

LAPPEENRANTA UNIVERSITY OF TECHNOLOGY  
Faculty of Technology  
Department of Mathematics and Physics

# MCMC Analysis Of Classical Time Series Algorithms.

The topic of this Master's thesis was approved by the departmental council of the Department of Mathematics and Physics on 15<sup>th</sup> July, 2009.

The examiners of the thesis were Professor Heikki Haario and PhD Tuomo Kauranne.  
The thesis was supervised by Professor Heikki Haario.

In Lappeenranta November 18, 2009

Isambi Sailon  
Punkkerikatu 2 A 6  
53850 Lappeenranta  
Phone: +358466443208  
Isambi.Sailon@lut.fi

## **Abstract**

Lappeenranta University of Technology  
Department of Mathematics and Physics

Isambi Sailon

### **MCMC Analysis Of Classical Time Series Algorithms**

Master's thesis

2009

66 pages, 47 figures, 11 tables

Key words: Time series, Autoregressive Moving Average Model (ARMA), Markov Chain Monte Carlo (MCMC), Reversible Jump Markov Chain Monte Carlo (RJMCMC)

Identification of order of an Autoregressive Moving Average Model (ARMA) by the usual graphical method is subjective. Hence, there is a need of developing a technique to identify the order without employing the graphical investigation of series autocorrelations. To avoid subjectivity, this thesis focuses on determining the order of the Autoregressive Moving Average Model using Reversible Jump Markov Chain Monte Carlo (RJMCMC). The RJMCMC selects the model from a set of the models suggested by better fitting, standard deviation errors and the frequency of accepted data.

Together with deep analysis of the classical Box-Jenkins modeling methodology the integration with MCMC algorithms has been focused through parameter estimation and model fitting of ARMA models. This helps to verify how well the MCMC algorithms can treat the ARMA models, by comparing the results with graphical method. It has been seen that the MCMC produced better results than the classical time series approach.

# Acknowledgements

First of all, I would like to express my gratitude to the Department of Mathematics for the scholarship during my studies, otherwise my studies would not be possible.

My sincere thanks go to Professor, Ph.D. Heikki Haario for close and continuous supervision and support. I also thank Ph.D. Tuomo Kauranne, for all his assistance rendered to me during my studies.

I am acknowledging and extending my heartfelt gratitude to Matylda Jabłońska for her contributions. She gave constructive ideas, comments, and directions on how to do this work. Without her, I could have not completed this work.

As for my friends and colleagues at the Department of Mathematics, I am greatly indebted. In particular I want to thank my friend Saunders Oliver for his material support since my Bachelor studies to date.

My sincere acknowledgments are addressed to my parents (Mr and Mrs Sailon Mbala-wata) for their love.

'Asanteni Sana'

Lappeenranta; December 18, 2010

Isambi Sailon

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objective of the Thesis . . . . .	1
1.2	Structure of the thesis . . . . .	2
<b>2</b>	<b>Theoretical background</b>	<b>3</b>
2.1	Time series . . . . .	3
2.2	Box-Jenkins Models . . . . .	4
2.2.1	Autoregressive Model . . . . .	4
2.2.2	Moving Average Model . . . . .	5
2.2.3	Autoregressive Moving Average Model . . . . .	6
2.3	Autocorrelation and Partial Autocorrelation . . . . .	6
2.3.1	Correlation, Covariance and Autocovariance . . . . .	6
2.3.2	Autocorrelation . . . . .	7
2.3.3	Partial Autocorrelation . . . . .	8
2.4	Residual Analysis . . . . .	9
2.5	Model Identification and Model Estimation . . . . .	11
2.5.1	Pre-analysis of data . . . . .	11
2.5.2	Model Identification . . . . .	17
2.5.3	Model Estimation . . . . .	18
<b>3</b>	<b>Theoretical background for MCMC</b>	<b>20</b>
3.1	Bayesian approach . . . . .	20
3.2	The Metropolis-Hastings Algorithm . . . . .	22
3.3	The Gibbs Sampler . . . . .	24
3.4	Adaptive MCMC Algorithms . . . . .	25

3.5	Reversible Jump MCMC . . . . .	26
<b>4</b>	<b>Parameter Estimation</b>	<b>30</b>
4.1	Series generation . . . . .	30
4.2	Box-Jenkins recursive estimation . . . . .	31
4.3	Matlab <i>garchfit</i> estimation . . . . .	33
4.4	Yule-Walker Equations . . . . .	34
4.5	Observation . . . . .	35
4.6	Parameter Estimation by MCMC . . . . .	36
<b>5</b>	<b>ARMA Model Fitting, Forecasting and Identification of <math>p</math> and <math>q</math> by RJMCMC</b>	<b>39</b>
5.1	Model identification for given examples . . . . .	39
5.2	Model Fitting by Classical Time Series . . . . .	41
5.3	Model fitting by MCMC . . . . .	43
5.4	Forecasting . . . . .	46
5.5	Forecasting Errors . . . . .	50
5.6	Forecasts by MCMC . . . . .	51
5.7	ARMA Model Identification by Reversible Jump MCMC . . . . .	57
<b>6</b>	<b>Real Data: Consumption of Electricity in Finland</b>	<b>59</b>
6.1	Decomposition of Data . . . . .	59
6.2	MCMC and Reconstruction . . . . .	61
<b>7</b>	<b>Conclusion</b>	<b>63</b>
	<b>References</b>	<b>65</b>

## List of Tables

1	Theoretical characteristics of ACF and PACF for basic ARMA models . . . . .	18
2	Average performance of different estimation methods. . . . .	36
3	Original and estimated parameters for ARMA(3,0). . . . .	40
4	Parameter estimates for chosen possible models for Series Y. . . . .	41
5	Measuring forecasting error. . . . .	50
6	Measuring forecasting error; the case with <i>garchsim.m</i> . . . . .	50
7	The Correlation Between Parameters of original ARMA(3,0). . . . .	52
8	The Correlation Between Parameters of ARMA(3,0). . . . .	53
9	The Correlation Between Parameters of ARMA(2,1). . . . .	54
10	The Correlation Between Parameters of ARMA(2,3). . . . .	56
11	The Correlation Between Parameters of ARMA(2,5). . . . .	57

## List of Figures

1	Residuals for ARMA(2,0).	10
2	Histogram of Residuals for ARMA(2,0).	11
3	Posterior probabilities.	21
4	Likelihood, Prior and Posterior.	21
5	Histogram for $\theta$ and $s$ .	23
6	Last 500 simulations of the chain.	24
7	ARMA(1,0).	30
8	ARMA(2,1).	31
9	Normalized histogram of $\theta$ estimates ARMA(1,0).	32
10	Normalized histogram of $\theta_1, \theta_2, \psi$ estimates ARMA(2,1).	32
11	Normalized histogram of $\theta$ estimates obtained from Matlab garchfit.m estimation (original $\theta = 0.7$ ).	33
12	Normalized histogram of $\theta_1, \theta_2, \psi$ estimates ARMA(2,1).	34
13	Normalized histogram of $\theta$ estimates obtained from MCMC estimation (original $\theta = 0.7$ ).	37
14	Normalized histogram of $\theta_1, \theta_2, \psi$ estimates ARMA(2,1) by MCMC.	38
15	Sample ACF and PACF for series X.	39
16	Sample ACF and PACF for series Y.	40
17	Original and Estimated data set.	42
18	Original and Estimated Data Set With different White Noise.	42
19	MCMC fitting of $ARMA(2,1)$ .	43
20	MCMC fitting of $ARMA(2,1)$ with different noises.	44
21	MCMC fitting of $ARMA(3,0)$ without new noises.	45
22	MCMC fitting of $ARMA(3,0)$ with new noises.	46
23	Original data and Forecasts of possible model of series X.	46

24	Original data and Forecasts of ARMA(2,3). . . . .	47
25	Original data and Forecasts of ARMA(2,5). . . . .	47
26	Original data and Forecasts of ARMA(2,1). . . . .	48
27	ARMA(3,0): Forecasts and the Original data. . . . .	48
28	ARMA(2,3): Forecasts for Sample = 100. . . . .	49
29	ARMA(2,5): Forecasts for Sample = 100. . . . .	49
30	Original ARMA(3,0): Scatter Plots for series X. . . . .	51
31	Original ARMA(3,0): Forecast Predictive Distribution for series X. . . . .	52
32	ARMA(3,0): Scatter Plots for series X. . . . .	52
33	ARMA(3,0): Forecast Predictive Distribution for series X. . . . .	53
34	ARMA(2,1): Scatter Plots for series Y. . . . .	54
35	ARMA(2,1): Forecast Predictive Distribution for series Y. . . . .	54
36	ARMA(2,3): Scatter Plots for series Y. . . . .	55
37	ARMA(2,3): Forecast Predictive Distribution for series Y. . . . .	56
38	ARMA(2,5): Scatter Plots for series Y. . . . .	56
39	ARMA(2,5): Forecast Predictive Distribution for series Y. . . . .	57
40	RJMCMC: fitting of the suggested models. . . . .	58
41	Finland Electricity consumption. . . . .	59
42	Weekly Electricity consumption and Trend. . . . .	60
43	Residuals after extracting Trends and Seasonality. . . . .	60
44	Autocorrelation and Partial Autocorrelation Functions of Residuals. . . . .	61
45	Histogram of the parameter by MCMC. . . . .	62
46	Residuals and Estimations by MCMC. . . . .	62
47	Data (blue)and MCMC spread (yellow) due to parameters variability. . . . .	63

# 1 Introduction

Model identification, estimation and forecasting have great importance in determining how good a model is and how well it performs. Models exist in different forms, they can be deterministic or stochastic. Box-Jenkins models are stochastic models as they contain white noises. In selecting the order of Box-Jenkins models, there is a need of accurate method. This applies to estimating the parameters. There have been different classical time series methods in identifying and estimating the Box-Jenkins models. An alternative for the classical time series methods can be the Markov chain Monte Carlo (MCMC) approach. These can be used both to identify the degree of the model and to estimate the parameters.

## 1.1 Objective of the Thesis

The objective of the thesis is to combine the modern MCMC approach with classical time series algorithms. The autoregressive moving average models (ARMA) are treated by classical time series algorithms and then followed by MCMC approach. The results are compared. Treatment, this is applied to stages that are employed in building the model. These stages are identification, estimation and forecasting of the models.

The identification part is said to be a difficult task as it involves graphical approach, hence it is subjective ([1]). We suggest to select the models by autocorrelation and partial autocorrelation functions. Suggestions are assumed to include a correct model. Suggesting the models from the graphs is accompanied by the use of behaviors of the models; for instance one may choose ARMA(2,1) because in the partial autocorrelation function, the graph vanishes after one lag while in the autocorrelation the graph vanish towards zero. Following previous studies on *AR* ([22]) and ARIMA ([21]) model selection uncertainty, we use the Reversible Jump MCMC (RJMCMC) to select the model from the set of suggested models ([18]). This is aimed for reducing some part of human subjectivity in model selection.

The estimation stage depends on the method used. The classical time series methods and MCMC methods are expected to be used such that the results estimated should be as close to the true values as possible. The task is to see whether the MCMC algorithms would estimate parameters as precisely as the classical time series methods would estimate them. This is accompanied by fitting the model. Apart from estimation stage, the forecasting stage is expected to be affected by the existence of white noises as well as the errors due to estimated parameters.

## 1.2 Structure of the thesis

To attain the objective of this thesis, the next section contains the theoretical background for time series. Several issues are addressed; these include Time Series definition, Box-Jenkins Models, Autocorrelation and Partial Autocorrelation, Residual Analysis, and Model Identification and Model Estimation.

As the thesis integrates the MCMC algorithms and classical time series, Section 3 deals with theoretical part of MCMC methods. It covers Bayesian approach, the Metropolis-Hastings Algorithm, the Gibbs Sampler, Adaptive MCMC Algorithms and Reversible Jump MCMC. In a subsection of RJMCMC, there is a derivation of the probability of accepting the new values in parameter chains.

The practical part of the thesis starts in Section 4 where it covers estimation of parameters using classical time series and MCMC methods. In classical time series, the recursive formulas, GARCH Toolbox and Yule-Walker equations are discussed, followed by simulations using MATLAB software. The results are compared with the MCMC outcome.

Model fitting and forecasting are in Section 5. It covers fitting by classical way and MCMC, forecasting and forecasting errors. The effects of white noises are addressed in this section, as well as their impact in fitting the model. Section 6 tells about the subjectivity of the model and RJMCMC. This section may be considered as the main part as it solves the issue of subjectivity.

Data used in sections of this thesis are generated directly from MATLAB software. However, Section 7 uses the real data of Finnish electricity consumption. The section deals with decomposition and reconstruction of data as the data have two seasonality types. The nature of graphs before decomposition, after decomposition and after reconstruction are discussed and seen in this section. The thesis ends with conclusion in Section 8.

## 2 Theoretical background

### 2.1 Time series

A time series is a sequence of observations based on a regular timely basis, *e.g.* hourly, daily, monthly, annually, *etc.* It is a set of observations generated sequentially in time. The special features of a time series are that the data are ordered with respect to time, and that successive observations are usually expected to be dependent ([7]). The order of an observation is denoted by a subscript  $t$ . We denote by  $x_t$  the  $t$ th observation of a time series while the preceding observation is  $x_{t-1}$ , and the next observation is  $x_{t+1}$ . ([7])

Time series are applicable in different fields. Examples of fields are economics (monthly employment figures), sociology (crime figures), meteorology (rainfall, temperature, wind speed), medicine (electrocardiograms and electroencephalograms), vibrating physical systems (such as the rise of a car traveling over the surface), seismology, oceanography, geomorphology, astronomy (star brightness, solar activity), and many others.

There are two types of time series; namely *Continuous* time series and *Discrete* time series. If the set of observation is continuous, the time series is said to be continuous. If the set of observation is discrete, the time series is said to be discrete. Continuous time series is the series whose measurements are taken at every moment of time; thus it exists at every point in time. An example of this series is the temperature in a given place.

A discrete time series is the series of observations where the measurements are taken at predetermined, and equally spaced time intervals (hourly, daily, monthly, or quarterly data). Discrete time series may arise in several ways. For instance, by sampling a continuous time series, that is, given a continuous time series, it is possible to construct a discrete time series by taking measurements at equally spaced intervals of time. The series formed is called *Instantaneously* recorded; example of the instantaneous series are daily temperature readings. Moreover, discrete time series may arise by accumulating or aggregating a realization for a predetermined time interval and forms a type called *accumulated* series; examples of such series are monthly rainfall, quarterly industrial production, daily miles traveled by an airline, or monthly traffic fatalities ([7]).

If the future values of the time series are determined by some mathematical function then the time series is said to be deterministic. An example of a mathematical function can be  $x_t = \cos(2\pi ft)$ .

If the future values that are described only in terms of a probability distribution, the time series is said to be non-deterministic (or statistical time series). A statistical phenomenon evolving in time according to probabilistic laws is called a stochastic process. In analyzing

a time series we regard it as a realization of a stochastic process. The observed time series is an actual realization of an underlying stochastic process. By realization, which is a path or trajectory, we refer to the sequence of observed data points, and not just a single point.

## 2.2 Box-Jenkins Models

There are many methods and approaches for formulating models. George Box and Gwilym Jenkins in their 1970 book ([1]) described Autoregressive Integrated Moving Average (ARIMA) models. Box-Jenkins models are mathematical models used typically for accurate short-term forecasts of 'well-behaved' data (that shows predictable repetitive cycles and patterns) and find the best fit of a time series to past values of this time series, in order to make forecasts. Box-Jenkins models require at least a moderately long time series for an effective fitting, and generally include autoregressive, moving average, and seasonal moving average terms, as well as difference and seasonal difference operators.

### 2.2.1 Autoregressive Model

The simplest Box-Jenkins model is the autoregressive model. The idea of an Autoregressive model is to use the past values of a time series as independent variables in predicting future values, that is, autoregressive model is the model where the current value of a variable  $x_t$  depends upon only the values that the variable took in previous periods plus an error term.

The autoregressive model is denoted by  $AR(p)$  where  $r$  is the order of the model. The order of model refers to the maximum time lag used, not the maximum power of a variable as in regression analysis. This order is the number of parameters that need to be estimated. The  $p^{th}$  order autoregressive model  $AR(p)$  is

$$x_t = C + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + u_t \quad (1)$$

Variable  $x_t$  is the series under investigation and  $C$  is a constant. For simplicity we can omit the constant term.  $\phi_1, \dots, \phi_p$  are autoregressive parameters that describe the effect of unit change in two consecutive time series observations (Example:  $\phi_1$  is the autoregressive parameter which describes the effect of a unit change in  $x_{t-1}$  on  $x_t$ ) and which needs to be estimated. The  $u_t$  are the random shocks or errors, or white noise disturbance terms (white noise series that reflects only purely random error) assumed to be normally and independently distributed with mean zero, constant variance ( $u_t \sim N(0, \sigma^2)$  and no significant autocorrelation).

The equation (1) can be written more compactly as

$$x_t = C + \sum_{i=1}^p \phi_i x_{t-i} + u_t$$

Using the lag operator, the equation (1) can be written as:

$$x_t = C + \sum_{i=1}^p \phi_i L^i x_t + u_t$$

or

$$\phi(L)x_t = C + u_t,$$

$$\text{where } \phi(L) = (1 - \sum_{i=1}^p \phi_i L^i).$$

### 2.2.2 Moving Average Model

An alternative model to an  $AR(p)$  model would be to consider past errors to see if they can improve on the time series representation of the data. The resulting model is called a Moving Average model. It is a model which is the linear combination of white noises, where  $x_t$  depends on the current and previous values of a white noise disturbance term. Thus the current value of the series  $x_t$  is expressed as a linear function of the current and previous errors or shocks. As with an autoregressive process, these random shocks (white noises) in a moving average process are assumed to be normally and independently distributed with mean zero and constant variance. The  $q^{th}$  order moving average model  $MA(q)$  is

$$x_t = C + \psi_1 u_{t-1} + \psi_2 u_{t-2} + \dots + \psi_q u_{t-q} + u_t \quad (2)$$

Again  $C$  is a constant, which can be omitted and  $x_t$  is the time series observation,  $\psi_1, \dots, \psi_q$  are moving average parameters, and  $u_t$  are the random shocks or errors or white noise disturbance terms. In sigma notation, the equation (2) is written as

$$x_t = C + \sum_{i=1}^q \psi_i u_{t-i} + u_t$$

In lag operator form, it is written as

$$x_t = C + \sum_{i=1}^q \psi_i L^i u_t + u_t$$

or

$$x_t = C + \psi(L) u_t,$$

$$\text{where } \psi(L) = 1 + \sum_{i=1}^q \psi_i L^i.$$

The *AR* and *MA* models differences can be seen by considering the effect of lagged  $u_t$  on the current  $x_t$  values. In a pure *MA* model, the effect of a shock persists for a specified number of time periods, then disappears suddenly whereas in a pure *AR* model the effect of shock declines gradually.

### 2.2.3 Autoregressive Moving Average Model

One may need a high-order model with many parameters to describe the dynamic structure of the data. To obtain a parsimonious parametrization, it is necessary to include both AR and MA terms in the model. The combination, of  $AR(p)$  and  $MA(q)$ , is called Autoregressive Moving Average model, abbreviated to  $ARMA(p, q)$  where  $p$  refers to the number of autoregressive parameters, and the  $q$  to the number of moving average parameters.

The  $ARMA(p, q)$  model is the forecasting model or process in which both  $AR(p)$  and  $MA(q)$  are applied or combined to a well-behaved time series data. The  $ARMA$  states that the current value of a given series  $x_t$  depends linearly on its own previous values plus a combination of current and previous values of a white noise error term.

The general  $ARMA(p, q)$  model equation is the combination of equations (1) and (2); thus the equation is

$$x_t = C + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \psi_1 u_{t-1} + \psi_2 u_{t-2} + \dots + \psi_q u_{t-q} + u_t \quad (3)$$

$C$  is a constant,  $\psi_1, \dots, \psi_n$  are moving average parameters,  $x_t$  is the series under investigation,  $\phi_1, \dots, \phi_p$  are autoregressive parameters and  $u_t$  are the random shocks (or errors or white noise disturbance term); if a model depicts the  $ARMA$  process governing the series, then the errors of the model should be white noise ([7]). If the lags  $p$  and  $q$  are large, very complex models can result. The usual strategy is to find a simple model with reasonably small  $p$  and  $q$ , which fits the data adequately, without over-fitting though.

## 2.3 Autocorrelation and Partial Autocorrelation

### 2.3.1 Correlation, Covariance and Autocovariance

The correlation between two random variables  $X$  and  $Y$  indicates the existence of strength and direction of variables,  $X$  and  $Y$ , and it is defined as

$$\rho_{xy} = \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sigma_x \sigma_y}, \quad (4)$$

where  $E$  is the expectation,  $\mu_x$  and  $\mu_y$  are the mean values of variables  $X$  and  $Y$  respectively,  $\sigma_x$  and  $\sigma_y$  are the standard deviations of variable  $X$  and  $Y$  respectively.

The covariance is a measure of how much two variables change together and it can be positive or negative. A positive covariance occurs when both variables are above or below their respective mean while for negative covariance, it occurs when one variable is above its mean while, at the same time, the other variable is below its mean. The covariance between two variables,  $X$  and  $Y$ , defined by the expectation, is determined by

$$\text{cov}(X, Y) = E[(X - \mu_x)(Y - \mu_y)] \quad (5)$$

The autocovariance is simply the covariance of the signal against a time-shifted version of itself ([8]). With time series  $x$ , autocovariance determines how  $x$  is related to its previous values, and for stationary series autocovariances depend only on the difference between  $t_1$  and  $t_2$ , so that  $\text{cov}(x_t, x_{t-i}) = \text{cov}(x_{t-k}, x_{t-k+1})$  ([2]). The covariance between  $x_t$  and its value  $x_{t+k}$ , separated by  $k$  intervals of time, is the autocovariance at lag  $k$  and is defined by ([1] )

$$\gamma_k = \text{cov}[x_t, x_{t+k}] \quad (6)$$

The asymptotic covariance matrix for the estimated  $AR$  parameters is given by

$$C = \frac{\sigma^2 R^{-1}}{N},$$

where  $\sigma^2$  is the variance of the white noise,  $N$  is the number of observations and

$$R = \begin{bmatrix} r(0) & r(1) & \cdots & r(p-1) \\ r(1) & r(0) & \cdots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{bmatrix} \quad (7)$$

is the autocovariance matrix, where  $r(k) = \gamma_k$ . An extended covariance matrix includes the estimation of the residual variance, giving

$$C = \begin{bmatrix} \frac{\sigma^2}{N} R^{-1} & 0 \\ 0 & 2\frac{\sigma^4}{N} \end{bmatrix}$$

For  $MA$ , the covariance matrix for estimated parameters is

$$C = \begin{bmatrix} \frac{\sigma^2}{N} R_{zz}^{-1} & 0 \\ 0 & 2\frac{\sigma^4}{N} \end{bmatrix},$$

where  $R_{zz}$  denotes the autocovariance matrix which looks like  $R_p$ .

