & \left[I_2 + \begin{pmatrix} -0.2 & 0.5 \\ 0.25 & -0.2 \end{pmatrix} B + \begin{pmatrix} 0.6 & 0.3 \\ -0.2 & -0.3 \end{pmatrix} B^2 \right] \\ & \left[I_2 + \begin{pmatrix} -0.2 & -0.4 \\ -0.2 & 0.3 \end{pmatrix} B^{24} \right] \left[I_2 + \begin{pmatrix} 0.1 & -0.4 \\ -0.25 & -0.25 \end{pmatrix} B^{48} \right] \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} = \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \end{aligned} \quad (3.11)$$

where a_t , $a_{t,1}$ and $a_{t,2}$ are independent innovations of variance 1 and $I_2 \in \mathbb{R}^{2 \times 2}$ is the identity matrix.

A first thing to say before looking at Figures 3.1-3.6, is the fact that the variance σ_a^2 of the noise does not influence the parameter estimate values. In fact, it has been found that for all the simulations done in this chapter, parameter estimates computed on a signal with innovation vector a and on a signal with innovation vector ka (where $k \in \mathbb{R}_0$) have the same values. Using an innovation vector ka is equivalent as generating the same innovation but with a variance

$k^2\sigma_a^2$. So we deduce that the value of σ_a^2 does not influence the statistical properties of the parameter estimates. This can be explained by the fact that the innovation is present in the matrix X through the values of y . So the change of variance in a_t is balanced in $(X^T X)^{-1} X^T$, and $\mathbb{E}[(X^T X)^{-1}]$ contains the factor $\frac{1}{\sigma_a^2}$.

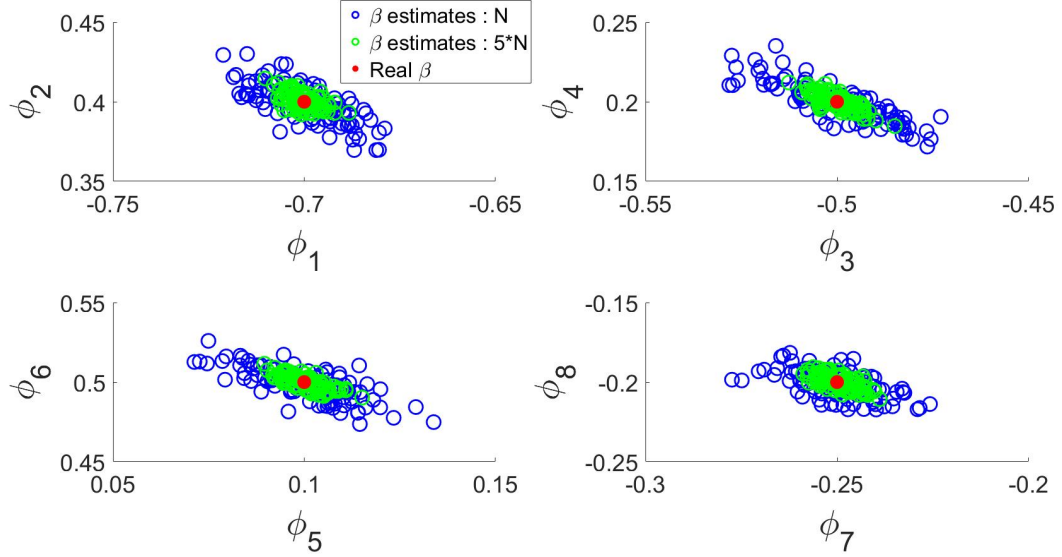


Figure 3.1: Scatter plot of the parameter estimates of the $AR(8)$ of eq 3.6 coming from 100 Monte Carlo realizations

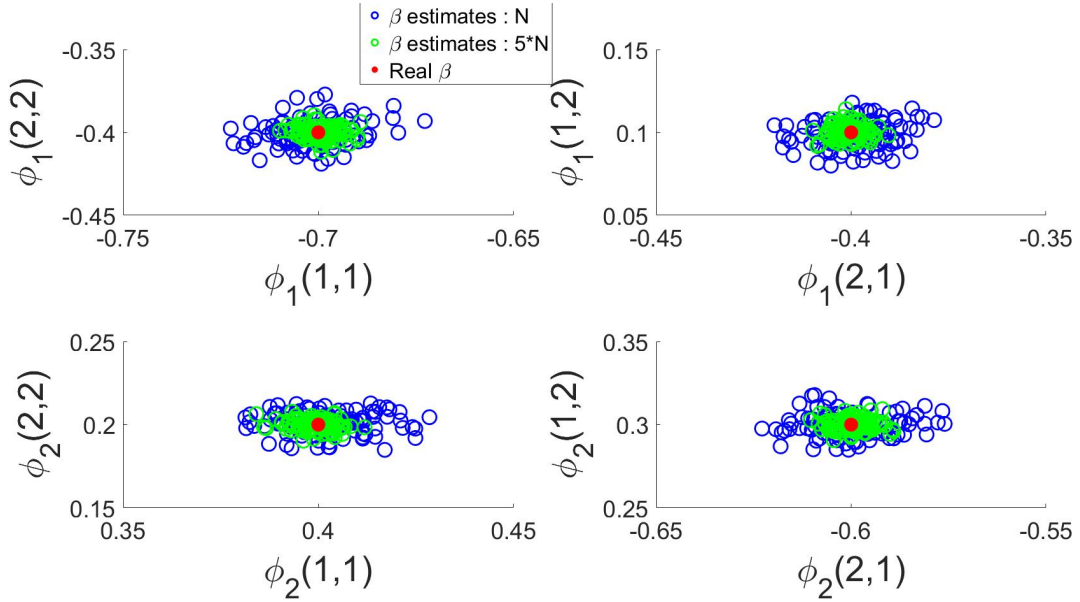


Figure 3.2: Scatter plot of the parameter estimates of the two variables $VAR(2)$ of eq 3.7 coming from 100 Monte Carlo realizations

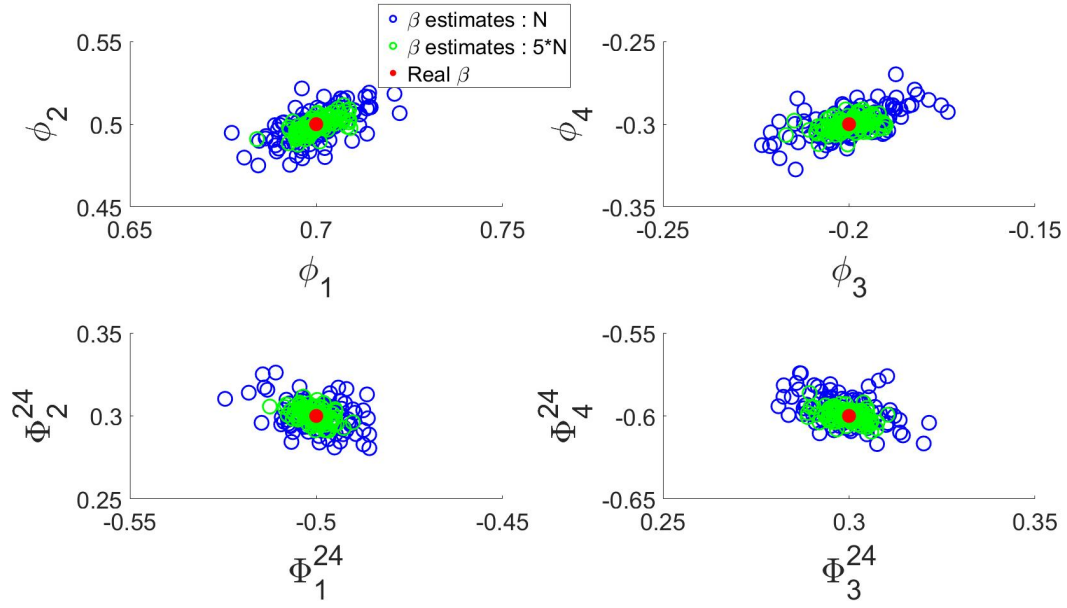


Figure 3.3: Scatter plot of the parameter estimates of the $SAR(4)(4)_{24}$ of eq 3.8 coming from 100 Monte Carlo realizations

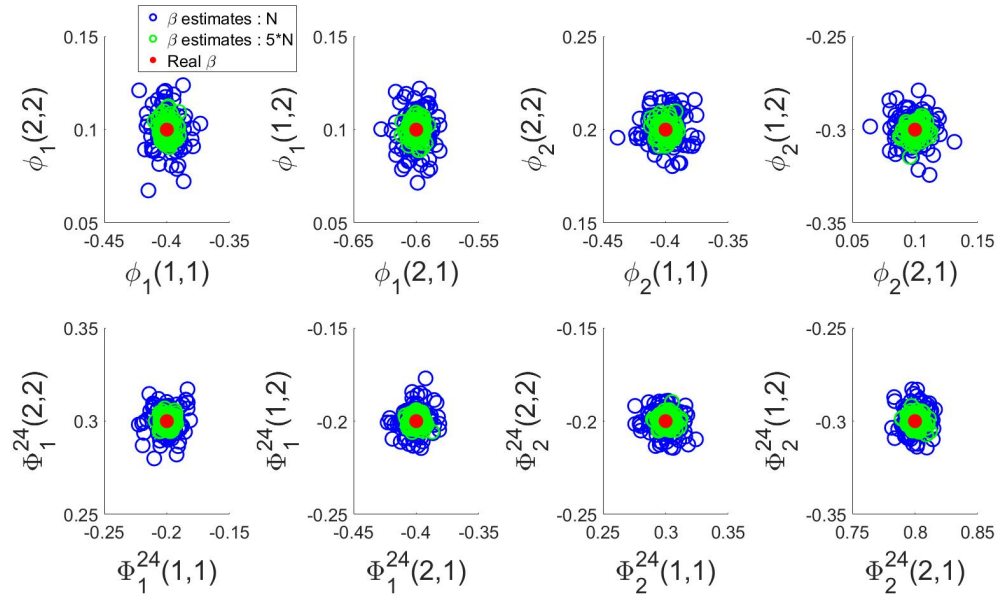


Figure 3.4: Scatter plot of the parameter estimates of the two variables $VSAR(2)(2)_{24}$ of eq 3.9 coming from 100 Monte Carlo realizations

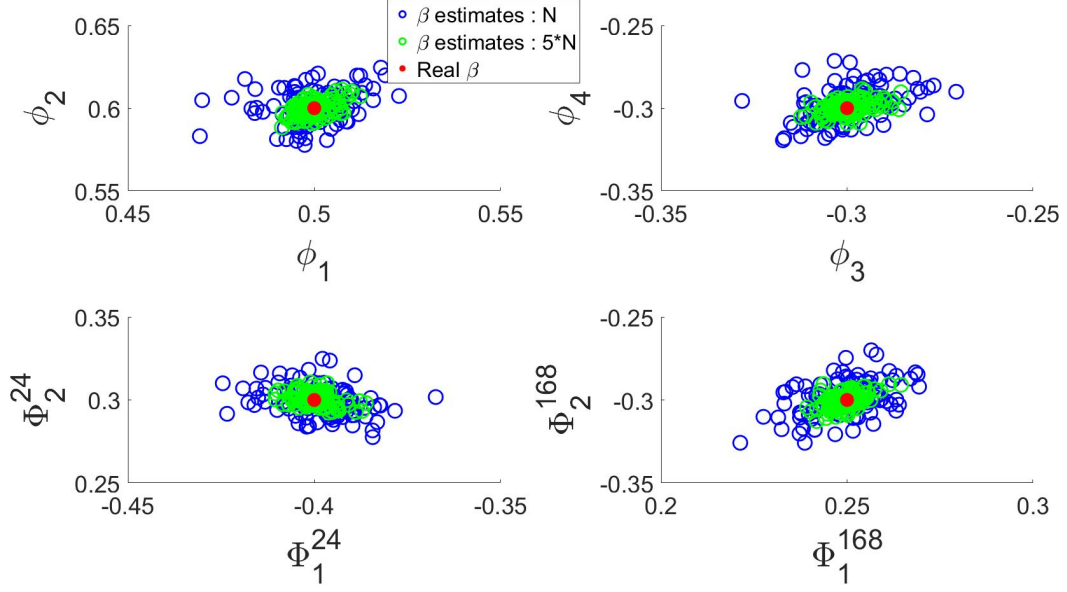


Figure 3.5: Scatter plot of the parameter estimates of the $SAR(4)(2)_{24}(2)_{168}$ of eq 3.10 coming from 100 Monte Carlo realizations

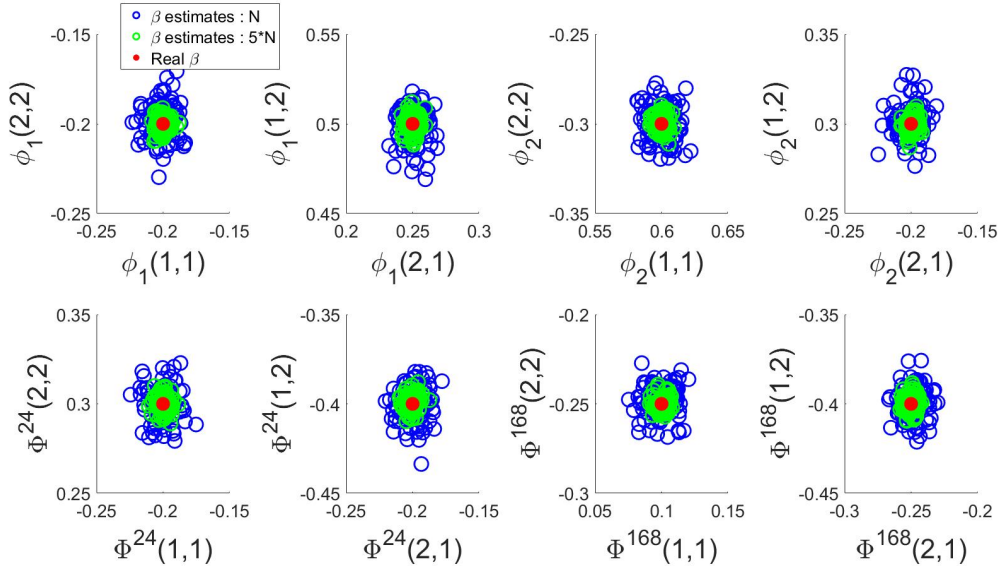


Figure 3.6: Scatter plot of the parameter estimates of the two variables $VSAR(2)(1)_{24}(1)_{168}$ of eq 3.11 coming from 100 Monte Carlo realizations

Looking at Figures 3.1-3.6, the results are quite similar. Concerning the number of data N used to train the model, the plots show that $Cov(\hat{\beta})$ decreases when N increases. Intuitively, it is explained by the fact that the more data are available, the more there is potentially new information to train the model. It also means that $\mathbb{E}[(X^T X)^{-1}]$ is a decreasing function of N , which is due to the fact that X depends on N and the product $X^T X$ is inverted in $Cov(\hat{\beta})$. The plots also confirm that the parameter estimates are unbiased.

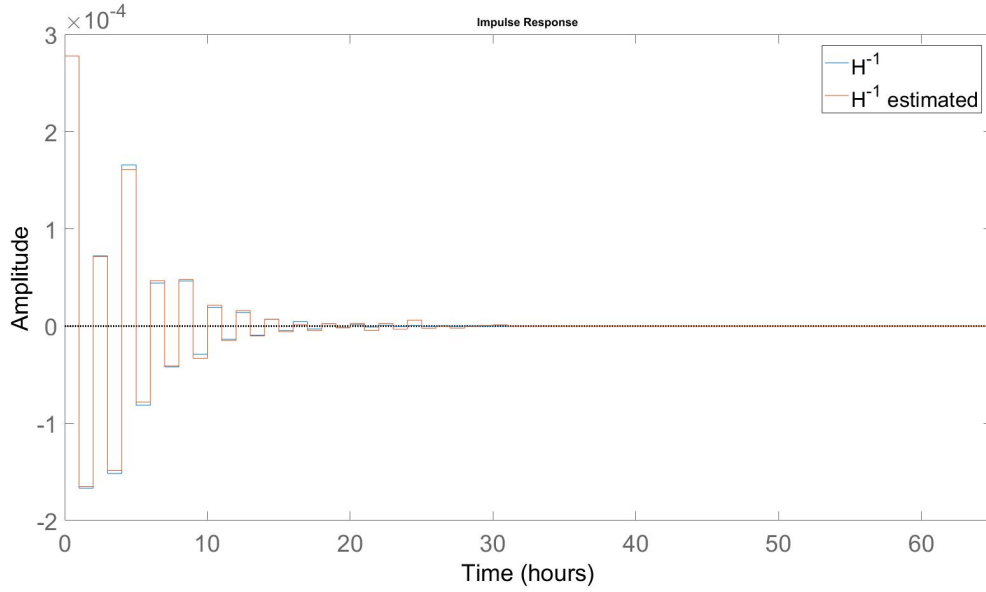
3.2.3 Mixed Autoregressive-Moving Average models

Regular models

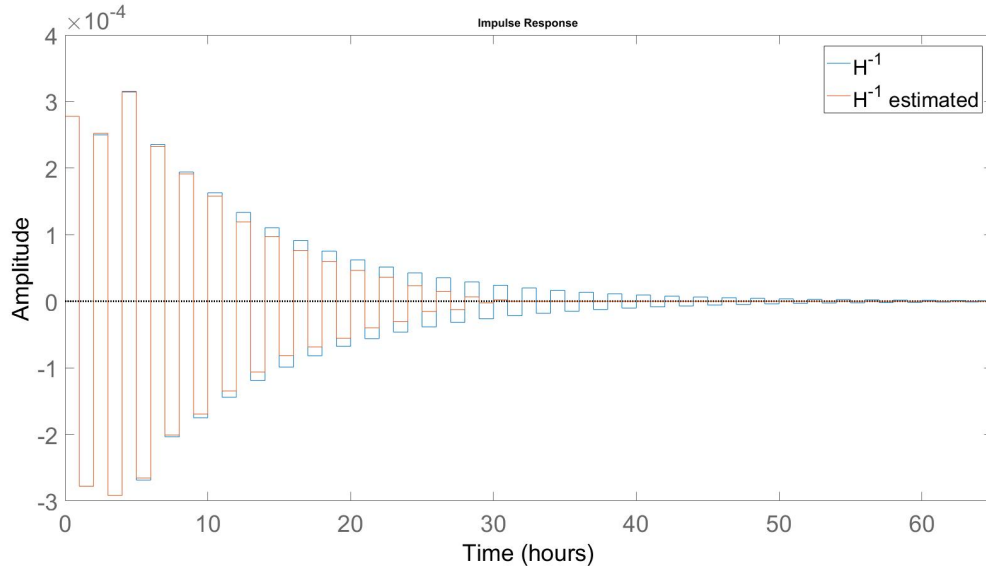
These models are estimated based on the procedure described in the section 2.3.2 (HR method). At the first step of the HR method, an AR model of sufficiently large order is estimated. It is thus necessary to do some tests in simulation to choose a good order according to the parameters of the model. At this purpose, an accurate estimation of the impulse response of H^{-1} (the inverse transfer function of the model) computed at the first step could be a good indication of the efficiency of this step. If this impulse response is shorter than the real one, it means that the orders of the intermediate AR need to be increased and inversely. But the problem with the impulse response of H^{-1} is the fact that it depends strongly on the values of the parameters. For models of same orders, the impulse response can decrease in very different speed to 0. As an example, Figure 3.7 shows two ARMA(4,4) models with the parameters resumed in Table 3.1. We see clearly that the order of the "best" large AR model will be different for the two models (Figure 3.7a has a shorter impulse response than 3.7b). However, only the value of 3 out of 8 parameters are different in the 2 models.

Parameters \ Models	Figure 3.7a	Figure 3.7b
ϕ_1	-0.7	-0.7
ϕ_2	0.4	0.4
ϕ_3	-0.5	-0.5
ϕ_4	0.3	0.3
θ_1	0.1	0.5
θ_2	-0.2	-0.4
θ_3	-0.1	-0.3
θ_4	-0.25	-0.25

Table 3.1: Model parameters corresponding at Figure 3.7



(a) Model with short impulse response



(b) Model with long impulse response

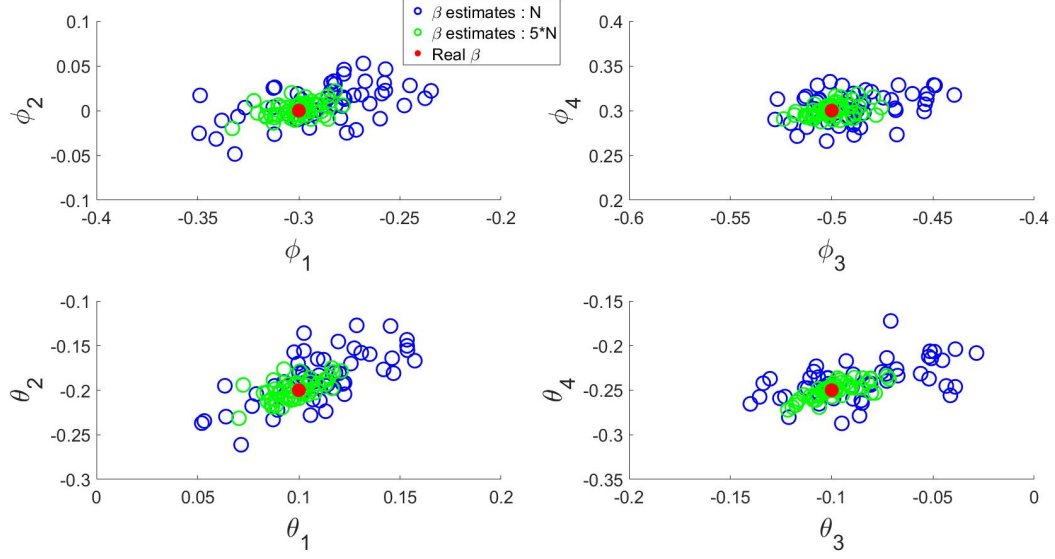
Figure 3.7: Plots of the impulse response of the transfer function H^{-1} for two $ARMA(4,4)$ models with parameters coming from Table 3.1. The estimation of H^{-1} (in red) is done using an $AR(\log(N)^{1.5})$ model with $N = 15000$

The same phenomena appear for seasonal and multivariate models. It is thus impossible to link directly the number of regular internal parameters (p_{large}) to the number of model parameters.

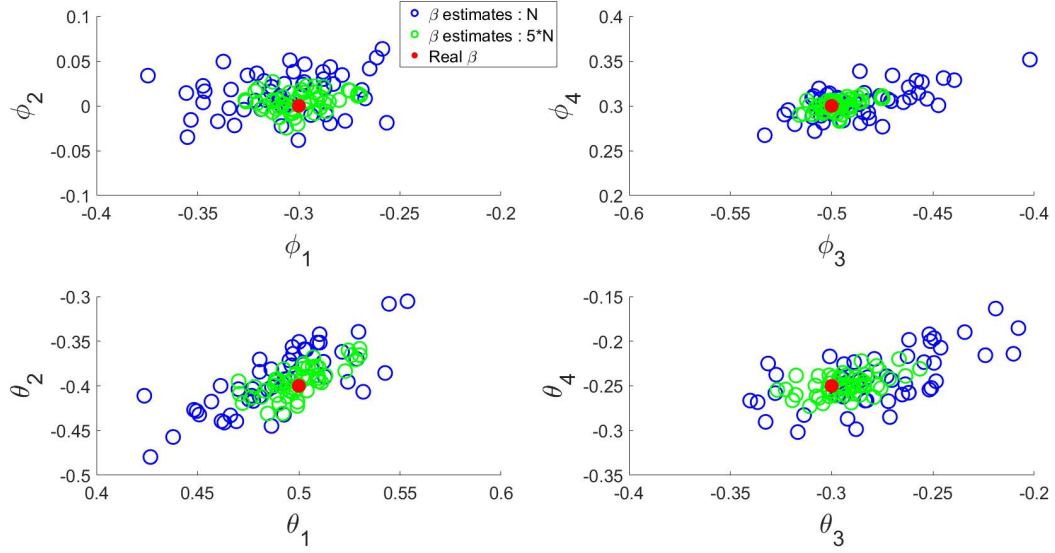
In general, to choose the value of the large orders of the intermediate AR impulse response model of H^{-1} , some simulations are conducted further and based on their results the choice is done and justified. Another possible way (not applied here) to compute these internal parameters would be to use some information criterion. These criteria are indicators of the relative quality

of the estimated model.

For non-seasonal models, there is only one internal parameter (p_{large}) used for the long AR and VAR models of step 1. A value for this parameter suggested is $p_{large} = \log(N)^{1.5}$ [11]. The size of the data used in the real life process of next chapter is approximately $N = 15000$ and Figure 3.7 gives in red the impulse response of the estimated H^{-1} at step 1 using this size of data (for both $ARMA(4, 4)$ models from Table 3.1). Parameter estimates for both models using $p_{large} = \log(N)^{1.5}$ are shown on Figure 3.8. Estimation of a signal of length 10000 is in blue and of length 50000 is in green. We can see that the covariance of the parameter estimates decreases when N increases. As for the AR models, the variance of the noise does not influence the value of the estimates.



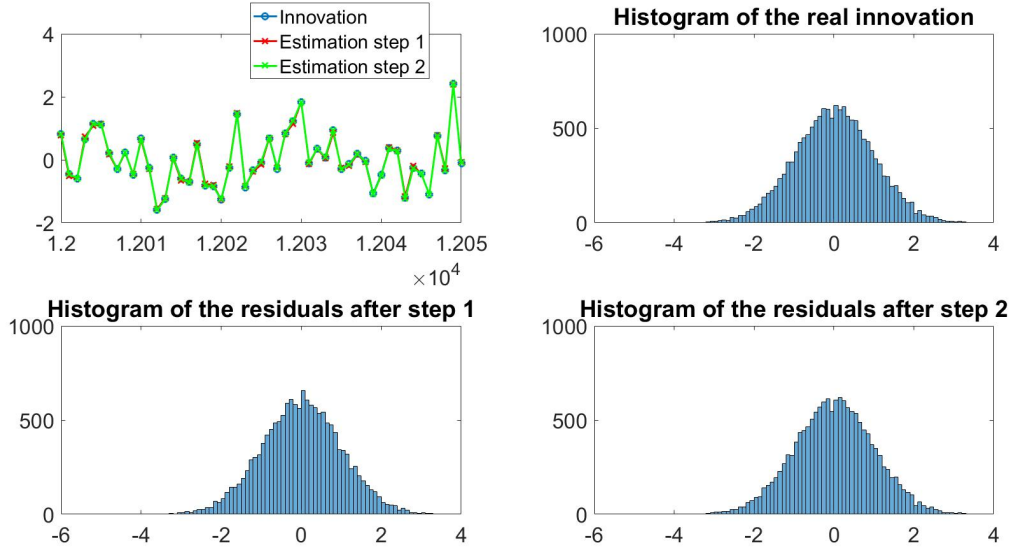
(a) Model with short impulse response



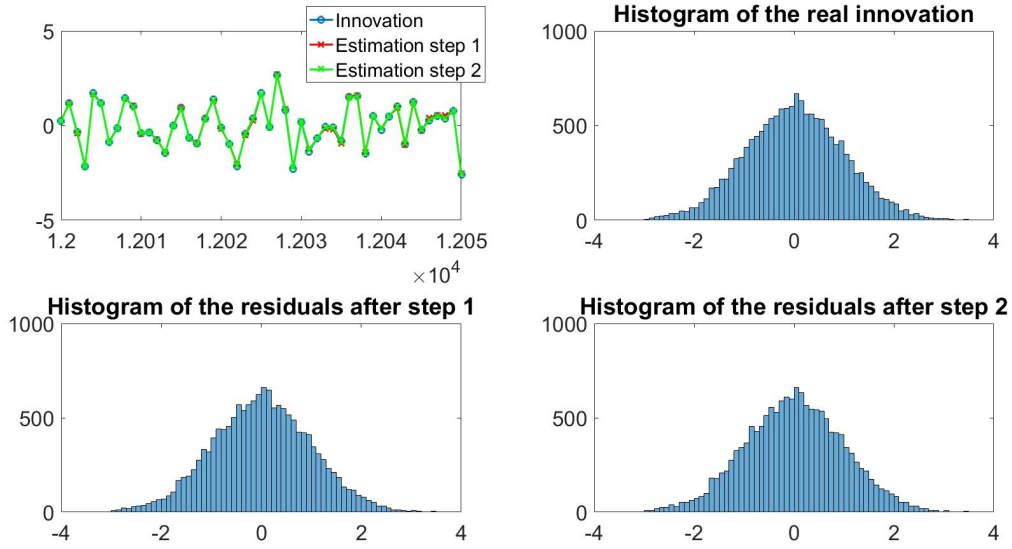
(b) Model with long impulse response

Figure 3.8: Scatter plot of the parameter estimates of two $ARMA(4, 4)$ coming from 50 Monte Carlo realizations. The model parameters are resumed in Table 3.1

Regarding the residuals (difference between the real measured output signal and the estimated one), a comparison between the real innovation and the residuals obtained after step one and step two is done at Figure 3.9. The estimation of the innovation seems quite accurate, for the first step as well as for the second step (both realizations and histograms of the residuals follow the same pattern as the ones of the innovation).



(a) Model with short impulse response



(b) Model with long impulse response

Figure 3.9: Plot of the residuals (upper left graph) and histogram of residuals obtained after step one and step two of the HR method compared with the real innovation of the signal coming from the two *ARMA* models of Table 3.1.

For VARMA models, the same results and discussions are obtained. We have the same problem with the length of impulse response for different models of the same order. An example of the impulse response estimation (in red) compared to the real impulse response (in blue) of an

ARMA(2,2) model of equation

$$\begin{aligned} \left[I_2 + \begin{pmatrix} -0.7 & 0.1 \\ -0.4 & -0.4 \end{pmatrix} B + \begin{pmatrix} 0.4 & 0.3 \\ -0.6 & 0.2 \end{pmatrix} B^2 \right] \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} \\ = \left[I_2 + \begin{pmatrix} 0.2 & 0.4 \\ 0.5 & 0.2 \end{pmatrix} B + \begin{pmatrix} -0.2 & 0.1 \\ -0.2 & -0.3 \end{pmatrix} B^2 \right] \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \end{aligned} \quad (3.12)$$

with two variables and size $N = 15000$ is shown at Figure 3.10. A Monte Carlo analysis of the same model is shown at Figure 3.11 as a plot of the residuals after the first and second step at Figure 3.12 ($p_{large} = \log(N)^{1.5}$). The results are similar to the ones obtained for the univariate models.

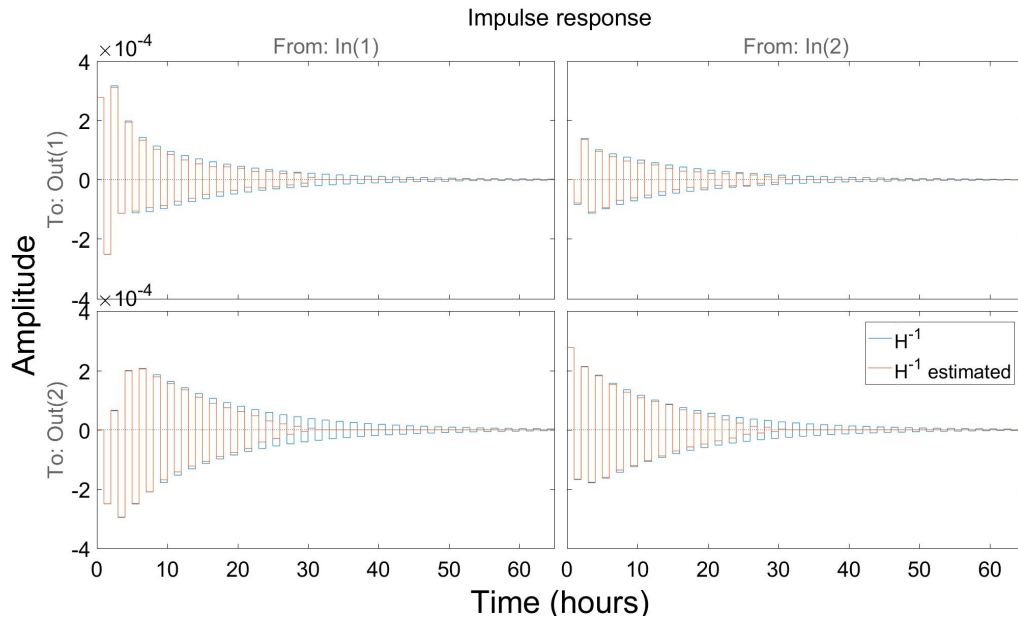


Figure 3.10: Impulse response of real transfer function H^{-1} and the estimated transfer function of the *VARMA*(2,2) model of eq 3.12 at step one of the HR estimation

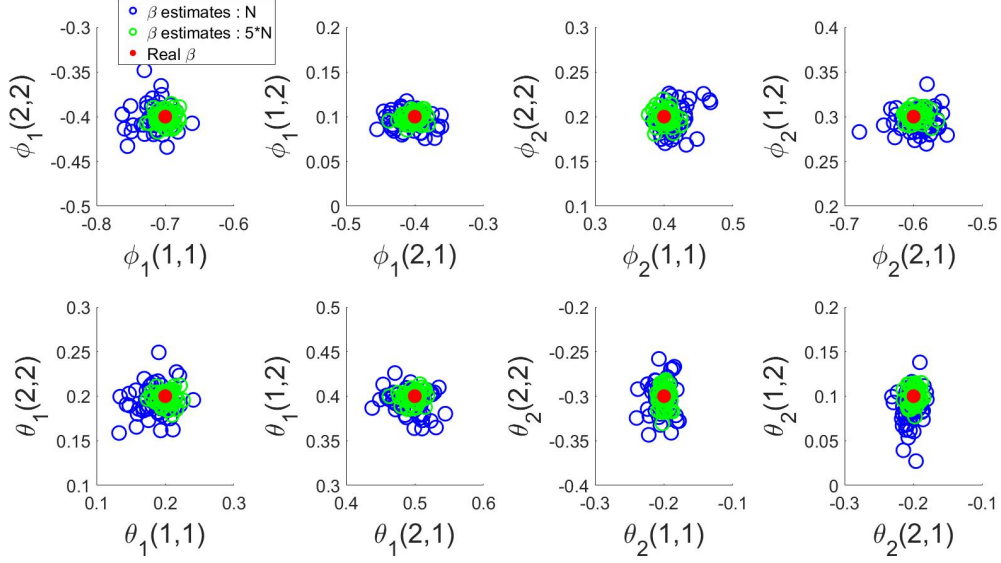


Figure 3.11: Scatter plot of the parameter estimates of a two variables $VARMA(2, 2)$ of eq 3.12 coming from 50 Monte Carlo realizations

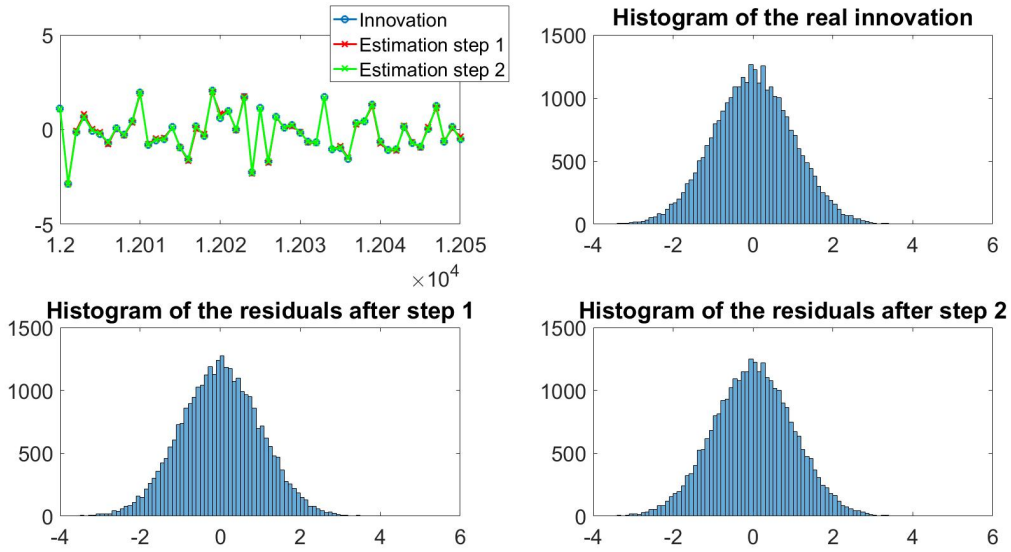


Figure 3.12: Plot of the residuals (upper left) and histogram of residuals obtained after step one and step two of the HR method compared with the real innovation of the signal coming from an $VARMA(2, 2)$ model of eq 3.12.

All these results indicate that the HR method gives good results for ARMA and VARMA models even if the inverse process H^{-1} has a different impulse response length. The internal parameter to estimate these models is

$$p_{large} = \lceil \log(N)^{1.5} \rceil$$

where $\lceil x \rceil$ is x rounded to the upper integer.

Single seasonal processes

In this case, the extended HR method described in Section 3.1 is used, and so there are two internal parameters at the first step : regular order p_{large} and seasonal order P_{large} .

The same problem as for the regular models is encountered. The impulse response of H^{-1} varies a lot as a function of the model parameters. Gómez [12] proposed to use a regular AR model of order $p_{large} = \log(N)^2$ but this gives not very good results in practice (especially for models with large seasons, as shown at Figure 3.20). After looking at the impulse response of H^{-1} and testing different values of p_{large} and P_{large} for models with different orders, it seems that $p_{large} = \log(N)^{1.4}$ and $P_{large} = P + Q$, where P and Q are the seasonal order of the $SARMA(p, q)(P, Q)_s$, give good results. These choice are motivated by the simulations that follows in this section.

So finally it is decided that :

$$\begin{aligned} p_{large} &= \lceil \min(\log(N)^{1.4}, \frac{s}{1.1}) \rceil \\ P_{large} &= P + Q \end{aligned} \quad (3.13)$$

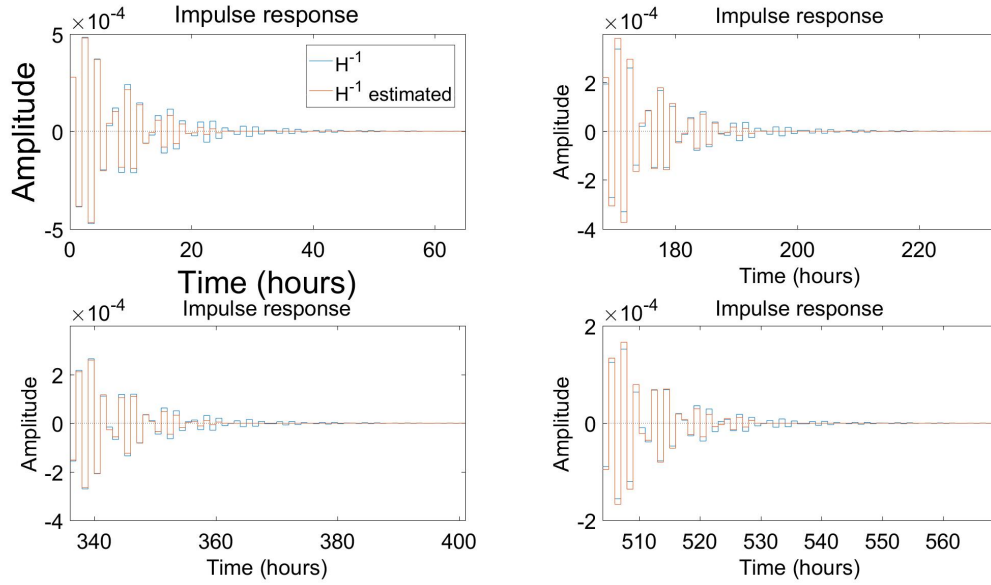
where s is the season of the model. The $\frac{s}{1.1}$ term comes from the fact that we do not want that the parameters of the non-seasonal part overlap the first parameters of the seasonal part. So to avoid this, a maximum limit is fixed arbitrarily for p_{large} . The value $\log(N)^{1.4}$ is slightly decreased compared to the ARMA case due to the fact that there are more parameters to estimate with the seasonal part.

At Figure 3.13 the impulse response of H^{-1} of the $SARMA(4, 4)(2, 2)_{168}$ model of equation

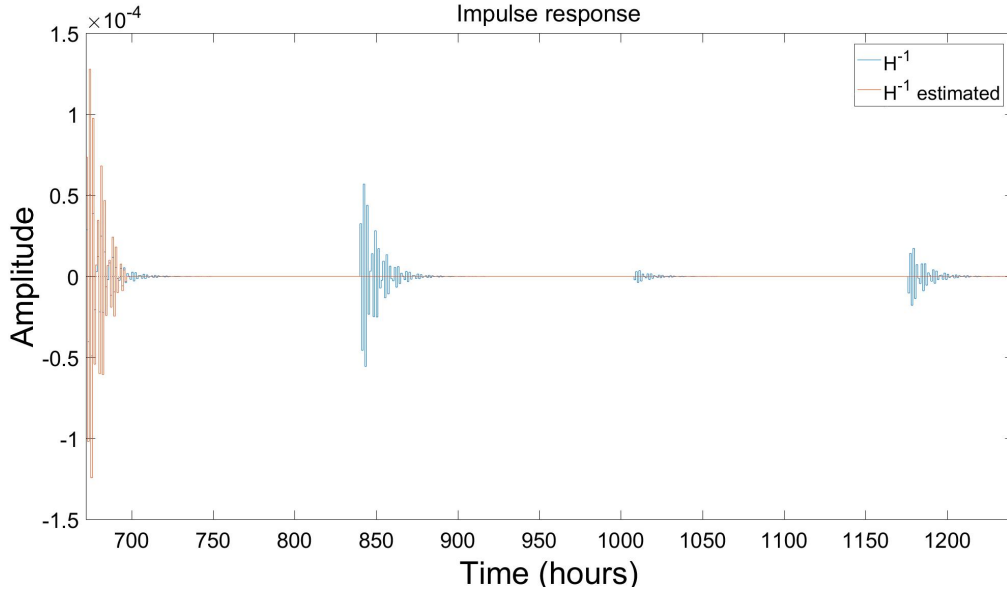
$$\begin{aligned} (1 - 0.8B + 0.5B^2 - 0.3B^3 + 0.2B^4)(1 - 0.2B^{168} + 0.3B^{336})y_t = \\ (1 + 0.6B - 0.4B^2 - 0.2B^3 + 0.3B^4)(1 + 0.5B^{168} - 0.4B^{336})a_t \end{aligned} \quad (3.14)$$

is shown and the estimation of H^{-1} obtained at the first step using the internal parameters described just before is depicted. We see that the estimated impulse response follows pretty well the real impulse response. For Figure 3.13b, the estimated impulse response becomes zero from approximately the lag 710, but the real impulse response of H^{-1} has not decreased to zero yet. It means that P_{large} could be possibly increased a little more, and the choice of this value is discussed further in this section with other simulations.

Concerning the parameter estimates, they can be observed at Figure 3.14 for the same model



(a) From time 1 to 569



(b) From time 672 to 1241

Figure 3.13: Impulse response of real transfer function H^{-1} and the estimated transfer function of a $SARMA(4,4)(2,2)_{168}$ model of eq 3.14 at step one in the HR estimation

used for Figure 3.13. In that case the parameter estimates are clearly biased, and it seems that the bias and covariance decrease slightly for increasing N .

To obtain a deeper intuition of the nature of the bias, we vary the values of p_{large} and P_{large} . At Figure 3.15 the estimates were computed using different P_{large} : $\frac{P+Q}{2}$, $P+Q$ and $4(P+Q)$. For $P_{large} = \frac{P+Q}{2}$ in green, the regular estimates are the least biased but have a bigger variance and the seasonal estimates have an important bias. When $P_{large} = 4(P+Q)P$ in magenta, seasonal estimates are least biased while the regular estimates move away from the real

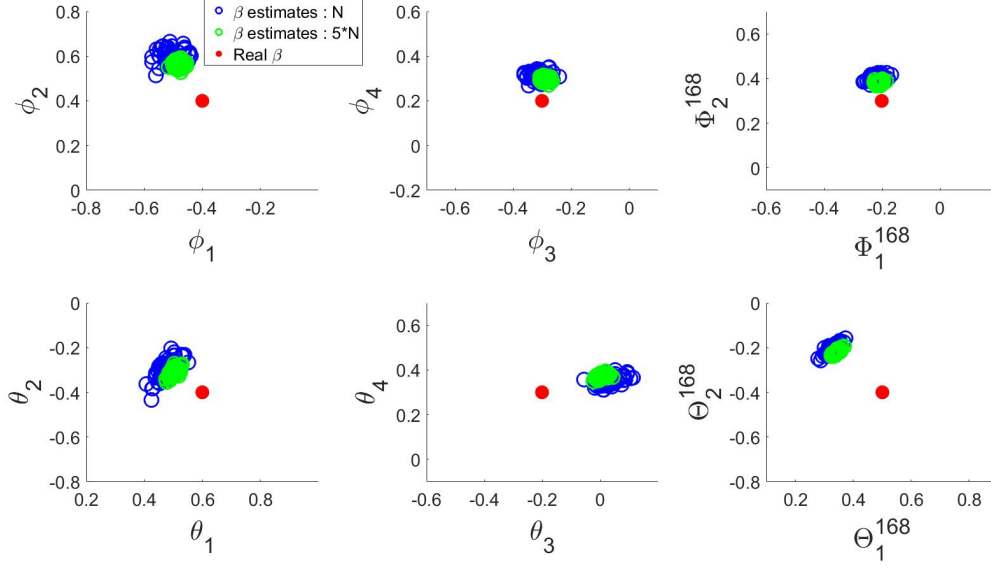


Figure 3.14: Scatter plot of the parameter estimates of $SARMA(4,4)(2,2)_{168}$ of eq 3.14 coming from 50 Monte Carlo realizations with different values of N

parameters. It seems then that taking $P_{large} = P + Q$ is a good compromise for estimating both the regular and the seasonal parameters. Concerning p_{large} , Figure 3.16 shows that increasing or decreasing its value does not influence fundamentally the variance of the estimates. However, if p_{large} becomes too low, the bias increases, so it is necessary to keep a value of p_{large} high enough.

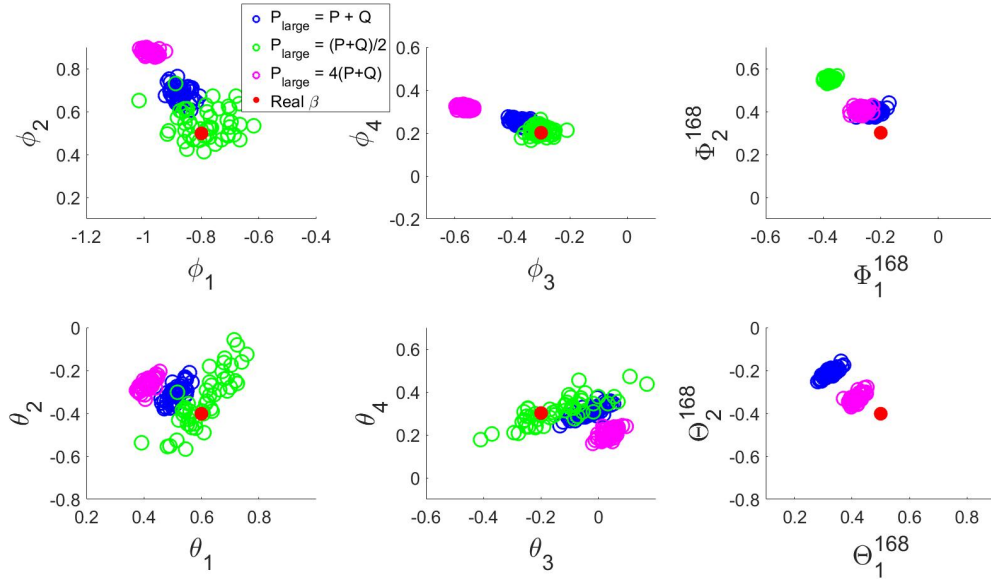


Figure 3.15: Scatter plot of the parameter estimates of $SARMA(4,4)(2,2)_{168}$ of eq 3.14 coming from 50 Monte Carlo realizations with $p_{large} = \log(N)^{1.4}$ and different values of P_{large}

Finally, if we look at the residuals obtained using the extended HR method with the chosen internal parameters of eq 3.13, it can be seen at Figure 3.17 (at the upper left) that the latter estimates pretty well the real innovation. But looking at the histograms, we observe that the distribution of the residuals after step 1 does not follow very well the distribution of the real

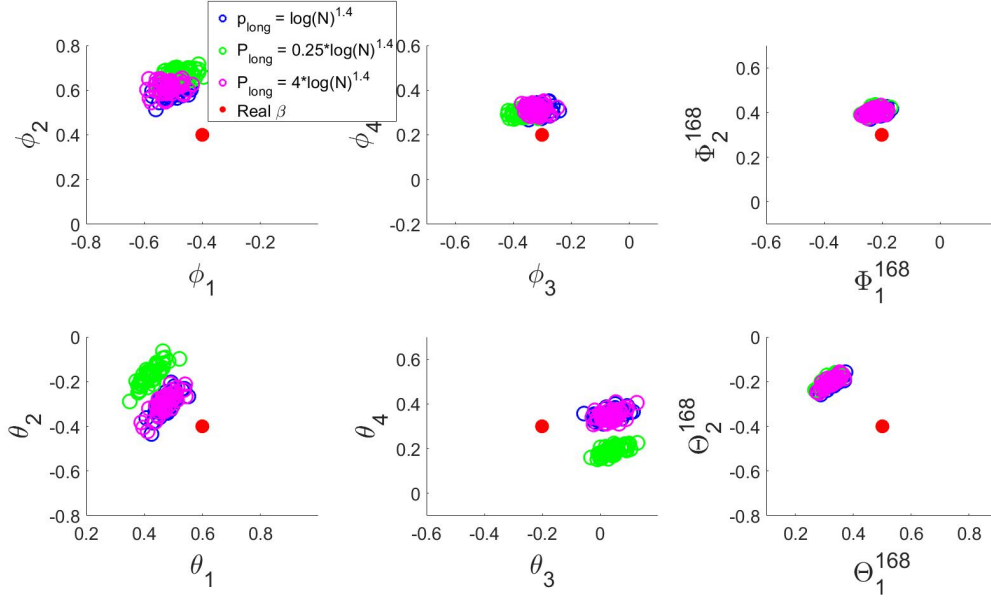


Figure 3.16: Scatter plot of the parameter estimates of $SARMA(4,4)(2,2)_{168}$ of eq 3.14 coming from 50 Monte Carlo realizations with $P_{large} = 2P$ and different values of p_{large}

innovation : too many residuals are equal to 0. Even if the distribution of the residuals after step 2 is better, there is still a peak at 0. This phenomenon shows that the final estimation is (strongly) dependent of the estimated model of step 1 (and particularly on the distribution of the residuals obtained at this step).

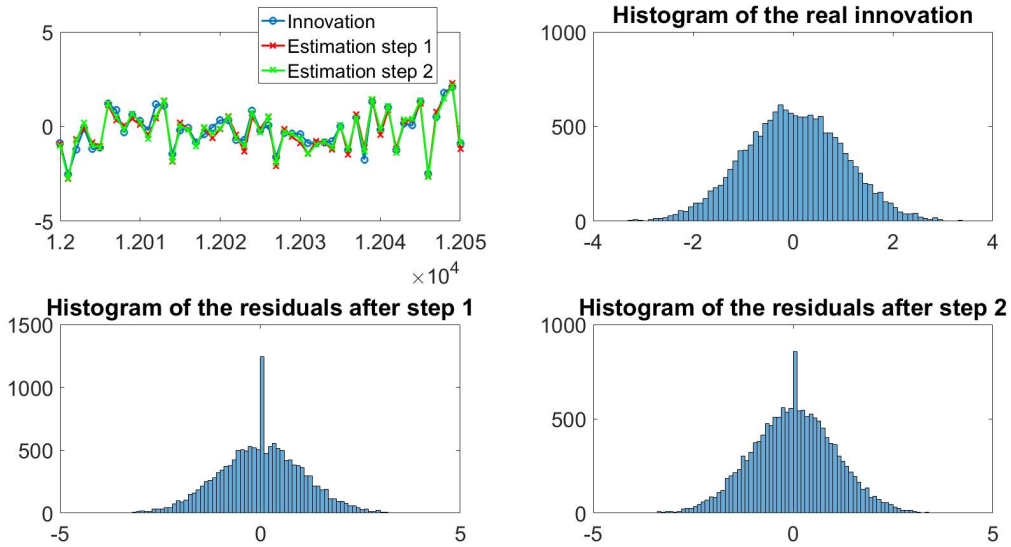


Figure 3.17: Plot of the residuals (upper left) and histogram of residuals obtained after step one and step two of the HR method compared with the real innovation of the signal coming from a $SARMA(4,4)(2,2)_{168}$ model of eq 3.14

Concerning VSARMA models, the same issues are encountered with the impulse response of H^{-1} . The same analysis as for the univariate case is done trying different internal parameters

with a $VSARMA(2, 2)(2, 2)_{168}$ models of equation

$$\begin{aligned}
& \left[I_2 + \begin{pmatrix} -0.4 & 0.1 \\ -0.4 & -0.4 \end{pmatrix} B + \begin{pmatrix} 0.4 & 0.3 \\ -0.1 & 0.2 \end{pmatrix} B^2 \right] \left[I_2 + \begin{pmatrix} -0.2 & -0.2 \\ -0.4 & 0.3 \end{pmatrix} B^{168} + \begin{pmatrix} 0.3 & -0.3 \\ 0.8 & -0.2 \end{pmatrix} B^{336} \right] \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} \\
&= \left[I_2 + \begin{pmatrix} 0.2 & 0.4 \\ 0 & 0.2 \end{pmatrix} B + \begin{pmatrix} -0.2 & 0.1 \\ -0.2 & -0.3 \end{pmatrix} B^2 \right] \left[I_2 + \begin{pmatrix} 0.1 & -0.1 \\ -0.3 & 0 \end{pmatrix} B^{168} + \begin{pmatrix} -0.5 & 0.2 \\ 0.5 & -0.25 \end{pmatrix} B^{336} \right] \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix}
\end{aligned} \tag{3.15}$$

with 2 variables. Once more, if p_{large} is large enough (and this is true for $\log(N)^{1.4}$), its value does not influence much the estimates (this is seen at Figure 3.18). Figure 3.19 represents parameter estimates using different values of P_{large} . This time, it is more difficult to say which P_{large} is the best. The parameter estimates with $P_{large} = 8(P + Q)$ in magenta seem to not be a good option because the parameter estimates obtained with $P_{large} = 2(P + Q)$ (in blue) and $P_{large} = P + Q$ (in green) are better in term of bias (for the most part of the model parameters). But it is the blue estimates that seem to have the lowest bias taking into account all the model parameters. The covariance seems to stay pretty like the same of each P_{large}

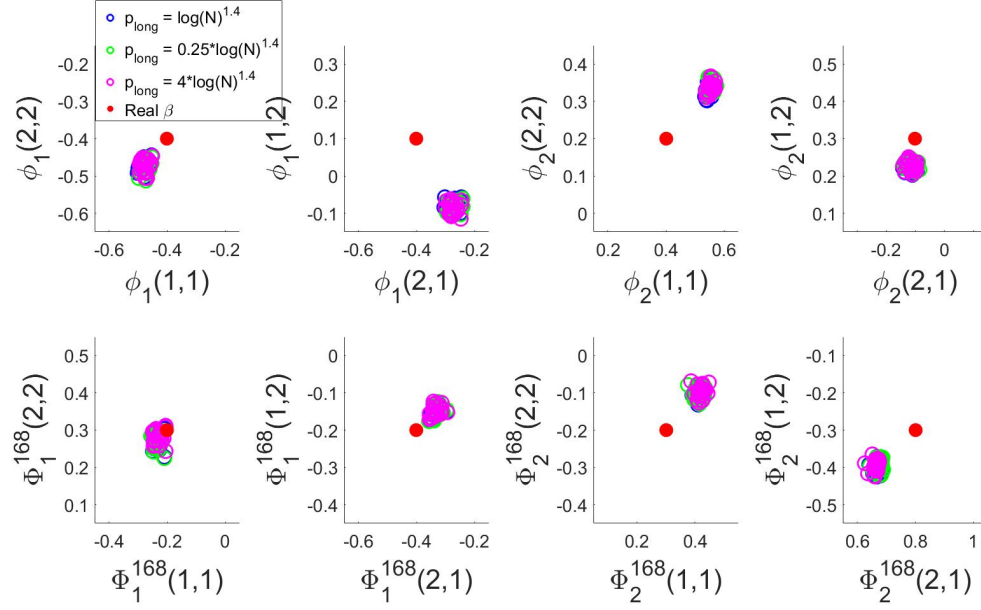
At the sight of the previous results, the chosen internal parameters (sufficiently large order of intermediate SAR) for the estimation of a SARMA model with K variables are

$$\begin{aligned}
p_{large} &= \lceil \min(\log(N)^{1.4}, \frac{s}{1.1}) \rceil \\
P_{large} &= K(P + Q)
\end{aligned}$$

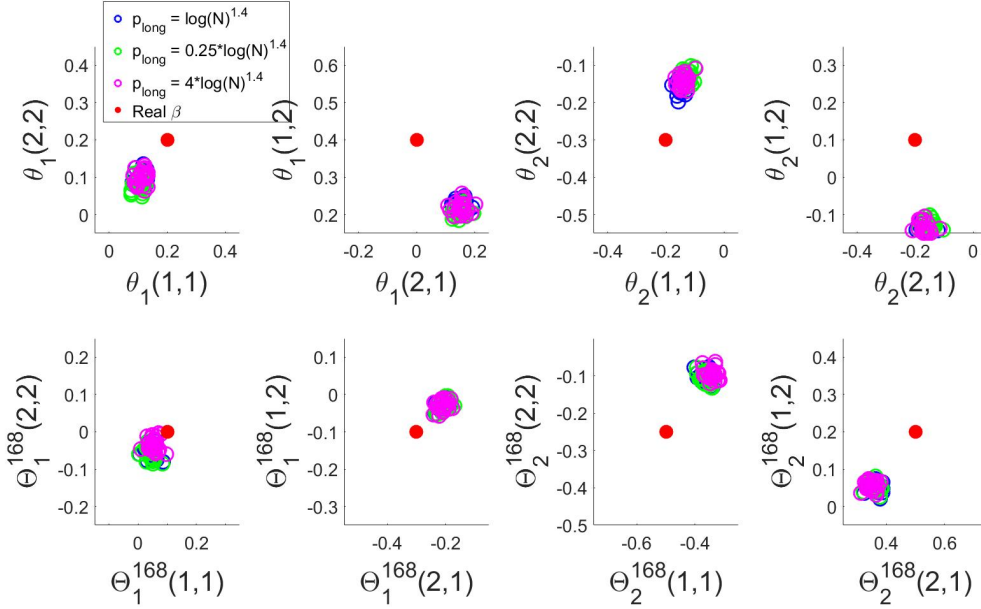
The value of p_{large} comes from the eq 3.13 when the value of P_{large} is derived from the best parameter estimates observe for the $SARMA$ and $VSARMA$ models studied here.

To compare the extended HR method with the classical one, Gómez et al. [13] let a free access to a set of matlab programs (called SSMMatlab) in which the classical HR method for regular and single seasonal models is implemented. For SARMA models, this implementation allows to compute only model with seasonal orders equal to 1, so little modifications in the code were done to estimate the $SARMA$ model of eq 3.14. Using these programs, Figure 3.20 shows the parameter estimates of the $SARMA$ model obtained using the extended HR method and the classical HR method. We see that the extended HR method gives parameter estimates which are really less biased (mostly for the seasonal parameters) and with smaller variance. Note that for the MA seasonal parameter, the classical HR method gives parameter estimates located near zero, due to the fact that the impulse response length of the AR at step 1 is not large enough.

Moreover, the computation time to estimate this model using a size data set of 15000 is, in



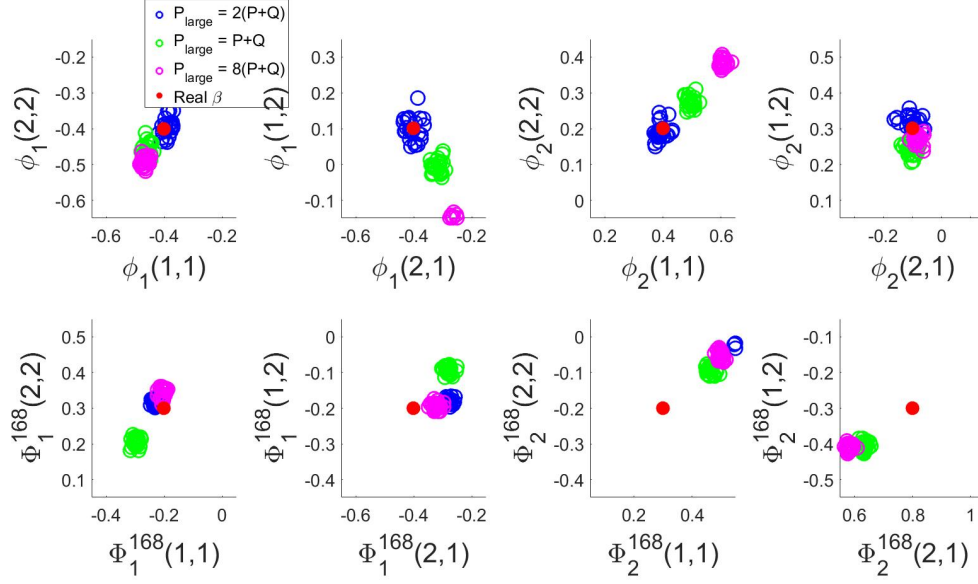
(a) Autoregressive parameters



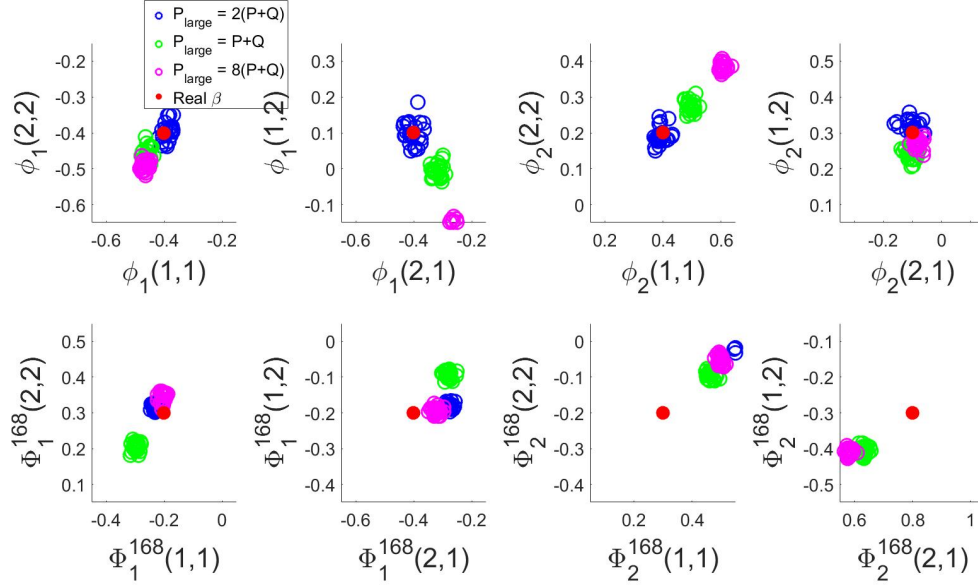
(b) Moving Average parameters

Figure 3.18: Scatter plot of the parameter estimates of $VSARMA(2, 2)(2, 2)_{168}$ of eq 3.15 coming from 25 Monte Carlo realizations with $P_{large} = 2(P + Q)$ and different values of p_{large}

average, more than 1500 times faster with the proposed implementation of the extended method than with SSMMatlab (this is not directly linked to the method, but more to the implementation of the programs itself). This is important for the model structure selection (presented in Section 3.3) which needs the estimation of a certain number of different models.