### 2.3.2 Autocorrelation

A certain type of correlation concept that portrays the dependence of two consecutive observations of time series is called *autocorrelation*. The autocorrelations are statistical

measures that indicate how a time series is related to itself over time. The autocorrelation describes the correlation between the process at different points in time. It represents a degree of similarity between a given time series and a lagged version of itself over successive time intervals. It is the same as calculating the correlation between two different time series, except that the same time series is used twice – once in its original form and once lagged by one or more time periods. Thus autocorrelation is the correlation coefficient, however, instead of correlation between two different variables, the correlation is between two values of the same variable at time period. The autocorrelation is given by

$$r_k = \frac{E[(x_t - \mu)(x_{t+k} - \mu)]}{\sigma_x^2}, \quad (8)$$

where  $k$  is the specified lag number,  $x_t$  are series observations,  $\mu$  is the series mean value and  $\sigma_x^2$  is the variance. The symbol of autocorrelation is  $r_k$  if it is referred to estimated autocorrelation at lag  $k$  while for the theoretical autocorrelation at lag  $k$ , the used symbol is  $\rho_k$ . The autocorrelation, also, can be defined in terms of autocovariance, that is, autocorrelation is a normalized autocovariance defined as, the autocorrelation at lag  $k$

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (9)$$

The autocorrelation can be used to detect non-randomness in data. Also it can be used to identify an appropriate time series model if the data are not randomly distributed, for which case the autocorrelations are usually plotted for many lags.

The autocorrelation will first test whether adjacent observations are autocorrelated; that is, whether there is significant correlation between observations 1 and 2, 2 and 3, 3 and 4, etc. This is known as lag one autocorrelation, since one of the pair of tested observations lags the other one by one period or sample. Similarly, it will test at other lags. For instance, the autocorrelation at lag four tests whether observations 1 and 5, 2 and 6, ..., 19 and 23, etc. are correlated. Estimates at longer lags have been shown to be statistically unreliable ([1]). In some cases, the effect of autocorrelation at smaller lags will influence the estimate of autocorrelation at longer lags. For instance, a strong lag one autocorrelation would cause observation 5 to influence observation 6, and observation 6 to influence 7. This results in an apparent correlation between observations 5 and 7, even though no direct correlation exists.

### 2.3.3 Partial Autocorrelation

It is possible to remove the intervening correlation between  $x_t$  and  $x_{t+k}$ . The partial autocorrelation removes the effect of shorter lag autocorrelation from the correlation estimate at longer lags. Partial autocorrelation is similar to autocorrelation, except that

when calculating it, the autocorrelation with all the elements within the lag are partialled out. The partial autocorrelation measures the degree of association between two random variables with the effect of a set of controlling random variables removed, that is, it measures the correlation between an observation  $k$  periods ago and current observation, after controlling for the observations at intermediate lags (all lags  $< k$ ), that is, the correlation between  $x_t$  and  $x_{t-k}$ , after removing the effects of  $x_{t-k+1}, x_{t-k+2}, \dots, x_{t-1}$ .

The partial autocorrelation is calculated by

$$r_{kk} = \frac{r_k - r_{k-1}^2}{1 - r_{k-1}^2}, \quad (10)$$

where  $r_k$  is the autocorrelation at lag  $k$ .

Therefore, autocorrelation is the correlation of a series with itself shifted by a particular lag of  $k$  observations, while partial autocorrelation is the correlation of a series with itself shifted by a particular lag of  $k$  observations, and controlling for the correlations for all shifts of 1 through  $k - 1$ . The partial autocorrelation function is a useful tool to identify  $AR(p)$  models while the autocorrelation function is used in identifying  $MA(q)$ . It is difficult to know the population values of autocorrelations and partial autocorrelations of the stochastic process, hence one uses the sample autocorrelations and partial autocorrelations functions to see if they are similar to those of the model ([7]).

## 2.4 Residual Analysis

The residual analysis investigates the quality of the fitted model. Residuals are necessarily correlated with each other even when the true errors are independent([7]). If the model is well specified and the parameter estimates are close to the true values, then the residuals have nearly the properties of white noise. They should behave roughly like independent, identically distributed normal variables with zero means and common standard deviations ([20]). If the trend is adequately modeled, a plot of residuals, Figure 1, for ARMA(2,0), suggests a rectangular scatter with no discernible trends whatsoever.

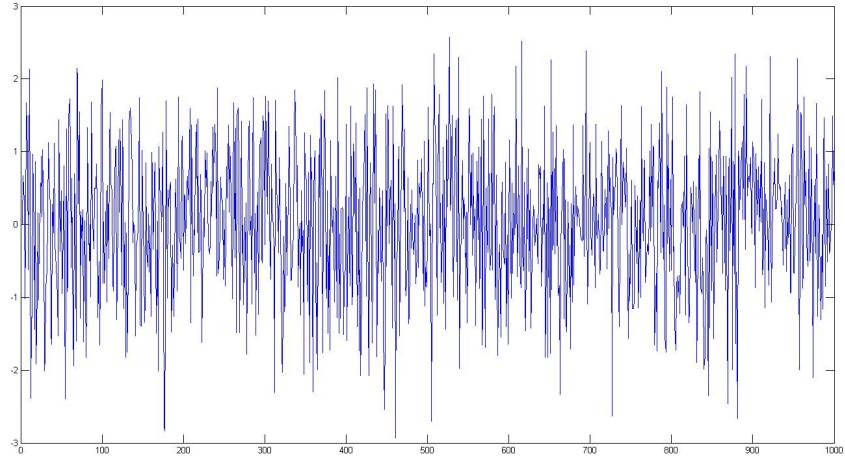


Figure 1: Residuals for ARMA(2,0).

As stated, for ARMA models the residuals behave roughly like independent, normal, random variables with zero mean and standard deviation  $s$  if the parameters estimated are approximately equals to real values;  $s$  is defined as

$$s = \sqrt{\frac{\sum_{t=1}^n (X_t - \hat{X}_t)^2}{n-p}} \quad (11)$$

The study of data patterns can be seen by standardizing the residuals which is done by dividing the residuals with residuals standard deviation,  $s$ . From the equation,  $p$  is the number of parameters while  $n$  is the number of observations. Standardization allows us to see residuals of unusual size easily ([20]).

Similarly, one can study the histogram of residuals which is seen in Figure 2 and the plot is somewhat symmetric and tails off at both the high and low ends, approximately, as a normal distribution does. This is the expected plot since the residuals behave like white noises.

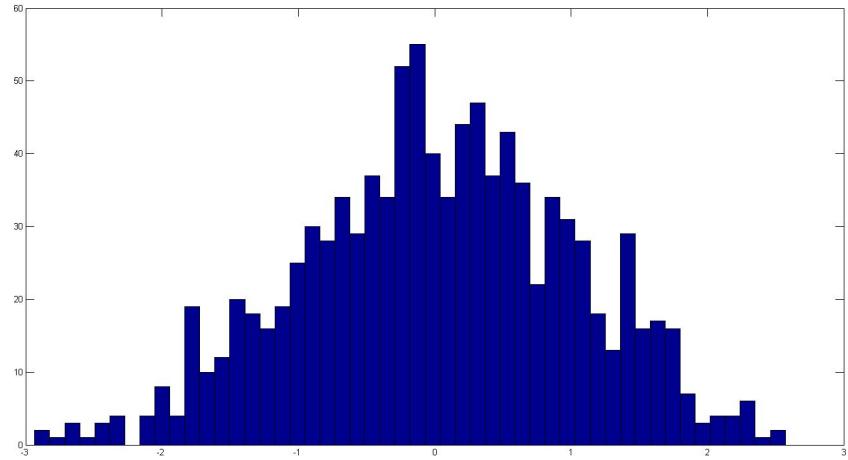


Figure 2: Histogram of Residuals for ARMA(2,0).

## 2.5 Model Identification and Model Estimation

### 2.5.1 Pre-analysis of data

Given a set of observations, we have to start looking at possible stationarity and seasonality. This is a stage where the data are analyzed if they are stationary (testing the stationarity) and then looking for any presence of seasonality. Thus it requires making sure that the variables to be modelled are stationary, and identifying the seasonality. Before looking the stationarity and seasonality, one may check the invertibility and the trend of *ARMA* models.

#### Invertibility

Consider the *MA*(1) defined as

$$x_t = u_t + \psi u_{t-1}$$

This can be written as

$$u_t = x_t - \psi u_{t-1}$$

Replace  $t$  by  $t-1$  gives

$$u_{t-1} = x_{t-1} - \psi u_{t-2}$$

and substituting this to  $u_t = x_t - \psi u_{t-1}$  yield

$$u_t = x_t - \psi(x_{t-1} - \psi u_{t-2}) = x_t - \psi x_{t-1} - \psi^2 u_{t-2}$$

If  $|\psi| < 1$ , the substitution can be done “infinitely” into the past and obtain the expression ([20])

$$u_t = x_t - \psi x_{t-1} - \psi^2 x_{t-2} - \psi^3 x_{t-3} - \dots$$

This expression can, also, be written as

$$x_t = \psi x_{t-1} + \psi^2 x_{t-2} + \psi^3 x_{t-3} + \dots + u_t$$

If  $|\psi| < 1$ , the  $MA(1)$  can be inverted into an infinite -order autoregressive model. Therefore a model is invertible if and only if  $|\psi| < 1$  ([20]); this generalizes that  $MA$  model is said to be invertible if it can be represented as a stationary infinite-order autoregression,  $AR(\infty)$ . If  $|\psi| > 1$ , then the model is non-invertible; however every non-invertible  $MA$  process can be generated by an invertible  $MA$  process. Thus, for  $MA(q)$ , the characteristic equation

$$1 = \psi_1 L + \psi_2 L^2 + \psi_3 L^3 + \dots + \psi_n L^n$$

should have roots that exceed one in modulus so that  $MA(q)$  model is invertible. For a general  $ARMA(p, q)$  model, we require both stationarity and invertibility.

## Trends and Stationarity

A *stationary* process has the property that the mean, variance and autocorrelation structure do not change over time. The basic idea of stationarity is that the probability laws that govern the behavior of the process do not change over time ([20]). It can be assessed by run sequence plot. Run sequence plot is the graph that displays observed data in a time sequence, so it is a graphical data analysis technique for preliminary scanning of the data. Thus a time series is said to be stationary if there is no systematic change in mean (no trend), if there is no systematic change in variance and if strictly periodic variations have been removed. Much of modern theory of time series is concerned with stationary time series, and for this reason time series analysis often requires one to transform a non-stationary series into a stationary one.

Most time series are not stationary, they have trends, cycles and season pattern. A time series may be viewed differently by different analysts. Trend has a tendency of slowing and changing the mean function. We can decompose the time series, that is, the decomposition of time series means de-constructing a time series into notional components ([8]). The decomposition identifies the trend and the seasonal components exhibited by data. Given the deterministic time trend model, the classical least squares method (regression method) can be used to estimate the unknown parameters. The computation of the partial derivatives (these partial derivatives are set to zero) with respect to the unknown parameters can be used to find the solution.

Different techniques can be used to remove trends. The techniques like fitting of parametric curves and spline function are mostly used. Trends may be treated as polynomial trends (examples: linear, quadratic), seasonal means, cosine trends, exponential trend, logistic trend. The linear or quadratic function is applied in the case of monotonic increase or decrease while the higher degree polynomials may be used in other cases.

Removing the trend is preceded by identifying the trend whether it is deterministic or stochastic. If a series has a deterministic trend, then we simply regress the series on an intercept and save the residuals; the residuals are de-trended. For the case of stochasticity, then differencing (regular differencing: a process of computing the difference between every two successive values) can produce an *ARMA* process, which is stationary and estimable.

### Making data series stationary

If the data set given is non-stationary we can make attempts to change the given data series to be stationary to guarantee the assumption that its mean is constant. It can be done by *differencing* transformations to remove the trend in mean. It is called a regular differencing. If the non-stationary series,  $x_t$ , is differenced  $d$  times before it changes to stationary, then it is said to be integrated of order  $d$  and  $d$  is the *order of integration*. The order of integration reports the minimum number of differences required to change the non-stationary series to the stationary series.

The first-order differences of time series values  $x_1, x_2, \dots, x_n$  are given by a new series  $y_1, y_2, \dots, y_{n-1}$ . Where  $y_1 = x_2 - x_1, y_2 = x_3 - x_2, \dots, y_{n-1} = x_n - x_{n-1}$ . The operator  $y_t = x_{t+1} - x_t = \nabla x_t$  is called the first difference and  $\nabla$  is the difference operator. First difference is often enough to convert a time series with a trend into a stationary time series. However, if first difference is not enough to remove the trend in mean then we try second differencing.

For second differencing, we have  $y_t = \nabla^2 x_t = \nabla x_t - \nabla x_{t-1} = x_t - 2x_{t-1} + x_{t-2}$ . Similarly we employ third differencing if second differencing is not enough, same we go to fourth differencing if third was not enough to remove the trend in mean, and so on.

### Testing Stationarity

There are statistical tests that are used to test the stationarity. Testing can be done after differencing the data, if originally one is sure that the data are non-stationary, or before differencing, if one is not sure if the data are non-stationary. A process is said to be non-stationary if it has a *unit root*. A process has a unit root if 1 is a root of

the characteristic equation (12). It can be shown, by solving the equation 12), that the  $AR(p)$  is stationary and ergodic provided the roots of the characteristic equation lie outside the complex unit circle.

Suppose the  $AR$  model is known, we check the stationarity by evaluating the roots of the characteristics equation, and if all of them lie outside the unit circle then the given model is stationary, that is, it is stationary if and only if the  $r$  roots of the  $AR$  characteristic equation each exceed 1 in absolute value (modulus). For  $AR(p)$ , equation (1), the characteristic equation is

$$1 = \phi_1 L + \phi_2 L^2 + \dots + \phi_p L^p \quad (12)$$

Hence, one should solve for  $L$  to see the nature of roots whether they lie inside the unit circle or outside the unit circle.

Take an example of  $AR(3)$  defined as  $x_t = 3x_{t-1} - 2.75x_{t-2} + 0.75x_{t-3} + u_t$  whose characteristic equation is

$$1 = 3L - 2.75L^2 + 0.75L^3$$

Solving for  $L$  gives  $L = 1$ ,  $L = \frac{2}{3}$  and  $L = 2$ , this is not stationary because out of its roots only one lies outside the unit circle.

## Dickey–Fuller test

Dickey and Fuller, ([9]), developed the basic test for unit roots and order of integration. The test is called Dickey–Fuller test for stationarity. The objective is to test the null hypothesis. However the Dickey–Fuller test tests whether a unit root is present in an autoregressive model. It examines the null hypothesis that  $\phi = 1$  (the process has a unit root, its current realization appears to be an infinite sum of past disturbances with some starting value  $y_0$ ) versus the one-side alternative  $\phi < 1$  (the process is stationary). The test statistics look as follows

$H_0$ : series contains a unit root

$H_1$ : series is stationary

$$DF = \frac{1 - \hat{\phi}}{\hat{SE}(1 - \hat{\phi})}, \quad (13)$$

where  $H_0$  and  $H_1$  are the null and alternative hypothesis respectively and  $SE$  is the Standard Error.

The Dickey–Fuller test statistic does not follow the t-distribution under the null, because the null is one of non-stationarity, but rather Dickey–Fuller test follows a non-standard distribution ([2]). For comparison there are critical values which were derived from experimental simulations; for instance, Monte Carlo experiments give estimates for critical

values. If the DF statistical value is smaller than the critical value then we reject the null hypothesis of a unit root. In most cases the Dickey-Fuller test verifies whether a unit root is present in an autoregressive model, that is, deals with AR models.

### Phillips-Perron test

A similar test to Dickey-Fuller test is called the Phillips-Perron test. Phillips and Perron developed a comprehensive theory of unit roots and non-stationarity. The Phillips-Perron test is similar to Dickey-Fuller test. It incorporates an automatic correction to the Dickey-Fuller procedure for autocorrelated residuals to be used. However, this one relaxes assumptions about lack of autocorrelation in the error term. The critical values used for comparison are the same as for Dickey-Fuller test.

Criticism of Dickey-Fuller and Phillips-Perron-type tests: Main criticism is that the power of the tests is low if the process is stationary but with a root close to the non-stationary boundary. When the process has the  $\phi$  value close to the non-stationarity boundary, *i.e.*  $\phi = 0.95$ , the problem arises. The process is, by definition, still stationary for Dickey-Fuller and Phillips-Perron tests. If the size of the sample is small, the tests often fail to distinguish for the values  $\phi = 1$  and  $\phi = 0.95$ . To avoid this failure of Dickey-Fuller and Phillips-Perron-type tests, there is another test called KPSS test done by Kwiatkowski, Phillips, Schmidt and Shin, 1992, ([10]). This test assumes stationarity as a null hypothesis:

$H_0$ : series is a stationary

$H_1$ : series is not stationary

$$KPSS = T^{-2} \sum_{i=1}^t S_i^2 / \sigma_T(l), \quad (14)$$

where  $S_t = \sum_{i=1}^t e_i$ ,  $e$  are residuals given as  $e = [e_1, e_2, \dots, e_T]'$ , and  $\sigma_T(l)$  represents an estimate of the long run variance of the residuals. We reject the stationary null when KPSS is large, since that is evidence that the series wanders from its mean.

### Seasonality

Seasonality means periodic fluctuations, that is, a certain basic pattern tends to be repeated at regular seasonal intervals. This can be assessed by a run sequence plot, a seasonal sub series plot, multiple box plots, or autocorrelation plot. Sometimes, after one has removed the trend in mean with regular differencing there is still a seasonal effect. With monthly data, we may be able to remove the seasonal effect, if there is any,

by the seasonal  $\nabla_{12}$  difference operator. If the time series is quarterly, we may be able to remove the seasonal effect by the  $\nabla_4$  operator.

### **Stages in building Box-Jenkins model**

After checking the stationarity and seasonality, basically, there are 4 stages in building a Box-Jenkins time series model. These are

1. Model Identification: By identification we mean the use of the data, and of any information on how the series was generated, to suggest a subclass of parsimonious models worthy to be tested or used ([1]). Model identification involves determining the order of the model required to capture the dynamic features of the data. Graphical procedures are used to determine the most appropriate specification.
2. Model Estimation: By estimation we mean efficient use of the data to make inferences about parameters conditional on the adequacy of the entertained model ([1]). Model estimation involves estimation of the parameters of the model specified in Model Identification. The most common methods used are maximum likelihood estimation and non-linear least-squares estimation, depending on the model.
3. Model Validation: This involves model checking (diagnostic checking), that is, determining whether the model specified and estimated is adequate. Thus: model validation deals with testing whether the estimated model conforms to the specifications of a stationary univariate process. It can be done by over-fitting and Residual diagnosis. Here we check the fitted model in its relation to the data with intent to reveal model inadequacies and so to achieve model improvement. Normally, the data set is divided into two parts, namely fitting part (used for model estimation) and validation part (last  $\frac{1}{20}$  or  $\frac{1}{10}$  of observation). We forecast using fitting data set and verify its performance by comparing with the true realization of the process. In Validation, we can use R2 or Q2 measures to estimate the goodness of the fit and cross valid prediction, respectively. If the validation indicates wrong model we go back to the first stage (Identification stage) if the data are with no doubts stationary, otherwise we get back to the data pre-analysis stage (([1]) and([7])).
4. Model Forecasting: The estimated model is used to generate forecasts and confidence limits of the forecasts, forecasting future realizations.  
Only the first two stages will be discussed, in this work; it should first be said that identification and estimation necessarily overlap.

### 2.5.2 Model Identification

Identification is inexact because it is difficult to decide which orders of models occur in practice by using purely mathematical argument. Indeed several models may be, roughly, equally valid for a given situation. This is a property of the behavior of the physical world, there is some subjectivity in decision. Model identification is a stage where statistical inefficient methods are used since there is no precise formulation of the problem. We use the graphical methods where the judgments are exercised. The task is to identify what is the appropriate model from general ARMA family. One chooses a model from the general *ARMA* family and selects its order. This is done when the data are stationary. Therefore, once stationarity and seasonality have been addressed, one needs to identify the order ( $r$  and  $n$ ) of the AR and MA terms. For doing this we use the autocorrelation and the partial autocorrelation plots. Here the sample autocorrelation plot and the sample partial autocorrelation plot are compared to the theoretical behaviour of these plots when the model order is known. The identification of models can also be done using standard errors for estimated autocorrelations and partial autocorrelations. ([1])

The ARMA model identification is based on autocorrelation and partial autocorrelation function values. Naturally, the model whose values are closest to calculated ones is chosen. Understanding the concept of autocorrelation can be tested by trying to conclude the mentioned theoretical values; here they are presented in Table (1)

Table 1: Theoretical characteristics of ACF and PACF for basic ARMA models

Model	Theoretical $r_k$	Theoretical $r_{kk}$
AR(0)	All zero	All zero
AR(1)	Vanish toward zero	Zero after 1st lag
AR(2)	Vanish toward zero	Zero after 2nd lag
MA(1)	Zero after 1st lag	Vanish toward zero
MA(2)	Zero after 2nd lag	Vanish toward zero
ARMA(1,1)	Vanish toward zero	Vanish toward zero

### 2.5.3 Model Estimation

We discussed a way of choosing the model, we were selecting the model and not the numerical values of the parameters. For pure AR models there exist simple estimation techniques, since there is a linear relationship between the autocorrelations and the AR parameters. This relationship can be inverted, and then the theoretical autocorrelations can be replaced by their estimates, to yield estimates of the AR parameters. Given AR( $r$ ) model, we can obtain a system of linear equations called the Yule-Walker equations by multiplying both sides of the equation by  $x_{t-k}$ , take expectation and then normalize. The estimates are obtained by solving the formed equation. The resulting values are called the Yule-Walker estimates. The Yule-Walker estimates always correspond to a stationary AR model (see the practical part about Yule-Walker Equations, for more explanation and parameterization).