(a) Autoregressive parameters



(b) Moving Average parameters

Figure 3.19: Scatter plot of the parameter estimates of $VSARMA(2,2)(2,2)_{168}$ of eq 3.15 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.4}$ and different values of P_{large}

Processes with double seasonality

For the double seasonal models, we study the particular ones with first seasonality of 24 and second seasonality of 168 (because the real process studied later exhibits this seasonality behaviour). Once again, the impulse response of H^{-1} for different double seasonal models is inspected. It was seen during this inspection that the impulse response linked to the first seasonality (24) decreases to zero only if the model orders of the first seasonality are 1, i.e. $P_1 = 1$ and $Q_1 = 1$. This is shown at Figure 3.21 with the models that follow : the $2SARMA(2,2)(2,2)_{24}(2,2)_{168}$ model is described by the equation above, and $2SARMA(2,2)(1,1)_{24}(2,2)_{168}$ is the same model

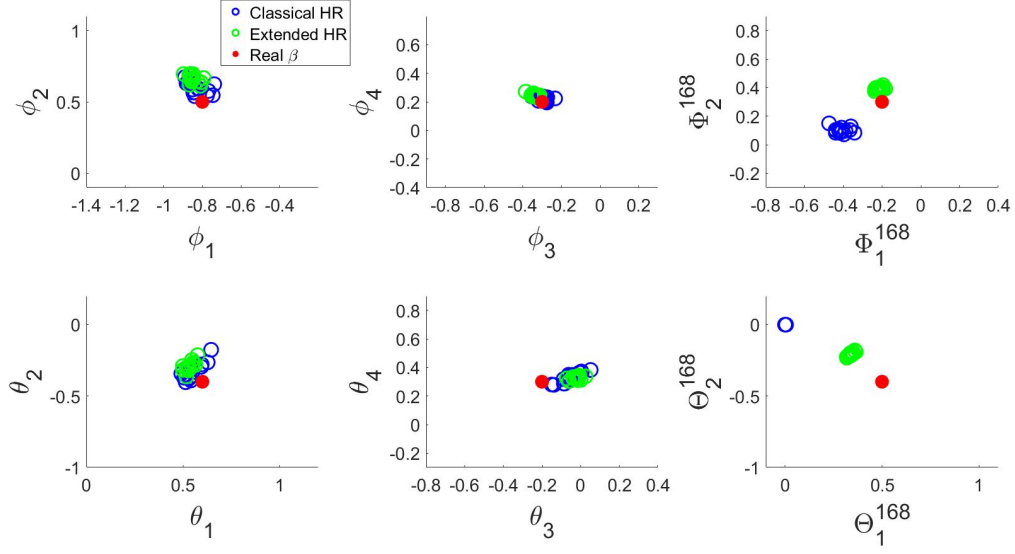
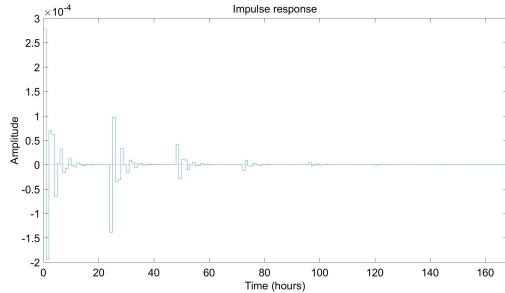


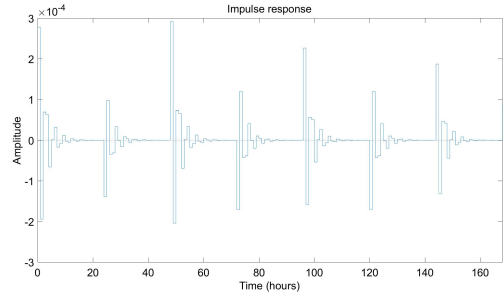
Figure 3.20: Scatter plot of the parameter estimates of $SARMA(4, 4)(2, 2)_{168}$ of eq 3.14 coming from 15 Monte Carlo realizations using the classical HR (in blue) and the extended HR (in green)

but with the last parameters linked to the first season removed.

$$(1 - 0.2B + 0.4B^2)(1 - 0.2B^{24} + 0.3B^{24})(1 + 0.5B^{168} + 0.7B^{336})y_t = (1 + 0.5B + 0.5B^2)(1 + 0.3B^{24} - 0.6B^{24})(1 - 0.4B^{168} - 0.4B^{336})a_t \quad (3.16)$$



(a) $2SARMA(2, 2)(1, 1)_{24}(2, 2)_{168}$



(b) $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$

Figure 3.21: Impulse response of H^{-1} for a $2SARMA(2, 2)(2, 2)_{24}(2, 2)_{168}$ models (right sub-figure) of eq 3.16 and for a $2SARMA(2, 2)(1, 1)_{24}(2, 2)_{168}$ of same equation with last parameters of the first season removed

So the value of the internal parameters concerning the first seasonality $P_{1_{large}}$ will be fixed at $3 \max(P_1, Q_1)$, with the constraint $P_{1_{large}} \leq \lfloor \frac{s_1}{s_2} - 1 \rfloor$ to avoid overlapping (in our case $P_{1_{large}} \leq 6$).

As for the previous models, it is difficult to deduce the sufficiently large regular order (p_{large}) regarding at the order of the $2SARMA$ model. The chosen value $p_{large} = \log(N)^{1.3}$ follows the same idea as for the single seasonal models, where we decrease a little bit p_{large} due to

the increasing number of parameters to estimate. The same constraint to avoid overlapping is also used : $p_{large} \leq \frac{s_1}{1.1}$. The double seasonal order P_{2large} is first fixed at $P_2 + Q_2$ like for the SARMA models. A resulting estimation of the impulse response of the estimated H^{-1} for the $2SARMA(2,2)(2,2)_{24}(2,2)_{168}$ of eq 3.16 is shown at Figure 3.22. As for the single seasonal models, we see that the impulse response of the estimated H^{-1} at Figure 3.22b attains zero before the impulse response of H^{-1} . But increasing too much P_{2large} results in a worse impulse response for the estimated H^{-1} . So Monte Carlo simulations are done to find the best values of the orders for the internal 2SAR model.

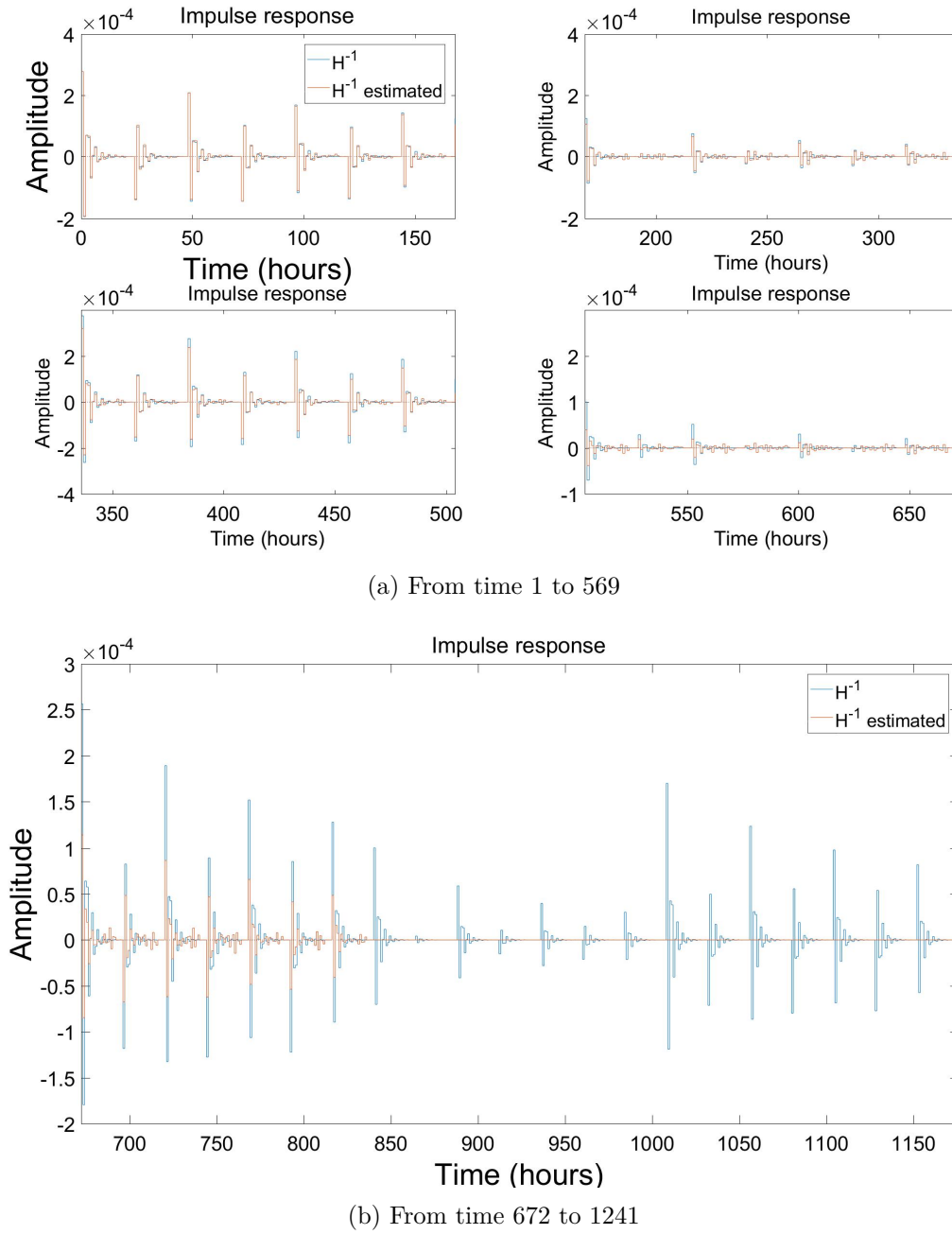


Figure 3.22: Impulse response of real transfer function H^{-1} and the estimated transfer function of the $2SARMA(2,2)(2,2)_{24}(2,2)_{168}$ model of eq 3.16 at the first step of the extended HR estimation

Figures 3.23 and 3.24 show the estimates using different values of p_{large} and P_{2large} . Looking at these plots, it seems that the values previously chosen (results in blue) are the best compromise for the estimation of regular and seasonal parameters. The previous discussion with Figure 3.21 seems sufficient to keep $P_{1large} = 3$ when P_1 and Q_1 are lower or equal to 1 and $P_{1large} = 6$ otherwise.

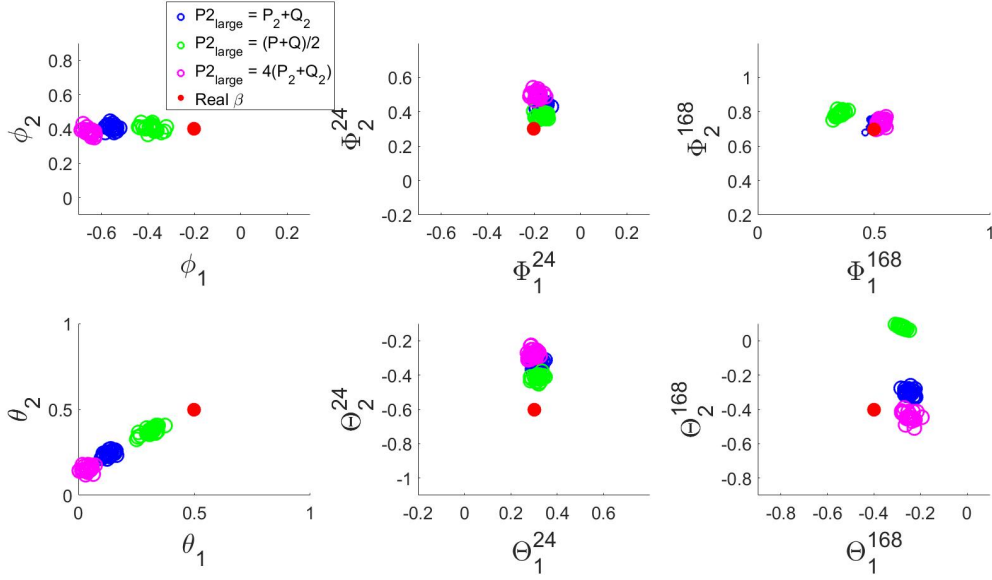


Figure 3.23: Scatter plot of the parameter estimates of $2SARMA(2,2)(2,2)_{24}(2,2)_{168}$ of eq 3.16 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.3}$ and different values of P_{2large}

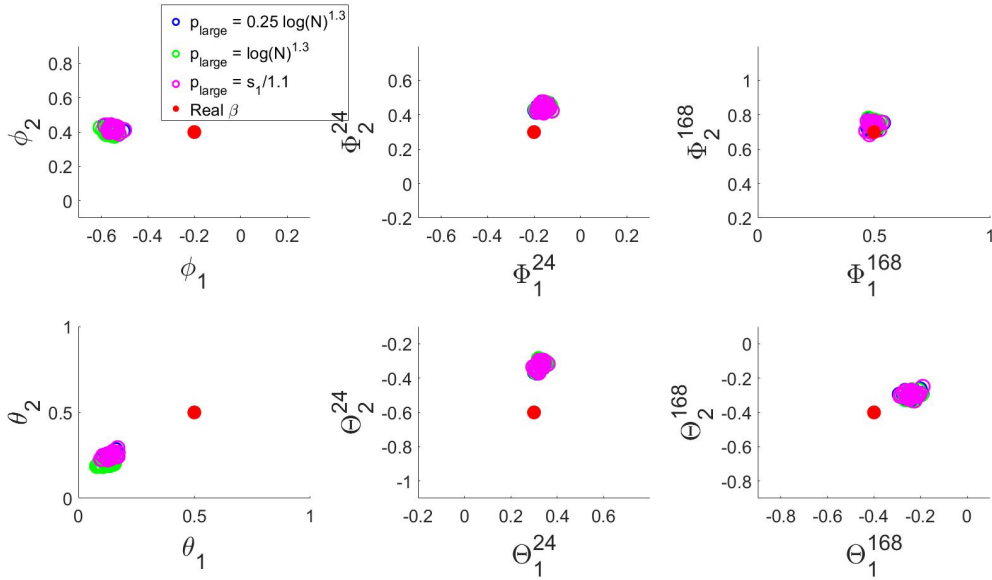


Figure 3.24: Scatter plot of the parameter estimates of $2SARMA(2,2)(2,2)_{24}(2,2)_{168}$ of eq 3.16 coming from 25 Monte Carlo realizations with $P_{2large} = 2P$ and different values of p_{large}

Using these internal parameters (see eq 3.18) to compute parameter estimates, we obtain the Monte Carlo plots of Figure 3.25, using once a signal of length $N = 10000$ in blue and then of length $N = 50000$ in green. We see that the covariance of the parameter estimates decreases slightly with N increases, but the bias seems to stay constant.

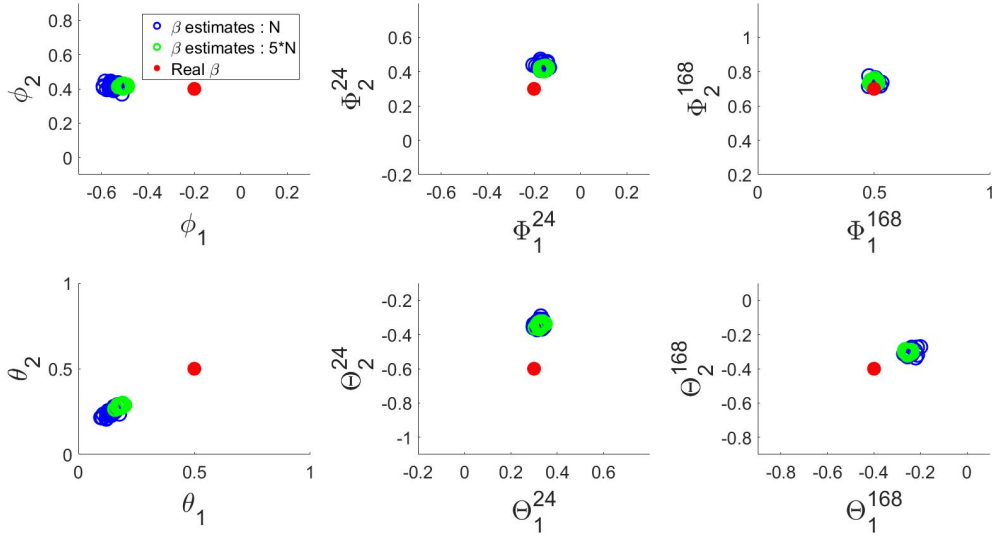
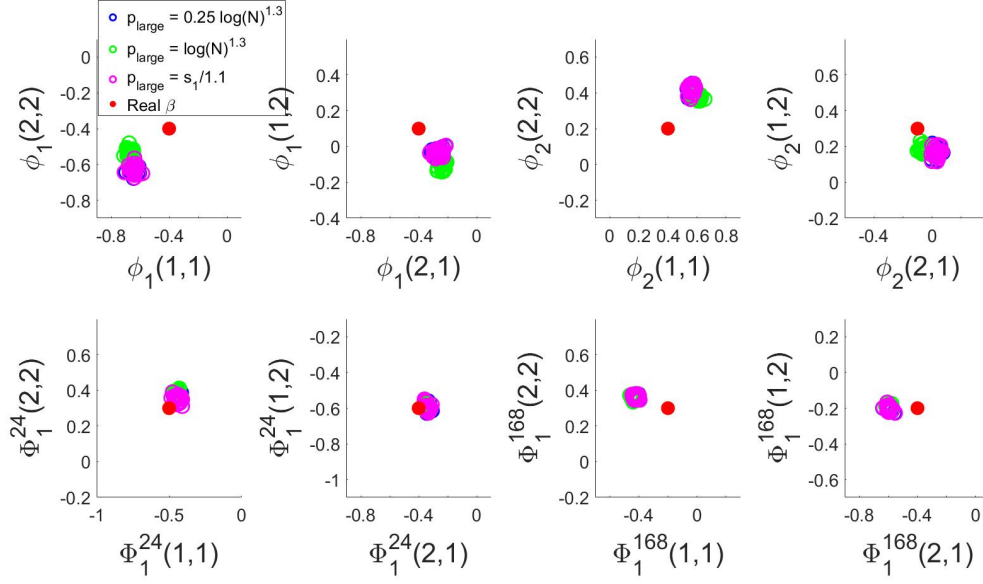


Figure 3.25: Scatter plot of the parameter estimates of a $2SARMA(2,2)(2,2)_{24}(2,2)_{168}$ of eq 3.16 coming from 25 Monte Carlo realizations

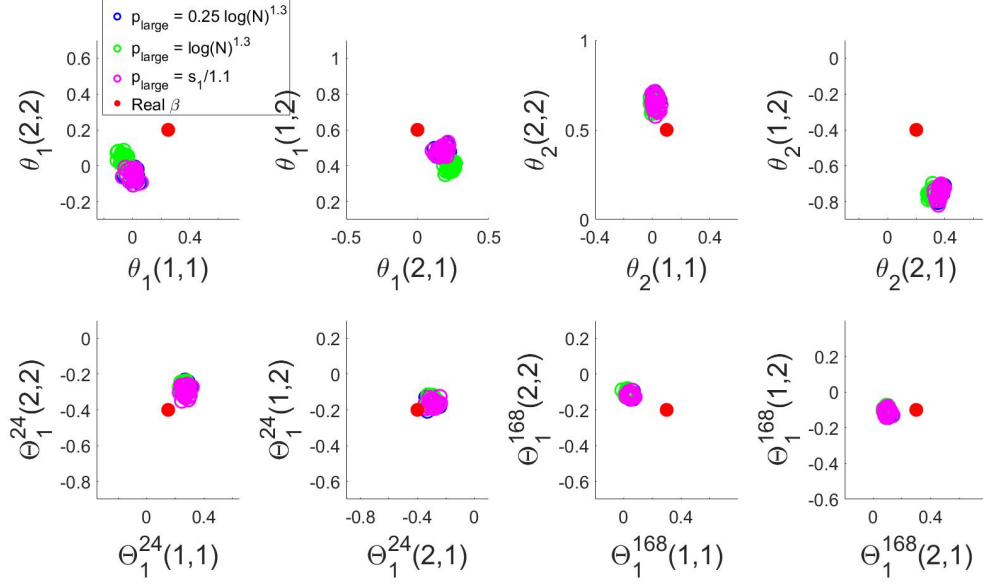
Concerning the multivariate double seasonal models, the same results are obtain as for the univariate case concerning the length of impulse response linked to the first seasonal orders. So the orders of the first seasonality of $V2SARMA$ models are fixed to $3 \max(P_1, Q_1)$ with the constraints that $P_{1large} \leq \lfloor \frac{s_1}{s_2} - 1 \rfloor$. The other parameters are chosen in the same way as for the $2SARMA$ models i.e. $p_{large} = \log(N)^{1.3}$ with $p_{large} \leq \frac{s_1}{1.1}$ and $P_{2large} = P_2 + Q_2$. For a $V2SARMA(2,2)(1,1)_{24}(1,1)_{168}$ with 2 variables of equation

$$\begin{aligned} & \left[I_2 + \begin{pmatrix} -0.4 & 0.1 \\ -0.4 & -0.4 \end{pmatrix} B + \begin{pmatrix} 0.4 & 0.3 \\ -0.1 & 0.2 \end{pmatrix} B^2 \right] \left[I_2 + \begin{pmatrix} -0.5 & -0.6 \\ -0.4 & 0.3 \end{pmatrix} B^{24} \right] \\ & \left[I_2 + \begin{pmatrix} -0.2 & -0.2 \\ -0.4 & 0.3 \end{pmatrix} B^{168} \right] \begin{pmatrix} y_{t,1} \\ y_{t,2} \end{pmatrix} = \left[I_2 + \begin{pmatrix} 0.25 & 0.6 \\ 0 & 0.2 \end{pmatrix} B + \begin{pmatrix} 0.1 & -0.4 \\ 0.2 & 0.5 \end{pmatrix} B^2 \right] \\ & \left[I_2 + \begin{pmatrix} -0.15 & 0.2 \\ 0.4 & 0.4 \end{pmatrix} B^{24} \right] \left[I_2 + \begin{pmatrix} 0.3 & -0.1 \\ 0.3 & -0.2 \end{pmatrix} B^{168} \right] \begin{pmatrix} a_{t,1} \\ a_{t,2} \end{pmatrix} \quad (3.17) \end{aligned}$$

the Monte Carlo method was used to compare the estimates of the same model using different internal parameters. This is shown at Figures 3.26, 3.27 and 3.28. The difference with single seasonal models is the fact that the best internal parameters are the same for the univariate and multivariate models (blue estimates). It can be seen that the parameter estimates obtained using $P_{1large} = 1$ are sometimes better, sometimes really worse (for example with Θ_1^{24}). So this value is not chosen for the final estimation method.

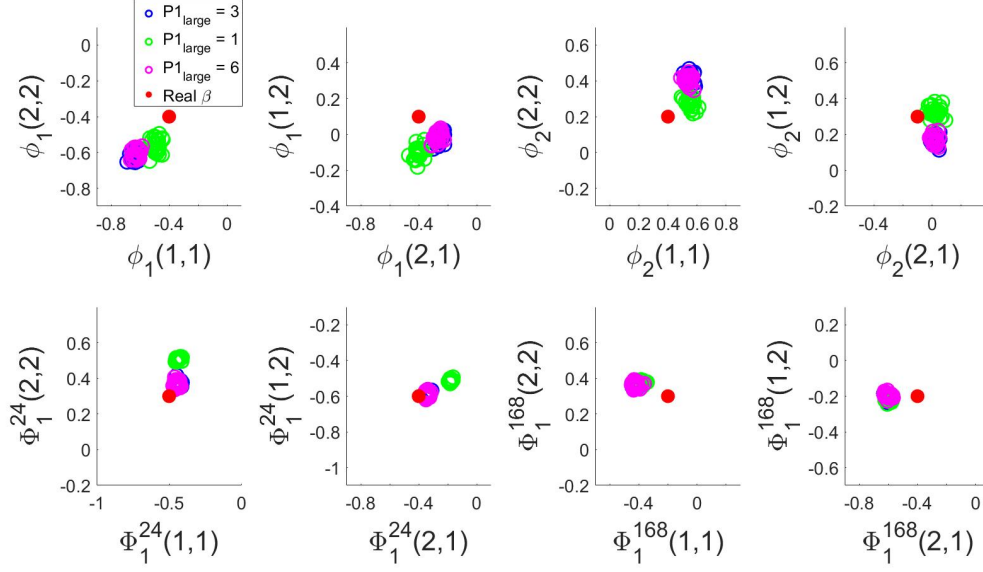


(a) Autoregressive parameters

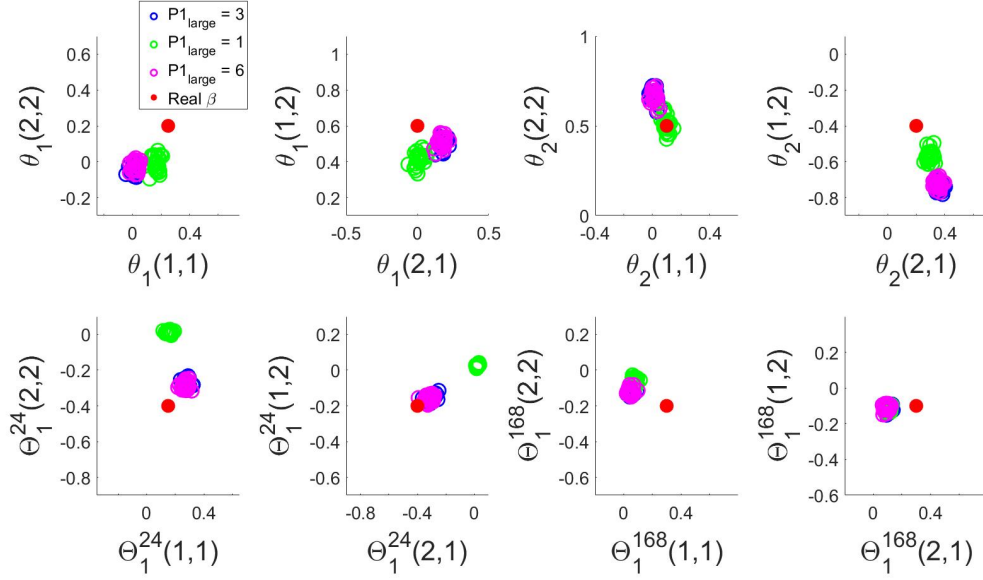


(b) Moving Average parameters

Figure 3.26: Scatter plot of the parameter estimates of 2-variables $V2SARMA(2,2)(1,1)_{24}(1,1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations with $P_{large} = 3$ and $P_{large} = 2$

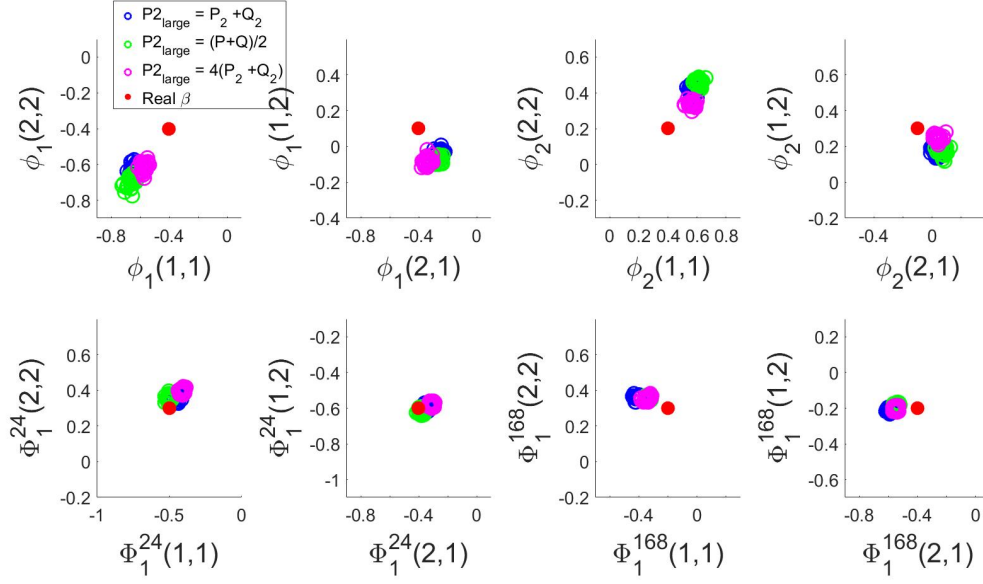


(a) Autoregressive parameters

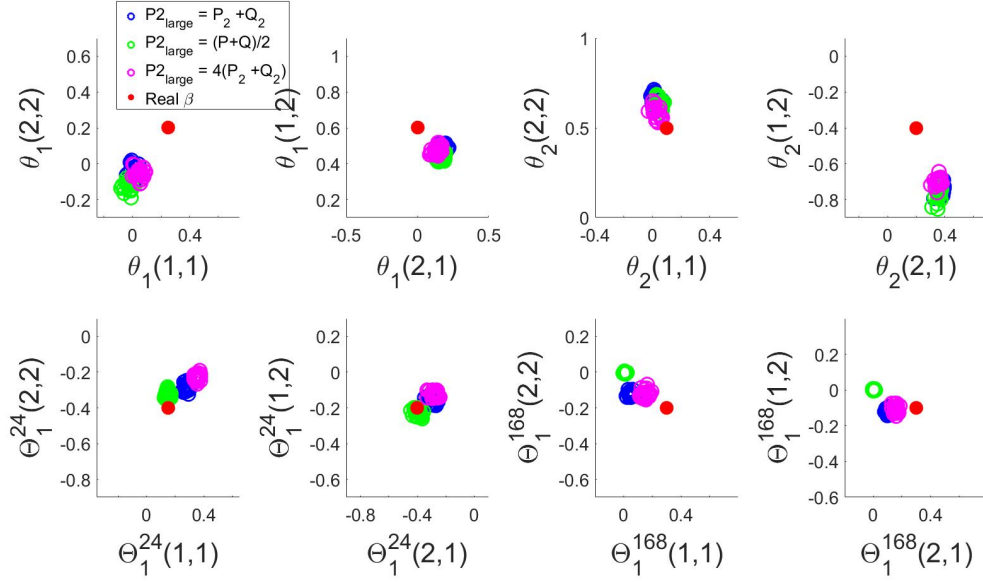


(b) Moving Average parameters

Figure 3.27: Scatter plot of the parameter estimates of the 2-variables $V2SARMA(2,2)(1,1)_{24}(1,1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.3}$ and $P_{2_{large}} = 2$



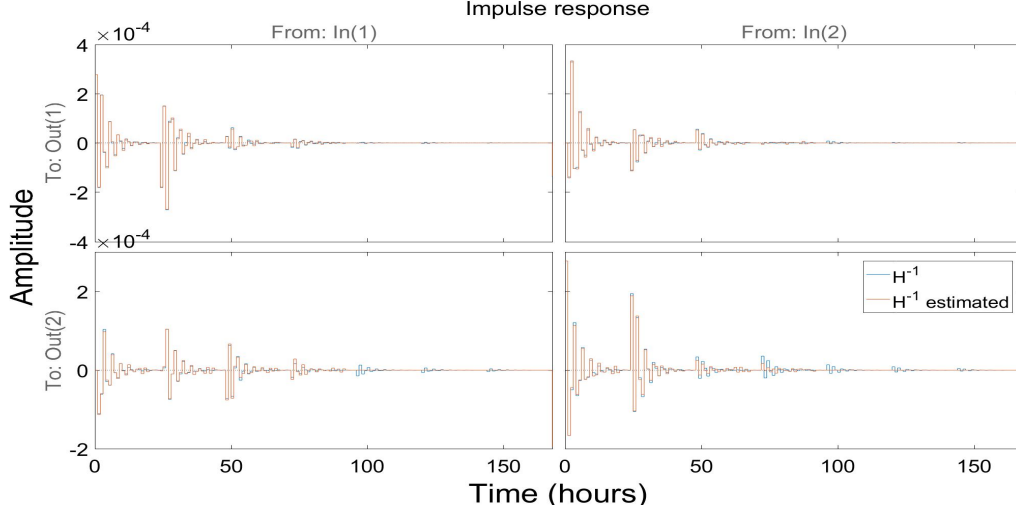
(a) Autoregressive parameters



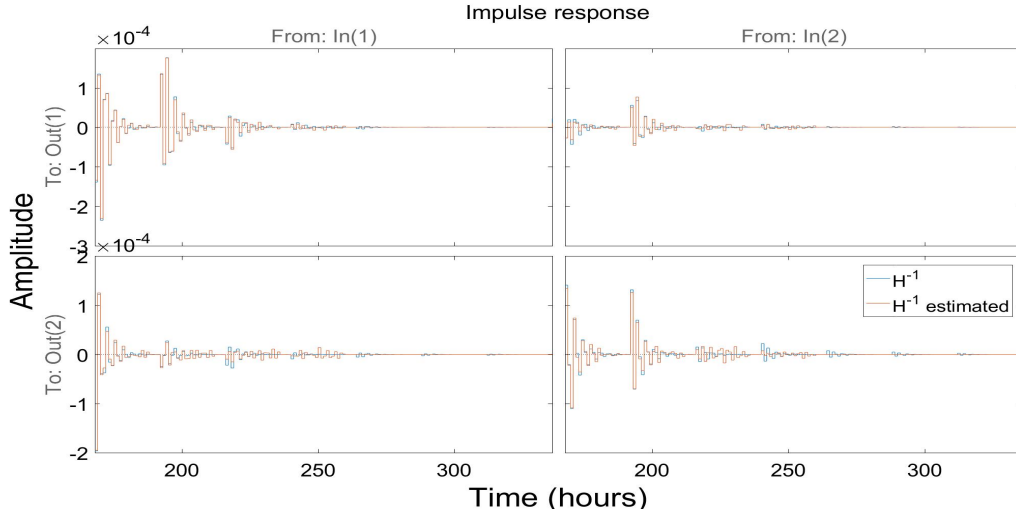
(b) Moving Average parameters

Figure 3.28: Scatter plot of the parameter estimates of the 2-variables $V2SARMA(2,2)(1,1)_{24}(1,1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations with $p_{large} = \log(N)^{1.3}$ and $P1_{large} = 3$

Using the same model, the resulting impulse response of the estimated H^{-1} of step 1 is shown at Figure 3.29. Parameter estimates (using the internal parameters of eq 3.18) obtained with signals of different length are shown on Figure 3.31.



(a) From time 1 to 168



(b) From time 168 to 336

Figure 3.29: Impulse response of real transfer function H^{-1} and the estimated transfer function of the $V2SARMA(2, 2)(1, 1)_{24}(1, 1)_{168}$ model of 2 variables of eq 3.17 at the first step of the extended HR estimation

Finally, the internal parameters for the estimation of uni- and multi-variate $2SARMA$ models are

$$\begin{aligned}
 p_{large} &= \lceil \min(\log(N)^{1.3}, \frac{s_1}{1.1}) \rceil \\
 P_{1_{large}} &= 3 \max(P_1, Q_1) \\
 P_{2_{large}} &= P_2 + Q_2
 \end{aligned} \tag{3.18}$$

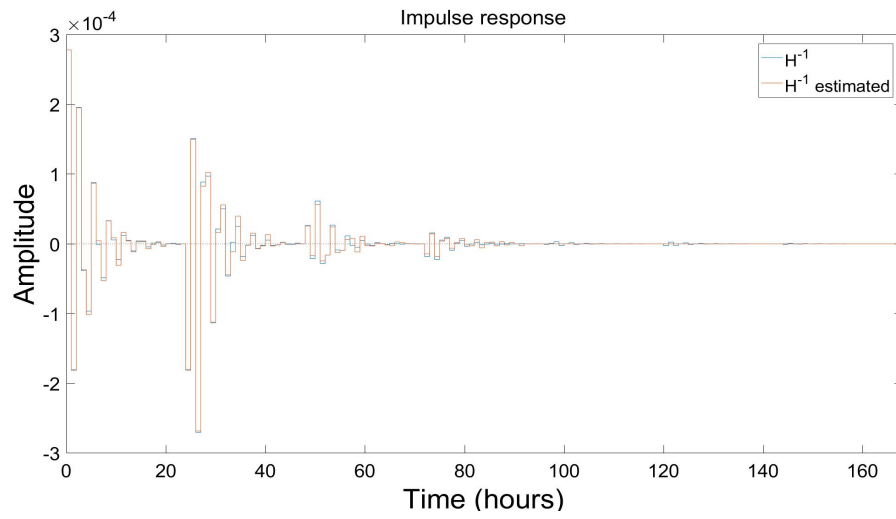
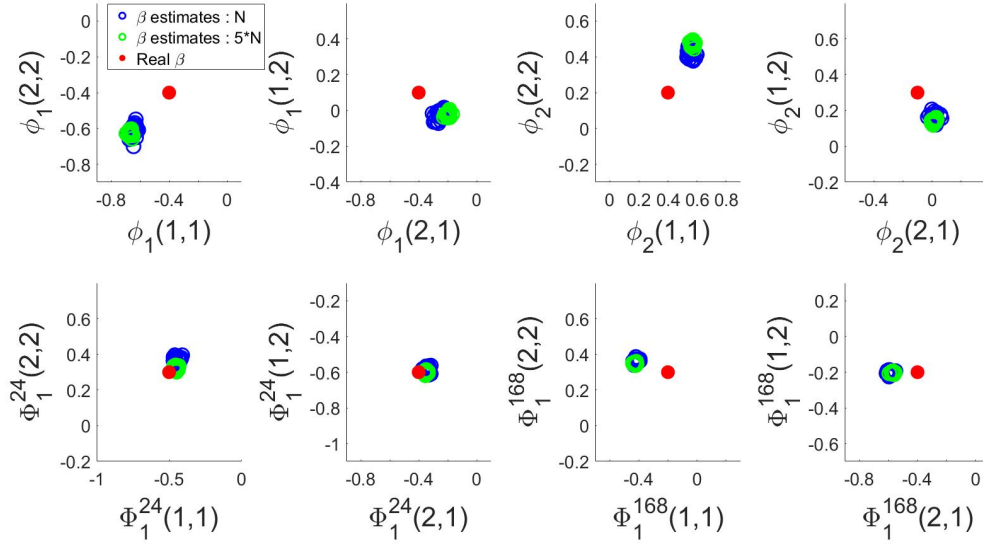
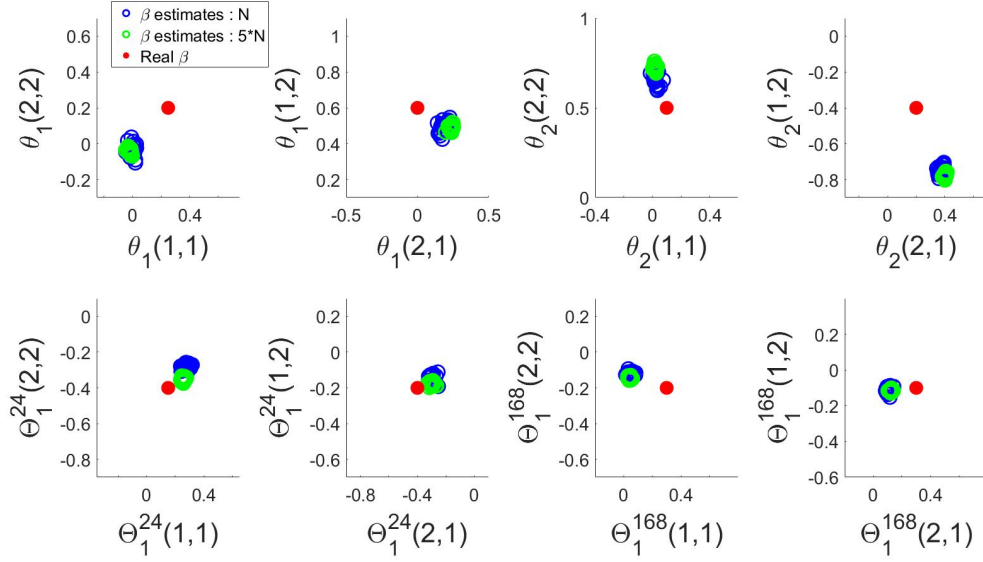


Figure 3.30: Zoom of the upper left graph of Figure 3.29a



(a) Autoregressive parameters



(b) Moving Average parameters

Figure 3.31: Scatter plot of the parameter estimates of a 2-variables $V2SARMA(2,2)(1,1)_{24}(1,1)_{168}$ model of eq 3.17 coming from 25 Monte Carlo realizations

3.3 Model structure selection

There exist different ways to select the model structure in case of time-series estimation. One possibility is to scan different candidate orders, compute parameter estimates for each model and compare the cost function obtained using these estimates.

Recall that for a measured output signal y_t for $t = 1, \dots, N$, the cost function is

$$\frac{1}{N} \sum_{t=1}^N \|y_t - \hat{y}_{t|t-1}\|_2^2 \quad (3.19)$$

$$\frac{1}{N} \sum_{t=1}^N \|H^{-1}y_t\|_2^2 \quad (3.20)$$

where $H^{-1} = \theta(B)^{-1}\phi(B)$ for an *ARMA* or a *VARMA*, $H^{-1} = \Theta(B^s)^{-1}\theta(B)^{-1}\phi(B)\Phi(B^s)$ for a *SARMA* or *VSARMA* and $H^{-1} = \Theta_{s_2}(B^{s_2})^{-1}\Theta_{s_1}(B^{s_1})^{-1}\theta(B)^{-1}\phi(B)\Phi_{s_1}(B^{s_1})\Phi_{s_2}(B^{s_2})$ a *2SARMA* or *V2SARMA*. The residuals are computed as $a_t = y_t - \hat{y}_{t|t-1}$.

But if the cost function is studied on the same data set as the one used to compute the parameter estimates, there is a risk of overfitting. The overfitting phenomenon appears when the model is too complex compared to the reality and the signal is more and more modelled instead of the underlying behaviour. In fact the model predicts "too well" the measured output, modeling both the process and the innovation which is independent of the process output. This results in very good performance on the data set used to estimate the model parameters, but poor results on new data.

Driven by that, for the selection of the model structure, two different sets are used. A training set which is used to train the data and to compute the parameter estimates, and a validation set containing data "never seen before" by the model. These unseen data are used to compute the cost function as estimations of what we should get with new data measured by the same process.

So it is expected that the cost function decreases on the training set when the complexity of the model increases. While on validation set a minimum should appear for the orders that correspond to the real system under estimations (or close to the real system assuming no model mismatch).

Considering that we have a data set of length N , a training and a validation set need to be selected. Classical proportions used for the training/validation sets are 80%/20% or 90%/10%. The idea here is to ensure that we have enough data on the training set to capture the behaviour of the real system and enough data on the validation set to verify the model performance. Recalling that for the real applications of the next chapter, approximately 15000 consumption energy samples are available, the simulations are based on this size of data. If 10% of the time

series is used to validate the model, it means that it will use 1500 hours ≈ 9 weeks ≈ 2 months. This seems enough if the weekly seasonality of the real system is not too high (below 6 or 7). This is also interesting because the most part of the data will be used to estimate the real system, and we have seen previously in this chapter that the accuracy of the parameter estimates increases with the number of data used for the estimation.

For the model structure selection, the internal parameters used in the extended HR method are fixed, such as they are independent of the orders of the model estimated. In this way it will not influence the model selection when a system with unexpected behaviour (with an impulse response of H^{-1} shorter or longer than expected) is modelled. Once the model structure is selected, the final parameter estimates are computed using the internal parameters decided previously in order to improve the performance of the estimation. These internal parameters are recalled in the following equations:

For *ARMA* and *VARMA* models :

$$p_{large} = \log(N)^{1.5} \quad (3.21)$$

For *SARMA* and *VSARMA* of K variables:

$$\begin{aligned} p_{large} &= \lceil \min(\log(N)^{1.4}, \frac{s}{1.1}) \rceil \\ P_{large} &= K(P + Q) \end{aligned} \quad (3.22)$$

For *2SARMA* and *V2SARMA*:

$$\begin{aligned} p_{large} &= \lceil \min(\log(N)^{1.3}, \frac{s_1}{1.1}) \rceil \\ P_{1_{large}} &= 3 \max(P_1, Q_1) \\ P_{2_{large}} &= P_2 + Q_2 \end{aligned} \quad (3.23)$$

For the model structure selection, the internal parameters are fixed. For regular models the value of p_{large} is given by equation 3.21, for single seasonal models they are given by equations 3.22 with $P = Q = 2$ and for double seasonal it is given by equations 3.23 with $P_1 = Q_1 = 2$ and $P_2 = Q_2 = 4$. These values seem to represent well the behaviour of general processes (too high internal parameters give poor estimates).

The real data will be estimated using directly mixed model, so the model structure selection is not done with autoregressive models (too simple model for the practical case).

To sum up, the model structure selection procedure consists of the following steps :

1. The internal parameters of the step 1 of the extended HR method are fixed at the values

given just before.

2. The model structure selection criterion is chosen. Here, the criterion is the cost function computed on the validation set, described as follows :

$$V_{val} = \frac{1}{N_{val}} \sum_{t=N-N_{val}+1}^N \|H^{-1}y_t\|_2^2$$

where N_{val} is the size of the validation set and H depends on the model structure (see Chapter 2).

3. A scanning through 2 (or 2 pairs of) model orders is done by varying their values while keeping the other orders fixed. The V_{val} is plotted and the minimum point, or the point where V_{val} stop to decrease (and is followed by a plate), is chosen as the one used for the model structure.
4. The previous step is repeated for different (pair of) model orders until all model orders are chosen.

Note that in the simulations that follow, the orders kept constant during the scanning are chosen as the ones corresponding to the real process.

3.3.1 Regular models

The scanning order is applied to an $ARMA(4, 4)$ model. The scanning goes from $p = 1, \dots, 10$ and $q = 1, \dots, 10$. The values of V_{val} are shown on Figure 3.32. The intersection of the 2 red lines shows the result for the estimation using the true model orders (the real orders of the estimated process). In this case, this intersection is the minimum point. During model estimation using larger orders than the real ones, there is a flat region which appears. This can be explained by the fact that more complex structures for this model give the same results as the exact order model.

The same scanning order is represented for a 2 variables $VARMA(3, 3)$ model. This is shown at Figure 3.33 and the same phenomena as for the univariate case are observed.

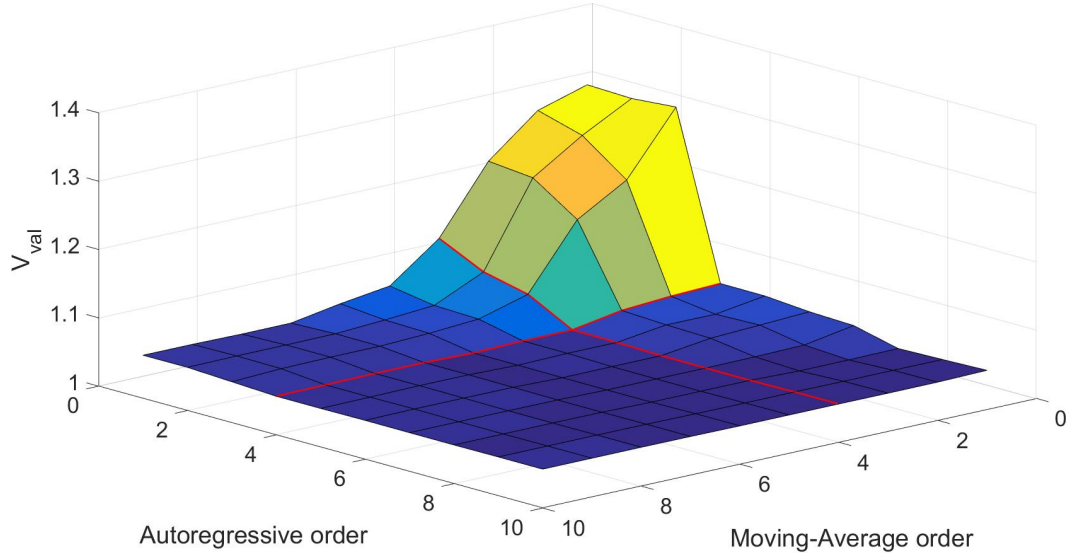


Figure 3.32: Plot of V_{val} obtained with different model orders used for the estimations of an $ARMA(4, 4)$

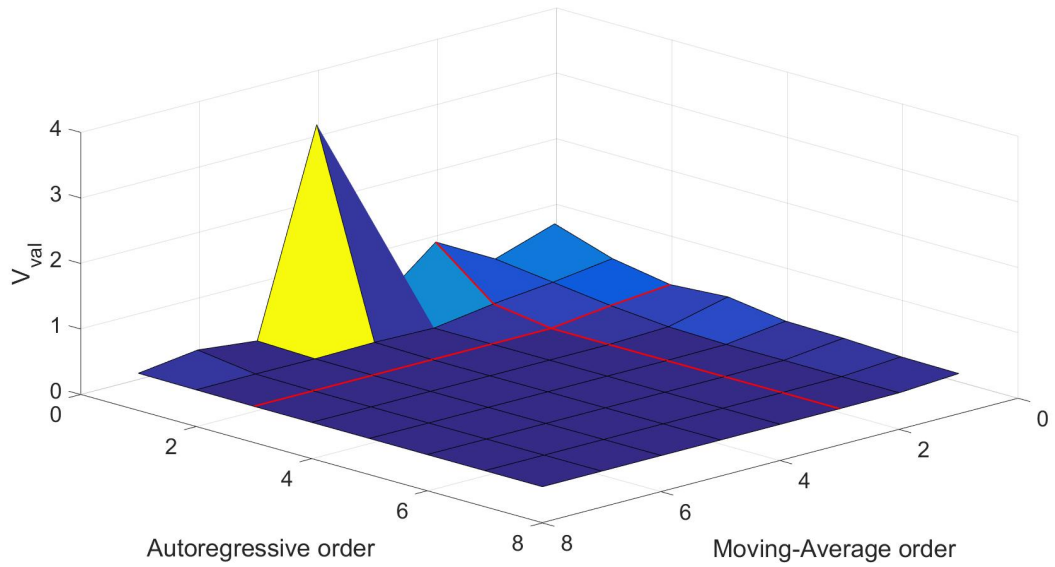


Figure 3.33: Plot of V_{val} obtained with different model orders used for the estimations of a $VARMA(3, 3)$ with 2 variables

3.3.2 Single seasonal models

To limit the number of plots, the scanning order with seasonal models is done with the same AR and MA orders. So a $SARMA(4, 4)(4, 4)_{168}$ is estimated using models of regular orders and seasonal orders from 1 to 8. The results are shown on Figure 3.34 and the same behaviour as for the regular models can be observed. Note that here V_{val} is not defined for the estimations with regular orders 2. In fact the estimates of these models give unstable model, so the resulting residuals go to infinity and it is the reason of these holes in Figure 3.34. Nevertheless, since this does not occur for the potentially best models (with minimum V_{val}), it will not affect the model structure selection.

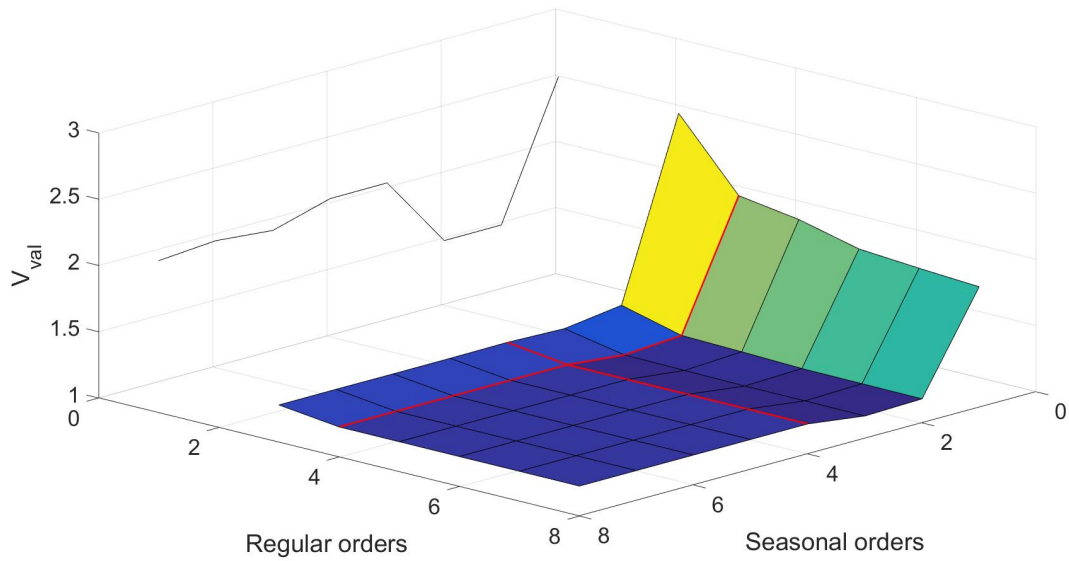


Figure 3.34: Plot of V_{val} obtained with different model orders used for the estimations of a $SARMA(4, 4)(4, 4)_{168}$

For the 2 variables $VSARMA(3, 3)(3, 3)_{168}$, Figure 3.35 shows the scanning order results, which have the same properties as before.

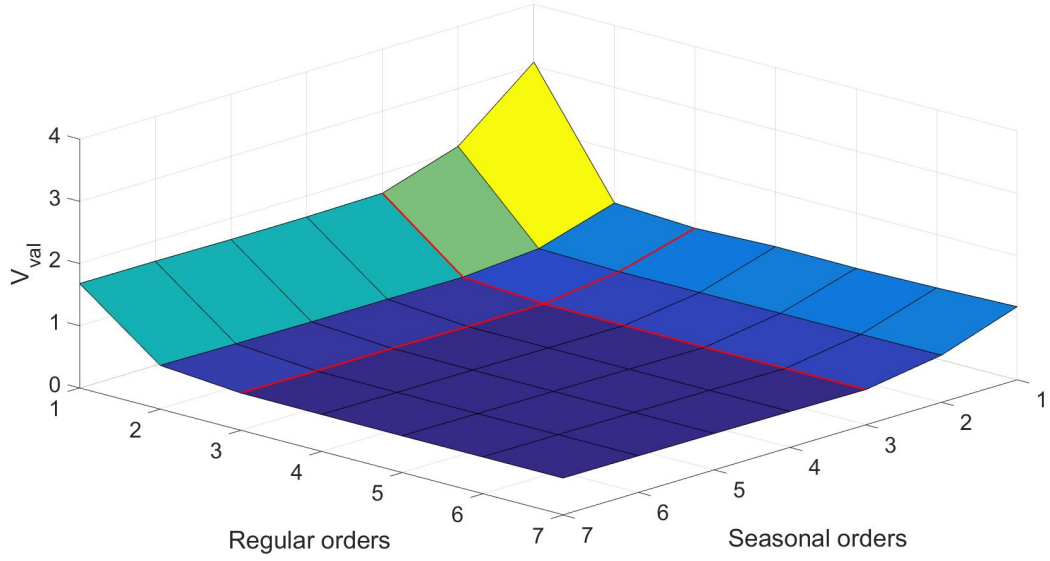


Figure 3.35: Plot of V_{val} obtained with different model orders used for the estimations of a $VSARMA(3,3)(3,3)_{168}$ with 2 variables

3.3.3 Double seasonal models

For the scanning order of a $2SARMA(3,3)(3,3)_{24}(3,3)_{168}$, three figures are shown: for Figure 3.36 the regular orders are fixed at 3, for Figure 3.37 the first seasonal orders are fixed at 3 and for the figure 3.38 the second seasonal orders are fixed at 3. For the aforementioned cases, the results correspond to what has been observed for the previous models.

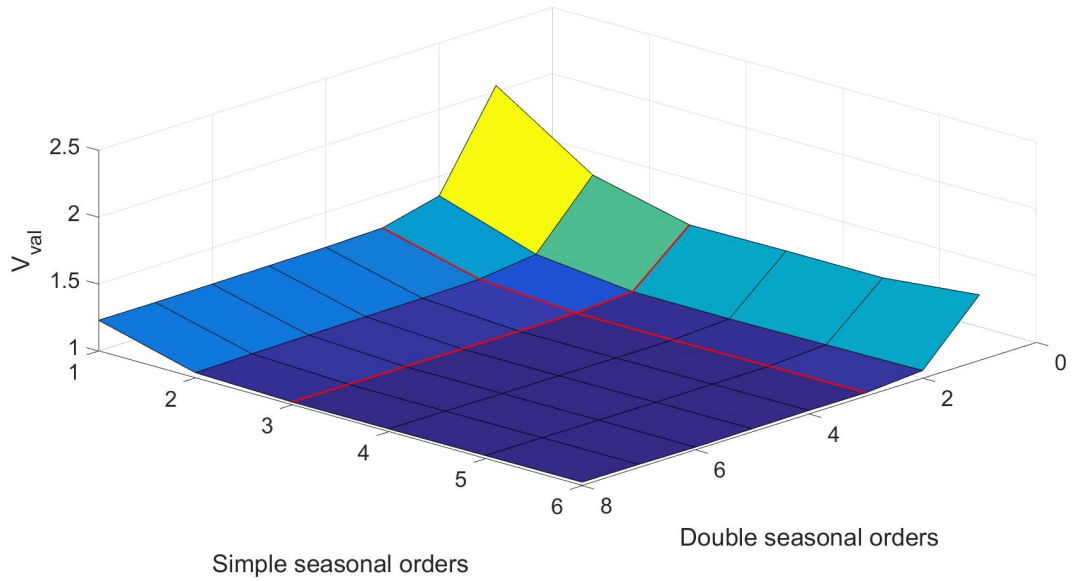


Figure 3.36: Plot of V_{val} obtained with different seasonal model orders used for the estimations of a $2SARMA(3,3)(3,3)_{24}(3,3)_{168}$

For the multivariate case, the scanning order is applied to a $V2SARMA(3,3)(3,3)_{24}(3,3)_{168}$

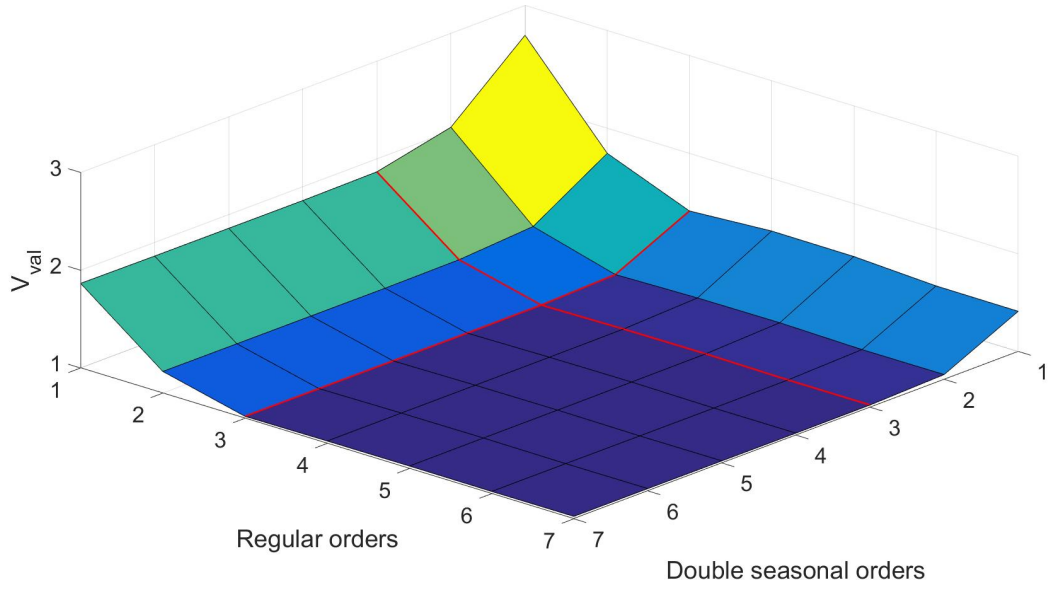


Figure 3.37: Plot of V_{val} obtained with different regular and double seasonal model orders used for the estimations of a $2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$

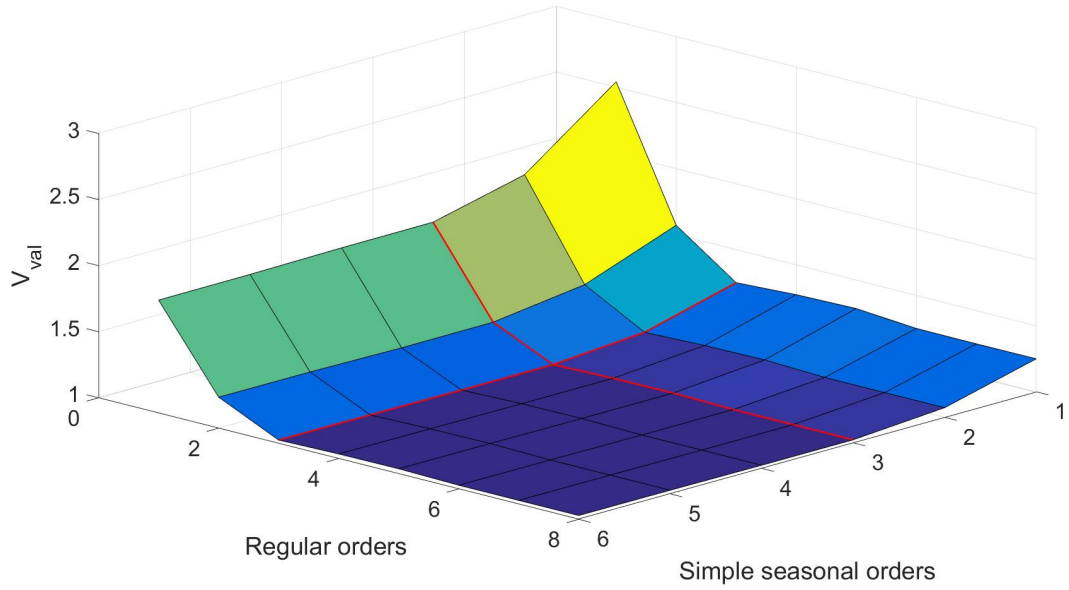


Figure 3.38: Plot of V_{val} obtained with different regular and simple seasonal model orders used for the estimations of a $2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$

of 2 variables. As for the univariate double seasonal model, 3 figures are provided: Figure 3.39 where the regulars orders are fixed at 3, Figure 3.40 where the first seasonal orders are fixed at 3 and Figure 3.41 where the second seasonal orders are fixed at 3. Concerning the values of V_{val} , same conclusions can be drawn as the ones done before.