The situation for the *MA* models is not like *AR* models because the theoretical relationship between the parameters and autocorrelations is not linear. For this case we use the *invertibility*. The *MA* model is said to be invertible if it can be represented as a stationary infinite-order autoregression,  $AR(\infty)$ .

For ARMA models, estimation of parameters proceeds by nonlinear methods. There are two methods available to estimate these parameters. One is called least squares method and the other is called the maximum likelihood method. In this study we will review only least squares methods.

The classical Box-Jenkins model estimation uses a recursive algorithm. The objective is to minimize the sum of squares of errors. The analysis of error is based on estimates, on residuals of the models ([7]). Residual is the difference between the observed and fitted values.

Example: AR(1) has  $u_t$  as the error and  $\hat{u}_t$  as the residual defined by

$$u_t = x_t - \theta_1 x_{t-1}$$

$$\hat{u}_t = x_t - \hat{\theta}_1 x_{t-1}$$

Under the least squares method one chooses those values of the parameters which will make the sum of the squared residuals as small as possible, that is the minimization of

$$S = \sum_{t=1}^n \hat{u}_t^2.$$

The model requires starting values for the data  $x_0, x_1, x_2, \dots, x_n$  and for the errors  $u_0, u_1, u_2, \dots, u_n$  and possibly also for seasonal terms.

In general, the model is nonlinear in the coefficients and if the residuals are truly white noise, then their ACF should have no spikes and the sample autocorrelations should all be small. The autocorrelation of the residuals would also yield valuable information about possible model inadequacies. If we assume for simplicity the ARMA(1,1) model

$$x_t = \phi x_{t-1} + \psi u_{t-1} + u_t$$

then the recursive algorithm considers all possible values of  $\phi$  and  $\psi$  and for them minimizes the sum of squares  $\sum_{i=1}^t \hat{u}_i^2$ . In particular, assume we have  $x_0, x_1, x_2, \dots, x_t$  as the observed stationary series. Initially  $u_0 = 0$ . Then we have

$$\begin{aligned} x_1 &= \phi x_0 + \psi \cdot 0 + u_1 \Rightarrow u_1 = x_1 - \phi x_0 \\ x_2 &= \phi x_1 + \psi \cdot u_1 + u_2 \Rightarrow u_2 = x_2 - \phi x_1 - \psi u_1 \\ x_3 &= \phi x_2 + \psi \cdot u_2 + u_3 \Rightarrow u_3 = x_3 - \phi x_2 - \psi u_2 \\ &\vdots \\ x_n &= \phi x_{n-1} + \psi \cdot u_{n-1} + u_n \Rightarrow u_n = x_n - \phi x_{n-1} - \psi u_{n-1} \end{aligned} \tag{15}$$

The parameters  $\phi$  and  $\psi$  are varied between (-1,1) (since the series is required to be stationary), and the sums  $\sum_{i=1}^t \hat{u}_i^2$  are calculated each time using any optimizer such as *fminsearch*. Then the proper estimates  $\phi$  and  $\psi$ , and the  $u_t$  noise values are those that give the minimal value of the  $\hat{u}_t$  sum of squares. The equation (15) generalizes to any *ARMA*( $p, q$ ) models.

### 3 Theoretical background for MCMC

#### 3.1 Bayesian approach

Bayesian approach is a statistical technique where observations are used to infer the probability that a hypothesis may be true by the use of Bayes formula; by quantifying the information of an observer about the model parameters, given the observed data. In the classical probabilities, if  $P(D) \neq 0$ , then the conditional probability is defined as ([17])

$$P(A/D) = \frac{P(A \cap D)}{P(D)} = \frac{P(D/A)P(A)}{P(D)} \quad (16)$$

$$P(A/D) \propto P(D/A)P(A) \quad (17)$$

$P(D)$  is the normalizing constant needed to make the total probabilities on the left sum to one. It is the marginal probability of the data which is defined as

$$P(D) = \sum_i P(D \cap A_i) = \sum_i P(D/A_i)P(A_i)$$

If we have an initial belief about the truth of  $A$  and we observe some data  $D$ , then we can find the revised belief about  $A$ , in the light of  $D$  by the use of Bayes formula (17). Here  $P(A)$ ,  $P(D/A)$  and  $P(A/D)$  are the prior distribution, the likelihood and the posterior distribution. Therefore, posterior  $\propto$  prior  $\times$  likelihood. The prior distribution describes the previous information about the model parameters, the likelihood describes the probabilities of observing a set of data and the posterior distribution defines the Bayesian solution to the parameter estimation.

The Bayesian inference can be performed as follows: enumerate all of the possible states of nature and choose a prior distribution, establish the likelihood function, which tells you how well the data we actually observed are predicted by each hypothetical state of nature, compute the posterior distribution by Bayes' formula. Thus the Bayesian approach is to choose a prior information that reflect the beliefs of observer about model parameters to be considered, and then updating the beliefs on the basis of data observed, resulting in the posterior distribution. Note that the interpretation is subjective.

Suppose there are  $N$  coins, from these, one coin has heads at both sides. A coin is selected at random and flipped  $k$  times resulting, all flipping, to heads. We are interested with the probability of choosing a two headed coin. To find this, we let  $A_k$  be the event that a coin lands  $k$ -times,  $H_1$  be the coin is two headed, and  $H_2$  be the coin is fair. Therefore  $P(H_1) = \frac{1}{N}$  and  $P(H_2) = 1 - \frac{1}{N}$ . The conditional probabilities are

$$P(A_k/H_1) = 1$$

$$P(A_k/H_2) = \frac{1}{2^k}$$

From total probability formula,

$$P(A_k) = \frac{2^k + N - 1}{2^k N}$$

$$P(H_1/A_k) = \frac{2^k}{2^k + N - 1}$$

If  $N = 1000000$  and  $k = 1, 2 \dots 30$ , the graph of the posterior probabilities is given in Figure 3.

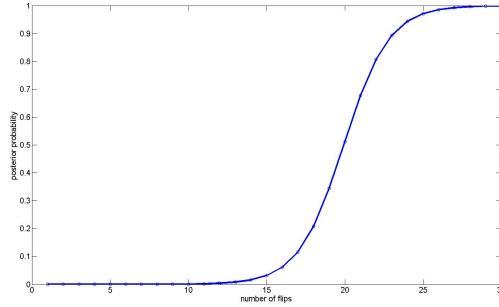


Figure 3: Posterior probabilities.

In a case of the Bayesian model  $X \sim N(\theta, \sigma^2)$  and  $\theta \sim N(\mu, \tau^2)$  with a data set (2.9441, -13.3618, 7.1432, 16.2356, -6.9178, 8.5800, 12.5400, -15.9373, -14.4096, 5.7115). If  $\sigma = 10000$ ,  $\mu = 20$  and  $\tau = 400$  such that the data are coming from  $N(\theta, 10000)$  and the prior on  $\theta$  is  $N(20, 400)$ , then the three densities are shown in Figure 35. The posterior is  $N(6.8352, 6.6667)$ .

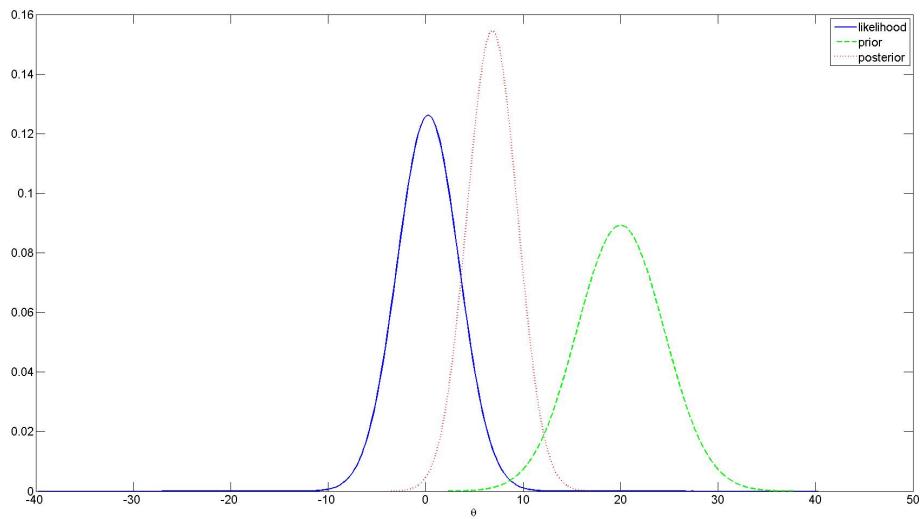


Figure 4: Likelihood, Prior and Posterior.

For the case of the probability densities, the formulae are similar; if  $\theta$  represents the parameters of the density and  $x$  is the observed data, then

$$P(\theta/x) = \frac{P(\theta \cap x)}{P(x)} = \frac{P(x/\theta)P(\theta)}{P(x)} \quad (18)$$

Computing the marginal probability of the data is the hardest practical problem of Bayesian inference because the integrals are over high-dimensional spaces. The research in Bayesian statistics contributed to tremendous broadening of the scope of Bayesian models such that the models that were not handled are now routinely solved by other methods such that MCMC methods.

### 3.2 The Metropolis-Hastings Algorithm

The MCMC algorithms have become extremely popular in statistics. They are a way of approximating sampling from complicated and higher dimensional probability distributions. The MCMC algorithms have transformed Bayesian inference by allowing practitioners to sample from posterior distributions of complicated models. The MCMC algorithms involve Markov chains with a complicated stationary distribution.

The Metropolis algorithm is fundamental to MCMC development. Suppose the target distribution is known, we need a chain  $\pi$  as its stationary distribution. The proposal distribution  $q(x, y) = q(x/y)$  is the one for a new value of a chain  $y$ , given that the chain is at the value  $x$ . If  $f$  is the stationary distribution, a Markov chain with  $q(x, y) = q(x/y)$  satisfies a detailed balanced equation  $q(y/x)f(x) = q(x/y)f(y)$ . If  $q(y/x)\pi(x) > q(x/y)\pi(y)$ , there exists a factor  $\rho(x/y) \leq 1$  such that

$$\rho(x/y) = \min \left\{ 1, \frac{q(x/y).\pi(y)}{q(y/x).\pi(x)} \right\} \quad (19)$$

If  $\pi$  is the initial distribution of the starting state and given the Detailed Balanced Equation  $\pi(\theta)p(\theta, \theta') = \pi(\theta')p(\theta', \theta)$  then the intensity of going from state  $\theta$  to state  $\theta'$  is the same as that of going from  $\theta'$  to state  $\theta$ . Therefore  $\int \pi(\theta)p(\theta, \theta') = \pi(\theta')$ . The Metropolis-Hastings algorithm (MH) is a general way of constructing a Markov chain. The following are the steps to be taken:

- Start with the arbitrary  $x_0$  from the support of the target distribution.
- Generate proposal  $y$  from  $q(y/x_n)$  at the stage  $n$ .
- Take  $x_{n+1} = y$  with  $\rho(x/y) = \min \left\{ 1, \frac{q(x/y).\pi(y)}{q(y/x).\pi(x)} \right\}$ , otherwise take  $x_{n+1} = x_n$ . This random acceptance is done by generating  $U(1, 0)$  and accept  $y$  if  $U \leq \rho(x/y)$ .

- Increase  $n$  and return to step 2.

Under some easily satisfied regularity conditions on the proposal density, the sequence of simulated draws,  $x_1, x_2, \dots$  will converge to a random variable that is distributed according to the posterior distribution.

If  $q(y/x) = q(x/y)$ , then  $\rho(x/y) = \min \left\{ 1, \frac{\pi(y)}{\pi(x)} \right\}$ . If  $q(x/y)$  does not depend on  $x$ , that is,  $q(y/x) = q(y)$  (if the proposal density is independent of the current value in the sequence) then the MH is the independence Metropolis which is similar to Acceptance-Rejection method.

Example [From Johnson and Albert (1999)]: a small company improved a product and wants to infer about the proportion of potential customers who will buy the product, if the new product is preferred to the old one. The company is certain that this proportion will exceed 0.5, and uses the uniform prior on  $[0.5, 1]$ . Out of 20 customers surveyed, 12 prefer the new product. Find the posterior for  $p$ .

Solution:

Since the support of  $p$  is  $[0.5, 1]$ , we transform the data by  $\theta = \log \left( \frac{p-0.5}{1-p} \right)$ , so that  $\theta \in (-\infty, \infty)$ . Hence  $p = \frac{1+exp\{\theta\}}{1+exp\{\theta\}}$  with jacobian  $\frac{\frac{1}{2}exp\{\theta\}}{(1+exp\{\theta\})^2}$  and the density is proportional to  $\frac{(\frac{1}{2}+exp\{\theta\})^{12}exp\{\theta\}}{(1+exp\{\theta\})^{22}}$ . The proposal distribution is normal  $N(\theta_n, s^2)$ , where  $\theta_n$  is a current state of the chain and  $s^2$  is to be specified. Figures 5 and 6 show the histogram.

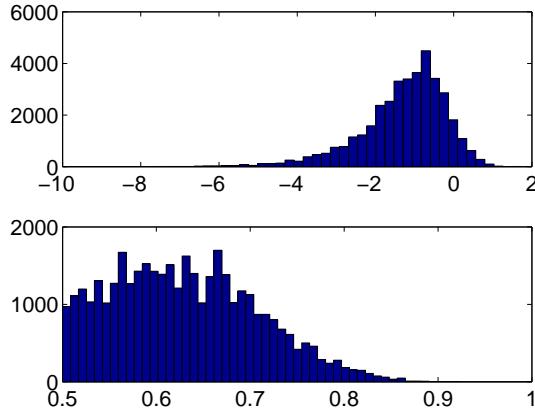


Figure 5: Histogram for  $\theta$  and  $s$ .

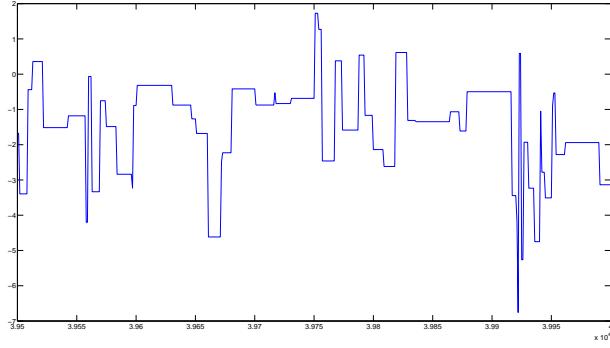


Figure 6: Last 500 simulations of the chain.

The choice of the proposal distribution depends on the nature of components. For instance, for discrete components it is common to choose the uniform distribution over the state space, an alternative is to use a distribution that is uniform over all values except the current one. For the case of continuous components, the Gaussian distribution (or multivariate Gaussian for compound components) centered on the current value is chosen and its alternative is the Cauchy distribution (or multivariate Cauchy), with heavier tails allowing occasional large jumps in the Markov chain. There are many different classes of ways of choosing the proposal density namely Symmetric Metropolis Algorithm, Random walk Metropolis-Hastings, Independence sampler, and Langevin algorithm.

### 3.3 The Gibbs Sampler

The Gibbs sampler is a popular MCMC algorithm for its computational simplicity and it is a special case for the MH. It aims to make sampling from a high-dimensional distribution more tractable by sampling from a collection of more manageable smaller dimensional distributions because of the problem of finding a proposal distribution for higher dimensional models. The idea behind is that we can set up a Markov chain simulation algorithm from the joint posterior distribution by simulating parameters from the set of conditional distributions.

Let  $\theta_{-i}$  be the set  $\theta \setminus \theta_i$  and  $\pi(\theta_i) = \pi(\theta_i | \theta_{-i}), i = 1, \dots, d$  be conditional distribution. Then

- Start with the arbitrary  $\theta^{(0)} = \theta_0^{(0)}, \dots, \theta_d^{(0)}$  for which  $\pi(\theta^{(0)}) > 0$ .
- Obtain  $\theta_1^{(t)}$  from the conditional distribution  $\pi(\theta_1 | \theta_2^{(t-1)}, \dots, \theta_d^{(t-1)})$ .
- Obtain  $\theta_2^{(t)}$  from the conditional distribution  $\pi(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \dots, \theta_d^{(t-1)})$ .
- ⋮

- Obtain  $\theta_p^{(t)}$  from the conditional distribution

$$\pi(\theta_p | \theta_1^{(t)}, \dots, \theta_{p-1}^{(t-1)}, \theta_{p+1}^{(t-1)}, \dots, \theta_d^{(t-1)}).$$

- Repeat from second step.

The algorithm is run until the convergence is attained; as the convergence is reached, the resulting value  $\theta^{(j)}$  is drawn from  $\pi(\theta)$ . The main requirement is that the sampling process is ergodic. An ergodic process will converge to the correct distribution given enough time. As the number of iterations becomes large, the Gibbs sampling algorithm converges.

The distinguishing feature of the Gibbs sampler is that first, it samples one variable conditioned on all the others, then a second variable, then a third variable, and so on, always conditioning on the most current values of the other variables. However, one needs to be able to draw a sample from each of the conditional distributions, otherwise one cannot use exact Gibbs. Therefore, this algorithm assumes that the conditional distributions are known, and the points created are accepted. There are techniques for doing this under some circumstances, such as importance sampling and slice sampling.

### 3.4 Adaptive MCMC Algorithms

The MCMC algorithms, such as the MH, are used in statistical inference, to sample from complicated high-dimensional distributions. However, it is difficult to find a proposal that fits the target distribution due to time-consuming, trial-and-error 'tuning' of the proposal. For instance, when dealing with the Gaussian proposal, tuning of associated parameters, proposal variances, is crucial to achieve efficient mixing, but can also be very difficult. The Adaptive MCMC algorithms attempt to deal with this problem by automatical 'learning' from better parameter values of MCMC algorithms while they run. They do not need to determine the recommended distribution of variables in advance, they use the history to tune the proposal distribution suitably. Thus AM attempts to adaptively tune the algorithm as it progresses for the purpose of improving the performance of the algorithm.

In the Adaptive Metropolis method (AM) ([11]) the proposal covariance is adapted by using the history of the chain generated so far. The algorithm for AM is given below ([12]).

- Start from an initial value  $\theta^0$  and initial proposal covariance  $C = C_0$ . Select a covariance scaling factor  $s$ , a small number  $\epsilon$  for regularizing the covariance, and an initial non-adapting period  $n_0$ .

- At each step, propose a new  $\theta^*$  from a Gaussian distribution centered at the current value  $N(\theta^{i-1}, C)$ .
- Accept or reject  $\theta^*$  according to the MH acceptance probability.
- After an initial period of simulation, say for  $i \geq 0$ , adapt the proposal covariance matrix using the chain generated so far by  $C = cov(\theta^0, \dots, \theta^i)s + I\epsilon$ . Adapt from the beginning of the chain or with an increasing sequence of values. Adaptation can be done at fixed or random intervals.
- Iterate from step 2 until enough values have been generated.

### 3.5 Reversible Jump MCMC

Model selection may be done by graphical method, ACF and PCF. Also, the selection can be done by Reversible jump MCMC (RJMCMC). This is done by MCMC jumping between the set of suggested models. The RJMCMC encompasses the MCMC algorithms, such as MH, but it allows different move types where the mixing of a reversible jump algorithm is improved by alternating the updates of standard MCMC and RJMCMC. The joint distribution of model dimension and model parameters is optimized to find the best pair of dimension and parameters that suits the observations. This is done by designing the moves for jumping between the dimensions. The RJMCMC permits jumps between parameter subspaces of different dimensions at each iteration, that is, the proposal distribution and the acceptance probability are formulated such that the chain performs reversible jumps between spaces of different dimensions.

The RJMCMC is used in various areas such as in modelling intensities of point process by step functions, in crack detection in electrically conducting media, in variables selection in regression and in finding the order of a given model such as ARMA models, to mention some. In this thesis, the RJMCMC is used for finding the order of the model and the then estimating the parameters of the model.

The Detailed Balanced Equation, for RJMCMC ([18]), can be formulated in general space, the states from positive probability to do a reversible move back, the state space for RJMCMC is  $(k, \theta^k)$  where  $k$  comes from the model space while  $\theta^k$  comes from the model parameter space. The factorization of the posterior distribution is  $\pi(\theta^k, k) = \pi(\theta^k | k)\pi(k)$ .

For the RJMCMC;  $i$  and  $j$  are assumed to be models such that there is a need of a reversible move between these models. There exist a bijective function,  $g_{ij}$ , that transforms the parameters,  $g_{ij}(\theta^{(i)}), u^{(i)} = (\theta^{(j)}, u^{(j)})$ , and retains the dimensions,  $d(\theta^{(i)}) + d(u^{(i)}) = d(\theta^{(j)}) + d(u^{(j)})$ , of the models;  $u$  is the random quantity for proposing the change in the

components. Since it is a bijective function, then there exist an inverse function defined by  $g_{ij}^{-1}$  that takes back the move, that is, it gives the move to other direction. We can derive the probability used in RJMCMC ([18]). This is done as follows:

Consider the general state space  $\chi$ . Suppose we are interested in some target distribution  $\pi$  defined in  $\chi$ . We need to construct a Markov Chain, which is reversible, with transition kernel  $\kappa$  having  $\pi$  as its invariant distribution.  $\pi$  is the probability measure on  $\chi$  with equilibrium equation

$$\int_{\chi} \pi(dx) \kappa(x, dx') = \pi(dx') \quad (20)$$

The chain satisfies (20) for all Borel sets  $C, C' \subset \chi$ , thus

$$\int_C \pi(dx) \kappa(x, C') = \int_{C'} \pi(dx') \kappa(x', C) \quad (21)$$

As in MH,  $x'$  is drawn from a proposal measure  $q(x, dx')$ .  $x'$  is accepted with probability  $\alpha(x, x')$ , otherwise  $x$  is retain and becomes a new state. Thus  $\kappa$  is defined as

$$\kappa(x, C') = \int_{C'} q(x, dx') \alpha(x, x') + R(x) \mathbf{1}_{x \in C'} \quad (22)$$

Where  $\mathbf{1}_.$  is an indicator function and  $R(x)$  is the probability of rejecting the proposed model and is given by

$$R(x) = \int_{\chi} q(x, dx')(1 - \alpha(x, x')) \quad (23)$$

Substituting equation (22) to equation (21) we have

$$\int_C \pi(dx) \int_{C'} q(x, dx') \alpha(x, x') + \int_{C \cap C'} \pi(dx) R(x) = \int_{C'} \pi(dx') \int_C q(x', dx) \alpha(x', x) + \int_{C' \cap C} \pi(dx) R(x') \quad (24)$$

But

$$\int_{C \cap C'} \pi(dx) R(x) = \int_{C' \cap C} \pi(dx) R(x') \quad (25)$$

Therefore equation (24) becomes

$$\int_{(x,x') \in C \times C'} \pi(dx) q(x, dx') \alpha(x, x') = \int_{(x,x') \in C \times C'} \pi(dx') q(x', dx) \alpha(x', x) \quad (26)$$

Green(1995) discusses that the  $\kappa$  is a mixture over a number of move types, so that a move of type  $m$ , taking the chain  $x'$  is proposed with probability  $q_m(x, dx')$ . Green assumes the existence of a symmetric measure  $\mu$  on  $\chi$  which dominates  $\pi(dx)q(x, dx')$ ; which has density  $f(x, x')$  with respect to  $\mu$ . So the equation (26) can be written as

$$\int_{(x,x') \in C \times C'} \alpha(x, x') f(x, x') \mu(dx, dx') = \int_{(x,x') \in C \times C'} \alpha(x', x) f(x', x) \mu(dx', dx) \quad (27)$$

Which holds for all Borel  $C, C'$  if

$$\alpha(x, x') = \min \left\{ 1, \frac{f(x', x)}{f(x, x')} \right\}$$

If the densities  $f$  is replaced by appropriate measure, then

$$\alpha(x, x') = \min \left\{ 1, \frac{\pi(dx')q(x', dx)}{\pi(dx)q(x, dx')} \right\} \quad (28)$$

Suppose  $\chi \subset \mathbb{R}^d$  and  $\pi$  has a density with respect to  $d$ -dimensional Lebesgue measure. We assume the move from  $x$  to  $x'$ . We generate  $r$  random numbers,  $u$ , which have known density  $g$ . The proposed new state is  $x'$  when  $(x', u') = t(x, u)$ ;  $t$  is a known deterministic function,  $u'$  is needed for revers transition. Consider the reverse transition which is made analogous to original proposal so that  $(x, u) = t'(x', u')$ ;  $u'$  are random numbers with known density  $g'$  and known deterministic function  $t'$ . The equation (26) becomes

$$\int_{(x, x') \in C \times C'} \pi(x)g(u)\alpha(x, x')dxdu = \int_{(x, x') \in C \times C'} \pi(x')g'(u')\alpha(x', x)dx'du' \quad (29)$$

Green(2003) noted that the transformation from  $(x, u)$  to  $(x', u')$  is invertible and differentiable with the basic calculus

$$dx'du' = |J| dxdu \quad (30)$$

Where  $J = \frac{\partial(x', u')}{\partial(x, u)}$  is the Jacobian of the transformation, from  $(x, u)$  to  $(x', u')$ , which comes from the dimension jump. Substitution of equation (30) to (29) gives the suitable choice

$$\alpha(x, x') = \min \left\{ 1, \frac{\pi(x')g'(u')}{\pi(x)g(u)} \right\} \quad (31)$$

Following the derivation, this can be stated in different language as follows:

Let  $q_{ij}(\theta^{(i)}, u^{(i)})$  be the probability density for the proposed move,  $p(i, j)$  be the probability for the move from  $i$  to  $j$ . Then the accepting probability is  $\alpha_{ij}(\theta^{(i)}, \theta^{(j)})$ , involving the model ratios, proposal ratios and Jacobian and it is defined as

$$\alpha_{ij}(\theta^{(i)}, \theta^{(j)}) = \min \left\{ 1, \frac{\pi_j(\theta^{(j)})p(j, i)q_{ji}(\theta^{(j)}, u^{(j)})}{\pi_i(\theta^{(i)})p(i, j)q_{ij}(\theta^{(i)}, u^{(i)})} \left| \frac{\partial(\theta^{(j)}, u^{(j)})}{\partial(\theta^{(i)}, u^{(i)})} \right| \right\} \quad (32)$$

After the derivation of the probability to accept the move, suppose we are in the model  $k_i$  with parameter vector  $\theta_i^{(k_i)}$ , the possible version of the RJMCMC algorithm is as follows:

- Choose a new model  $j$  by drawing it from distribution  $p(i, .)$ . Propose a value for the parameter  $\theta^{(j)}$  by generating  $u$  from the distribution  $q_{k_i j}(\theta_i^{k_i}, u)$ .
- Accept the move with probability  $\alpha_{ij}(\theta^{(i)}, \theta^{(j)})$  where  $j = k_{i+1}$  and  $\theta^{(j)} = \theta_{i+1}^{k_{i+1}}$ .
- If the move is not accepted then  $k_i = k_{i+1}$  and  $\theta^{(k_i)} = \theta_{i+1}^{k_{i+1}}$ , that is, stay in the current model.

From the algorithm, the RJMCMC is a more automatic order selection technique that allows simulation from target distribution on spaces of varying dimension. It generates a Markov chain jumping between models in parameter spaces of different dimensions with an acceptance probability that is designed to preserve Detailed Balanced Equation within each move.

## 4 Parameter Estimation

The theoretical part of Box-Jenkins model examined the stages to build a model from a set of observations. The practical results section moves on to examining the ways that are used to estimate the parameters of ARMA models.

### 4.1 Series generation

For the purpose of this study, a simple series of at least 1000 observations with coefficients such that the series are stationary was generated. Consider some ARMA models with different values of  $r$  and  $n$

$$\text{ARMA}(1,0) \rightarrow x_t = \theta x_{t-1} + u_t$$

$$\text{ARMA}(1,1) \rightarrow x_t = \theta x_{t-1} + \psi u_{t-1} + u_t$$

$$\text{ARMA}(2,0) \rightarrow x_t = \theta_1 x_{t-1} + \theta_2 x_{t-2} + u_t$$

$$\text{ARMA}(2,1) \rightarrow x_t = \theta_1 x_{t-1} + \theta_2 x_{t-2} + \psi_1 u_{t-1} + u_t$$

$$\text{ARMA}(2,2) \rightarrow x_t = \theta_1 x_{t-1} + \theta_2 x_{t-2} + \psi_1 u_{t-1} + \psi_2 u_{t-2} + u_t$$

Example: ARMA(1,0); here we fix the value of  $x_0 = 3$ ,  $\theta = 0.7$ , the standard deviation is 1 and the mean is 0 (White noise is normally distributed  $N(0,1)$ ). The sample size (number of observations) taken is 1000. Figure (7) presents an example realization of ARMA(1,0) series with the mentioned specification.

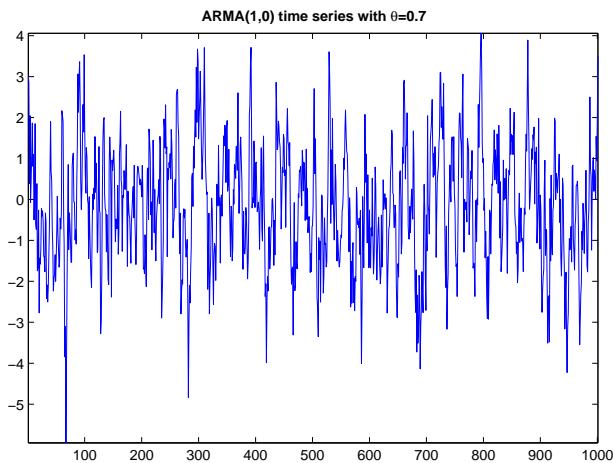


Figure 7: ARMA(1,0).

Example: ARMA(2,1); here we fix the value of  $x_0 = 3$ ,  $x_1 = 2.5$ ,  $\theta_1 = 0.58$ ,  $\theta_2 = -0.4$ ,  $\psi = 0.6$ , the standard deviation is 1 and the mean is 0 (White noise is normally distributed  $N(0,1)$ ). Figure (8) presents an example realization of such series with the sample size (number of observations) taken 1000.

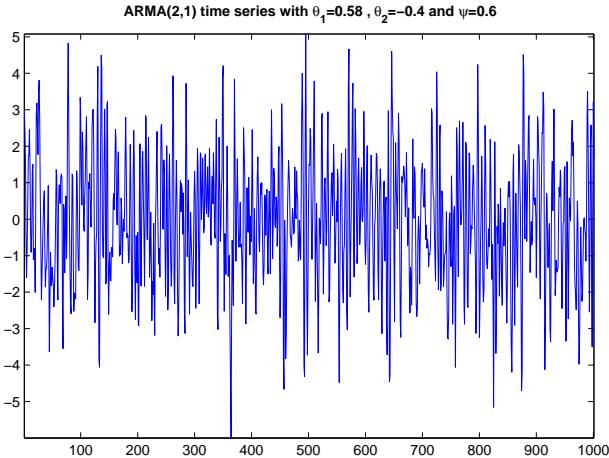


Figure 8: ARMA(2,1).

#### 4.2 Box-Jenkins recursive estimation

The purpose of this section is to perform the Box-Jenkins recursive calculation to estimate model parameters and compare results with the originally defined model parameters. We use the minimization of the sum of the squares of residuals. The formulas in equation (15) are used.

Two cases are taken into account; the first case is when the estimation is done for a single realization while the second case is when there are 1000 realizations, this is the case to verify the general accuracy of the estimation method. For the first case, only the estimates are given while for the second case the results are presented in normalized histograms.

Example: ARMA(1,0); The original  $\theta = 0.7$  and  $x_0 = 3$ , with the standard deviation 1 and the mean is 0 (White noise is normally distributed  $N(0,1)$ ). The estimation for one realization gave the result which is 0.71. For the case of 1000 realizations, Figure (9) presents a normalized histogram of estimates of  $\theta$  for different realizations of ARMA(1,0). The estimated valued, indeed, is approximately equals to the original value (0.7), this can be seen by looking the average performance value in Table (2).

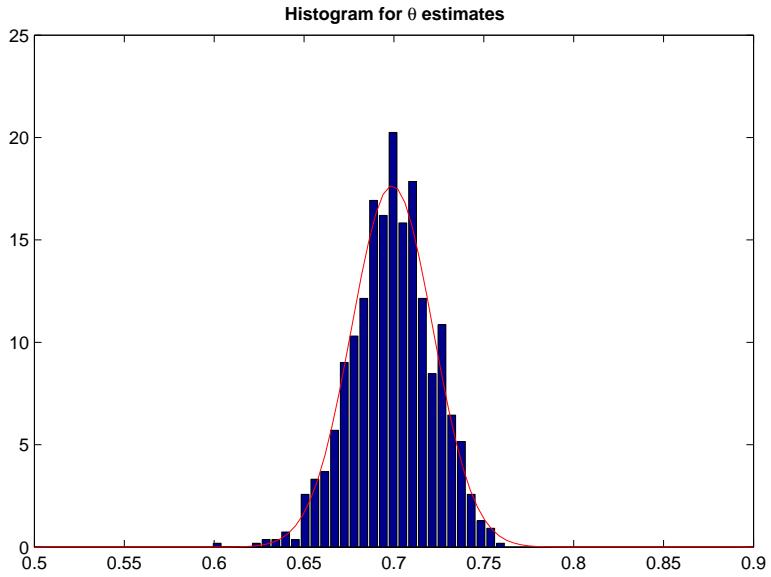


Figure 9: Normalized histogram of  $\theta$  estimates ARMA(1,0).

Example: ARMA(2,1); the original  $\theta_1 = 0.58$ ,  $\theta_2 = -0.4$ ,  $\psi = 0.6$ ,  $x_0 = 3$ ,  $x_1 = 2.5$ , with the standard deviation 1 and the mean is 0 (White noise is normally distributed  $N(0,1)$ ). In case of a single realization, the estimates are 0.5100, -0.3600 and 0.6400 which look like the original parameters, not exactly but approximately. For 1000 realizations: the results indicate that the estimates are approximately equal to original values of the parameters. This is represented in Figure (10) that shows a normalized histogram of estimates of ARMA(2,1).

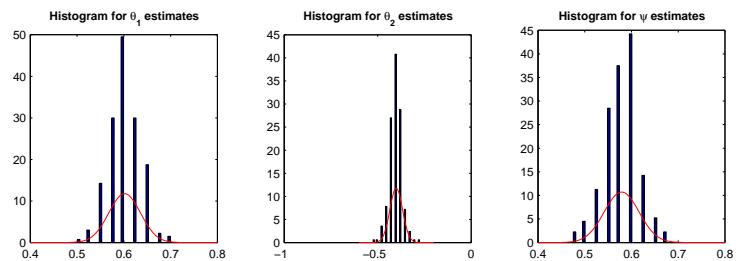


Figure 10: Normalized histogram of  $\theta_1, \theta_2, \psi$  estimates ARMA(2,1).

Note that the problem of estimation of parameters of nonlinear equation (3) seems linear with respect to  $\theta$  and  $\psi$ . But if we generate independent new values for the noise in all (3) and estimate parameters using linear regression (we use a "backslash operation" in Matlab), the consecutive character of the noise is not taken into account, and the result is clearly wrong.

### 4.3 Matlab *garchfit* estimation

GARCH is the Generalized Autoregressive Conditional Heteroskedasticity Process. GARCH Toolbox provides an integrated computing environment for modeling the volatility of univariate economic time series; it goes by first is to estimate a best-fitting autoregressive model; secondly, compute autocorrelations of the error term and lastly, test for significance. In this work, only ARMA model is considered.

The Matlab GARCH Toolbox (function `garchfit.m` in particular) provides the environment where one can use ready made functions to estimate the parameters of models with known orders, and compares the estimates with the original model values. It creates the specification structure for model, and stores the model orders and estimated parameters in it. The GARCH Toolbox contains the parameters that define a model and it controls estimation process in a specification structure.

During estimation by GARCH Tool box, *garchfit* function is preceded by *garchset* function which provides the main user-interface for specifying GARCH Toolbox parameters, and is the preferred method for creating and modifying model specification structures. The specification structure is made up of general parameters, conditional mean parameters, conditional variance parameters, equality constraints parameters, and optimization parameters.

Example: ARMA(1,0); the original parameters are  $\theta = 0.7$  and  $x_0 = 3$ , with the standard deviation 1 and the mean is 0 (White noise is normally distributed  $N(0,1)$ ). When one realization is done the parameter estimate was 0.6799, this value is close to original parameter value 0.7. For 1000 realizations, the estimation looks good as Figure (11) shows that the estimated parameter and the real parameter are approximately equal.

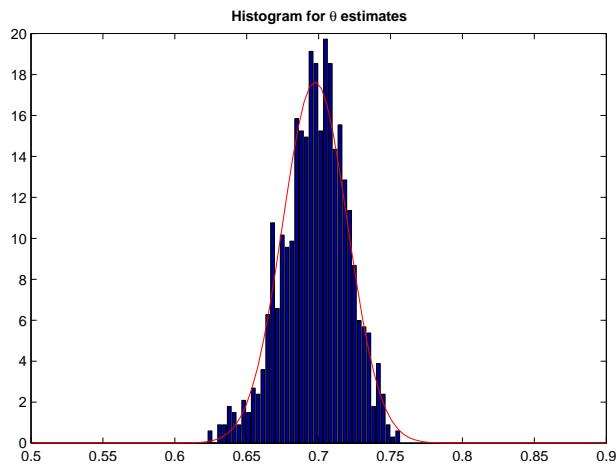


Figure 11: Normalized histogram of  $\theta$  estimates obtained from Matlab `garchfit.m` estimation (original  $\theta = 0.7$ ).

Example: ARMA(2,1); the original parameters are  $\theta_1 = 0.58$ ,  $\theta_2 = -0.4$ , and  $\psi = 0.6$ . The initial values are  $x_0 = 3$ , and  $x_1 = 2.5$ , with the standard deviation 1 and the mean is 0 (White noise is normally distributed  $N(0,1)$ ). For the case of single realization the method produces the estimates 0.6037, -0.4059 and 0.5618 for  $\theta_1$ ,  $\theta_2$ , and  $\psi$ . For 1000 realizations, Figure (12) presents histograms of parameter estimates for model ARMA(2,1) and the figure indicates a good fit of the estimate averages to the real values.

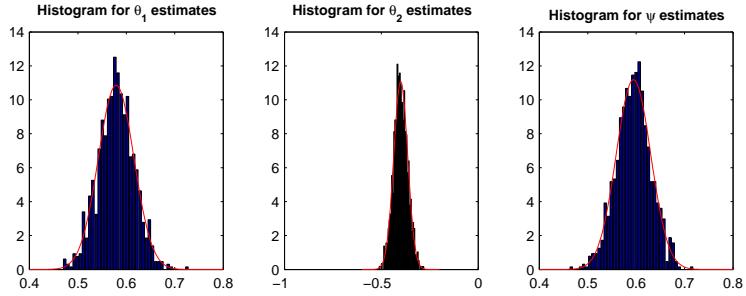


Figure 12: Normalized histogram of  $\theta_1, \theta_2, \psi$  estimates ARMA(2,1).

#### 4.4 Yule-Walker Equations

The Yule-Walker Equations is a common method for estimating the autoregressive process parameters. It is said to be one of the biased estimation methods. Consider the equation (1) which can be multiplied, throughout, by  $x_{t-k}$  where  $k = 1, 2, \dots, r$ ; and lead to;

$$x_t x_{t-k} = \phi_1 x_{t-1} x_{t-k} + \phi_2 x_{t-2} x_{t-k} + \dots + \phi_r x_{t-r} x_{t-k} + u_t x_{t-k} \quad (33)$$

Taking the expectation of the equation (33), this yield to the following equation

$$E(x_t x_{t-k}) = \phi_1 E(x_{t-1} x_{t-k}) + \phi_2 E(x_{t-2} x_{t-k}) + \dots + \phi_r E(x_{t-r} x_{t-k}) + E(u_t x_{t-k})$$

But  $E(x_{t-r} x_{t-k}) = \gamma_{k-r}$ , then we have the following

$$\gamma_k = \phi_1 \gamma_{k-1} + \phi_2 \gamma_{k-2} + \dots + \phi_r \gamma_{k-r} \quad (34)$$

The Expectation  $E(u_t x_{t-k})$  vanishes because, for  $k > 0$ ,  $x_{t-k}$  involves the shocks up to time  $t - k$ ; the shock of the current time is uncorrelated with the previous values of the process hence  $E(u_t x_{t-k}) = 0$ . Normalization follows and this is done by dividing throughout the equation (34) by autocovariance at lag 0 ( $\gamma_0$ ). Note that the autocorrelation at lag 0 is 1, that is,  $\rho_0 = 1$  because  $k = 0$ , which is substituted to equation (9). The resulting equation, after normalization, is

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \dots + \phi_r \rho_{k-r} \quad (35)$$

When  $k = 1, 2, \dots, r$  is substituted to the equation (35), the following set of equations is obtained

$$\begin{aligned}\rho_1 &= \phi_1 + \phi_2\rho_1 + \dots + \phi_r\rho_{r-1} \\ \rho_2 &= \phi_1\rho_1 + \phi_2 + \dots + \phi_r\rho_{r-2} \\ &\vdots \\ \rho_r &= \phi_1\rho_{k-1} + \phi_2\rho_{k-2} + \dots + \phi_r\end{aligned}\tag{36}$$

The equations (36) are called Yule-Walker equations. The Yule-Walker equations relate the parameters of  $AR(p)$  to its serial correlations. The equations show that the auto-correlations of a process follow the same difference equations as the process itself. The estimates  $\hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_r$  can easily be obtained by replacing the theoretical autocorrelation,  $\rho_k$ , by the estimated autocorrelation,  $R_k$ . Therefore, the estimates are obtained by solving the linear system of equations (37) which is written in a matrix form, equation (38)

$$\begin{aligned}R_1 &= \phi_1 + \phi_2R_1 + \dots + \phi_rR_{r-1} \\ R_2 &= \phi_1R_1 + \phi_2 + \dots + \phi_rR_{r-2} \\ &\vdots \\ R_r &= \phi_1R_{k-1} + \phi_2R_{k-2} + \dots + \phi_r\end{aligned}\tag{37}$$

The system of equations (37) can be written as

$$\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_r \end{bmatrix} = \begin{bmatrix} 1 & R_1 & \cdots & R_{r-1} \\ R_1 & 1 & \cdots & R_{r-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{r-1} & R_{r-2} & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \vdots \\ \hat{\phi}_r \end{bmatrix}\tag{38}$$

From the equations, it is easy to see that, for the case of AR(1), the Yule-Walker equations give the parameter estimate value as the value of autocorrelation at lag 1, this can be verified by direct inversion method. To see how the Yule-Walker equations estimate the parameters, take an example of  $ARMA(3, 0)$ . The original parameters were 0.3000, -0.7000 and 0.5000 (see Table (3)). The estimated values by MATLAB GARCH Toolbox are shown in Table (3). By using Yule-Walker equations, the estimated values are 0.3108, -0.6839 and 0.5050. These estimates are approximately equal to the estimates obtained by GARCH Toolbox. The estimates do not differ much compared to the original parameters.

## 4.5 Observation

To compare the estimated and original value, one may check the average performance; this is the average difference between the estimates and real parameters. It, also, helps to compare the accuracy of the method. Table (2) shows the values of the average differences from recursive method (15) and GARCH Toolbox.

Table 2: Average performance of different estimation methods.

Model	ARMA(1,0)	ARMA(2,1)	
Recursive	-0.0014	theta1	-0.0015
		theta2	0.0009
		phi	0.0014
GARCH Toolbox	-0.0024	theta1	0.0009
		theta2	-0.0009
		phi	-0.0510

From the table we can see that the recursive estimation method gives better results: the multiple runs give results that are close to the original parameters values, hence one can use this method in parameter estimations.