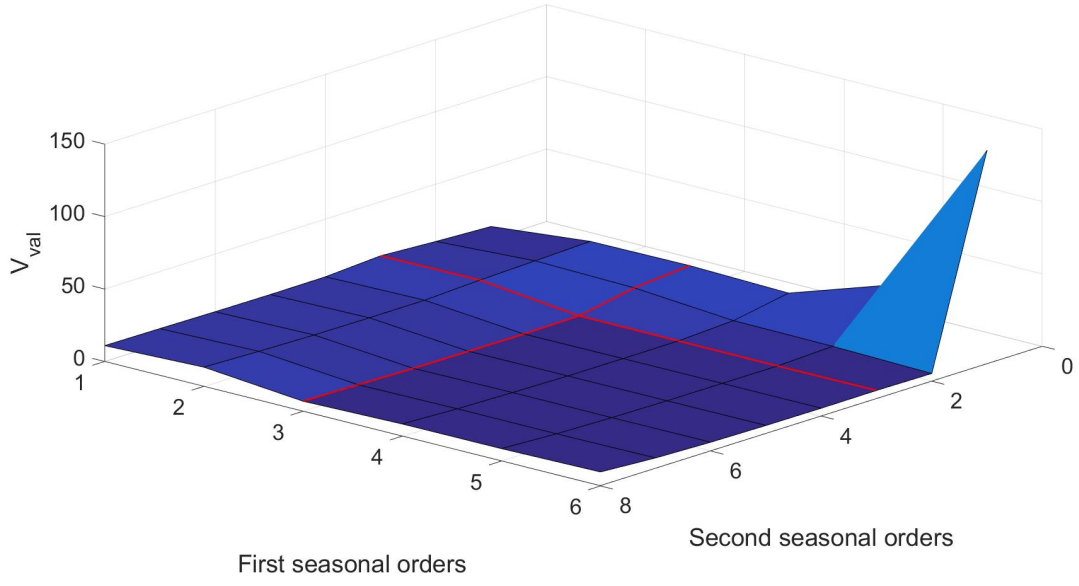


Figure 3.39: Plot of V_{val} obtained with different seasonal model orders used for the estimations of a $V2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$ with 2 variables

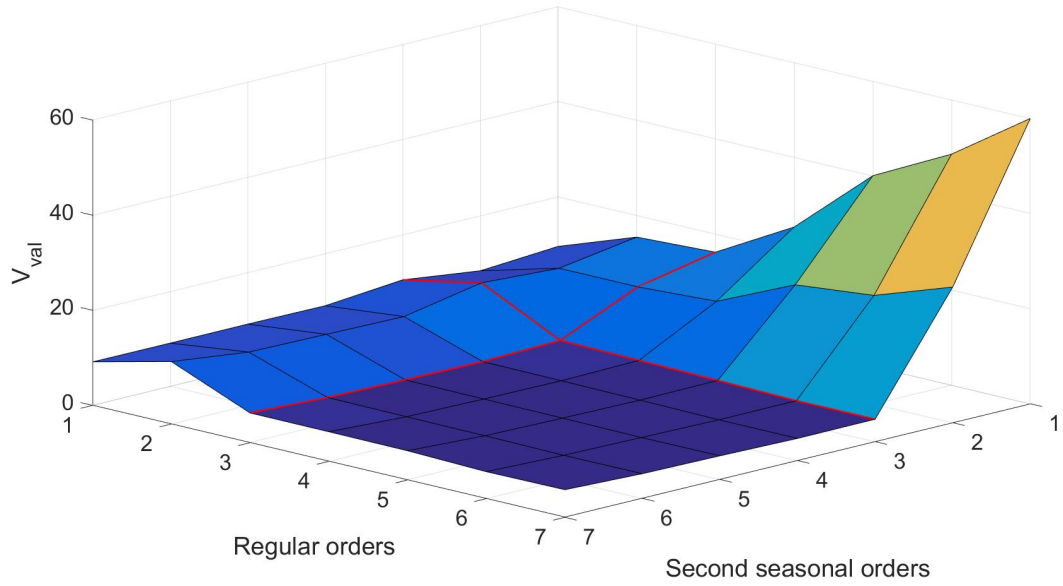


Figure 3.40: Plot of V_{val} obtained with different regular and double seasonal model orders used for the estimations of a $V2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$ with 2 variables

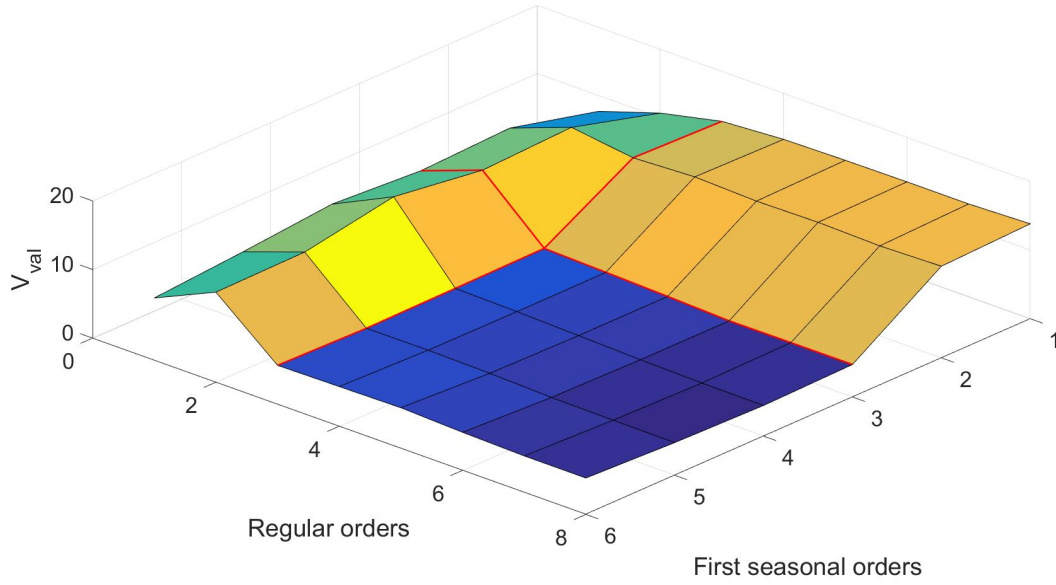


Figure 3.41: Plot of V_{val} obtained with different regular and simple seasonal model orders used for the estimations of a $V2SARMA(3, 3)(3, 3)_{24}(3, 3)_{168}$ with 2 variables

3.4 Conclusion

In this chapter, a proposed extended HR method is described to improve the performances (in the sense of MSE) of single seasonal models. This extended method is also used to model processes with double seasonal behaviour. During the Monte Carlo simulations, values of the internal parameters of the extended HR method have been discussed and then chosen to optimize the accuracy of the obtained estimates.

Then, an analysis of the theoretical MSE in terms of bias and variance is carried out for this method. The results obtained in simulations show that the variance of the noise does not influence the parameter estimates and the accuracy of the parameter estimates increases (i.e. the MSE decreases) with the number of data.

Finally, a method for the model structure selection has been thoroughly studied. The simulation results using training and validation sets show that the model order selection method can be potentially applied successfully to the real application studied in the following chapter.

Chapter 4

Application of the methods to a real life process

This chapter deals with the application of the modeling procedures presented so far (the extended HR and the model structure selection methods) to a real life electricity consumption modeling case. This is done for the consumption of three different rooms separately (using univariate models), for two pairs of rooms and finally for two triplets (using multivariate models in the last two cases). To model and predict the consumption of the first room, a detailed step-by-step description of the methodology is provided while a brief presentation including the important steps is presented for the other cases.

The different steps of the modeling procedure that can be used as a guideline for modeling any time-series coming from a real life process, are the following :

- An analysis of the data is performed to check whether there is some particular behaviour to take into account (e.g. seasonality) or to correct (e.g. missing data) before constructing the model structure.
- The set of model structures chosen to describe the real process is defined.
- The specific parameters used in the model estimation method (here extended HR method) are specified ($p_{large}, \dots$).
- The model structure selection procedure is applied using the scanning order method described in Section 3.3.
- Diagnostics using the results obtained with the model chosen after the model selection procedure are provided. Following these diagnostics, some modifications are eventually applied to the chosen model structure to improve the performance of the end result.

- The (possibly modified) model is applied and its performance is analysed.
- Robustness of the estimated model with respect to the length of the data is studied. This is done by computing the parameter estimates on a small training set and analysing the results when this training set is progressively increased. The same procedure is repeated for the validation set.
- Following the results obtained in the previous step, a new model is estimated (if necessary) using different training and validation sets. This is the final model used to predict the real process.
- The multi step-ahead predictor is applied to analyze the accuracy of the estimated model for predictions of several steps ahead.

Keep in mind that for the aim of privacy protection, the real data are not be presented as such. To allow readers to have an idea of the shape of the real consumption (or consumption estimates), the following transformation is applied before plotting :

$$y = \frac{y_{original}}{c}$$

where $y_{original}$ are the real data, y are the normalized data (used in the plots), c is a constant preserved unknown for the readers.

Before going further, recall that for the model structure selection procedure, a criterion needs to be chosen. As in Chapter 3, the cost function for a specific model H is used at this purpose. In this chapter, all the model performance indices are computed on the validation set, and the same holds for the cost function. Then the computation of the cost function on the validation set used as criterion is as follows :

$$V_{val} = \frac{1}{N_{val}} \sum_{t=N-N_{val}+1}^N ||H^{-1}y_t||_2^2 \quad (4.1)$$

where N_{val} is the size of the validation set, N is the number of output signal measurements available and y_t is the electricity consumption for $t = 1, \dots, N$.

Note that V_{val} is computed on the unnormalized data ($y_{original}$). Concerning the other metrics used further, they are all constructed in such a way that their values are unaltered by the transformation (same values for y and $y_{original}$).

4.1 Room 1

The first univariate model is obtained to predict the electricity consumption of one room.

4.1.1 Data analysis

Figure 4.1 shows the general shape of the consumption in this first room. A zoom on two random weeks shows a double seasonal behaviour (see Figure 4.2) with $s_1 = 24$ (one day) and $s_2 = 168$ (a week). Figure 4.3 shows the autocorrelation function of the consumption of this room, which consolidates us in the seasonal behaviour.

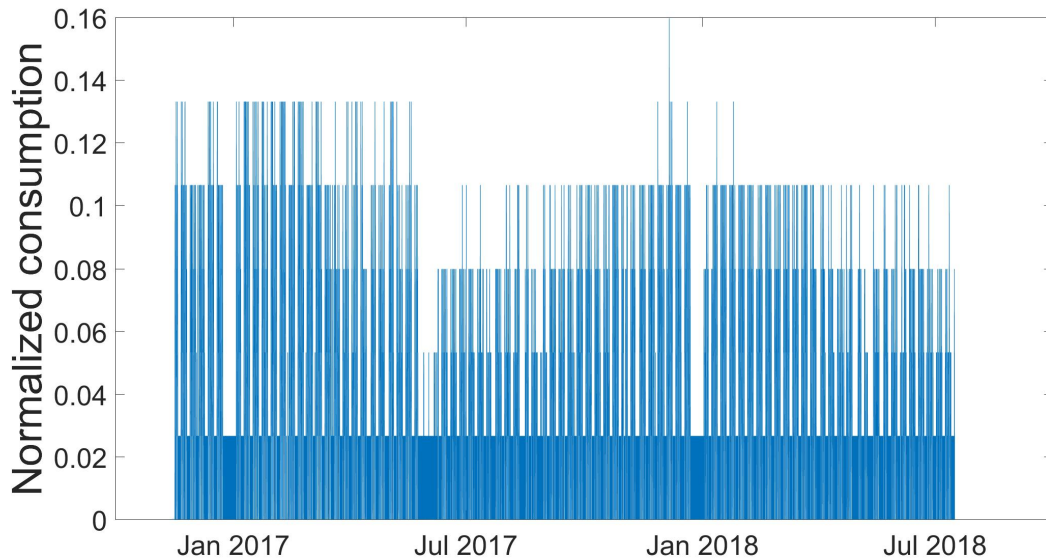


Figure 4.1: Plot of the normalized electricity consumption of Room 1

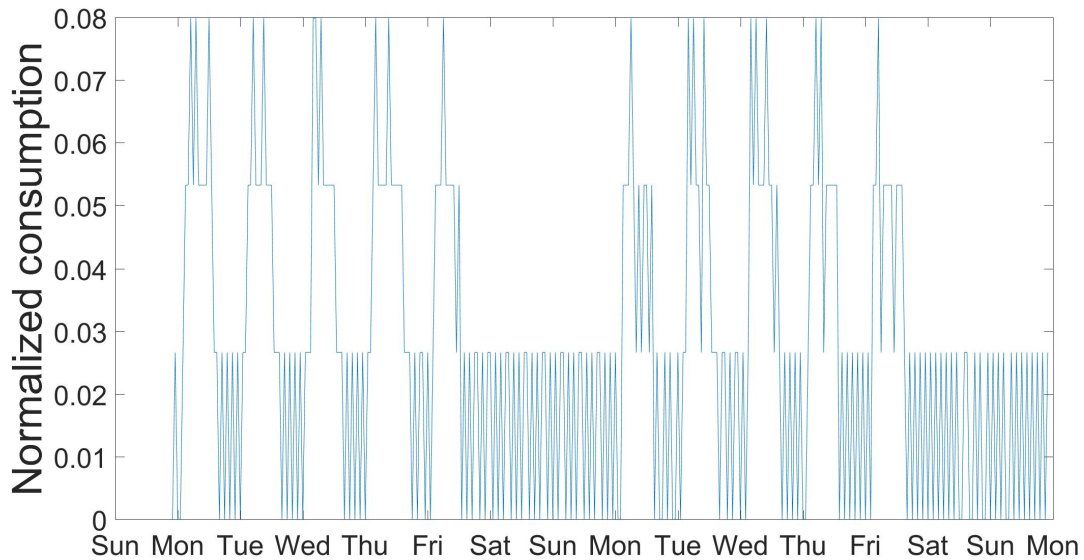


Figure 4.2: Plot of the normalized electricity consumption of Room 1 for 2 random weeks

If we take a closer look to Figure 4.1, two holes can be seen around the end of the years 2017 and 2018. In fact it corresponds to the Christmas holiday weeks, for which the general behaviour is broken. There are two possibilities to manage this kind of pattern: either these

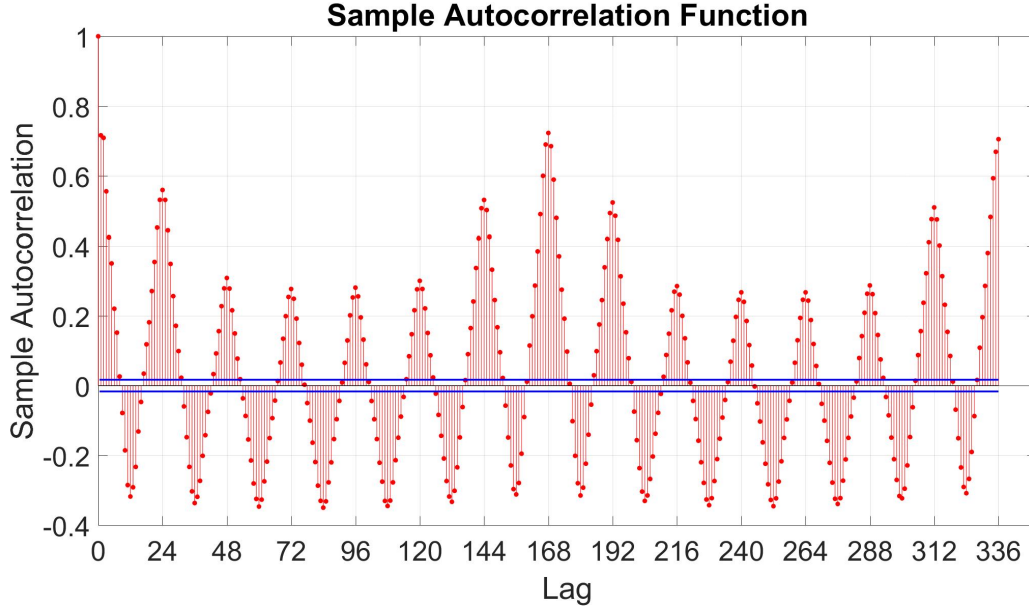


Figure 4.3: Plot of the autocorrelation function of the electricity consumption of Room 1

weeks are removed from the time-series (like if they were not existing), or they are replaced by some estimation (for example, the mean of the other weeks of the time-series or some estimations using a Kalman filter). Here the first possibility is chosen, because it allows the link of the first week of January with previous data (without including the Christmas week). If the other possibility was used, the first model parameters of the first week of January would be used with data created artificially. This means that these parameters would not be directly link to real data. Furthermore there is only one Christmas holiday week per year, corresponding to 2 weeks out of the 85 weeks of data available in our case. So simply removing them is not bound to deteriorate the model performance.

For reasons of completeness we should mention that another possibility to deal with this issue could be to consider the Christmas weeks as unknown (like if there were no available data) and to use model capable to handle missing data, such as the Kalman filter.

4.1.2 The model structure

When we look at Figures 4.1 and 4.2, an homogeneous non-stationary behavior can be identified (similar repeated behaviour during the week days and kind of a cut off during the weekend). Like it was explained in Section 2, an integrated model can be used to capture this kind of behaviour. So the weekly differencing operator $1 - B^{168}$ is applied to the room consumption to remove this non-stationary behaviour. The resulting signal z is shown on Figure 4.4. The autocorrelation function of this deseasonalized signal can be seen at Figure 4.5. Compared to Figure 4.3, we see that already an important part of the signal has been modelled. Trying to deseasonalize a second time with respect to the day gives the autocorrelation function of Figure 4.6. Comparing this

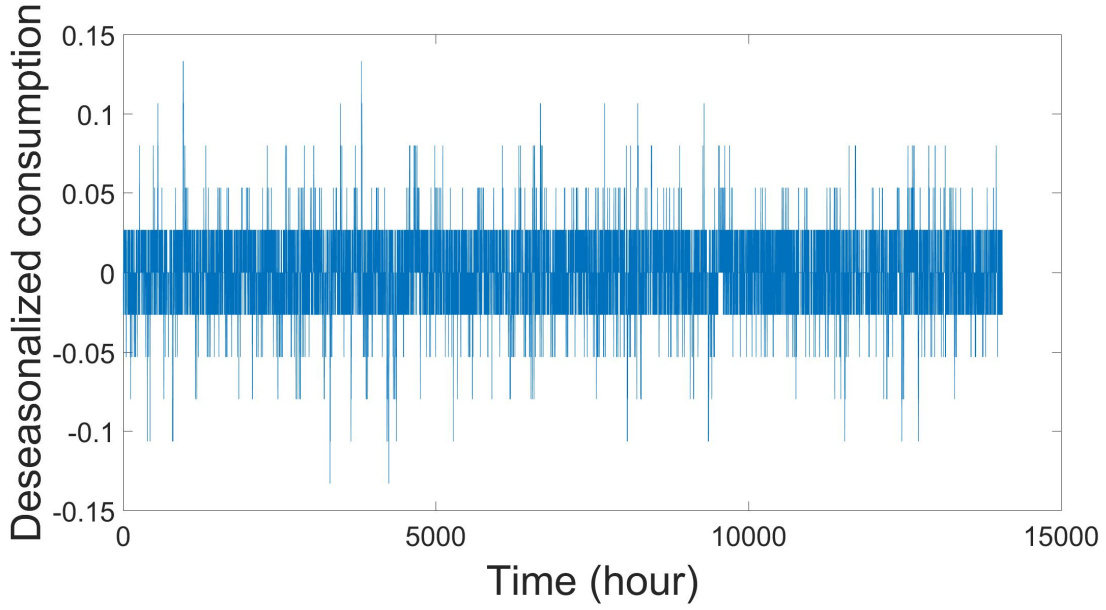


Figure 4.4: Plot of the deseasonalized electricity consumption $z(t) = (1 - B^{168})y(t)$ of Room 1

figure with Figure 4.4, it seems better to deseasonalize once with regard to the week than twice.

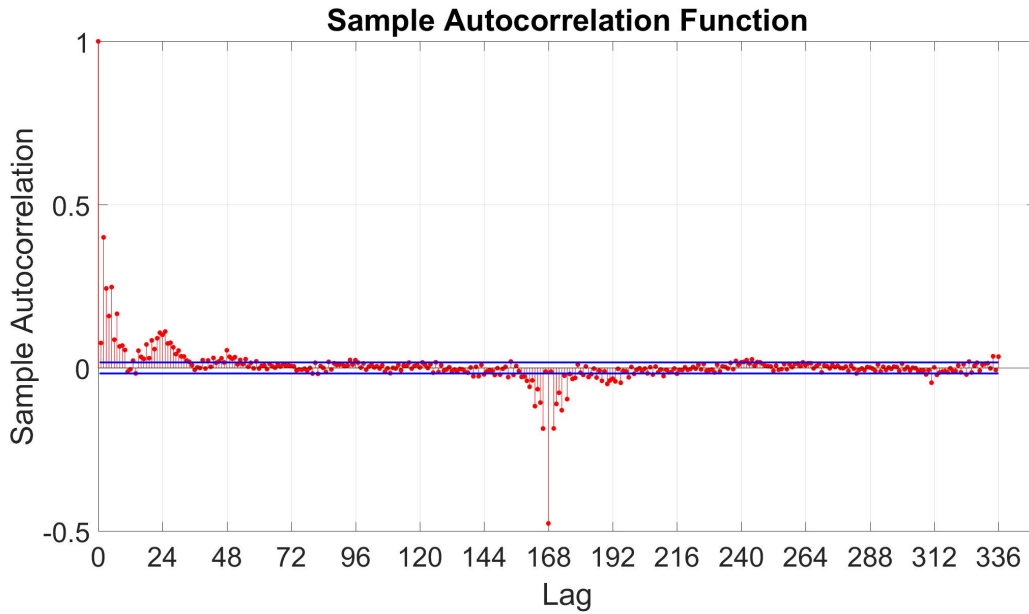


Figure 4.5: Plot of autocorrelation function of z until 2 week lags (336 hours) for Room 1

The initial idea to capture the process dynamics with a double seasonal model is confirmed by the autocorrelation of the transformed signal output z shown on Figure 4.5. The lags 24, 48, etc as well as the lags 168, etc are correlated with z_t . So the model structure used for the model estimation is a $2SARIMA(p, q, 0)(P_1, Q_1, 0)_{24}(P_2, Q_2, 1)_{168}$.

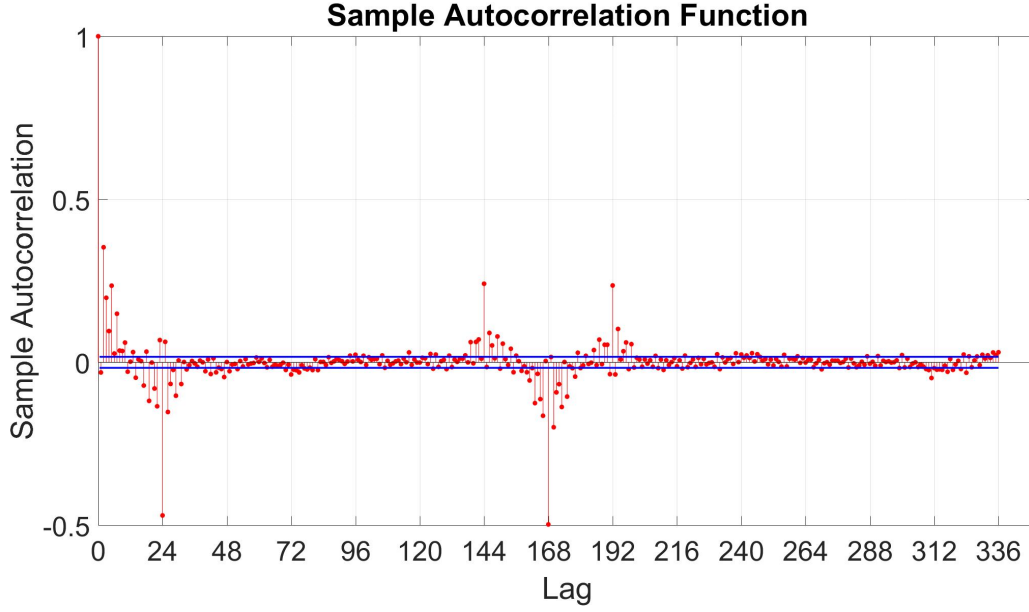


Figure 4.6: Plot of autocorrelation function until 2 week lags (336 hours) of Room 1 electricity consumption deseasonalized twice (with regard to the week and day)

4.1.3 Estimation method

The method used for the estimation of the model is the extended Hannan-Rissanen method. Arguments for this choice have been explained previously, and the internal parameters used are as follows ($s_1 = 24$):

- For the scanning order :

$$p_{long} = \lceil \min(\log(N)^{1.3}, 22) \rceil$$

$$P_{1_{long}} = 6$$

$$P_{2_{long}} = 8$$

- For the diagnostics and the estimation of the final model :

$$p_{long} = \lceil \min(\log(N)^{1.3}, 22) \rceil$$

$$P_{1_{long}} = 3 \max(P_1, Q_1)$$

$$P_{2_{long}} = P_2 + Q_2$$

4.1.4 Scanning order procedure

To choose the model structure, first the size of the training and validation sets need to be selected. Looking at the last weeks of the data, some differences between successive weeks can be seen and so one or two weeks will probably not be enough to verify if the behaviour of the process

has been well estimated. The proportion 90%/10% seems to be a good idea, but the junction of these two sets happens in the middle of May 2018. The problem in that case is the presence of several days off during this month, which are linked to unusual non-stationary behaviour and be practically translated as noise in the model. In other words, it's an abnormal comportment of the process, i.e. if our model would capture this behaviour it would have become way more complex. Since the idea is to verify the model performance using the validation set, we do not want this kind of perturbation in it. So the final proportion chosen is 92%/8% allowing to have 7 weeks from June in the validation set. We thus have $N_{val} = \lceil 0.08N \rceil$ with $N = 15000$ approximately.

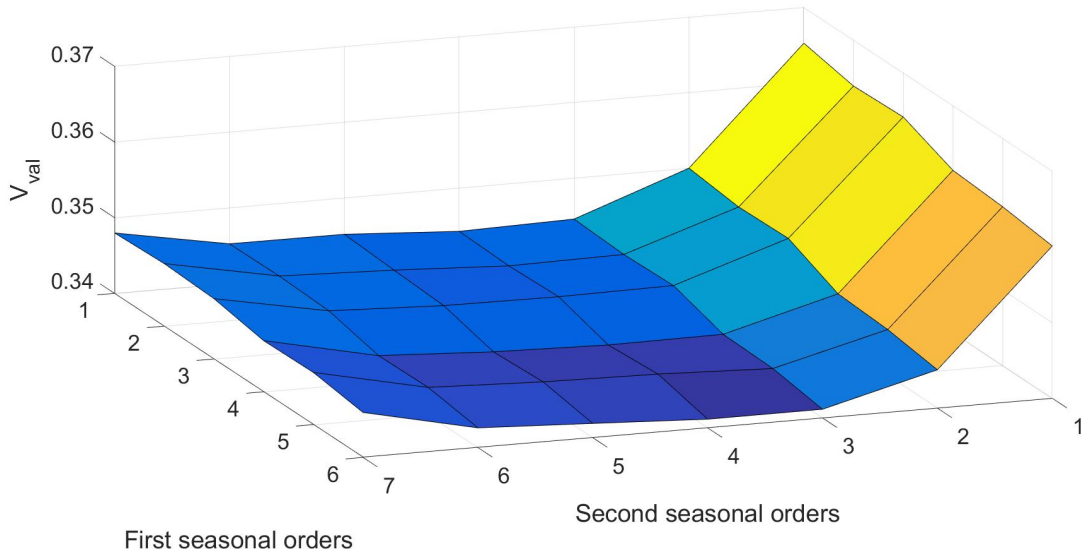


Figure 4.7: Plot of the scanning order for Room 1 with $p = q = 15$ and with the seasonal AR and MA orders the same

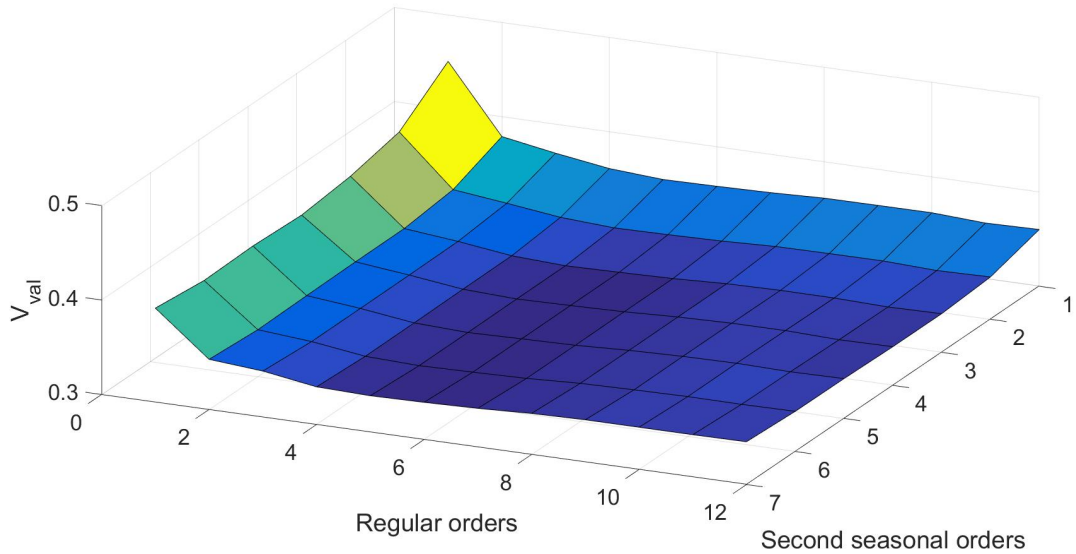


Figure 4.8: Plot of the scanning order for Room 1 with $P_1 = Q_1 = 6$ and with the AR and MA orders the same for the regular and second seasonal orders

Now that the training and validation sets are chosen, let's begin the scanning order procedure

to find the appropriate model structure. The first scanning is done for a model where the regular orders (p and q) are fixed to 15 (chosen after different tests). Both seasonal orders vary from 1 to 7, while keeping the regular AR and MA orders constant. The resulting cost function on the validation set is shown at Figure 4.7. It seems that the best pair of second seasonal parameters (P_2 and Q_2) is 3 no matter the first seasonality orders. To see if this minimum is the same when the regular orders vary, the scanning is done with regular and second seasonal orders, keeping first seasonal orders equal to 6, through Figure 4.8 (MA and AR orders stay equal). Similar to Figure 4.7, the minimum of the second seasonal orders seems to not depend on the regular ones. Finally, the regular orders are fixed to 6 (local minimum at Figure 4.8) and the first seasonal orders to 6 (local minimum at Figure 4.7). Then a scanning is done with different values of P_2 and Q_2 . The result is shown on Figure 4.9 and the best pair for minimizing V_{val} is $P_2 = 3$ and $Q_2 = 5$.

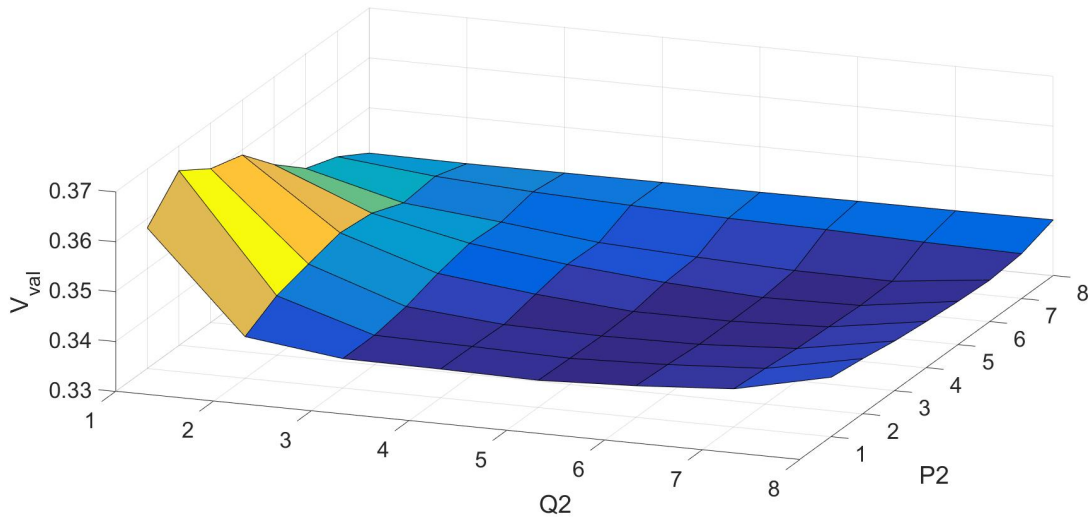


Figure 4.9: Plot of the scanning order for Room 1 with $p = q = 6$ and $P_1 = Q_1 = 6$

The next step is to scan the regular and first seasonal orders (still with same AR and MA). This is done at Figure 4.10. The minimum of V_{val} is around the point (5,5) for the regular and first seasonal pair of orders, respectively. There is another local minimum at (23,6), but we will come back to it at the diagnostics part in the following subsection. Nevertheless it seems that the value of the regular orders will probably not influence significantly the best P_1 and Q_1 .

Finally, to minimize V_{val} , the regular orders are fixed at 5 and then the scanning is done with P_1 and Q_1 . This is shown at Figure 4.11 and we see that minimum is achieved for $P_1 = Q_1 = 6$. The last scanning is done for p and q at Figure 4.12 when fixing the other orders at the best values found before. The best p and q seems to be 6 and 5, respectively.

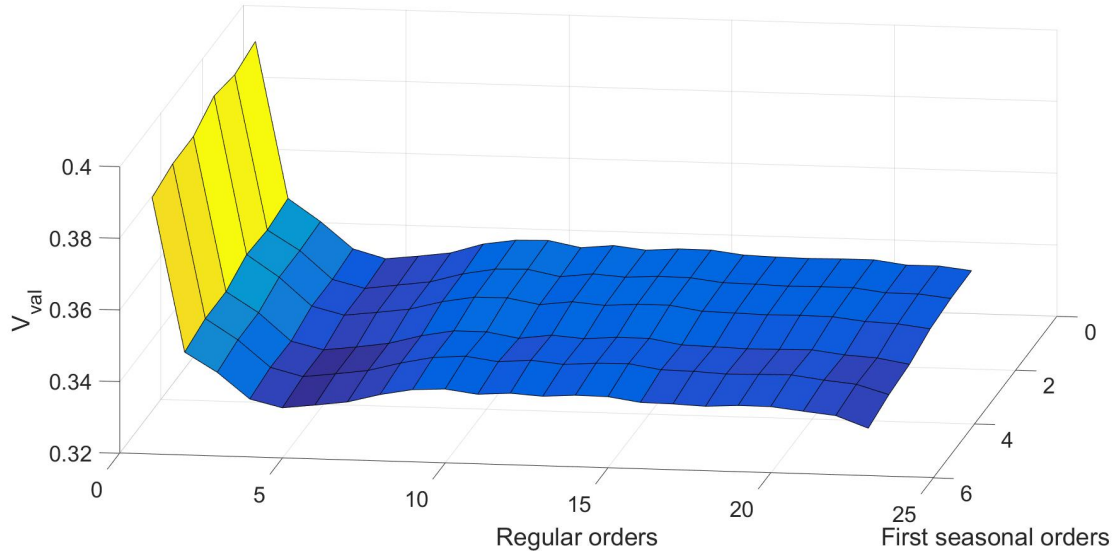


Figure 4.10: Plot of the scanning order for Room 1 with $P_2 = 3$ and $Q_2 = 5$ and with the AR and MA orders equal for the regular and first seasonal orders

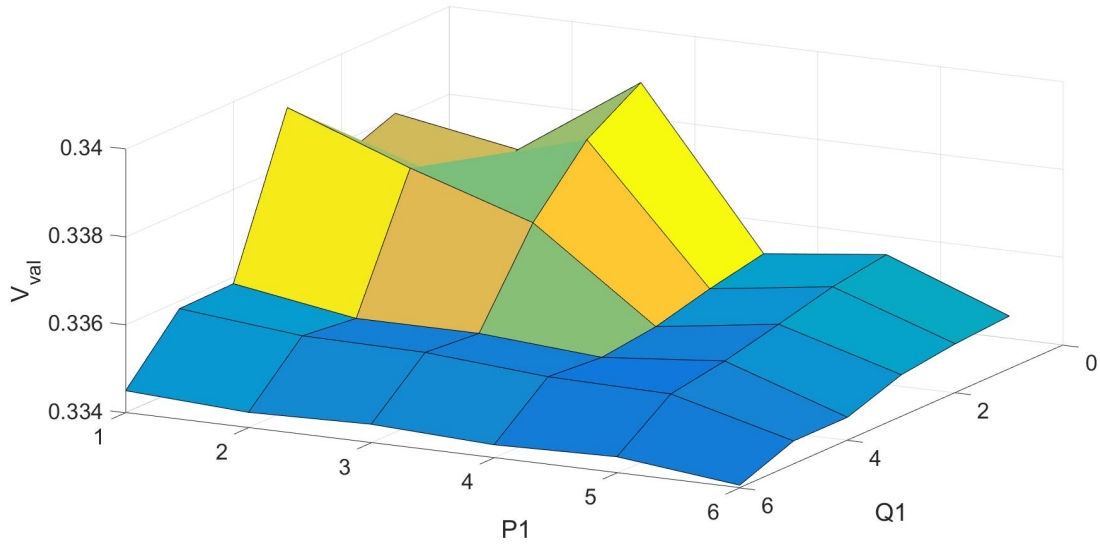


Figure 4.11: Plot of the scanning order for Room 1 with $p = q = 5$, $P_2 = 3$ and $Q_2 = 5$

4.1.5 Diagnostics

The next step in order to choose the final model is to check whether this model performs as expected and whether it can be improved. Recall that we want our model to predict as well as possible the consumption of electricity in the room. If the model was perfect, the estimated residuals would be i.i.d., given no model mismatch (the only thing not seen by the model is the white noise). So the diagnostics are done using the autocorrelation of the estimated residuals. This autocorrelation is computed on the training set, because the model tries to find the real pattern on this set. So for the perfect model, the autocorrelation of the residuals on the training

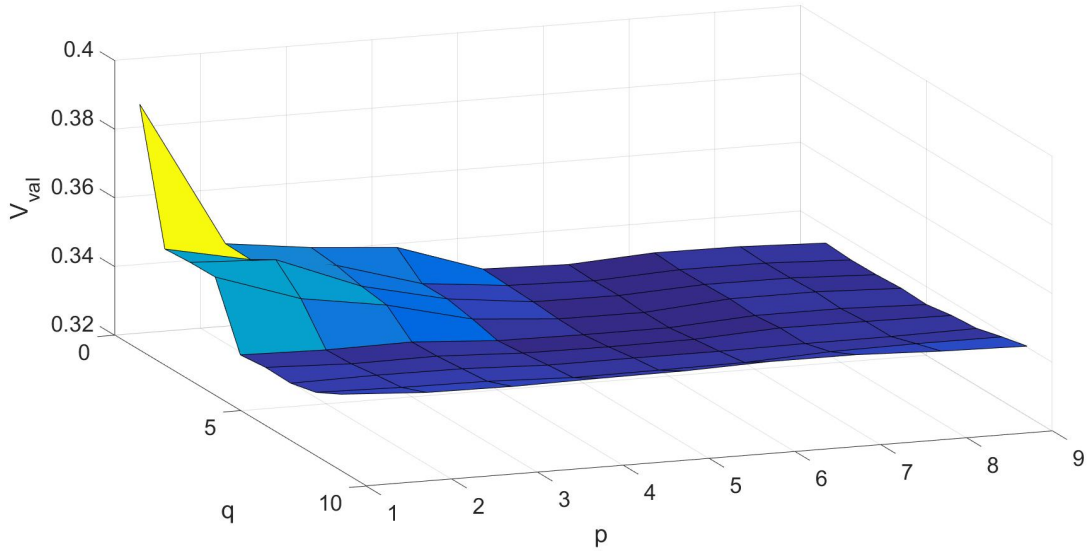


Figure 4.12: Plot of the scanning order for Room 1 with $P_1 = Q_1 = 6$, $P_2 = 3$ and $Q_2 = 5$

set would be 0 everywhere apart from the 0th lag, corresponding to the variance of the noise realization.

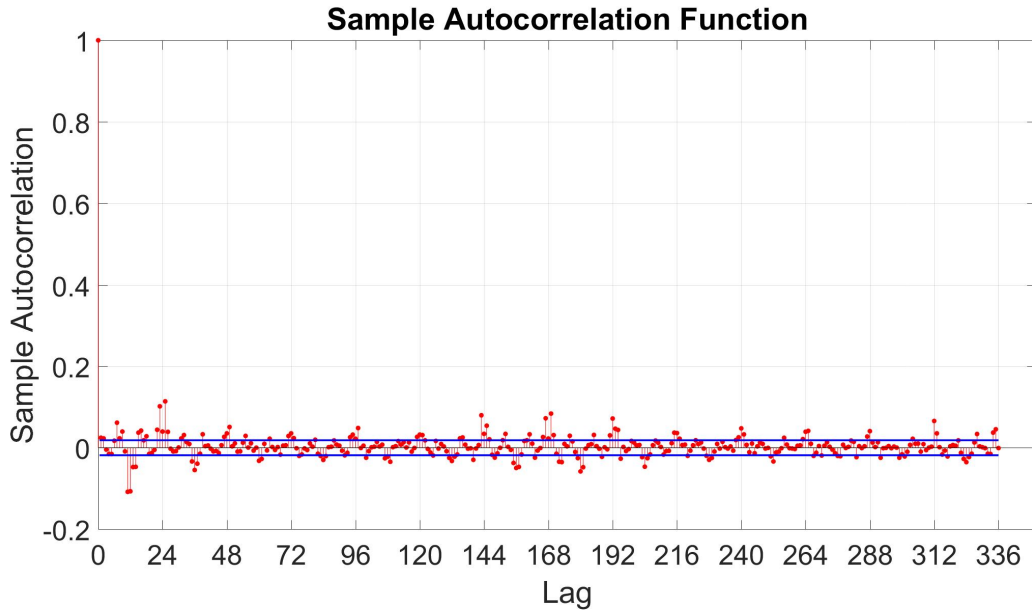


Figure 4.13: Plot of autocorrelation function of the residuals obtained using the model $2SARMA(6, 5, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ of Room 1

Recall that the chosen model after the scanning order is a $2SARMA(6, 5, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$. The residual autocorrelation function obtained with this model is plotted on Figure 4.13. The blue lines correspond to the 95% confidence bounds : if the autocorrelation is between these two lines then we can accept the statistical hypothesis that the value of the residuals is 0 with a probability of 0.95.

The first thing that can be noted is the fact that some lags from 7 to 23 are out of the

confidence bound, pointing to possibly unmodeled dynamics. To try to deal with that, the values of p and q can be increased to model dependencies between y_t and $y_{t-7}, y_{t-8}, \dots, y_{t-23}$. In fact, if we remember the local minimum on Figure 4.10 with regular orders of 23 and first seasonal orders of 6, we can repeat the scanning order for higher values of p and q (shown on Figure 4.14). We see that a better local minimum is found for $p = q = 23$. Using these values for the model, all the lags until 22 are now inside the confidence bound (Figure 4.15). Nevertheless the autocorrelation at the lag 23 stays outside the bound. It seems that the model structures proposed in this thesis cannot describe the dynamic linked to the apparition of this peak in the autocorrelation at lag 23. So even if these values occur due to a model model mismatch, we can accept here that the autocorrelation of the residuals resembles the one of a white noise signal.

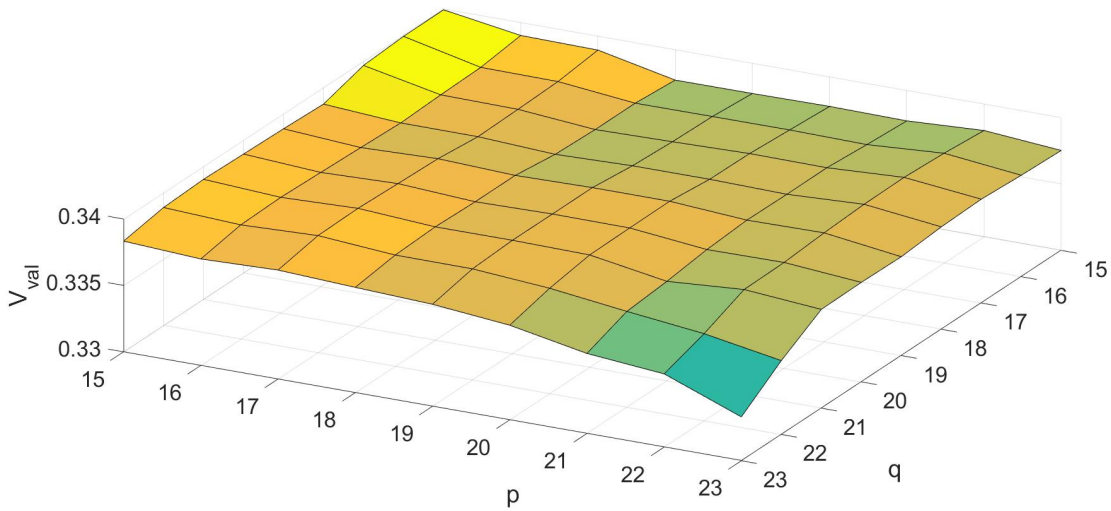


Figure 4.14: Plot of the scanning order for Room 1 with $P_1 = Q_1 = 6$, $P_2 = 3$ and $Q_2 = 5$

Concerning the daily lag and its multiples (24, 48, ..., 144), it seems that some of these are a little bit higher than the confidence bound. This phenomenon is already attenuated with the increase of the regular orders. Looking at the values of the daily parameter estimates, we see that $\Phi_6^{24} = 0.0018$, which is really low comparing to the other daily parameter estimates which have an absolute value mean of 0.0517. So this parameter can be omitted and $P_1 = 5$. It results with only the lag 72 which is a little bit higher than the confidence bound. Once again we suppose that this phenomenon appears due to model mismatch.

For the weekly lags (168, 336), the autocorrelations are in the bound and increasing the double seasonal orders does not improve the autocorrelation function or the cost function on the validation set (V_{val}).

4.1.6 Results given by the model

At the seen of the diagnostics, a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ model is chosen to describe this process linked to Room 1. Using this model, a cost function of 0.3358 on the validation set is obtained as well as the training set residual autocorrelation function on Figure 4.15.

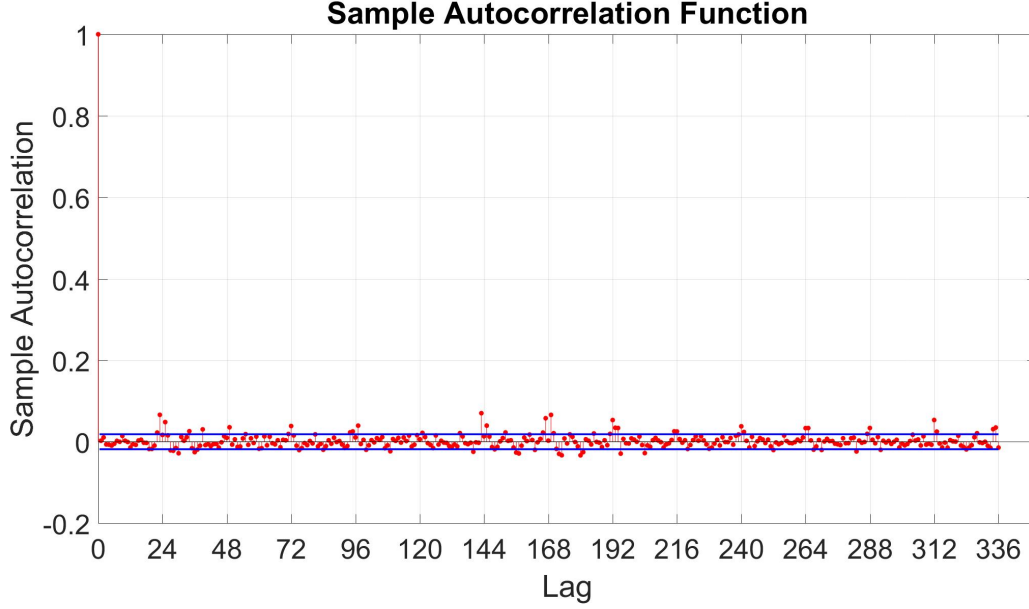


Figure 4.15: Plot of autocorrelation function of the residuals obtained using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ of Room 1

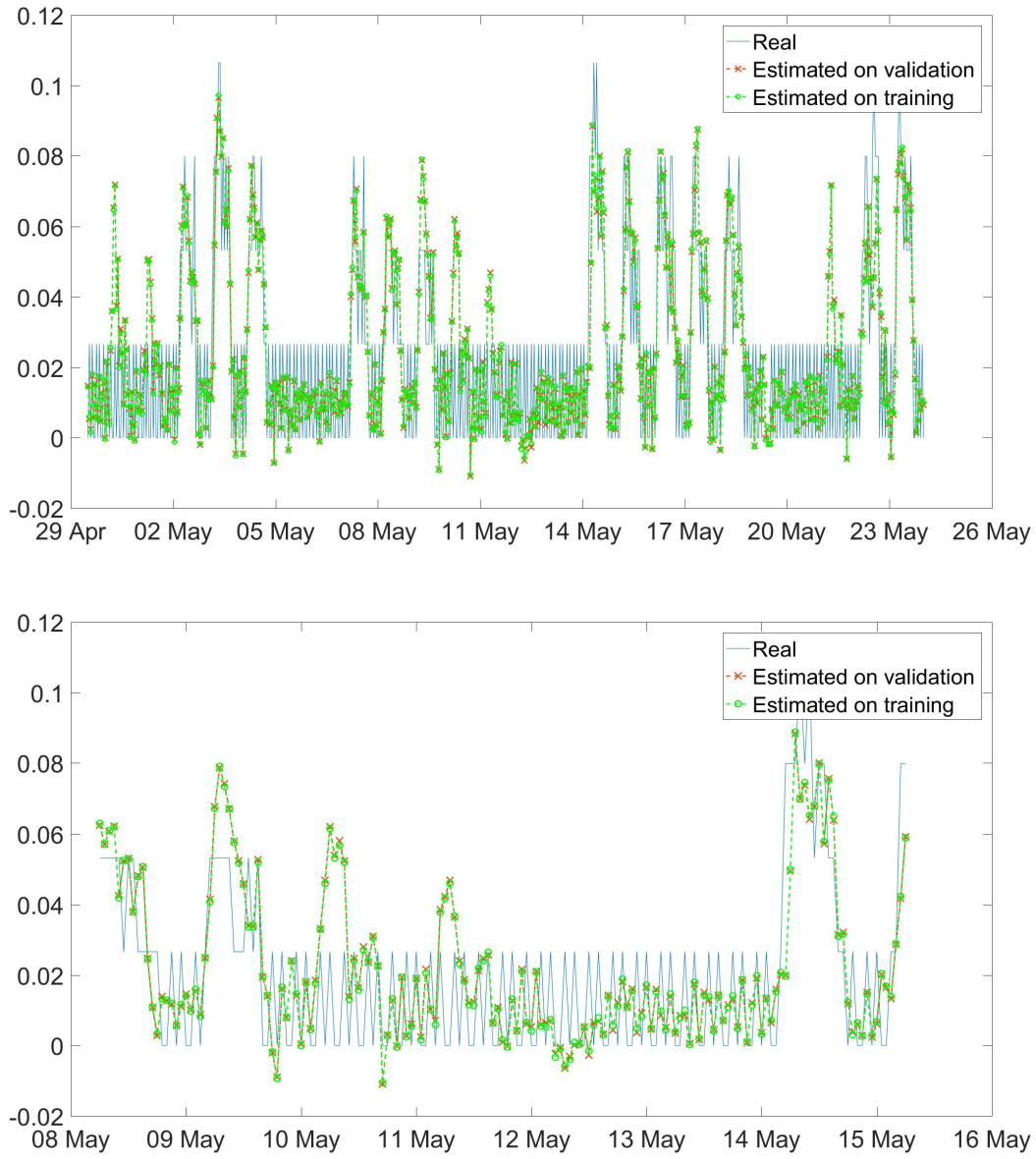
To have a better idea of the performance of the model, an other metric is used. In this work, we call **accuracy metric**, the metric that compares the derived predictions with naive ones to check whether a very simple predictor could be used to provide acceptable forecasts instead of our derived model. This accuracy metric is given by

$$\frac{\text{var}(y_t - \hat{y}_{t|t-1})}{\text{var}(y_t - y_{t-s_2})} \quad (4.2)$$

It compares the variance of the residuals obtained with the one-step ahead predictor of the estimated model with the variance of the residuals obtained using the consumption of the previous week (or season in general) as prediction. If the result is lower than one, it means that the model gives better results than the naive predictor which only looks at the consumption of the previous week. It is thus a good indication that the model is build in a good direction. The estimated model here gives an accuracy metric on the validation set of 0.4834. Note that the accuracy metric has the same value for y and $y_{original}$.

Talking about the issue concerning the days off, Figure 4.16 represents the consumption during May 2018 with regard to the estimated consumption when May is included in the training

set (ratio 92%/8%) in green and when it is included in the validation set (ratio 87%/13%) in red. From that figure we can conclude that the estimations are not very different on the training and validation set for the days off. So including May in the validation set is not bound to affect the results.



(a) Zoom

Figure 4.16: Plot of the real Room 1 normalized consumption for May 2018 and the estimation using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ when May 2018 is included in the training set or in the validation set

4.1.7 Robustness with respect to the number of data

Until now, all the available data were used to train or to validate the model. But it could be interesting to see how the model behaves if there were more or less data available, indicating if the model could be improved with more training data, or if the estimation of performance, could be better with a bigger validation set.

The first thing that can be studied is the cost function evaluated on the validation set (V_{val}) for the same model structure, trained on training sets of different size. At this purpose, the validation set is kept the same, and the minimum size of the training set is fixed to 15 weeks (approximately 4 months). The junction between the training and validation sets is also kept the same, so the training always ends when the validation begins (the two sets are consecutive). The size of the training set is increased by adding weeks at the beginning of it and the cost function on the validation set is calculated and plotted on Figure 4.17.

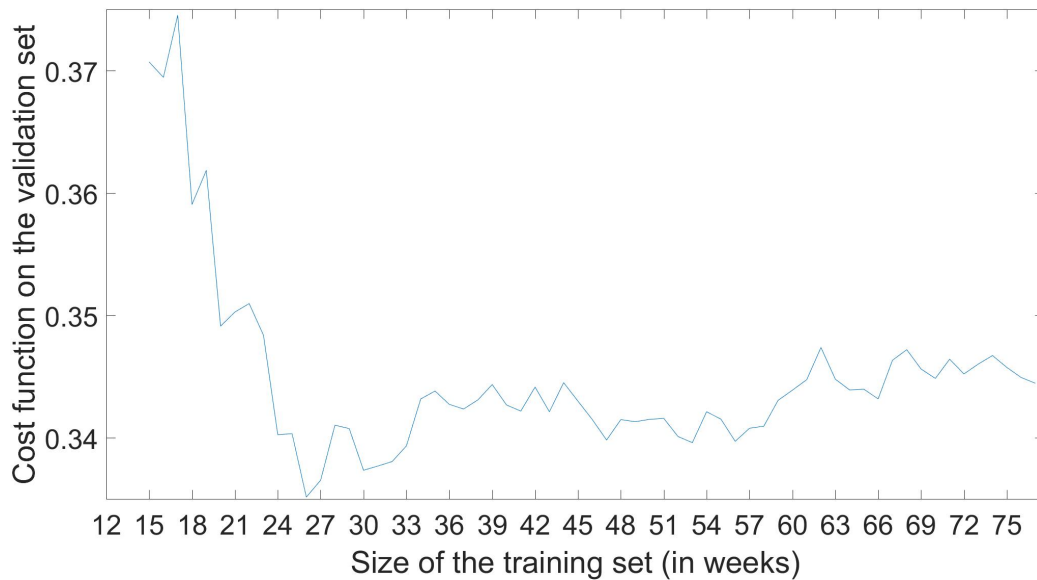


Figure 4.17: Plot of the cost function computed on the validation set (8% of the data) using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1, for different size of training set

It can be observed that the cost function decreases sharply until 26 weeks and increases slowly after that. Alternatively, we would expect that the cost function decreases when the size of the training set increases (because the simulation-based study shows that parameters estimates are better computed when there is more data), or at least that it does not increase. There are two possible ways to explain that : either the 3 months preceding the validation set already contain all the information of the real process and more data does not bring new information, or there is something wrong with the validation set (for example, it is not large enough). We will see later that by choosing a larger (and then a more representative) validation set, this phenomenon is not observed any more.

Fixing now the training set, the size of the validation set is modified. The same junction between the validation and training set is still kept, and more and more data are included in the validation until it contains 8% of them. The resulting cost function computed on the different validation sets are plotted on Figure 4.18. We see that the cost function is not really "stable", in the sense that it decreases when the size increases and it seems that it could continue to decrease if there were more data available. This is not good news because we would like our validation set to represent as good as possible the behaviour of the real system, and so that adding new data to it does not change a lot the information inside it. So V_{val} should not vary a lot when there is enough data contained inside the validation set.

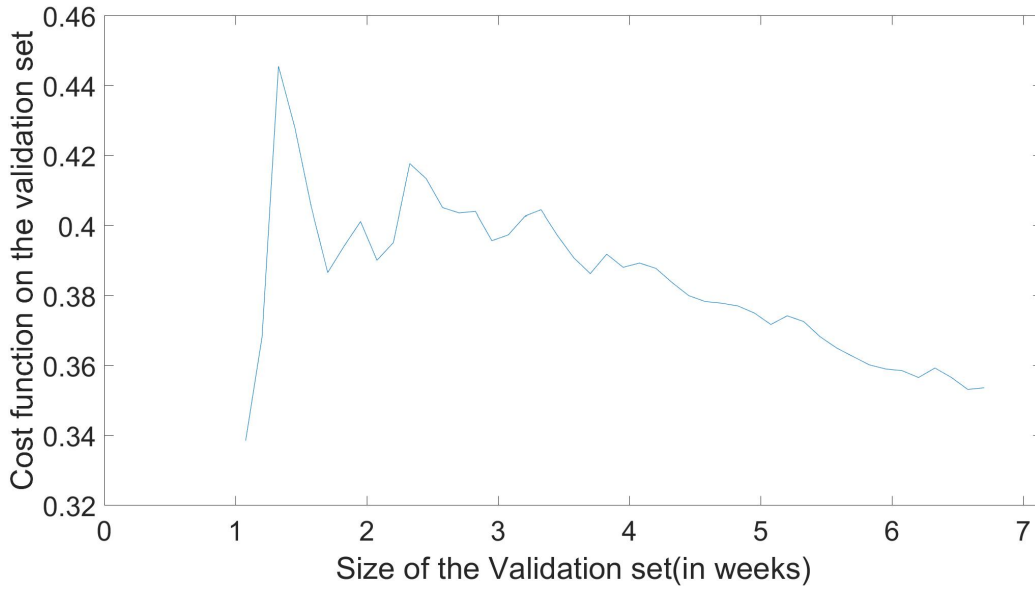


Figure 4.18: For a fixed training set (92% of the data), plot of the cost function computed on validation sets of different size using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1

These observations question the previous choice of the training/validation set proportion. And like the cost function does not decrease if we decrease the number of data (using the same validation set, see Figure 4.17), and May seems to not be a problem for the validation set (see Figure 4.16), the training/validation ratio is changed to 80%/20%. The results for the model robustness using this ratio, shown on Figures 4.19 and 4.20, seems to be more promising. Indeed the cost function decreases until 36 weeks and then stays constant with increasing training size. Moreover it varies less when the size of validation set is enough large.

4.1.8 Final model

From the robustness analysis point of view, the choice of the final structure model and the computation of its performance is done using the training/validation ratio of 80%/20%. The model structure selection and the diagnostics are repeated using this ratio. But like the methodology was already exposed in details, the final model structure obtained is directly

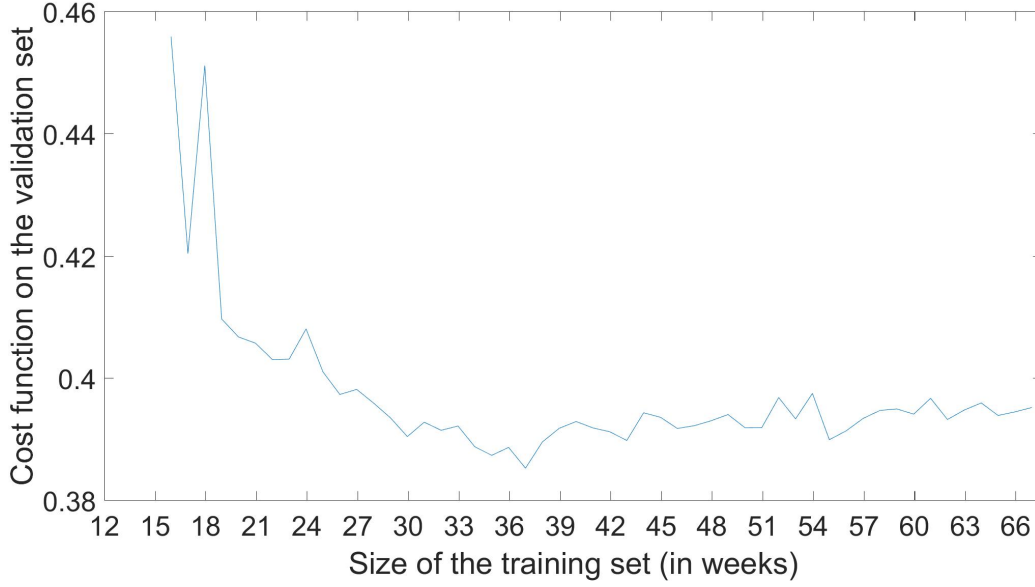


Figure 4.19: Plot of the cost function computed on the validation set (20% of the data) using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1, for different size of training set

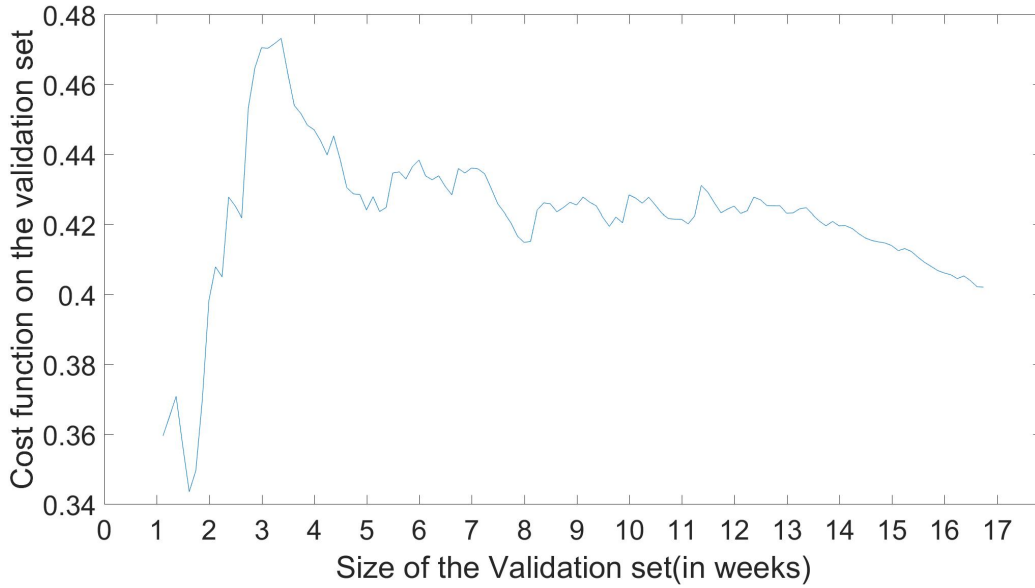


Figure 4.20: For a fixed training set (80% of the data), plot of the cost function computed on validation sets of different size using a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(3, 5, 1)_{168}$ on Room 1

given : a $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$. The resulting autocorrelations are plotted at Figure 4.21 and $V_{val} = 0,3882$. We see that the model structure is really similar to the one chosen previously, but the cost function is higher. It means that the new chosen training and validation sets do not change a lot the model structure but probably give a better estimation of the performance given by the model on a totally new data set. The accuracy metric of eq 4.2 obtained for this model on the validation set is 0.4672.

A part of the obtained normalized predictions are shown on Figure 4.22 for the training set

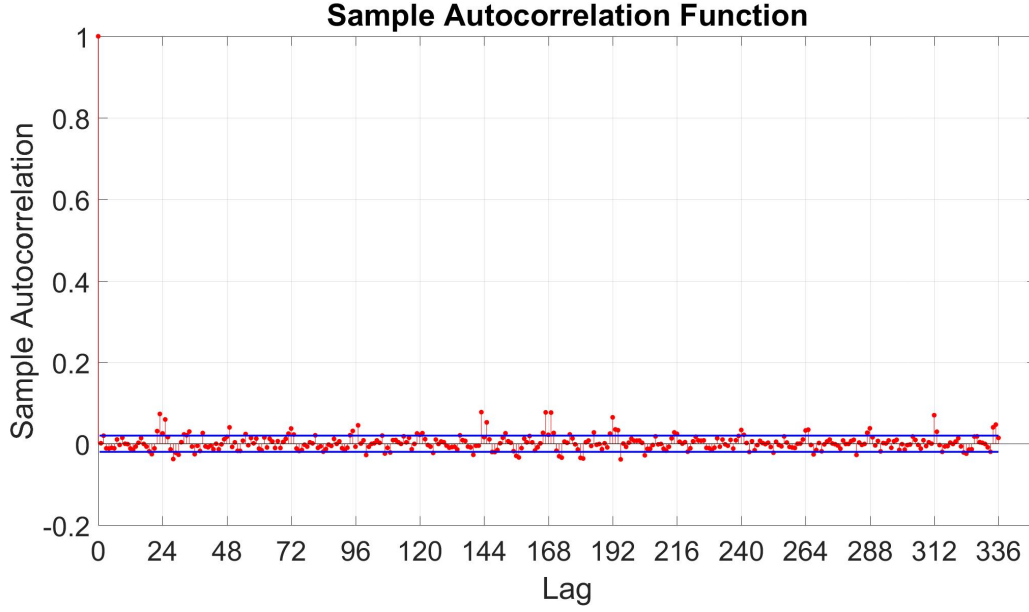


Figure 4.21: Plot of autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $2SARMA(22, 15, 0)(4, 6, 0)_{24}(3, 3, 1)_{168}$ of Room 1

(in magenta) and for the validation set (in red). Clearly these predictions are closed to the true behaviour of the real electrical consumption.

Given that, it can be the case that an integer number is needed to be returned as prediction of the electricity consumption (usually the case for large values of consumptions), so it is also interesting to round the predictions to obtain integer values for the consumption. At this end, once the predictions are computed, they are rounded to the upper integer if their decimals ≥ 5 and to the lower integer otherwise. The predictions of Figure 4.22 are rounded on Figure 4.23.

Using the rounded values, new metrics can also be explored to give an idea of the performance of the model in that case. For this model of Room 1, the percentage of incorrect rounded predictions on the validation set is 45.5%. It means that less than half of the predictions are incorrect, and so that it is difficult to predict exactly the rounded consumption. For these incorrect predictions, the average error committed is 105% of the lowest possible error that can occur (errors of 1 unit). In fact 95.5% of the errors are minimum (1 unit away from the true value). Keep in mind that all the metrics here are not altered by the normalization of the data, i.e. they stay the same on y and $y_{original}$.