GARCHFIT.M: Using the built-in MATLAB function garchfit.m, the results are close to the original parameter values, indicating that it is good to use the `garchfit` function to do the estimation.

The Yule-Walker equation approximate well the autoregressive parameters. The values estimated are close to the real one. However the Yule-Walker estimates are known to often be highly biased. It is possible to have the bias expression of Yule-Walker estimates.

Note that ([7]), if the iterations fail to converge to the correct value, this can depend both on the guess values used to start the estimation and on the extend of over specification in the model. Nonlinearity disappears as soon as the model contains no MA terms.

## 4.6 Parameter Estimation by MCMC

The study of the recursive and GARCH methods have shown that they estimate the parameters of *ARMA* models well. Similarly, the Yule-Walker equations, in Section 2, estimated the *AR* parameters well. In this subsection, the study is focused on the MCMC algorithm estimating the parameters or *ARMA* models and verify their reliability.

Example; consider the model *ARMA*(1,0) with original parameter  $\theta = 0.7$  and  $x_0 = 3$ , while White noise are normally distributed  $N(0,1)$ . We run MCMC chain using a single realization with the objective function given by (15). Figure (13) shows the normalized histogram for the estimate, from which we can conclude that the estimate looks fine. Its mean is 0.7025 and the maximal parameter estimate and minimal parameter estimate have distances from the mean as 0.3805 and 0.2799 respectively. The same distances are proportional to standard deviations with values 3.8764 and 2.8517 for maximal and minimal parameter estimates respectively. The R2 value is around 0.97. One can compare this figure with Figure (9) and Figure (11); they look approximately similar, differing

from the axes scales. The y-axis scale for the *MCMC* estimation is determined by the proposed initial standard deviation.

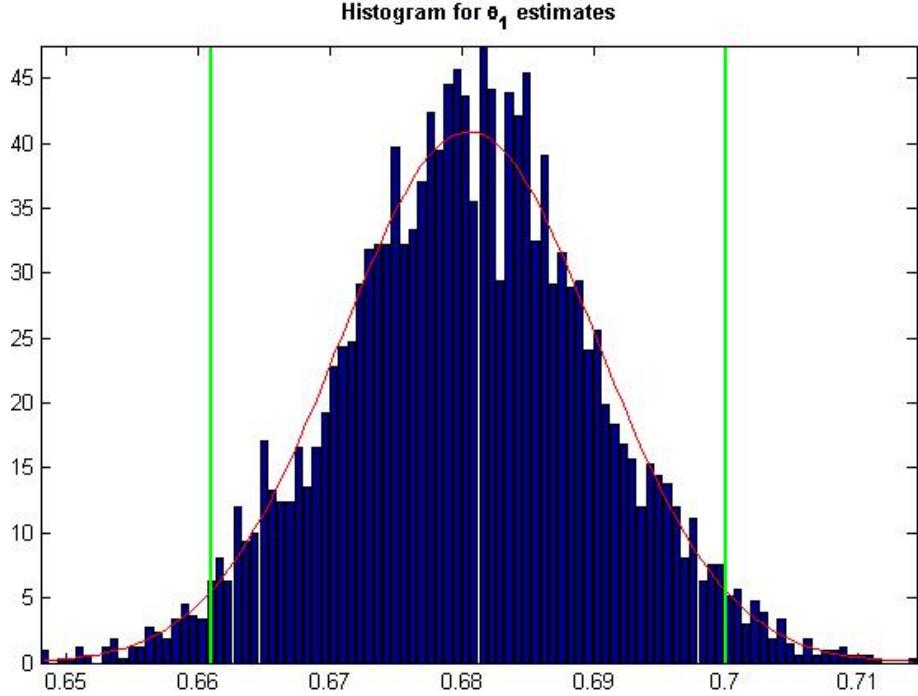


Figure 13: Normalized histogram of  $\theta$  estimates obtained from MCMC estimation (original  $\theta = 0.7$ ).

For the case of *ARMA*(2, 1), the original parameters are  $\theta_1 = 0.58$ ,  $\theta_2 = -0.4$ ,  $\psi = 0.6$ ,  $x_0 = 3$ ,  $x_1 = 2.5$ , and White noise being normally distributed  $N(0,1)$ . The normalized histograms of the estimates are seen in Figure (14). The figure shows that the estimates are good, approximately equal to original parameters. The means of  $\theta_1$ ,  $\theta_2$  and  $\psi$  are 0.5805, -0.3993 and 0.5995 respectively, with minimal parameter estimates having distances from their respective means as 0.0374, 0.0298 and 0.0343 while the maximal parameter estimates have distances from their respective means, from their respective means, as 0.0370, 0.0338 and 0.0309. The minimal parameter estimates have the distances, with respect to their distances from the mean, proportional to standard deviations with values 3.8434, 3.2512 and 3.3774 respectively. For the maximal estimates, the values are 3.8010, 3.6892 and 3.0462 respectively. As with the *ARMA*(1,0), the R2 value is around 0.97. One may compare the Figure (14) with the Figure (10) and the Figure (12); they look approximately similar, except the axes scales differ. The MCMC generates, from a single realization, that corresponds to a real time series measurements, practically, the same parameters posteriors as those from repeated estimation with synthetic data.

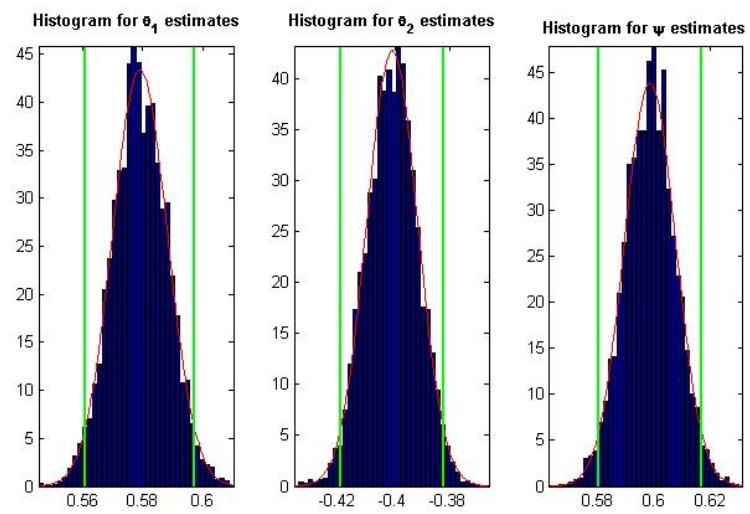


Figure 14: Normalized histogram of  $\theta_1, \theta_2, \psi$  estimates ARMA(2,1) by MCMC.

## 5 ARMA Model Fitting, Forecasting and Identification of $p$ and $q$ by RJMCMC

### 5.1 Model identification for given examples

The identification part, selection of  $p$  and  $q$  of *AR* and *MA* respectively, is one of the most difficult of the Box-Jenkins steps because it is subjective in nature. For this study, two cases were performed; the first case is when the data set, series X, from the known model (ARMA(3,0)) was generated, aiming to identify the order of the model. The data were given to a person who does not know the true model. The person pre-analyzed the data testing for stationarity and then drew the autocorrelation and partial autocorrelation plots for identifying the model order. Figure (15) shows the sample ACF and sample PACF drawn.

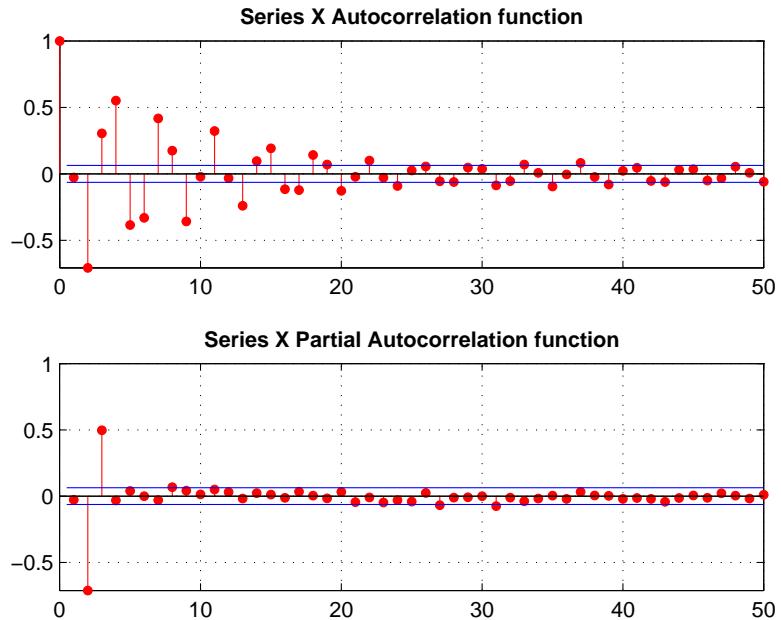


Figure 15: Sample ACF and PACF for series X.

Similarly, for the second case the data set, series Y, was generated from the known model (ARMA(2,1)) and given to the person who does not know the true model. As in the first case, the person checked the stationarity and drew the plots. Figure 16 shows the sample ACF and sample PACF drawn.

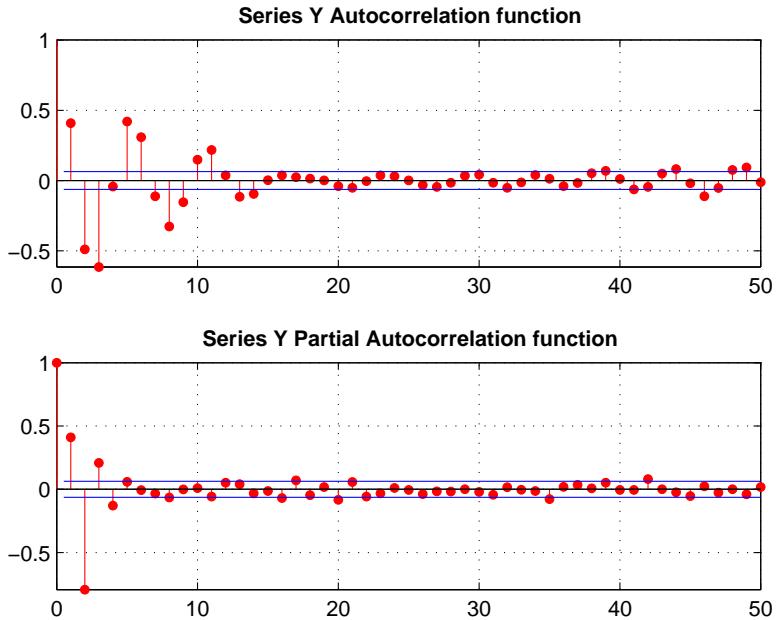


Figure 16: Sample ACF and PACF for series Y.

The person was asked to tell the model where the data came from. He identified the model by studying the graphs sample ACF and PACF. He suggested the possible model for the first case as the ARMA(3,0) because the ACF of Series X decays towards zero, while the PACF has the only significant values only up to lag 3. Table (3) collects the real and estimated parameters for model ARMA(3,0).

Table 3: Original and estimated parameters for ARMA(3,0).

	Original parameters	Estimated parameters
AR(1)	0.3000	0.3098
AR(2)	-0.7000	-0.6861
AR(3)	0.5000	0.5062

For the second case, the possible models suggested were ARMA(2,3) and ARMA(2,5), whereas the true one used for data generation was ARMA(2,1). The suggestions were based on the visualization of the graph. For the AR components, 2 was chosen because the PACF has the most significant value at lag 2. For the MA components, 3 and 5 were chosen because of the lags; that is there is lag 3 shows the highest significance, whereas the farthest significant value seems to be the one at lag 5.

The parameters relations for the second case are shown in the Table (4). We can see that for the suggested models, estimates of MA components from second up to 5th one are

significantly lower than the other ones, while values for AR(1), AR(2) and MA(1) remain on a similar level for all 3 models. Therefore, we suggest that this may be a possible way of dealing with order uncertainty. When not being sure about proper number of model components, one can estimate a few different suggestions and pick the most reliable one based on the estimates' significance.

Table 4: Parameter estimates for chosen possible models for Series Y.

parameters	Original	ARMA(2,1)	ARMA(2,3)	ARMA(2,5)
AR(1)	0.6	0.65	0.66	
AR(2)	-0.7	-0.72	-0.74	
MA(1)	0.3	0.28	0.27	
MA(2)	—	-0.095	-0.09	
MA(3)	—	0.02	0.01	
MA(4)	—	—	0.02	
MA(5)	—	—	-0.02	

If the model has no MA components, one can easily determine the order of it by studying the graph of ACF and PACF. This is noticed to this person who gave a correct model. However, if the model contains both AR and MA components it is a difficult task to identify the order from the graph. A person suggested different models while the data came from a single model. Therefore, there is a need for the other method apart from graphical method that should be used to identify the model that contain the MA and AR components. In the next sub section the suggested models are fitted and forecasts are analyzed. This is done by both ways, classical time series and MCMC approach. The fitting and forecasting and forecasting errors. The RJMCMC subsection addressed as the part that tells the RJMCMC machinery approach that is used to verify whether it is possible to identify the true model from the sample of models.

## 5.2 Model Fitting by Classical Time Series

To study the effect of white noises, consider series X. Two cases are considered. One case is when the estimated data set are generated while the white noise are kept constant, that is, the noise used in generation of X are used as they are, no new noise is generated. The initial points for original and estimated data are the same. The estimated parameters are obtained by using a GARCH Toolbox, however, one can use the recursive formulas. Figure (17) shows the original and estimated model. The estimate captures the original one such that the value of the variance of residual is small though they differ after every run/simulation because of white noise having unfixed values.

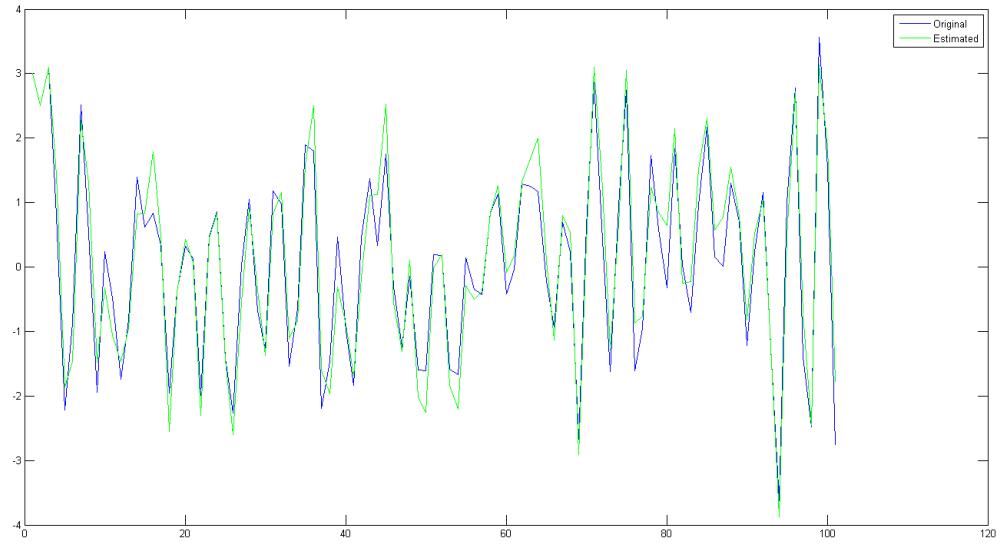


Figure 17: Original and Estimated data set.

When new noise is used in simulating the model with the estimated parameters, Figure (18) shows the relationship between the simulated and original data. The estimated graph does not capture the original one, however, at the beginning the graphs are almost in the same path. This indicate that the *ARMA* models estimates deviate much when new noises are used, if the prediction time interval is large enough. The noises come from the white noise,  $u_t$ , and from the parameters estimated. However, as seen, the noises that affect much the fitting are  $u_t$  noises.

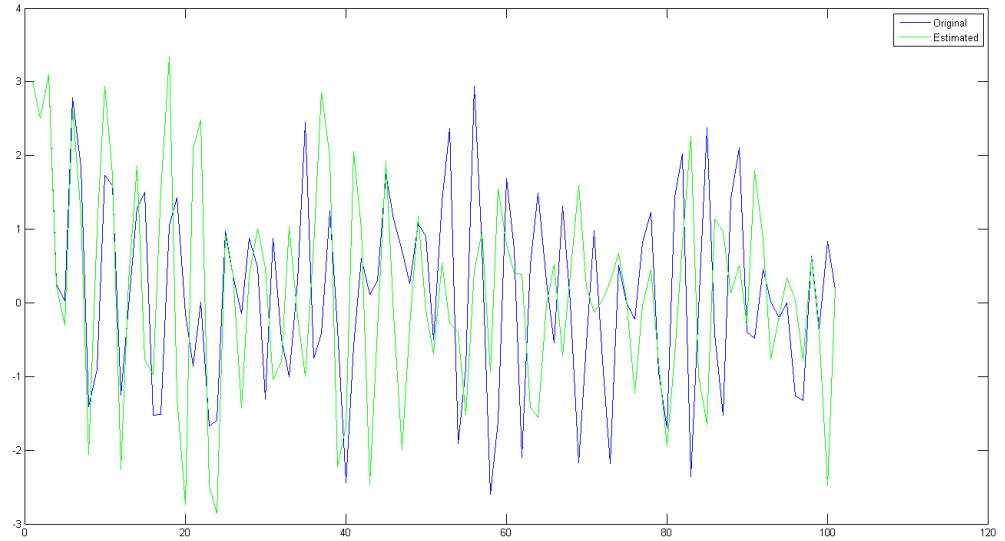


Figure 18: Original and Estimated Data Set With different White Noise.

### 5.3 Model fitting by MCMC

In this subsection, the MCMC is used to fit the model. The objective function used is the recursive sum of residuals (15). Two models ( $ARMA(2, 1)$  and  $ARMA(3, 0)$ ) are taken into account. The results by MCMC can be seen in Figure (19) for  $ARMA(2, 1)$ . The figure shows that the estimated data fits well the original data. This concludes that the MCMC confirms good  $ARMA$  fit. However, as soon as new noise values are used, the fitted paths construct a lot wider envelope around the original data, this is shown in the Figure (20). Figure (19) has two parts, the second part shows a zoomed portion of the first part. The zoomed part shows clearly (for visibility) the concentration of the fits to the original model. Note that in both models, there are 5000 *MCMC* samples.

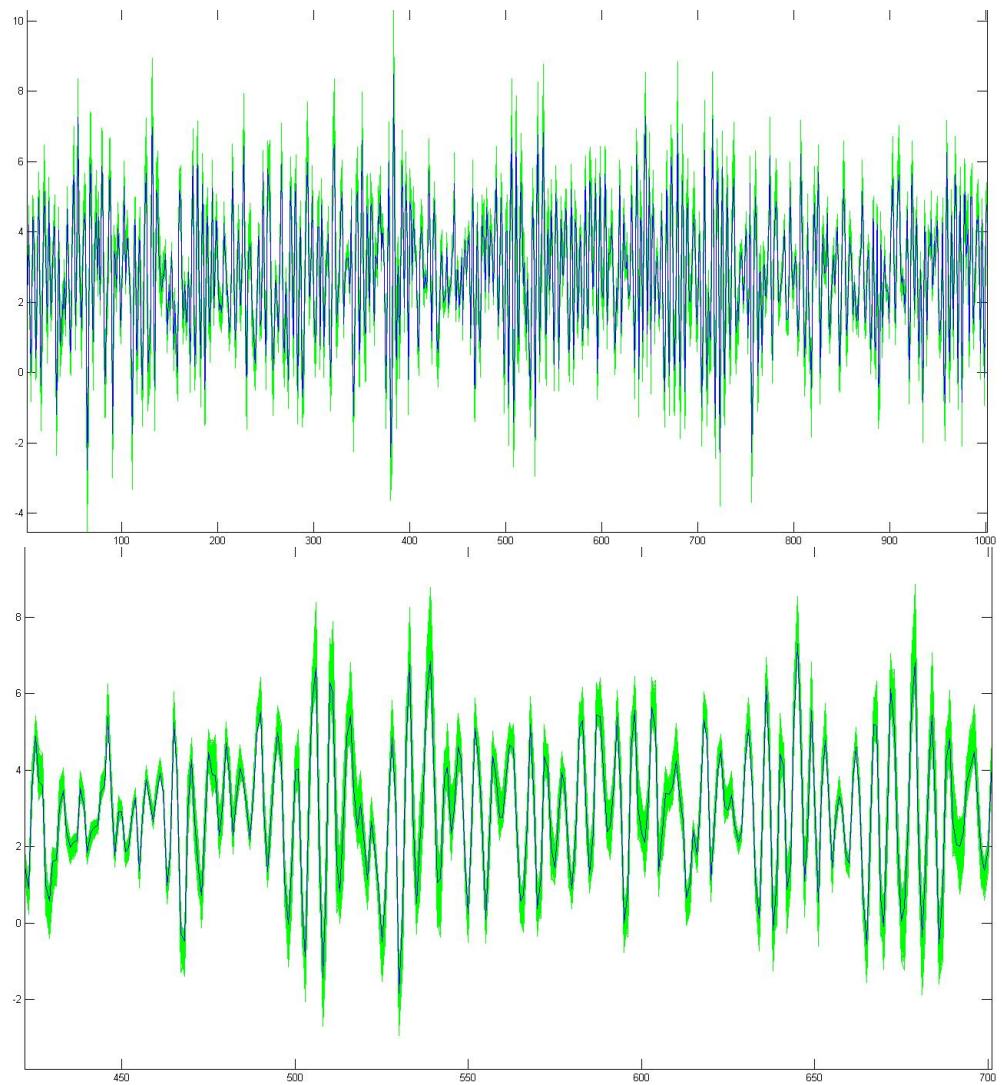


Figure 19: MCMC fitting of  $ARMA(2, 1)$ .