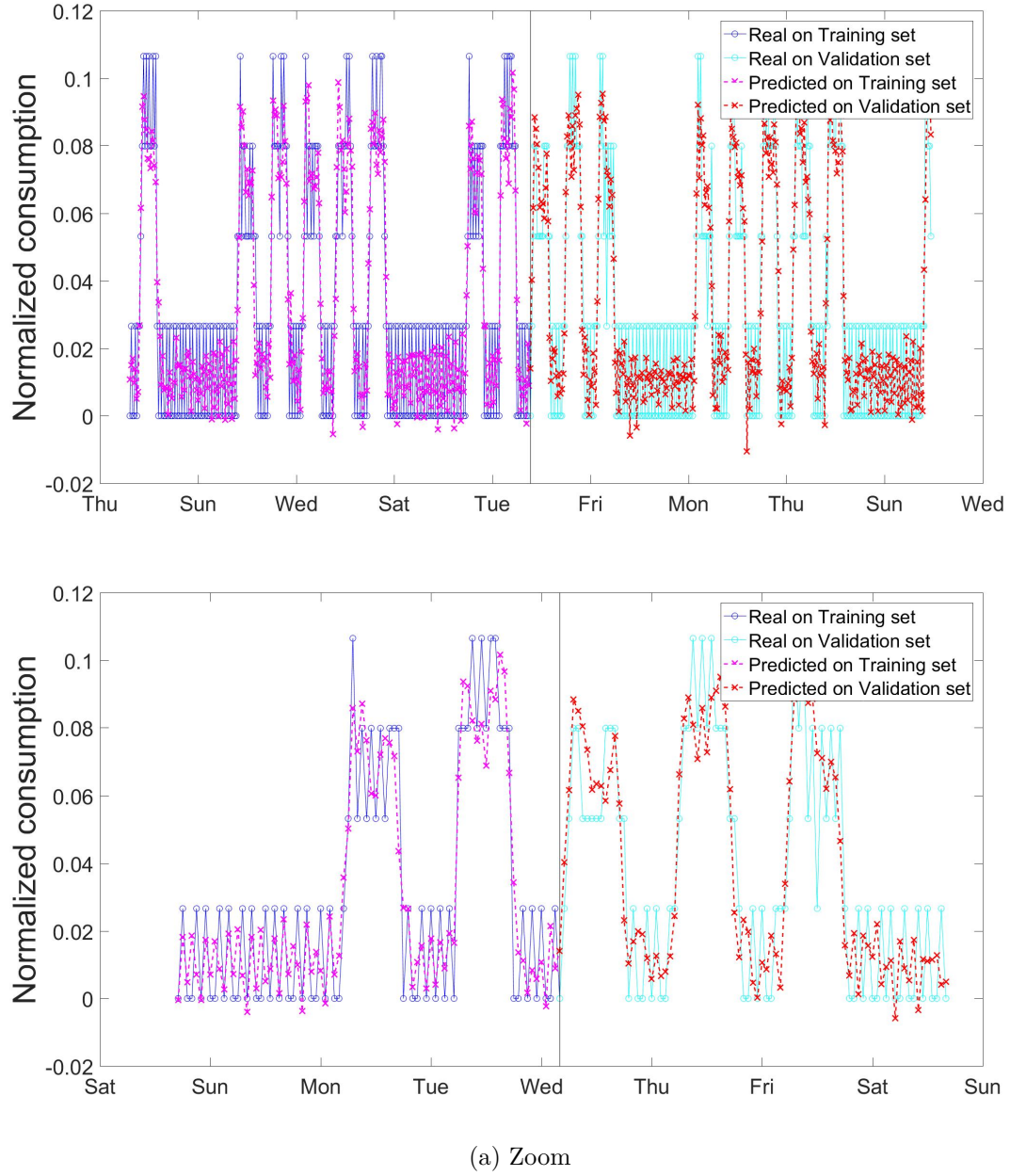
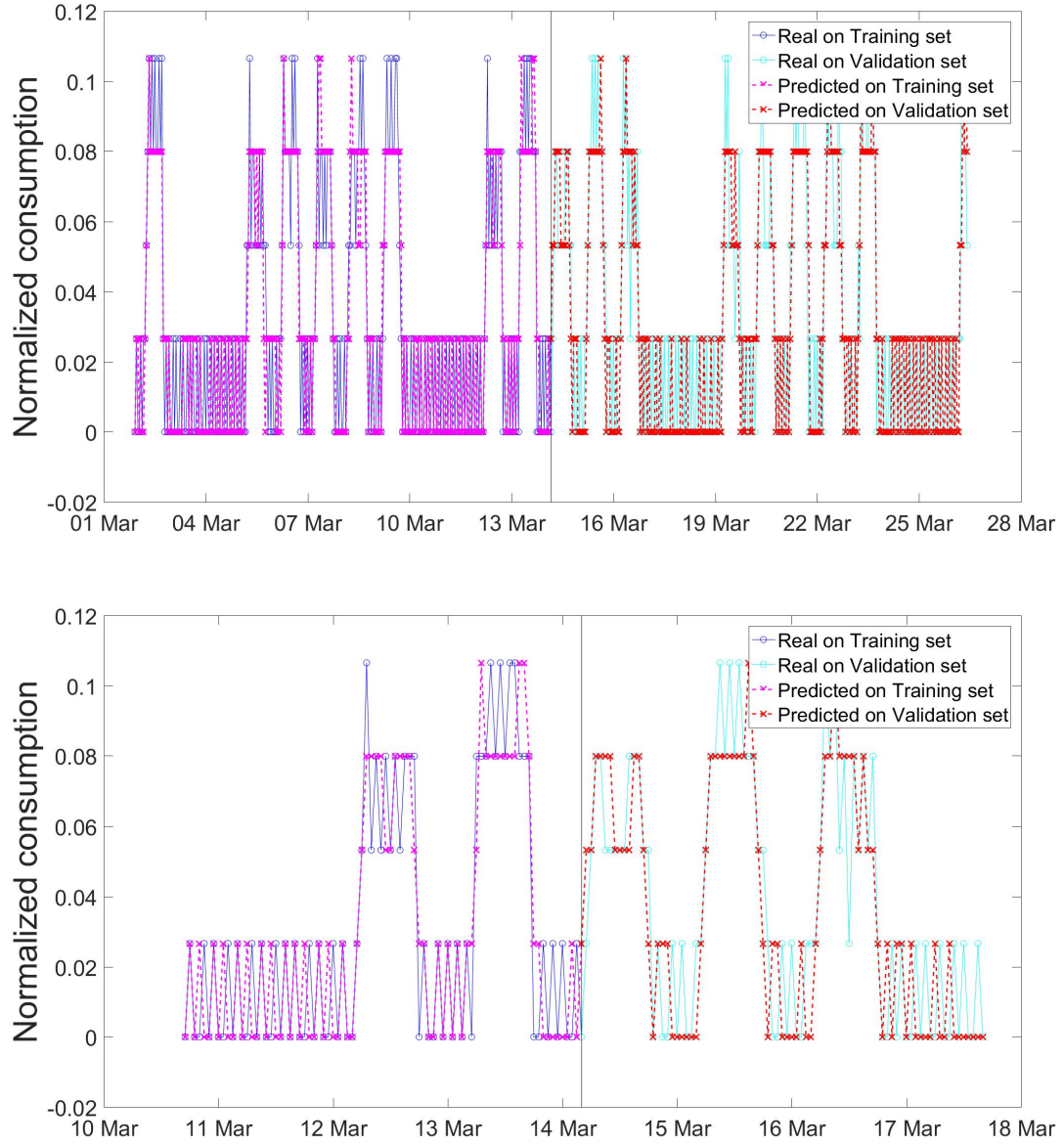


Figure 4.22: Plot of the real and estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$



(a) Zoom

Figure 4.23: Plot of the real and rounded estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$

4.1.9 Multi-step ahead predictions

Now that the final model is chosen and estimated, another interesting thing to check is the ability of the model to provide multi-step ahead predictions. To compute the k -step ahead predictor $\hat{y}_{t|t-k}$, first an estimation of the innovations is done as for the computation of the one-step ahead predictor : $\hat{a}_i = y_i - \hat{y}_{i|i-1}$ for i until $t - k$. $\hat{a}_{t-k+1}, \hat{a}_{t-k+2}, \dots, \hat{a}_t$ cannot be estimated and are thus fixed to zero. After that, an estimation of y_{t-k+1} is computed as the one-step ahead predictor $\hat{y}_{t-k+1|t-k}$. Using this estimation, the 2-step ahead predictor $\hat{y}_{t-k+2|t-k}$ is calculated by following the same principle as for the one-step ahead predictor. Then the 3-step ahead predictor $\hat{y}_{t-k+3|t-k}$ is computed using the estimations $\hat{y}_{t-k+2|t-k}$ and $\hat{y}_{t-k+1|t-k}$ of respectively y_{t-k+2} and y_{t-k+1} . The same methodology is recursively applied until $\hat{y}_{t|t-k}$ is obtained [2, 6].

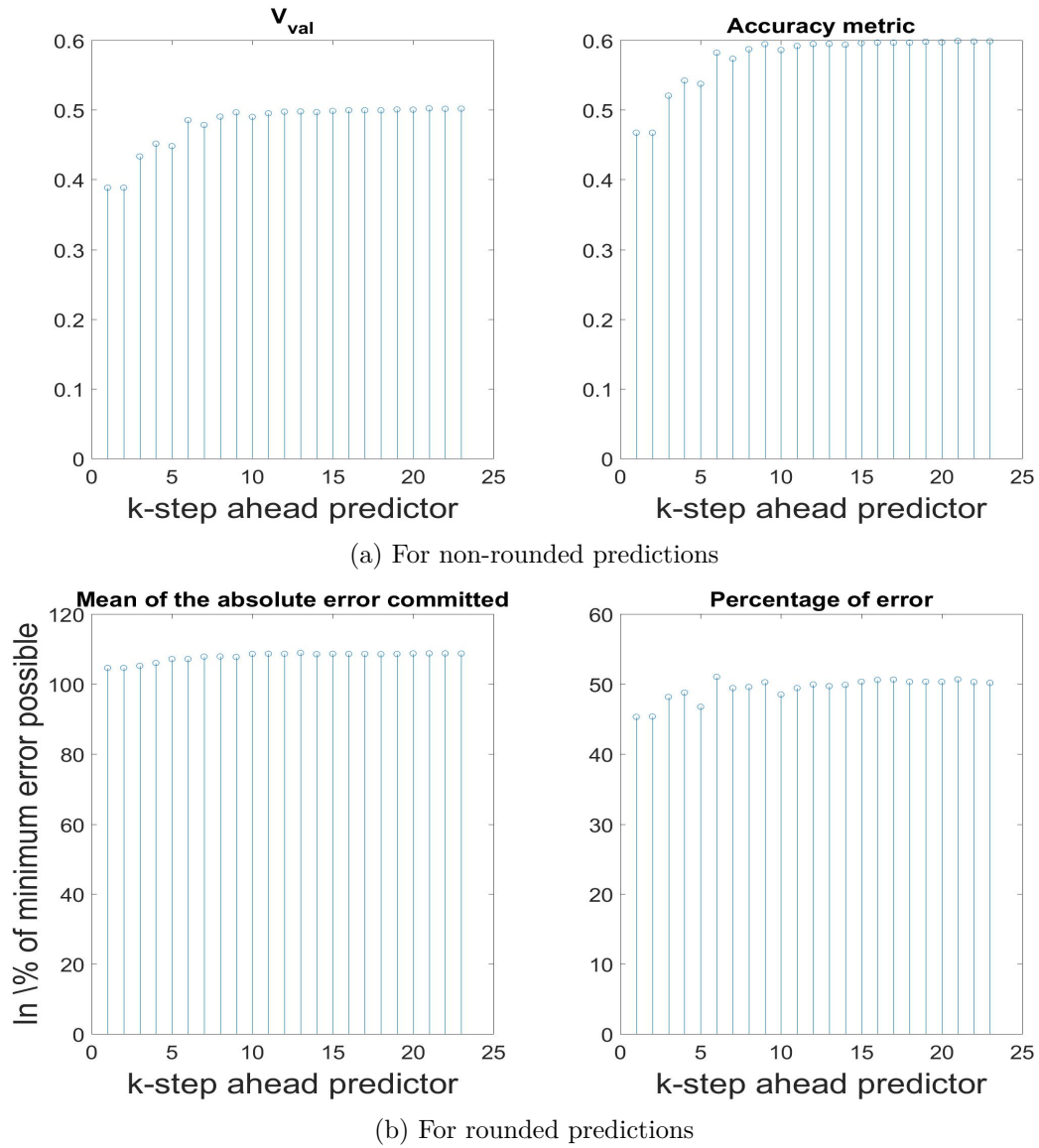


Figure 4.24: Plot of the of the metrics on the validation set for k -step ahead predictors ($k = 1, \dots, 23$) using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$ of Room 1

On Figure 4.24 the predictors from one hour to one day are exposed. The 2-step ahead

predictor presents similar performance as the one-step ahead, for the 4 metrics. It means that the estimated model can predict well both the 2 next hours following the end of the available electricity consumption. Concerning the other predictors, the metrics becomes worse with increasing k , with a larger increase before $k = 10$ and then it seems to tend to a constant or to increase only slightly.

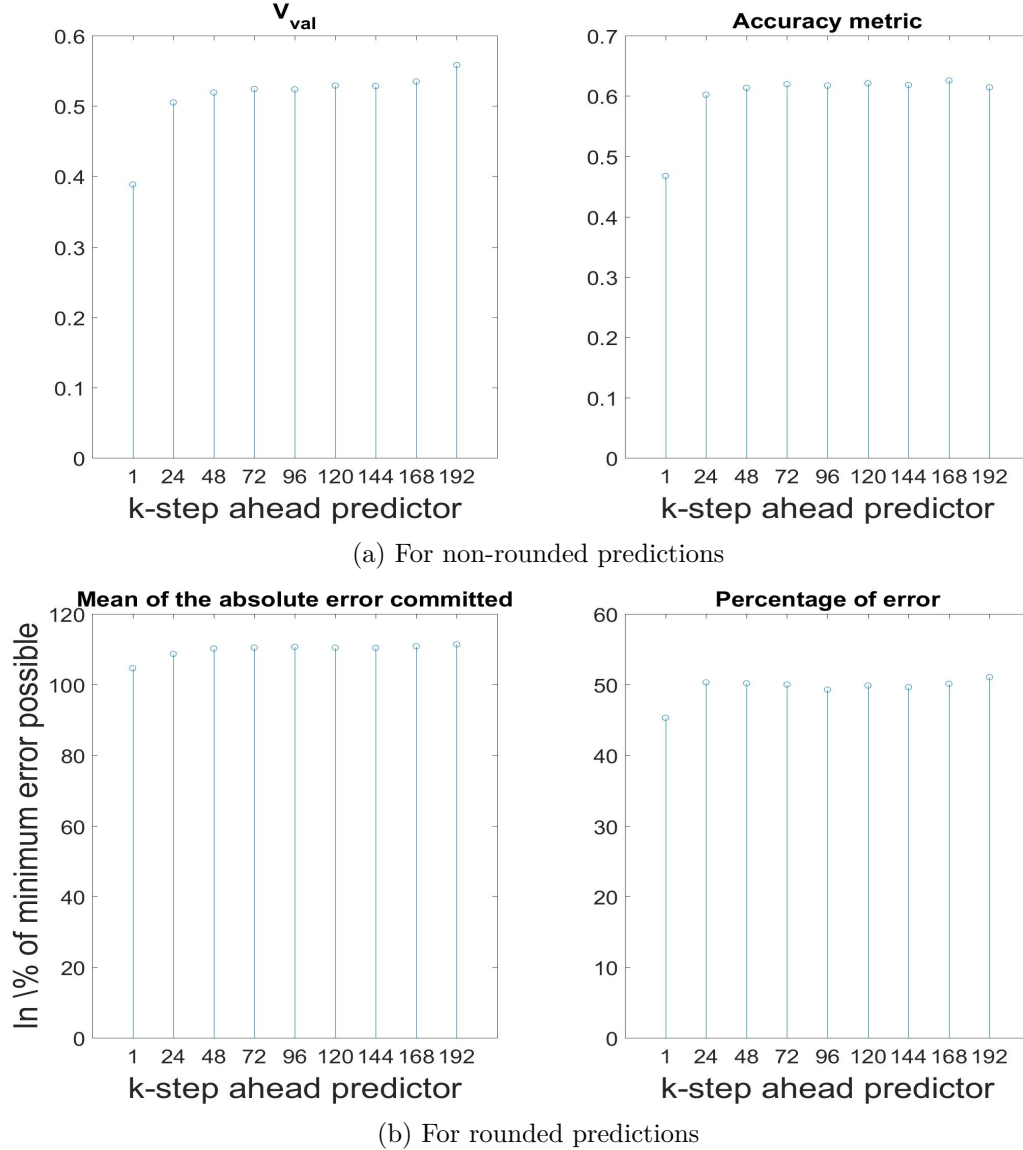


Figure 4.25: Plot of the of the metrics on the validation set for daily k -step ahead predictors ($k = 1, 24, 48, \dots, 192$) using the model $2SARMA(23, 23, 0)(5, 6, 0)_{24}(4, 5, 1)_{168}$ of Room 1

About the prediction of the electricity consumption for several days, the same metrics are plotted on Figure 4.25. They all increase sharply from 1 hour to 1 day, but continue their ascent slower with the number of days increasing, with a little peak for $k = 192$ (except for the accuracy metric).

An important thing to note is the fact that the accuracy metric stays below 1 for all predictors, meaning that even for predictor with high k , our predictions are better than the naive ones. Note

also that from $k = 192$ and on, the naive predictor of the accuracy metric is the consumption 2 weeks ago (like the predictor is more than 1 week away from the last known consumption). This explains the fact that the accuracy metric does not increase for $k = 192$.

4.2 Room 2

The modeling procedure is repeated for another room (Room 2). The detailed selection of the model structure and analysis are not given since it is the same as for Room 1.

For Room 2, the final model chosen is a $2SARMA(23, 20, 0)(5, 5, 0)_{24}(2, 7, 1)_{168}$. The autocorrelation function on the training set residuals and the real and estimated consumption are shown on Figure 4.26 and 4.27, respectively. Similar to the first room, the residuals are not completely i.i.d., but increasing the orders of the model does not improve the autocorrelation function or it does at the detriment of V_{val} . The estimated consumption seems to follow the real consumption, both for the validation and the training sets.

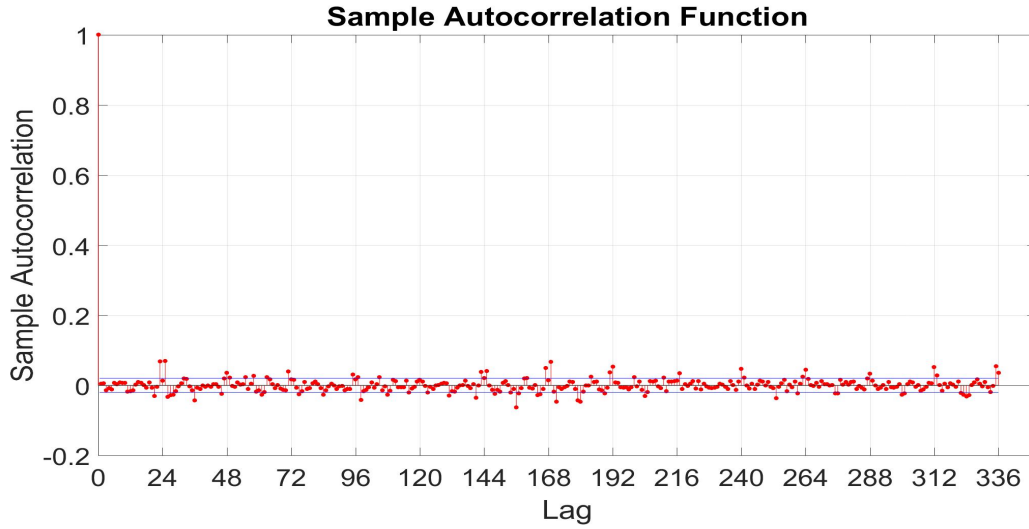


Figure 4.26: Plot of autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $2SARMA(23, 20, 0)(5, 5, 0)_{24}(2, 7, 1)_{168}$ of Room 2

The value of V_{val} obtained with this model is 0.8057, the accuracy metric is 0.2465, the percentage of incorrect rounded predictions is 45, the mean of the absolute error of the rounded incorrect predictions is 126% of the minimum error and the percentage of minimum error among all the errors is 81.

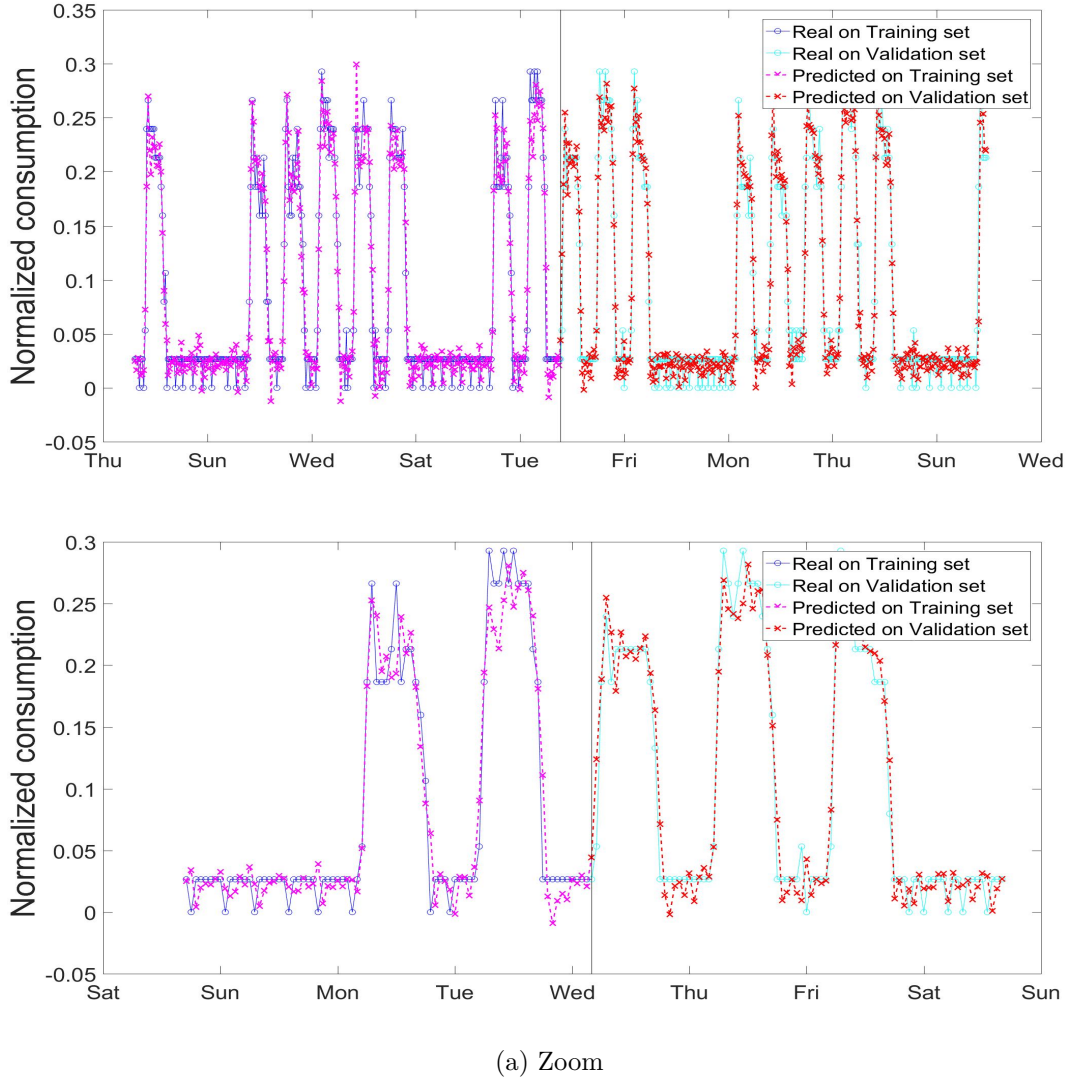


Figure 4.27: Plot of the real and estimated normalized consumption of Room 2 for the training (80%) set and validation (20%) set using the model $2SARMA(23, 20, 0)(5, 5, 0)_{24}(2, 7, 1)_{168}$

4.3 Room 3

For the third choice of single room (Room 3), a $2SARMA(21, 23, 0)(6, 6, 0)_{24}(7, 7, 1)_{168}$ is chosen as final model. The autocorrelation function on the training set residuals and the real and estimated consumption are shown on Figure 4.28 and 4.29, respectively.

The value of V_{val} obtained with this model is 0.8060, the accuracy metric is 0.2627, the percentage of incorrect rounded predictions is 50, the mean of the absolute error of the rounded incorrect predictions is 121% of the minimum error and the percentage of minimum error among all the errors is 83.5.

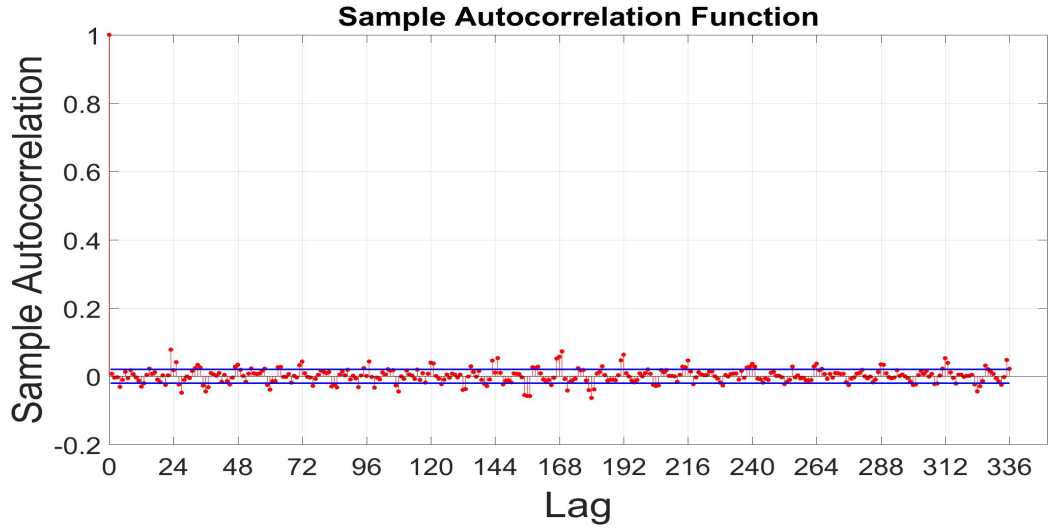
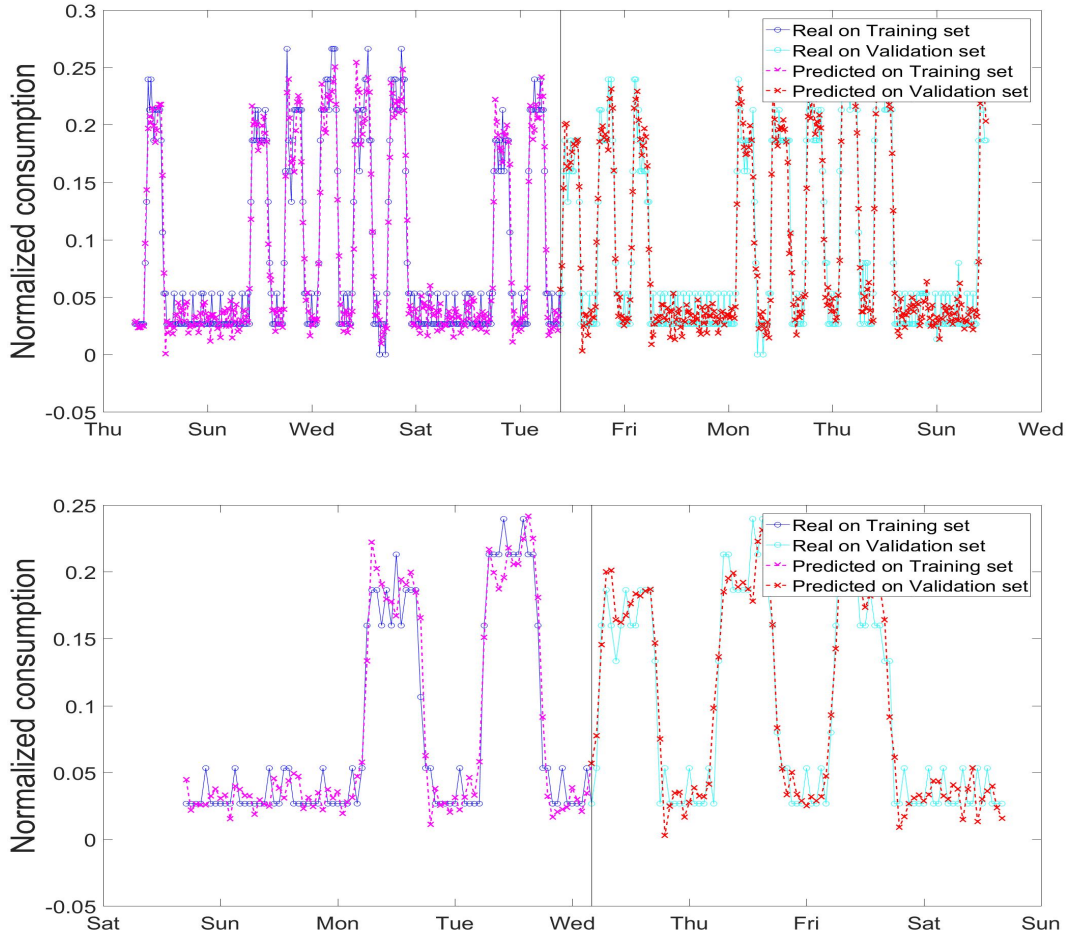


Figure 4.28: Plot of autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $2SARMA(21, 23, 0)(6, 6, 0)_{24}(7, 7, 1)_{168}$ of Room 3



(a) Zoom

Figure 4.29: Plot of the real and estimated normalized consumption of Room 3 for the training (80%) set and validation (20%) set using the model $2SARMA(21, 23, 0)(6, 6, 0)_{24}(7, 7, 1)_{168}$

4.4 Multivariate: Room 1 + another room

The first multivariate model contains Room 1 and another room located at the same floor. The final model chosen is a $V2SARMA(22, 20, 0)(6, 6, 0)_{24}(4, 7, 1)_{168}$. The autocorrelation function of the training set residuals and the real and estimated normalized consumptions are shown on Figure 4.30 and 4.31, respectively, for Room 1.

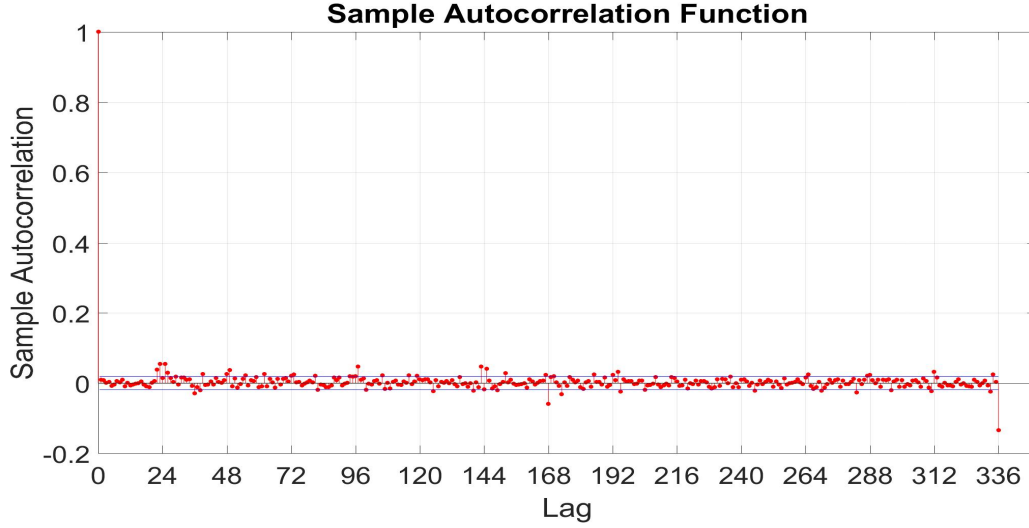
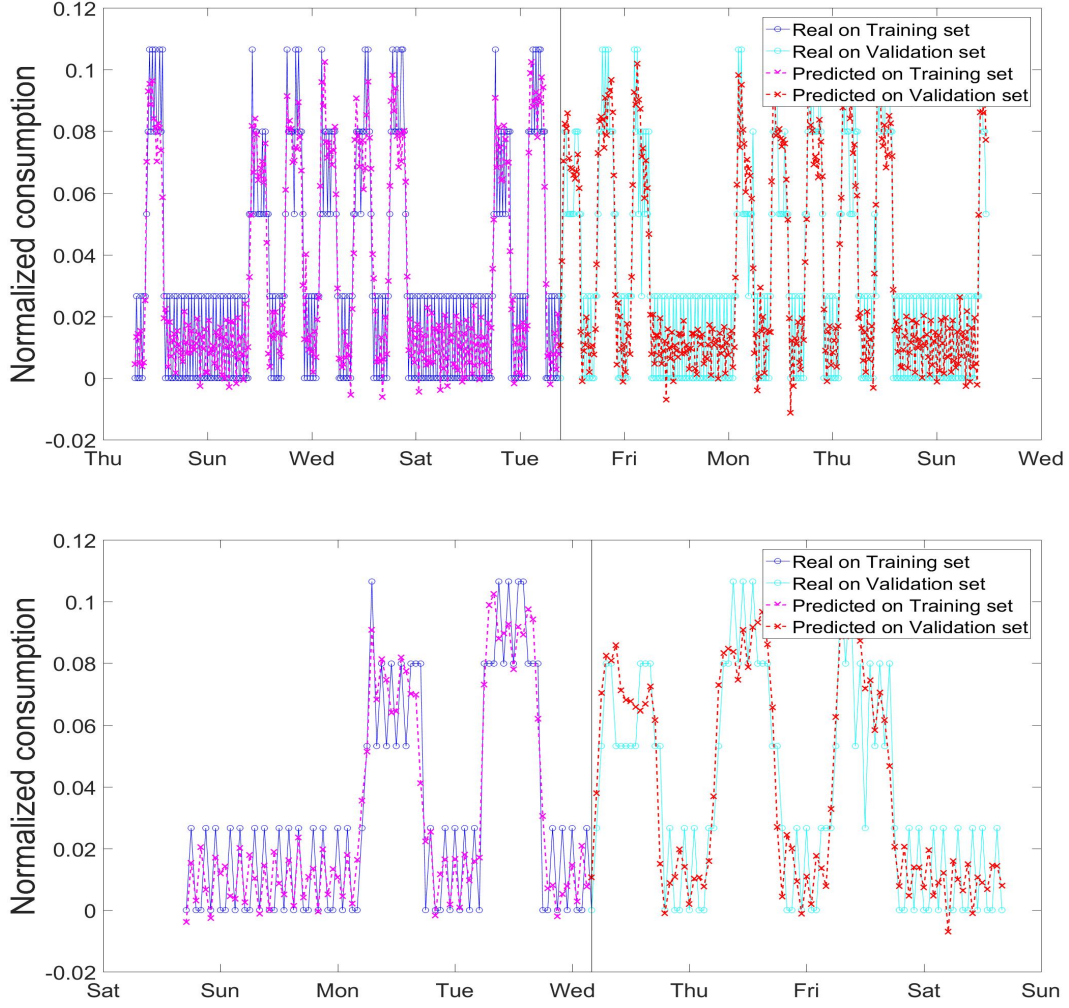


Figure 4.30: Plot of Room 1 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(22, 20, 0)(6, 6, 0)_{24}(4, 7, 1)_{168}$ with 2 variables

The model with the 2 variables gives $V_{val} = 0.3424$, a percentage of incorrect rounded predictions of 46, a mean of the absolute error of the rounded incorrect predictions is 110% of the minimum error and the percentage of minimum error among all the errors of 92. The accuracy metric is computed on the predictions of the Room 1, and its value is 0.4294.



(a) Zoom

Figure 4.31: Plot of the real and estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using the model $V2SARMA(22, 20, 0)(6, 6, 0)_{24}(4, 7, 1)_{168}$ with 2 variables

4.5 Multivariate: Room 2 + Room 3

The second multivariate model contains Room 2 and Room 3, which are on the same floor. The final model chosen is a $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$. The autocorrelation function of the training set residuals and the real and estimated normalized consumption are shown on Figure 4.32 and 4.33, respectively, for Room 2.

This model with the 2 variables gives $V_{val} = 0.5528$, a percentage of incorrect rounded predictions of 45, a mean of the absolute error of the rounded incorrect predictions is 122% of the minimum error and the percentage of minimum error among all the errors of 83. The accuracy metric is computed on the predictions of the Room 2 and Room 3 independently, and are 0.2383 and 0.2385, respectively.

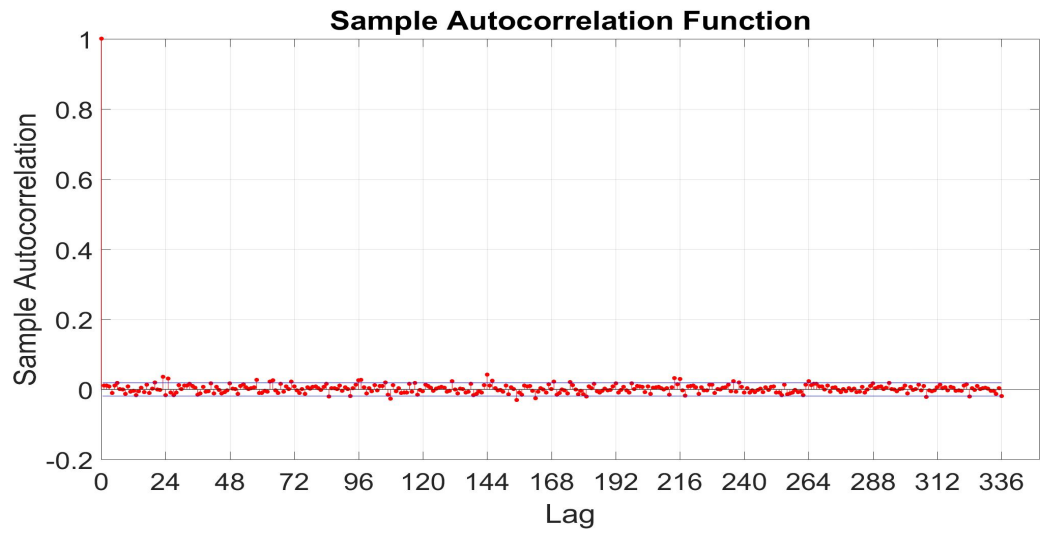
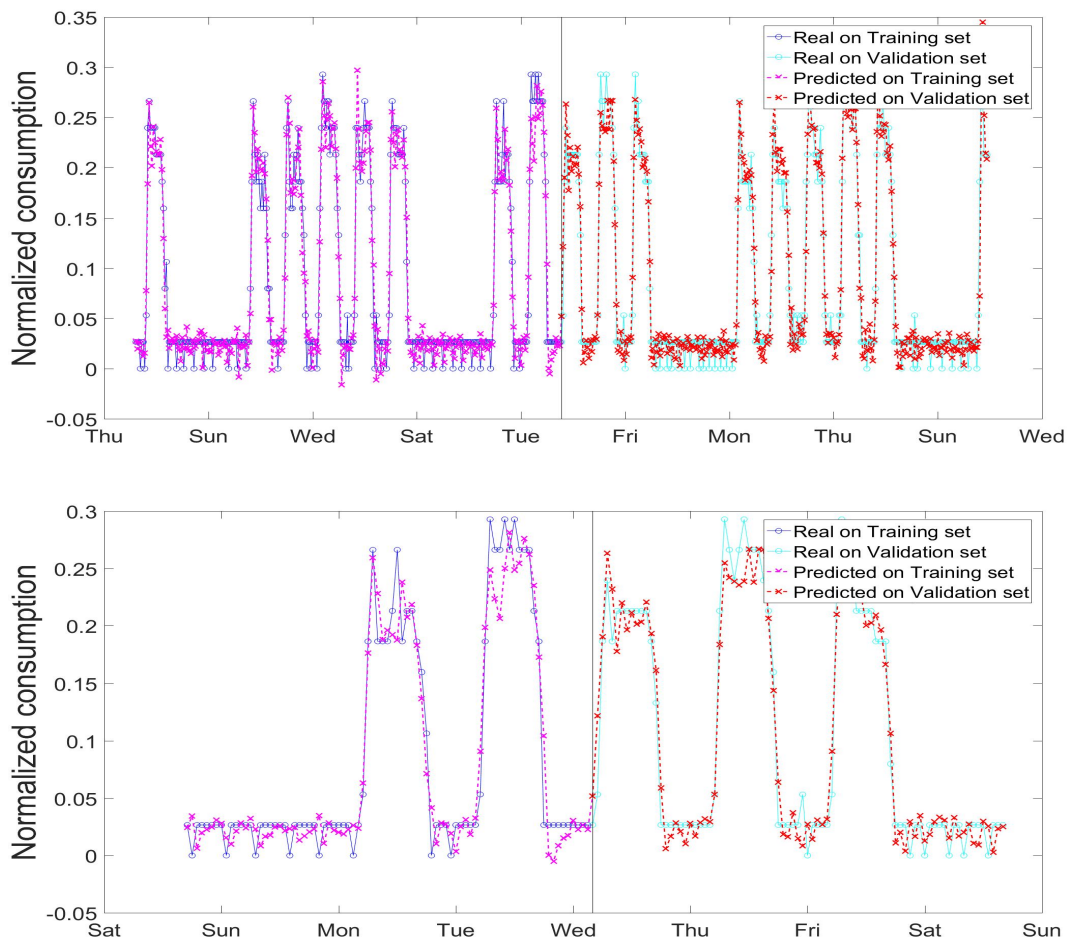


Figure 4.32: Plot of Room 2 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 2 variables



(a) Zoom

Figure 4.33: Plot of the real and estimated normalized consumption of Room 2 for the training (80%) set and validation (20%) set using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 2 variables

4.6 Multivariate: Room 1 + two other rooms

To test multivariate models with 3 variables, a model is fitted for Room 1 and two other rooms with one at the same floor as Room 1. The final model chosen to describe this multivariate process is a $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$. The autocorrelation function of the training set residuals and the real and estimated normalized consumptions are shown on Figure 4.34 and 4.35, respectively, for Room 1.

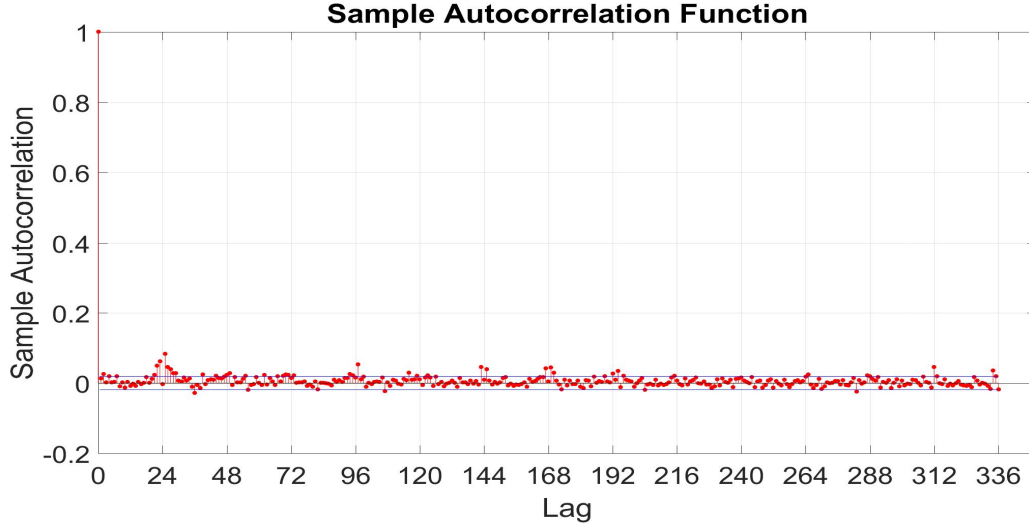
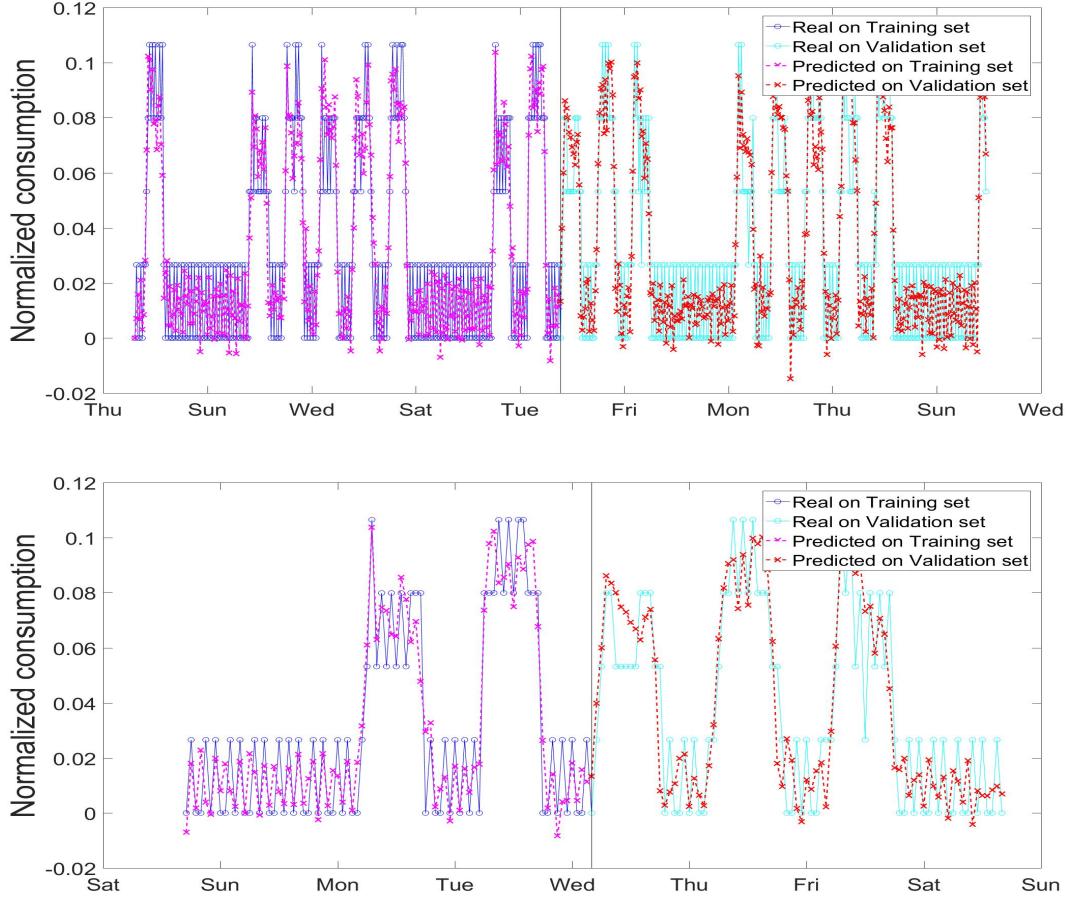


Figure 4.34: Plot of Room 1 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 3 variables

The model with the 3 variables gives $V_{val} = 0.3320$, a percentage of incorrect rounded predictions of 48, a mean of the absolute error of the rounded incorrect predictions is 113% of the minimum error and the percentage of minimum error among all the errors of 89.5. The accuracy metric is computed on the predictions of the Room 1, and its value is 0.4331.



(a) Zoom

Figure 4.35: Plot of the real and estimated normalized consumption of Room 1 for the training (80%) set and validation (20%) set using the model $V2SARMA(23, 23, 0)(6, 6, 0)_{24}(3, 5, 1)_{168}$ with 3 variables

4.7 Multivariate: Room 2 + Room 3 + another room

The last application is also a 3-variable model, using Room 2 and Room 3 (which are at the same floor) plus another room located at a different floor (this room was also include in the previous 3-variate model). The final model chosen to describe this multivariate process is a $V2SARMA(23, 22, 0)(6, 4, 0)_{24}(3, 3, 1)_{168}$. The autocorrelation function of the training set residuals and the real and estimated normalized consumptions are shown on Figure 4.36 and 4.37, respectively, for Room 2.

This model with the 3 variables gives $V_{val} = 0.4569$, a percentage of incorrect rounded predictions of 49.5, a mean of the absolute error of the rounded incorrect predictions is 119% of the minimum error and the percentage of minimum error among all the errors of 85. The accuracy metric is computed on the predictions of the Room 2 and Room 3 independently, and are 0.2330 and 0.2409, respectively.

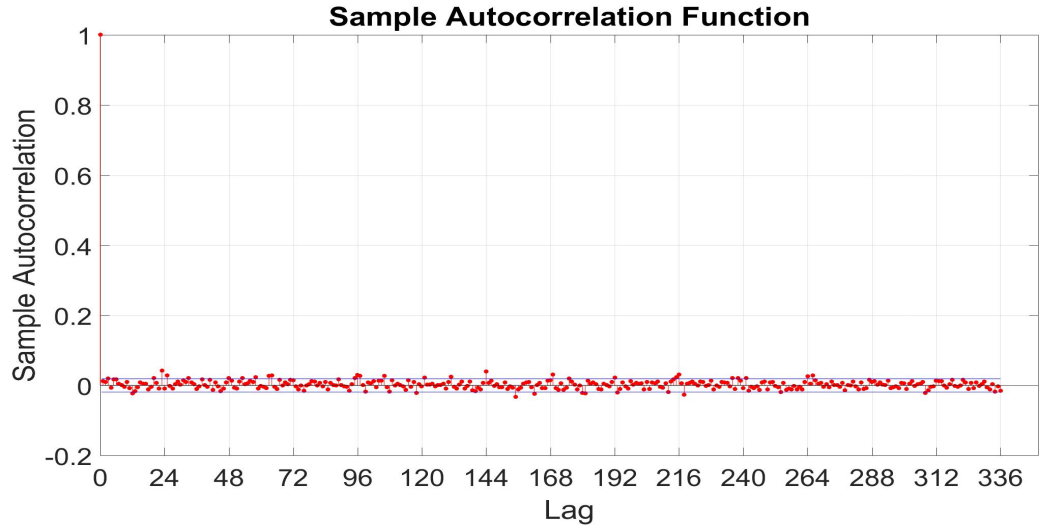
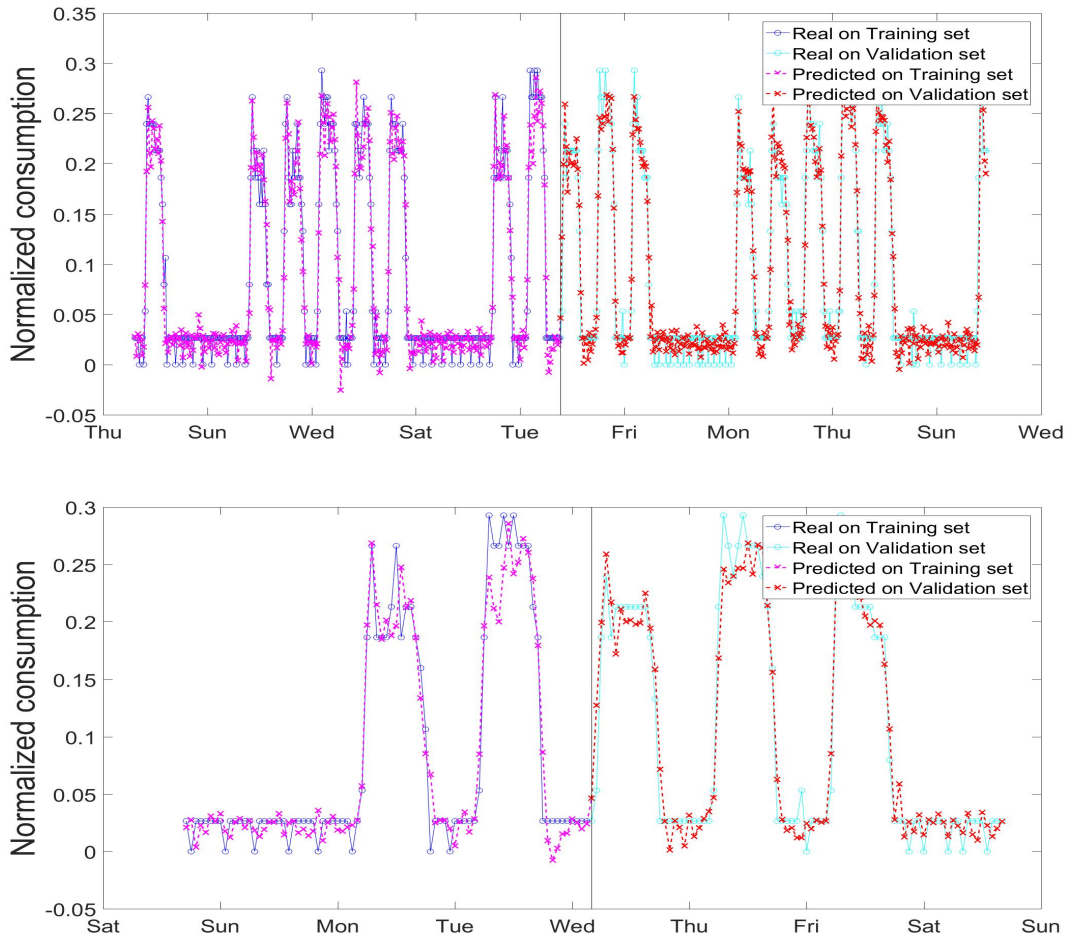


Figure 4.36: Plot of Room 2 autocorrelation function (computed on the 80% training set) of the residuals obtained using the model $V2SARMA(23, 22, 0)(6, 4, 0)_{24}(3, 3, 1)_{168}$ with 3 variables



(a) Zoom

Figure 4.37: Plot of the real and estimated normalized consumption of Room 2 for the training (80%) set and validation (20%) set using the model $V2SARMA(23, 22, 0)(6, 4, 0)_{24}(3, 3, 1)_{168}$ with 3 variables

4.8 Comparison of the results

Before ending this chapter, it is interesting to compare the different estimated models presented so far. This comparison is done using the independent performance of a particular room in the models. It means that the multivariate models, the metrics computed on the same room in the different models are compared. It is important to clarify that all the metrics presented here show a better model for prediction when they are lower.

Table 4.1 summarizes, for the three single rooms studied, cost function (V_{val}) (eq 4.1) and accuracy metric (eq 4.2) obtained using the different estimated models of each process studied in this chapter. The accuracy is considered "good" when it is below 1, which is the case for each room and model here. For the cost function, it is more difficult to give a bound. Keep in mind that the cost function in this table is the cost function of the room independently of the other rooms included in the same model.

		Model used for estimation		
		Univariate	2 variables	3 variables
Room 1	V_{val}	0.3882	0.3567	0.3594
	Accuracy metric	0.4672	0.4294	0.4332
Room 2	V_{val}	0.8057	0.7788	0.7628
	Accuracy metric	0.2477	0.2383	0.2330
Room 3	V_{val}	0.8060	0.7318	0.7375
	Accuracy metric	0.2627	0.2385	0.2409

Table 4.1: Values of the cost function and accuracy metric obtained for the 3 rooms studied, with the different estimated models of Chapter 4

Comparing the performance of the univariate models with the models having 2 variables, we see that it is better in the 2 variable case. It can be explained by the fact that in the 2 variable models, the rooms are located on the same floor. So it seems logical that their consumption are linked in a certain way, and that knowing the consumption of one helps to predict the consumption of the other. However, for the 3 variable models, the metrics are pretty like the same as for the 2 variable ones. This phenomenon can be explained by the fact that the third room is not located on the same floor. Then, taking into account in the model a room located at a different floor does not improve the performance of the model.

Table 4.2 summarizes metrics computed with rounded predictions for each room. These metrics constitute more specific information, providing more about the behaviour of the estimated models. First the accuracy metric (eq 4.2) informs us whether the use of the rounded predictions is interesting, comparing to naive predictions (taking the consumption of the previous week at the same moment). Then the percentage of incorrect predictions gives an idea of the proportion of correct predictions. Since during the nights and the weekends (an important part of the data)

the consumption seems to vary randomly between 2 values, a proportion of incorrect predictions of 50 % or less is considered acceptable. Finally the mean of the errors committed when the predictions are not exact allows to have an idea of the amplitude of the error committed. This metric is expressed in percentage of the minimum error possible. If this value is 100 %, it means that all the errors are minimum (of 1 unit). The more this value increases, the bigger are the errors committed. The exact same observations as in the table 4.1 can be done.

		Model used for estimation		
		Univariate	2 variables	3 variables
Room 1	Accuracy metric	0.6245	0.5627	0.5541
	Proportion of incorrect predictions (in %)	45.5	41.5	40.5
	Mean of error for incorrect predictions	105	104	105
Room 2	Accuracy metric	0.2722	0.2513	0.2513
	Proportion of incorrect predictions (in %)	45	43	44.5
	Mean of error for incorrect predictions	126	125	122
Room 3	Accuracy metric	0.2846	0.2600	0.2619
	Proportion of incorrect predictions (in %)	50	46.5	47.5
	Mean of error for incorrect predictions	121	120	119

Table 4.2: Values of metrics with rounded predictions for the 3 rooms studied, with the different estimated models of Chapter 4

4.9 Conclusion

In this chapter, successful application of the model structure selection as well as the extended HR model estimation method presented in Chapter 3 have been presented for a real life process. For one of them, a complete description of the method for choosing the model structure (data analysis, diagnostics, robustness with regard to the length of data,...) was described, allowing to have a clear idea of how the model orders are chosen.

The final results for univariate and multivariate models indicate that the behaviour of the underlying electricity consumption processes can be efficiently predicted, while different validation indices are provided to measure the accuracy of the predictions. The autocorrelation function of the residuals resembles in all cases the one of a white noise realization, while further improvement might be possible with a change in the model structure to make it more complex.

The comparison between models shows that rooms on the same floor can be used in the same model to improve the independent performance of each room. Nevertheless, if the rooms are located on different floors, it is possible that the performance of each room using the multivariate model decreases.

Chapter 5

Conclusions and future work

Nowadays, time-series prediction is a large field with increasing number of applications. In order to predict a time-series, there exists a large number of possible models to describe real life processes. For these models, there exists a large number of methods to compute estimations of them.

The contributions of this thesis are the following :

- The Hannan-Rissanen method was extended for estimation of single and double seasonal ARMA models with one or multiple variables. This extension uses in the first step the seasonality behaviour of the model to compute better innovation estimates than with the classical HR method.
- An identification procedure of the internal parameter orders (sufficiently large orders of intermediate AR model of step 1) of the extended HR method was also proposed. This procedure modify the internal parameter orders in function of the orders of the estimated model and the size of data.
- A statistical analysis of the method was done, from which we conclude that the HR method is more accurate (in the sense it is less biased and the variance decreases) with the increase of the number of available data to train the model. It was also found that the variance linked to the innovation does not influence the statistical properties of the parameter estimates.
- A model structure selection method was described and its performance was verified on simulation-based studies. Its principle is to scan different values of model parameter orders to evaluate the cost function on a validation set (V_{val}). The orders such that V_{val} is minimized are chosen for the model structure of the model.

- Applications of the extended HR method and the model structure selection were applied on real time-series coming from the measurement of the electricity consumption in one or multiple sites. The final consumption estimations gave sufficiently good results looking at the accuracy metric and the rounded prediction metrics.

One objective of this thesis is to open the way for time-series prediction using the (extended) HR method. At this purpose, a suggestion of possible future works can be made :

- Incorporating exogenous inputs in the models described in Chapter 2 and adapting the extended HR method to incorporate these inputs in the model estimation. For example, the weather condition could influence the electricity consumption and then a model taking this external inputs could give better results for the predictions.
- Adapting the extended HR method to work with missing data could improve the performance of the method.
- A description of other method for the identification of the internal parameters of the first step of the extended HR method ($p_{large}, P_{large}, \dots$). For example, using an information criterion (as Aikake, Bayesian, Deviance,...) for the selection of a model structure for the AR model of step 1 could result on a better estimation of the innovation at this step and then better parameter estimates (in term of MSE) after the second step. Another improvement could be to adapt the third step of the HR method [11] for prediction of seasonal models.
- Using the parameter estimates obtained with extended HR as initial values for computation of ML parameter estimates. As explained before, methods using the likelihood function need to solve a non-linear optimization problem, so initial values are needed. The ML methods could then be used complementary to the HR method to improve the accuracy of the parameter estimates and of the predictions.

Bibliography

- [1] Gianluca Bontempi. Machine learning strategies for time series prediction. *European Business Intelligence Summer School, Hammamet, Lecture*, 2013.
- [2] George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*. Prentice-Hall, Inc., 1994.
- [3] Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.
- [4] Sofiane Brahim-Belhouari and Amine Bermak. Gaussian process for nonstationary time series prediction. *Computational Statistics & Data Analysis*, 47(4):705–712, 2004.
- [5] José Ramón Cancelo, Antoni Espasa, and Rosmarie Grafe. Forecasting the electricity load from one day to one week ahead for the spanish system operator. *International Journal of forecasting*, 24(4):588–602, 2008.
- [6] Haibin Cheng, Pang-Ning Tan, Jing Gao, and Jerry Scripps. Multistep-ahead time series prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 765–774. Springer, 2006.
- [7] Fong-Lin Chu. Forecasting tourism demand with arma-based methods. *Tourism Management*, 30(5):740–751, 2009.
- [8] Johann Du Preez and Stephen F Witt. Univariate versus multivariate time series forecasting: an application to international tourism demand. *International Journal of Forecasting*, 19(3): 435–451, 2003.
- [9] Jean-Marie Dufour and Denis Pelletier. Practical methods for modelling weak varma processes: Identification, estimation and specification with a macroeconomic application. Technical report, Technical report, Département de sciences économiques and CIREQ, Université . . . , 2005.
- [10] A Girard, CE Rasmussen, and R Murray-Smith. Gaussian process priors with uncertainty inputs: multiple-step-ahead prediction. *Technical Report TR-2002-119, Dept. of Computer Science*, 2002.

- [11] Víctor Gómez. *Multivariate time series with linear state space structure*. Springer, 2016.
- [12] Victor Gómez. Linear time series with matlab and octave. 2018.
- [13] Victor Gómez et al. Ssmmatlab, a set of matlab programs for the statistical analysis of state-space models. *Online at <http://www.sepg.pap.minhap.gob.es/sitios/spgg/en-GB/Presupuestos/Documentacion/Paginas/SSMMATLAB.aspx>*, 2012.
- [14] Edward J Hannan and Jorma Rissanen. Recursive estimation of mixed autoregressive-moving average order. *Biometrika*, 69(1):81–94, 1982.
- [15] EJ Hannan and L Kavalieris. Regression, autoregression models. *Journal of Time Series Analysis*, 7(1):27–49, 1986.
- [16] Fumio Hayashi. Econometrics princeton university press. *Princeton University Press*, page 49, 2000.
- [17] Simon S Haykin et al. *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall,, 2009.
- [18] Zan Huang and Dennis KJ Lin. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, 21(2):286–303, 2009.
- [19] Yanting Li, Yan Su, and Lianjie Shu. An armax model for forecasting the power output of a grid connected photovoltaic system. *Renewable Energy*, 66:78–89, 2014.
- [20] Guy Mélard, Roch Roy, and Abdessamad Saidi. Exact maximum likelihood estimation of structured or unit root multivariate time series models. *Computational statistics & data analysis*, 50(11):2958–2986, 2006.
- [21] Konstantinos Metaxoglou and Aaron Smith. Maximum likelihood estimation of varma models using a state-space em algorithm. *Journal of Time Series Analysis*, 28(5):666–685, 2007.
- [22] Norizan Mohamed, Maizah Hura Ahmad, Zuhaimy Ismail, et al. Double seasonal arima model for forecasting load demand. *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics*, 26:217–231, 2010.
- [23] Andrew Ng. Neural networks and deep learning-coursera, 2017.
- [24] Anthony Papavasiliou. *Coupling renewable energy supply with deferrable demand*. PhD thesis, UC Berkeley, 2011.
- [25] M Hashem Pesaran and Richard J Smith. A generalized r^2 criterion for regression models estimated by the instrumental variables method. *Econometrica (1986-1998)*, 62(3):705, 1994.

- [26] Robert R Richardson, Michael A Osborne, and David A Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017.
- [27] Gilbert Thomas Walker. On periodicity in series of related terms. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 131(818):518–532, 1931.
- [28] George Udny Yule. Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer’s sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226(636-646):267–298, 1927.