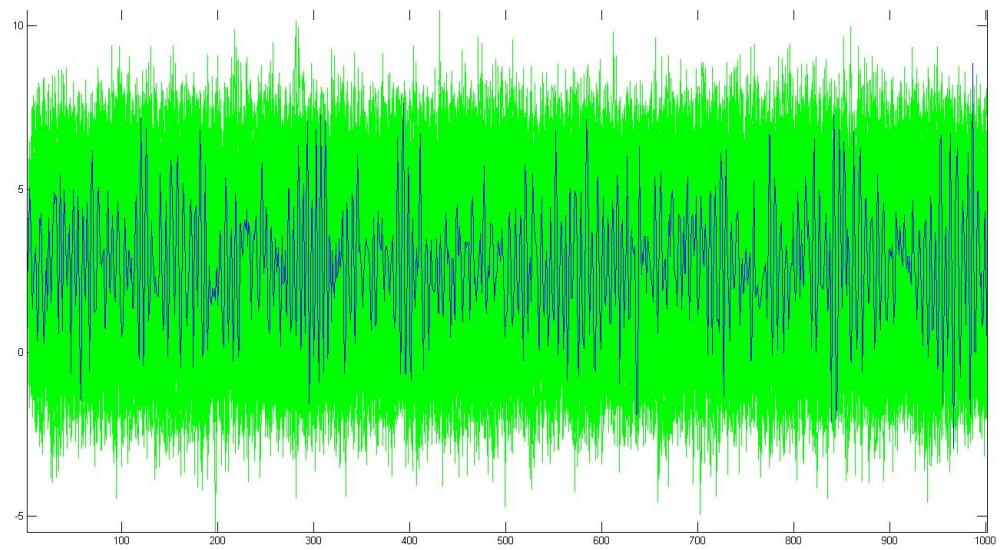


Figure 20: MCMC fitting of  $ARMA(2, 1)$  with different noises.

The MCMC fitting is done, also, to the X data. As in Figure (19), the Figure (21) shows how the MCMC fits better the model, without new noises. Figure (22) shows the simulations when the noises are regenerated; the enveloping, by estimated paths, of original model is seen in this model. The model, even though it has MA components or not, still has poor prediction power for long enough time window.

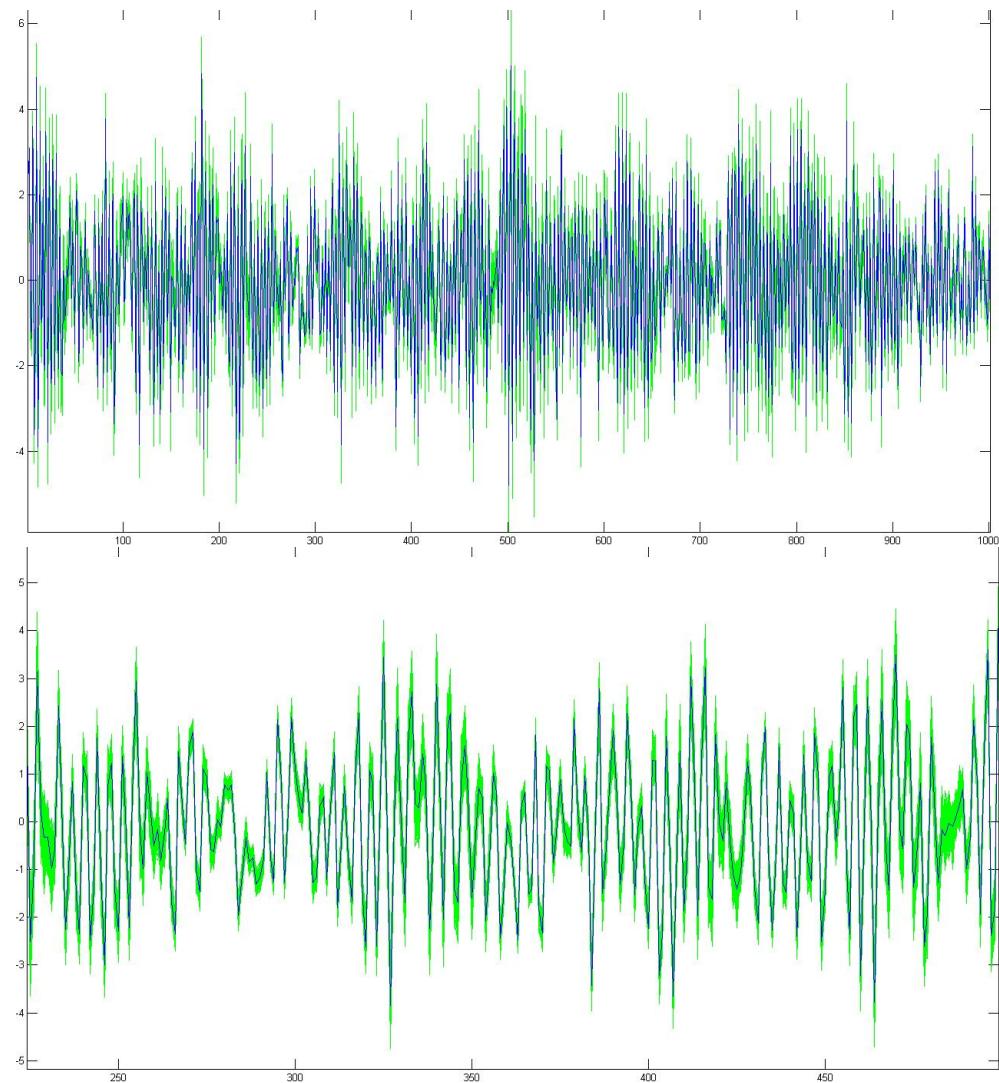


Figure 21: MCMC fitting of  $ARMA(3,0)$  without new noises.

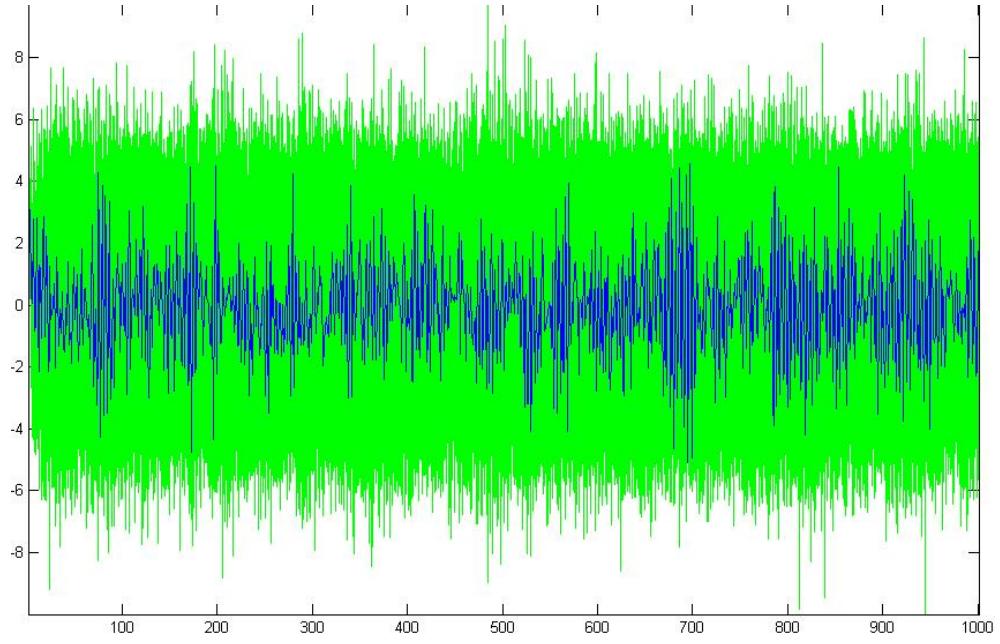


Figure 22: MCMC fitting of  $ARMA(3,0)$  with new noises.

#### 5.4 Forecasting

The objective of building a time series model is to be able to forecast the values for that series at future times. For the series X, the possible model was fitted by leaving out the last 30 observations, which then are used for forecasting. The forecast and the original data is seen in Figure (23) (the blue line shows the original data while the other are forecasts). From the figure, it shows the further we go, the more the forecasts spread.

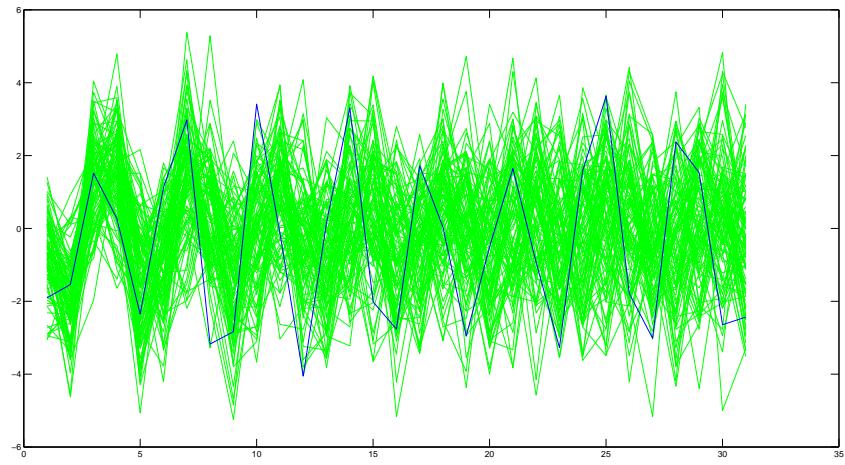


Figure 23: Original data and Forecasts of possible model of series X.

When it comes to series Y, for the first model possible by ACF and PACF,  $ARMA(2,3)$ ,

the forecast can be seen in Figure 24. Similarly, the graph for the other possible model, ARMA(2,5) showing the forecasts is seen in Figure (25). Figure (26) shows the forecast for the real model, ARMA(2,1), all obtained by an iterative computation with use of estimated parameters. The forecasts start well, that is they follow the original model at the beginning but as they go, they do not capture well (the forecasts spread).

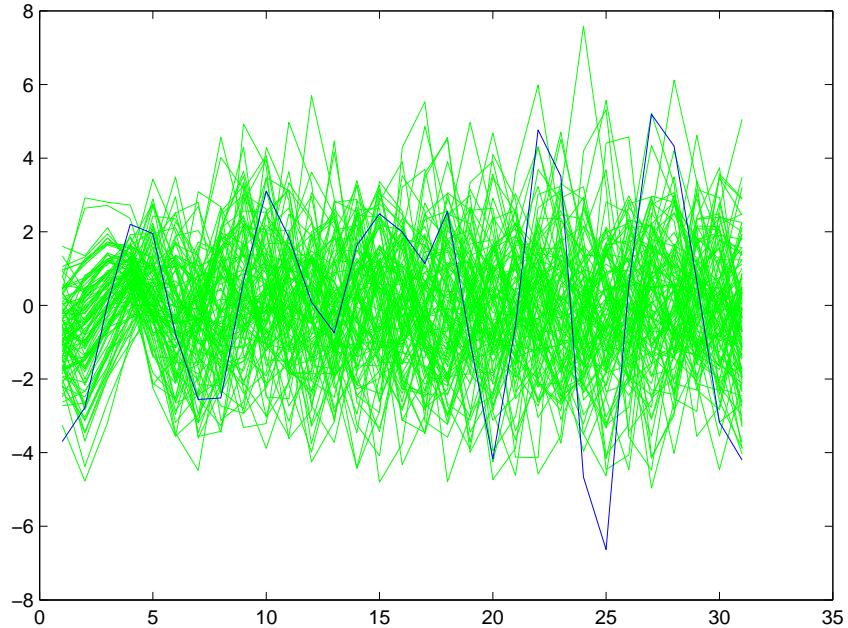


Figure 24: Original data and Forecasts of ARMA(2,3).

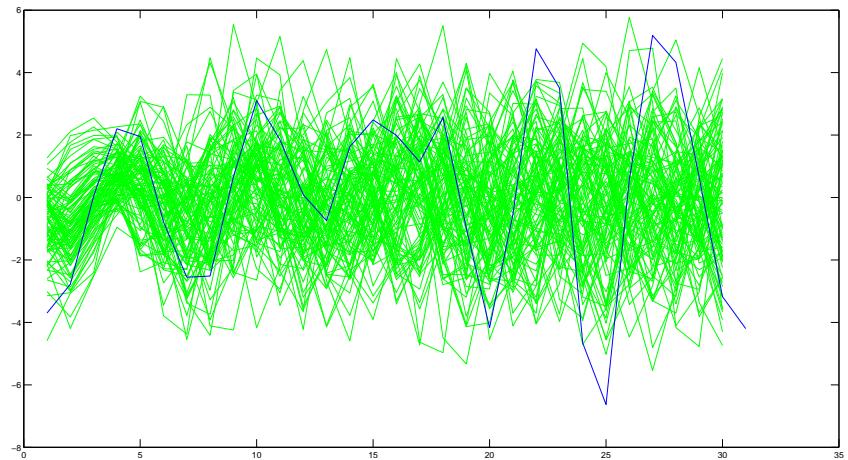


Figure 25: Original data and Forecasts of ARMA(2,5).

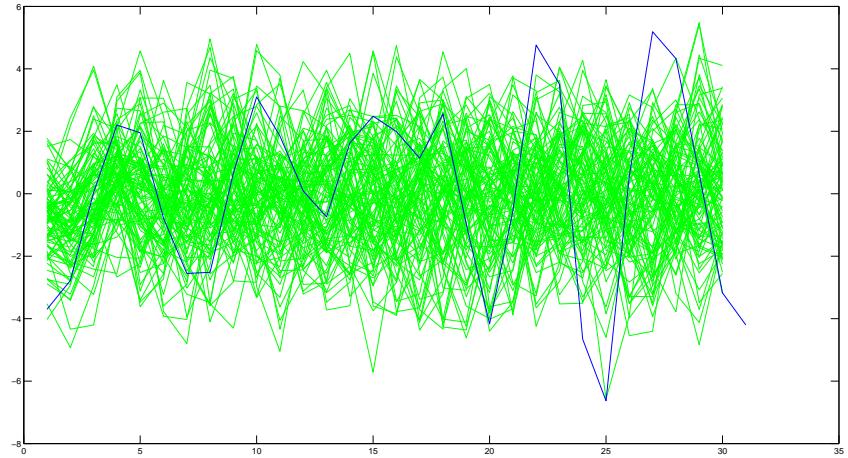


Figure 26: Original data and Forecasts of ARMA(2,1).

The forecast can also be done with the help of a built-in MATLAB function, here the simulation of data is done with the help of `garchsim.m`. The study here is to forecast 30 steps ahead and only the suggested models are presented. To study this, the model possible, through studying ACF and PACF, for the first case ( $ARMA(3,0)$ ) is taken as well as all possible models for the second case, ( $ARMA(2,3)$ , and  $ARMA(2,5)$ ). The sample size is 100. The resulting forecasts are seen in the Figure (27), (28) and (29) respectively, where the blue line shows the original data and the other ones are forecasts. As in iterative method, the forecasts start well and they spreads as they go further. They do not follow the original data.

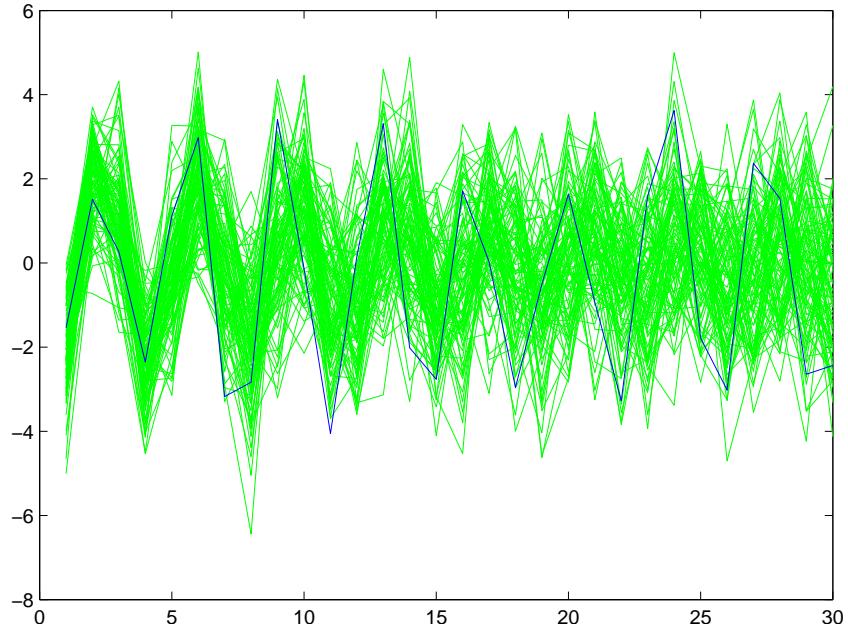


Figure 27: ARMA(3,0): Forecasts and the Original data.

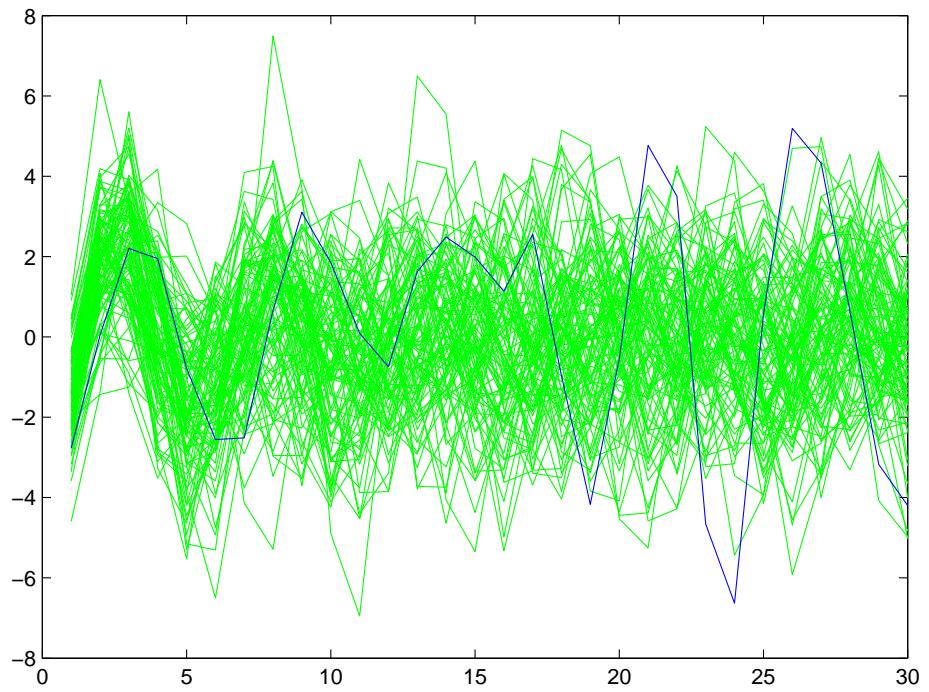


Figure 28: ARMA(2,3): Forecasts for Sample = 100.

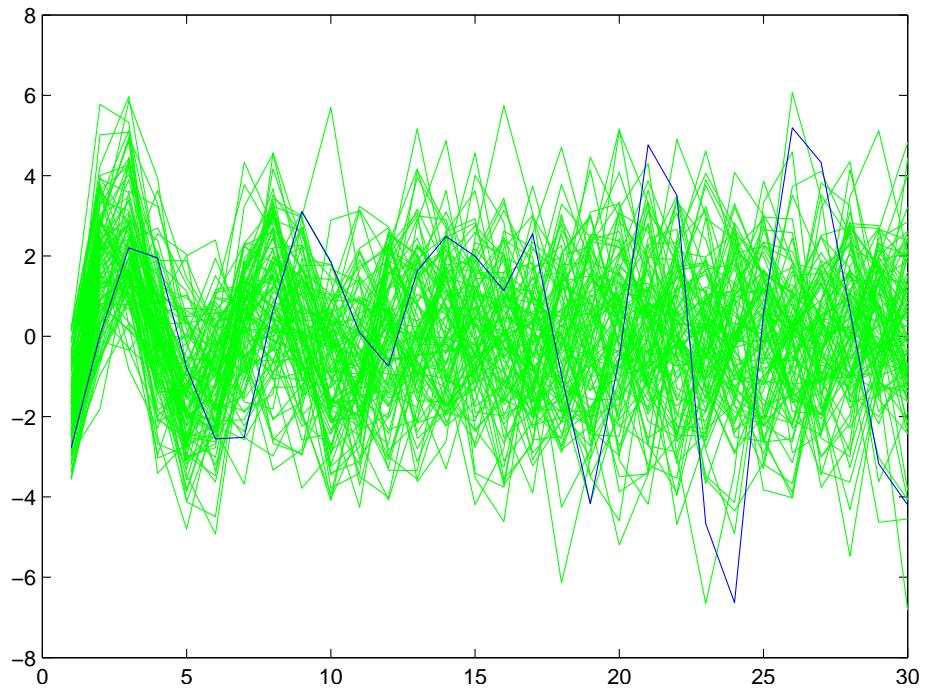


Figure 29: ARMA(2,5): Forecasts for Sample = 100.

## 5.5 Forecasting Errors

The forecasts seem to vary depending on departure from the true model, that is, the further the orders the more the forecasts mismatch with the original data. The forecasting errors are measured; these are root mean square error (RMSE), the mean absolute deviation (MAD), the mean squared error (MSE), the mean absolute percentage error (MAPE) and the mean percentage error (MPE). For the first case, without the use of *garchsim.m*, the errors (rounded to 2 decimal places) are show in Table (5), while Table (6) shows these errors by *garchsim.m* (excluding the errors for original models). The MAD is the mean of the absolute errors that measures the forecast accuracy; by averaging the absolute deviations of observations from their forecast (forecast errors). The MSE is the average of the square of the residuals, from which its square root is the RMSE. The MAPE is the expression of forecasting errors in a percentage form while the MPE determines how the forecasting method is biased, whether is consistently low or high. If MPE is close to zero then the forecasting method is unbiased.

Table 5: Measuring forecasting error.

Models	RMSE	MAD	MSE	MAPE	MPE
ARMA(3,0)	2.98	2.40	9.16	3.20	3.44
ARMA(2,3)	3.42	2.91	11.88	2.48	2.76
ARMA(2,5)	3.42	2.91	11.89	2.46	2.79

Table 6: Measuring forecasting error; the case with *garchsim.m*.

Models	RMSE	MAD	MSE	MAPE	MPE
Original ARMA(3,0)	2.53	2.14	6.50	1.70	2.13
ARMA(3,0)	2.52	2.12	6.41	1.64	2.11
Original ARMA(2,1)	3.25	2.72	10.67	1.73	1.86
ARMA(2,3)	3.25	2.88	10.63	1.88	2.17
ARMA(2,5)	3.25	2.75	10.68	1.86	2.16

Note that the errors were obtained after more than one compilation of algorithm. Each compilation had its error values. The error values picked are those that are close to each other. From the errors, the forecasts seem not to deviate much from each possible model. For instance, there are small differences between the errors of ARMA(2,1) compared to its possible models. However, all forecasts seem to perform poorly, mainly due to error terms (white noise component) typical for ARMA models.

## 5.6 Forecasts by MCMC

The MCMC method helps to construct predictive distributions of data and forecast parameters, *i.e.* it may be used for verification of prediction reliability. In this work/study, the MCMC determine the correlation of parameters and forecast predictive distribution. For the case of ARMA(3,0) the scatter plot for series Y is seen in Figure (30), where as the forecast predictive distribution is Figure (31). The scatter plot looks like there exist correlation between parameters. This is evidenced by studying the values of correlation of parameters in Table (7).

For the case of forecast predictive distribution, the capturing to the original series differs. For the possible model of series X, *ARMA*(3,0), the scatter plot and the forecast predictive distribution of this possible model are Figure (32) and Figure (33). The correlation values between parameters are in Table 8. The correlation of the parameters for the original model *ARMA*(3,0) and its possible model are the same (see Table 7 and Table (8)) even though their AR components and MA components are different (see Table (3)). Similarly the forecast predictive distributions of original *ARMA*(3,0) and its possible model are the same, see Figures (31) and (33).

When it comes to the comparison of original series against forecast predictive distribution (Figure (31)), one can notice that the predicted paths look alike the real data only in the beginning of forecasting horizon. The further it goes, the more the predicted values mismatch from the true ones. Also, two more phenomena are visible. Firstly, the variance of parameter chains is significantly lower than the forecast residuals, *i.e.* the spread of predictive paths is smaller than the differences between the original and predicted time series observations. Secondly, the envelope of realizations produced by MCMC is wider along the forecasting horizon, than within the training data fit.

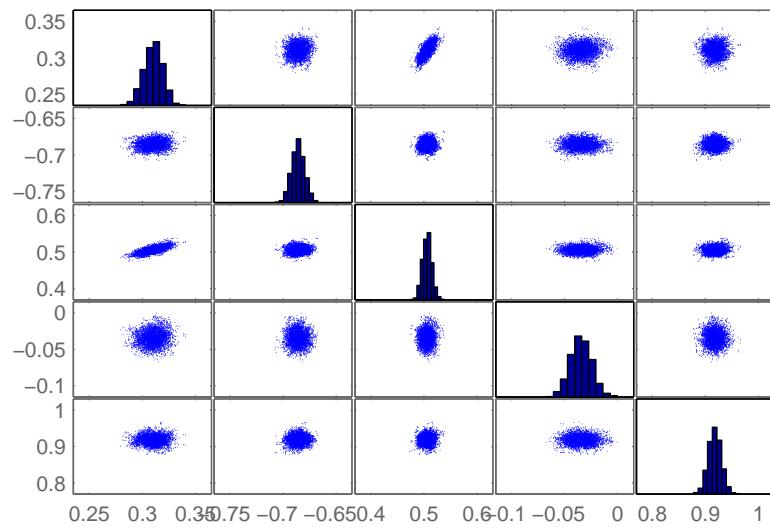


Figure 30: Original ARMA(3,0): Scatter Plots for series X.

Table 7: The Correlation Between Parameters of original ARMA(3,0).

AR(1)	AR(2)	AR(3)	C	K
1.0000	0.0936	0.7113	0.0993	0.0241
0.0936	1.0000	0.0804	-0.0075	0.0530
0.7113	0.0804	1.0000	0.0254	0.0735
0.0993	-0.0075	0.0254	1.0000	-0.0119
0.0241	0.0530	0.0735	-0.0119	1.0000

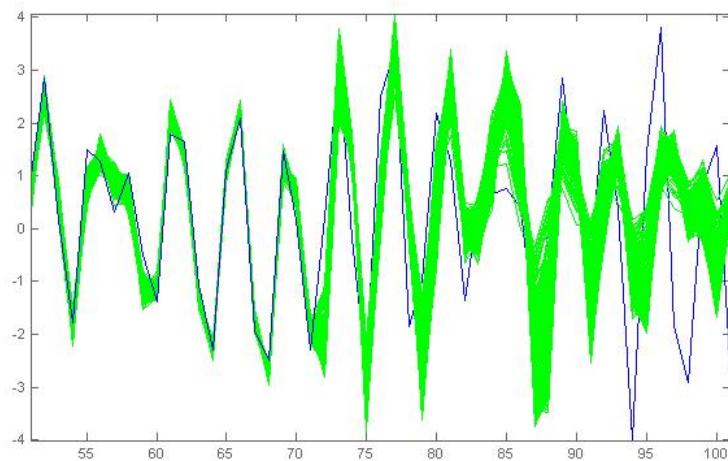


Figure 31: Original ARMA(3,0): Forecast Predictive Distribution for series X.

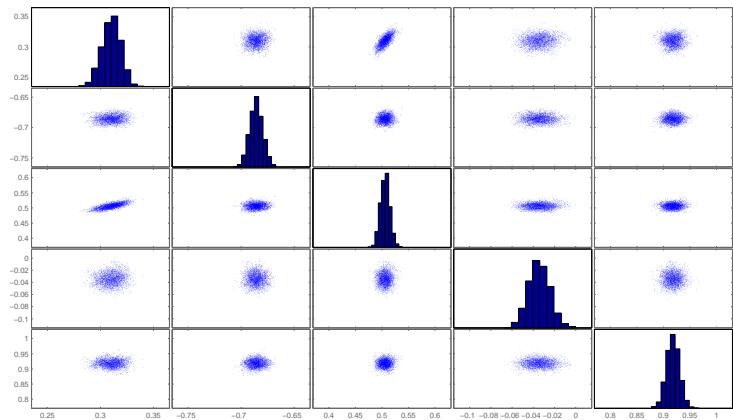


Figure 32: ARMA(3,0): Scatter Plots for series X.

Table 8: The Correlation Between Parameters of ARMA(3,0).

AR(1)	AR(2)	AR(3)	C	K
1.0000	0.0936	0.7113	0.0993	0.0241
0.0936	1.0000	0.0804	-0.0075	0.0530
0.7113	0.0804	1.0000	0.0254	0.0735
0.0993	-0.0075	0.0254	1.0000	-0.0119
0.0241	0.0530	0.0735	-0.0119	1.0000

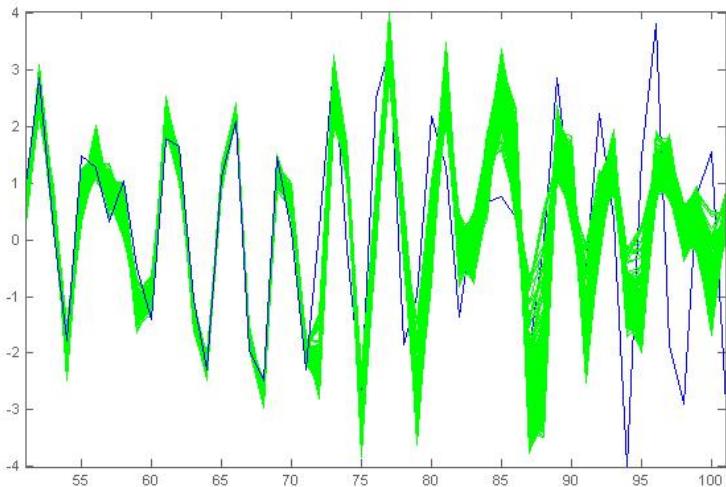


Figure 33: ARMA(3,0): Forecast Predictive Distribution for series X.

Figure (34), and Figure (35) show the scatter plots and the forecast predictive distributions of the original model  $ARMA(2,1)$  whose parameters are 0.6 and -0.7 for AR and 0.3 for MA. The Table 9 is the correlation table that shows the values of correlation between each parameter with itself and then with other parameters.

Regarding the forecasts, it is visible that the predicted MCMC paths are different in values already from the beginning of the forecasting horizon. When compared to the ARMA(3,0) case, it can be easily explained by the more noise included in the model. Note that ARMA(2,1) model contains not only the series lagged values, but also the lagged white noises.

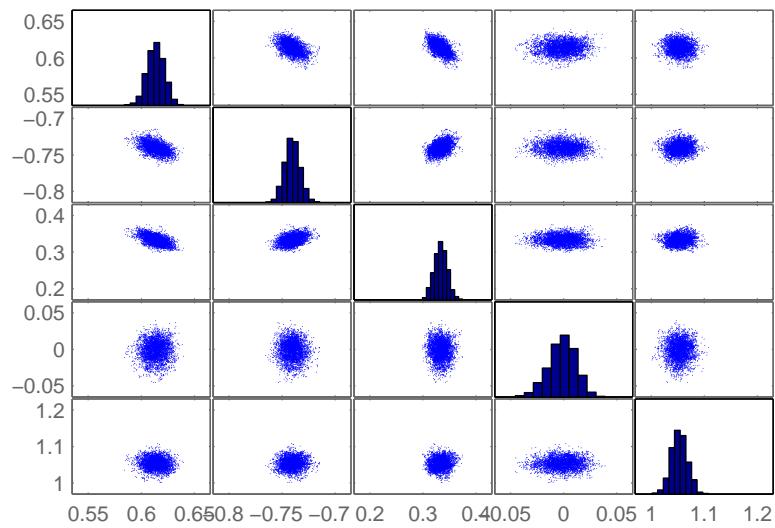


Figure 34: ARMA(2,1): Scatter Plots for series Y.

Table 9: The Correlation Between Parameters of ARMA(2,1).

AR(1)	AR(2)	MA	C	K
1.0000	-0.4883	-0.5686	0.0582	-0.0474
-0.4883	1.0000	0.4023	-0.0401	0.0638
-0.5686	0.4023	1.0000	-0.0514	0.0832
0.0582	-0.0401	-0.0514	1.0000	0.0301
-0.0474	0.0638	0.0832	0.0301	1.0000

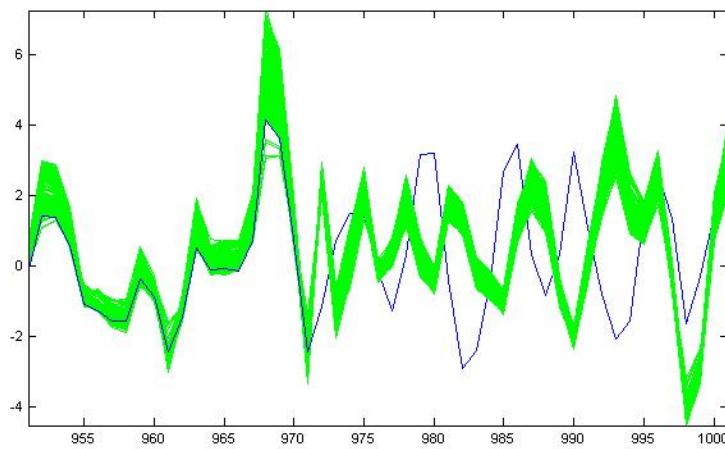


Figure 35: ARMA(2,1): Forecast Predictive Distribution for series Y.

The scatter plots tell whether the parameters are correlated with each other. If the

cumulated part looks circular or ellipsoidal with the axes vertically or horizontally, then there is no correlation. For these series, look like there is no correlation of parameters. The forecast predictive distribution is seen not able to capture the real data.

The two possible models identified during the study of  $ARMA(2, 1)$  are  $ARMA(2, 3)$  and  $ARMA(2, 5)$ . With  $ARMA(2, 3)$ , the scatter plot for series Y is in Figure (36), while the forecast predictive distribution can be seen in Figure 37. Similarly, Figure (38) and Figure (39) are for  $ARMA(2, 5)$  showing the scatter plot and the forecast predictive distribution. The scatter plots show that there exist some negative correlation between some parameters. For instance, Figure 39, there is negative correlation between parameters whose plots are in ellipsoidal shape bent by certain angle clockwise. See the shapes between the ( $AR(1)$ ) and ( $MA(1)$ ), ( $AR(2)$ ) and ( $MA(3)$ ) parameters. Similarly shapes with ellipsoidal shape bent by certain angle anticlockwise has positive correlation, for instance see the shapes between the ( $AR(1)$ ) and ( $MA(4)$ ) parameters. The same study and interpretation can be done in Table 10. The correlation between a parameter with itself and with other parameters is tabulated in Table (10) for  $ARMA(2, 3)$  and Table (11) for  $ARMA(2, 5)$ .

Similarly to the case of true model  $ARMA(2,1)$ , the forecasts predictive distributions for both  $ARMA(2,3)$  and  $ARMA(2,5)$  mismatch from the true values immediately from the first predicted step. Also, as visible particularly for the former of the two models, the envelope created by MCMC realizations is wider than within the true data fit, though still a lot narrower than the forecast residuals.

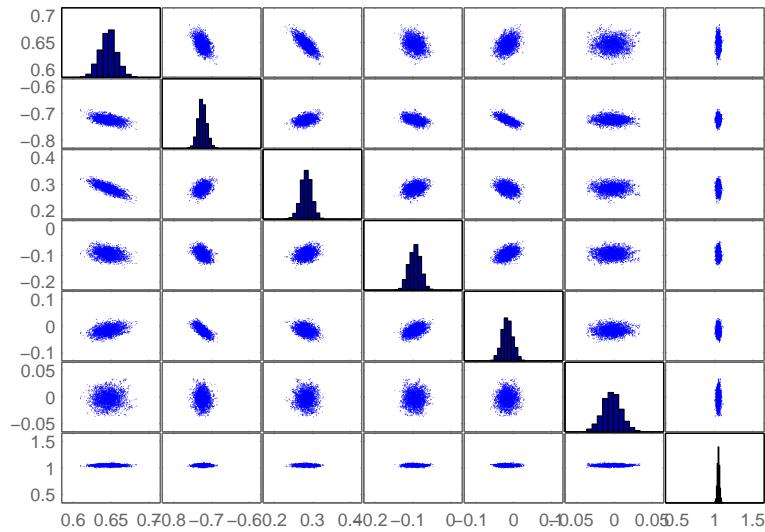


Figure 36:  $ARMA(2,3)$ : Scatter Plots for series Y.

Table 10: The Correlation Between Parameters of ARMA(2,3).

AR(1)	AR(2)	MA(1)	MA(2)	MA(3)	C	K
1.0000	-0.4490	-0.7651	-0.2794	0.3457	0.0418	0.0414
-0.4490	1.0000	0.3586	-0.4562	-0.7234	-0.0573	0.0022
-0.7651	0.3586	1.0000	0.2970	-0.3490	-0.0086	-0.0817
-0.2794	-0.4562	0.2970	1.0000	0.4460	0.0193	-0.0371
0.3457	-0.7234	-0.3490	0.4460	1.0000	0.0480	-0.0387
0.0418	-0.0573	-0.0086	0.0193	0.0480	1.0000	0.0218
0.0414	0.0022	-0.0817	-0.0371	-0.0387	0.0218	1.0000

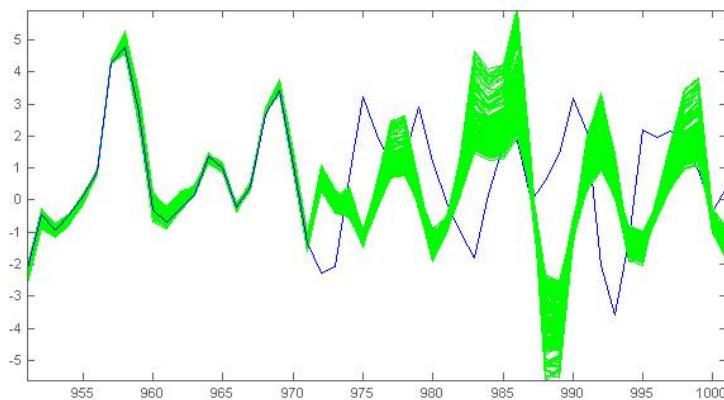


Figure 37: ARMA(2,3): Forecast Predictive Distribution for series Y.

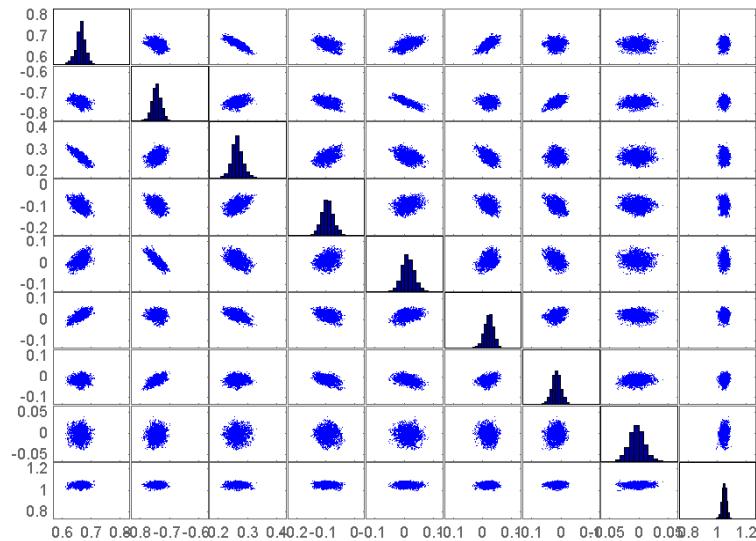


Figure 38: ARMA(2,5): Scatter Plots for series Y.

Table 11: The Correlation Between Parameters of ARMA(2,5).

AR(1)	AR(2)	MA(1)	MA(2)	MA(3)	MA(4)	MA(5)	C	K
1.0000	-0.3878	-0.8396	-0.4584	0.4681	0.6694	0.0494	-0.0946	0.0495
-0.3878	1.0000	0.3081	-0.4504	-0.8279	-0.1101	0.5715	0.1253	0.0464
-0.8396	0.3081	1.0000	0.4813	-0.3968	-0.5482	-0.0340	0.0529	-0.0214
-0.4584	-0.4504	0.4813	1.0000	0.3225	-0.4068	-0.4633	-0.0485	-0.0792
0.4681	-0.8279	-0.3968	0.3225	1.0000	0.2950	-0.4622	-0.0526	-0.0308
0.6694	-0.1101	-0.5482	-0.4068	0.2950	1.0000	0.3534	-0.1030	0.0401
0.0494	0.5715	-0.0340	-0.4633	-0.4622	0.3534	1.0000	0.0646	0.0421
-0.0946	0.1253	0.0529	-0.0485	-0.0526	-0.1030	0.0646	1.0000	0.1075
0.0495	0.0464	-0.0214	-0.0792	-0.0308	0.0401	0.0421	0.1075	1.0000

The correlation between the first parameter of *AR* components, (*AR*(1)), with first parameter of *MA* components, *MA*(1), is highly negative compared with other correlations. It is followed by the correlation between the first parameter of *AR* components, (*AR*(2)), with first parameter of *MA* components, *MA*(3). There is less correlation of *K* parameter with other parameter, see the values in *K* column Table 11.

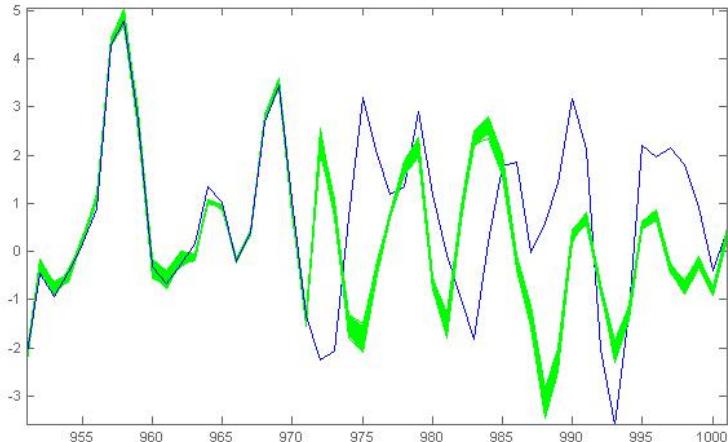


Figure 39: ARMA(2,5): Forecast Predictive Distribution for series Y.

## 5.7 ARMA Model Identification by Reversible Jump MCMC

As stated early, the autocorrelation and partial autocorrelation functions are subjective in determining the order of the model. The Reversible Jump MCMC tries to identify the model from the set of suggested models. The determination of the order by RJMCMC can be done by looking at how well the model fits when compared to the other models, or it can be done by looking at the error of standard deviations. More details about RJMCMC

can be found in its subsection. To check how RJMCMC works, the  $ARMA(2,1)$  model is identified and then the data are generated. Two suggested models ( $ARMA(1,1)$ , and  $ARMA(2,1)$ ) are used, where the second model is a true model.

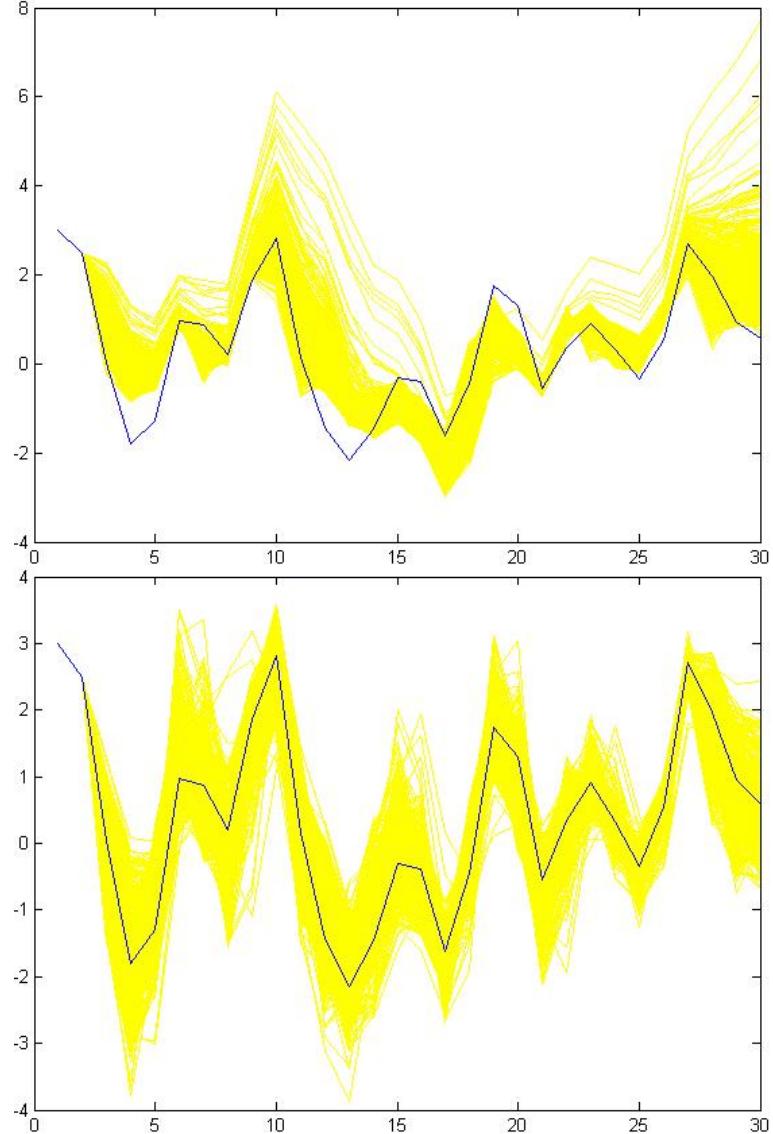


Figure 40: RJMCMC: fitting of the suggested models.

The first figure (from Figure (40)) shows the fitting of the first model,  $ARMA(1,1)$ , while the second one shows the fitting of the second model,  $ARMA(2,1)$ . Since the fitting of the second model is better than the fitting of the first one then one can easily conclude that the data are from second model.

We can also use frequencies to select the best model. The RJMCMC gives the frequencies or probability of each model. From these frequencies of models, we choose a model whose frequency is higher than the other models. The chosen model is the best model.

## 6 Real Data: Consumption of Electricity in Finland

### 6.1 Decomposition of Data

In this section where the data used are the real data from electricity market. This original data are daily consumption. The data had 5 column representing 5 places; West Denmark, East Denmark, Finland, Sweden, and Norway. In this section, only one column is taken, column Finland. The original data for Finland are shown in Figure (41). The data were tested for stationarity by Augmented Dickey-Fuller unit root test based on zero drift AR model, and then the data are treated by decomposition method. The decomposition of data refers to the statistical method that deconstruct data into notional components ([8]).

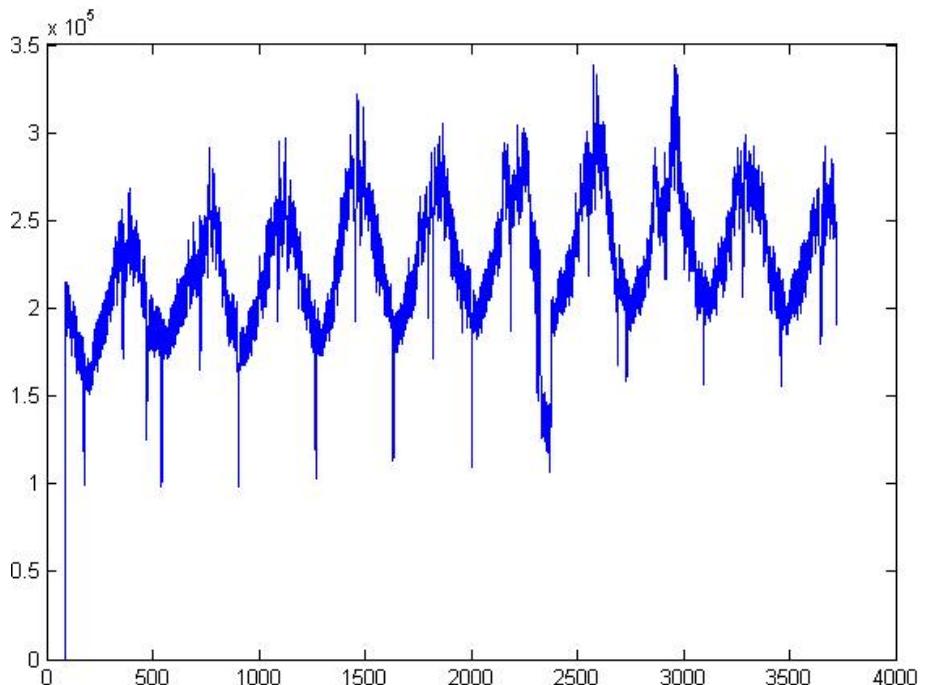


Figure 41: Finland Electricity consumption.

The data were grouped into weekly, that is, computation of weekly sums of electricity consumption by Finland is performed and then leaving data with zeros since the first 4 months had 0 values. Grouping the data is because the data have two seasonality, weekly seasonality and annually seasonality. We need only one seasonality, we take away the weekly seasonality. Precisely  $9*51$  weeks were taken into account and the final data were processed with decomposition, running the *MCMC* and then reconstruction was performed. Figure (42) shows the plot for weekly consumption. The straight line, in Figure (42), is the trend obtained by fitting the weekly data by least squares method. We

Compute the seasonal multipliers for each day of the year showing the relation between given observation and value of trend line in given time point. We calculate it for every week of each year and then average over year as result we get 51 seasonal multiplier - one for each week of a year. The process is repeated 9 times to have matrix size equal to trend line horizon. We decompose the series, that is, subtract/remove trends and seasonal components. Plot the 'random' rest of the series (call then residuals or remainings), as shown in Figure (43).

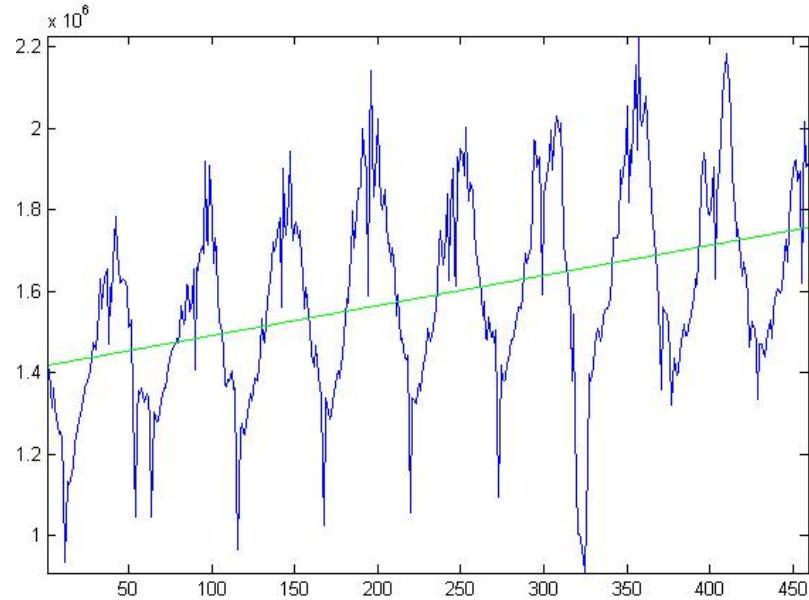


Figure 42: Weekly Electricity consumption and Trend.

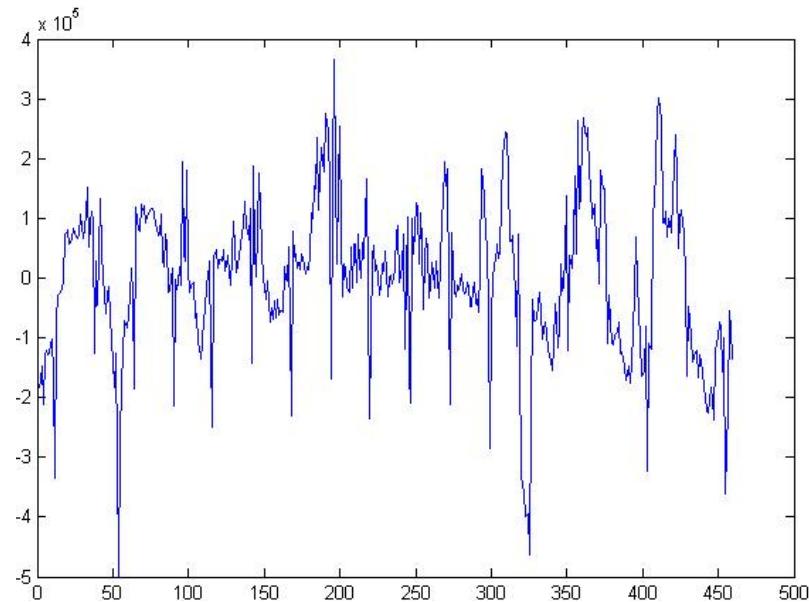


Figure 43: Residuals after extracting Trends and Seasonality.

The Identification of order, after checking the stationarity of decomposed data, is done by studying the autocorrelation and partial autocorrelation function. From Figure (44), it shows that the possible model is  $AR(1)$ . Therefore, we decide to use model  $AR(1)$  to model the residuals.

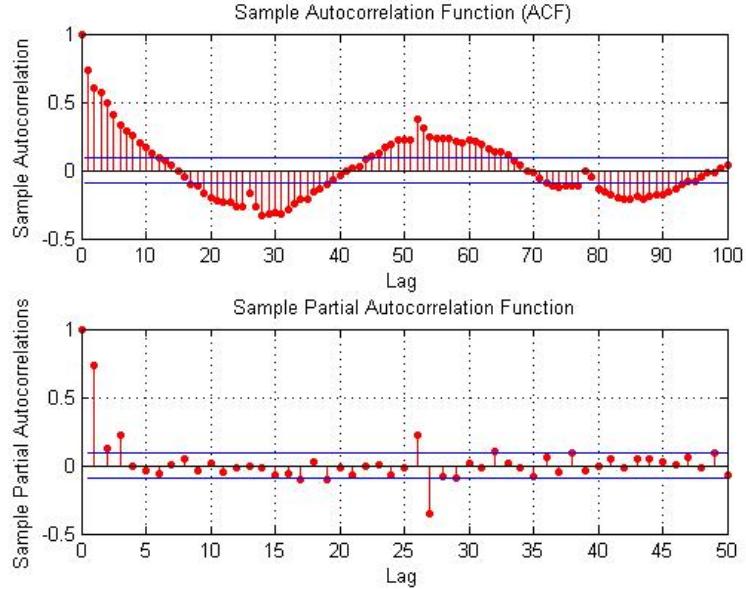


Figure 44: Autocorrelation and Partial Autocorrelation Functions of Residuals.

## 6.2 MCMC and Reconstruction

Since the model is decided ( $AR(1)$ ), then we employ the *MCMC* to study the distributions. The initial parameter for *MCMC* was defined by choosing any initial. The parameter Histogram is seen in Figure (45). The histogram looks fine as the mean obtained is close to the expected value. One may compare the mean from histogram by the estimated value obtained using the GARCH Toolbox. The values are approximately the same.

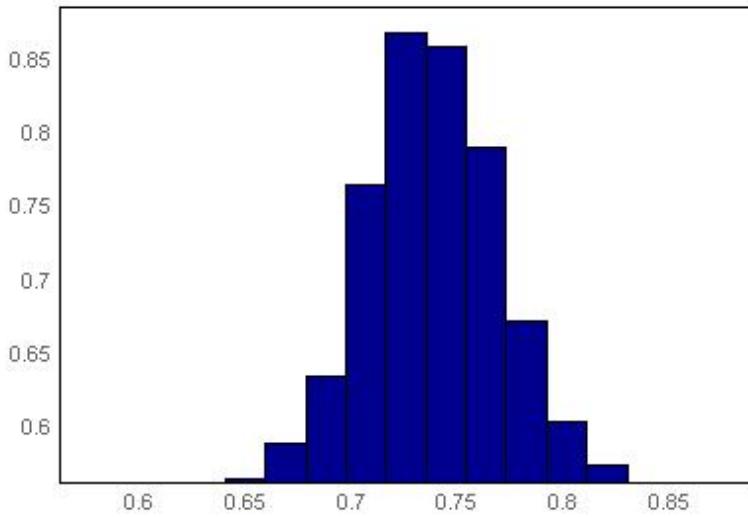


Figure 45: Histogram of the parameter by MCMC.

The *MCMC* results are plotted, as seen in Figure (45), Figure (46) shows the fitting of the decomposed data (blue) against the results obtained by *MCMC* (yellow). As seen the *MCMC* results capture well the residuals, indicating that the method works well.

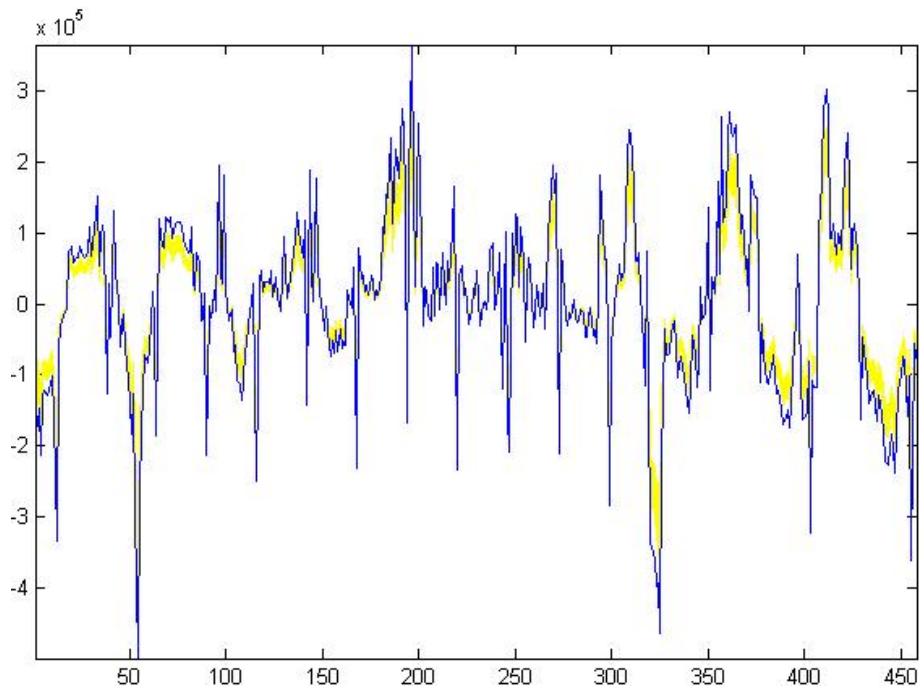


Figure 46: Residuals and Estimations by MCMC.

Reconstruct the consumption series by adding the seasonal multipliers on top of the residual realizations obtained from *MCMC*. The results is seen in Figure (47). As trends and seasonality are put back, the reconstructed *MCMC* paths of residuals follow well the real data.

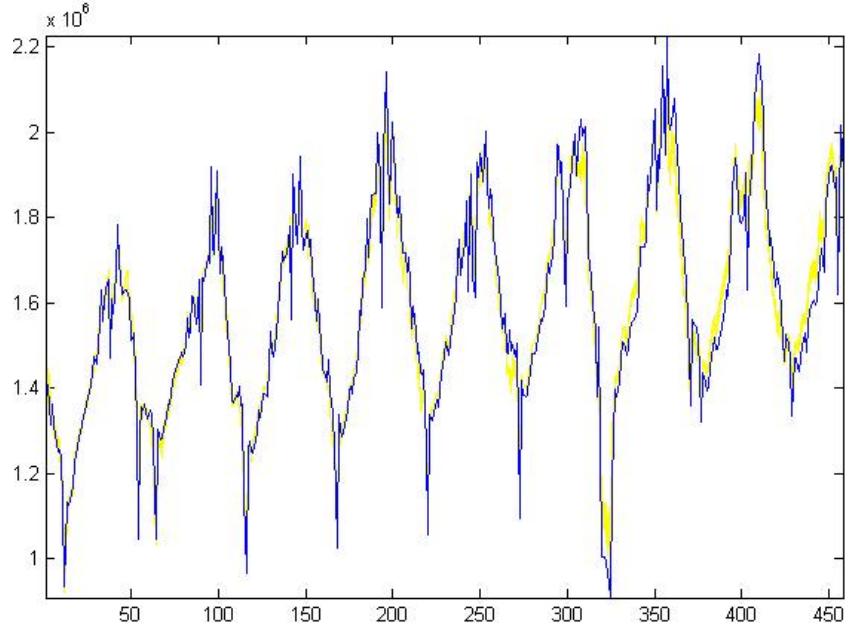


Figure 47: Data (blue) and MCMC spread (yellow) due to parameters variability.

## 7 Conclusion

The aim of this thesis was to integrate the *MCMC* algorithms and the classical time series. The first stage was to estimate the parameters of known *ARMA* models using time series methods, and also use the *MCMC* algorithms to estimate the parameters of the same models. For the case of models without *MA* components, the estimates were done by Yuler-Walker equations. For all methods used in estimating the parameters, the results were significant as all methods produced better estimates as these estimates are approximately to the original parameters. From these results, then one is recommended to use the *MCMC* algorithm to estimate the parameters for time series models.

The thesis focused to the model fitting. The fitting was done by using the classical time series way and the *MCMC* way. The fitting was ,by (15), deterministic while the noises were varied.

Forecasting the model together with forecast errors are the part that has been addressed in this thesis. Depending to the nature of the model equation the accumulation of noises

lead to poor forecasts. If the model contain *MA* components, the noises are highly accumulated as each data generated has error and the error is transferred or moved to another consecutive data. As evidenced, the plots of these forecasts has visible noises. Unfortunately, the *MCMC* has shown the same impact. The forecast predictive distribution failed to capture the real model while the scatter plots gave different results depending to the presence of *MA* components.

Model identification was, also, the aim of the thesis. In most cases, the order of the time series models are identified by graphical methods. As seen the autocorrelation and partial autocorrelation functions have been used to identify the order of *ARMA* models. Since, the graphical identification is subjective ([1]) the use of reversible jump *MCMC* has been employed. However, the RJMCMC needed one to suggest the models. The suggestion of models depended on the autocorrelation and partial autocorrelation functions. Hence, the autocorrelation and partial autocorrelation functions are still important in identification of order of models. The RJMCMC gives the percentage or frequencies of the expected model from which one chooses it. It gives the model depending to the frequency no matter the correct model is present in the suggested models or is not there.

To validate the use of the methods studied in this thesis, the preceding section have real data from electricity consumption. The data were treated by decomposing and then reconstructed. As seen the decomposition and reconstruction have been done so as to study the impact of trends and seasonality. The plots obtained help to interpret and predict the consumptions. The validation of the methods has been addressed also to the part of subjectivity when a person was given two series, from different models, and asked to suggest the model. It was hard to identify the one of the series that had *MA* components as the person suggested many models compared to the number of suggestions of the other series. This indicates that the model becomes complex once there are *MA* components.

The findings still demand the use of the autocorrelation and partial autocorrelation functions in model identification together with the RJMCMC. There is possibility of one not to suggest the correct model, at this point the RJMCMC gives wrong answer. How to find the order of time series model without employing the graphical methods such as autocorrelation and partial autocorrelation functions would also be another thesis topic for further studies.

## References

- [1] Box, G. E. P, Jenkins, G. M. and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*, revised edition. Prentice Hall, Engewood Cliffs, NJ.
- [2] Brooks, C. (2002). *Introductory econometrics for finance*. Cambridge University Press, United Kingdom.
- [3] Enders, W. (2004). *Applied Econometric Time Series*, second edition. Wiley, United States
- [4] Granger, C, W, J. (2001). *Forecasting Economic Time Series*, 2nd edition. Academic Press, INC, United States.
- [5] Hanke, J. E, Reitsch, A. G, and Wichern, D, W. (1994). *Business Forecasting*, seventh edition. Prentice Hall, Upper Saddle River, NJ.
- [6] Robert, P, and Casella, G. (2004). *Monte Carlo Statistical Methods*, second edition. New York, USA: Springer-Verlag.
- [7] Vandaele, W (1983). *Applied time series and Box-Jenkins model*. Academic Press, INC, United States.
- [8] <http://www.wikipedia.org>
- [9] Dickey, D. A and Fuller, W. A. (1979). Distribution of the estimates for autoregressive time series with a unit root. *Journal of Empirical Finance* **1**: 83-106
- [10] Phillips, P. C. B. (1987). Time series regression with a unit root. *Econometrica* **55**: 863-898
- [11] Haario, H., Saksman, E., Tamminen, J. (2001). An adaptive Metropolis algorithm. Bernoulli (Vol. 7(2)). pp. 223-242. Available at: <http://citeseer.ist.psu.edu/haario98adaptive.html>
- [12] Laine, M. (2008). Adaptive MCMC methods with applications in environmental and geo-physical models. PhD thesis. Lappeenranta University of Technology.
- [13] Solonen, A. (2006). Monte Carlo Methods in Parameter Estimation of Nonlinear Models. Masters Thesis. Lappeenranta University of Technology.
- [14] Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82, 711–732.
- [15] Green, P. J. (2003). Trans-dimensional Markov chain Monte Carlo. In P. J. Green, N. L. Hjort, and S. Richardson (Eds.), *Highly Structured Stochastic Systems*, Oxford Statistical Science Series, No. 27, pp. 179–198. Oxford University Press, Oxford.
- [16] Jabłońska, M (2008). Analysis of outliers in electricity spot prices with example of New England and New Zealand markets. Masters Thesis. Lappeenranta University of Technology.
- [17] James, G (2004). *Advanced Modern Engineering Mathematics*, third edition. Malaysia: Springer-Verlag.
- [18] Hastie, D (2005). Towards Automatic Reversible Jump Markov Chain Monte Carlo. PhD thesis. University of Bristol
- [19] Laine, M. (2000). Applications of reversible jump MCMC. University of Helsinki.

- [20] Cryer, J. D, Chan, K (2008). *Time Series Analysis With Applications in R*, Second edition. Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA
- [21] Ehlers, R. S and Brooks, S. P (2004) Bayesian Analysis of Order Uncertainty in ARIMA models. Technical Report 2004/’05-B, Federal University of Paraná.
- [22] Troughton, P. T and Godsill, S. J (1997). A Reversible Jump Sampler for Autoregressive Time Series, Employing Full Conditionals to Achieve Efficient Model Space Moves Technical Report CUED/F-INFENG/TR.304, University of Cambridge
- [23] Cryer, J. D (1986). *Time Series Analysis*. Boston, Duxbury Press, cop, United Kingdom.