65504

FORECASTING FOR AUTOCORRELATED DATA USING STATISTICAL SIGNAL PROCESSING

by

Özcan KILINÇCI

65504

February, 1997

İZMİR

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

FORECASTING FOR AUTOCORRELATED DATA USING STATISTICAL SIGNAL PROCESSING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
in Partial Fulfilment of the Requirements for
the Degree of Master Science in Industrial Engineering

by
Özcan KILINÇCI

February, 1997
İZMİR

M.Sc. THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc.Prof.Dr. G.Miraç Bayhan
(Advisor)

.....
(Committee Member)

.....
(Committee Member)

Approved by the
Graduate School of Natural and Applied Sciences

Prof.Dr. Macit Toksoy
Director

ACKNOWLEDGEMENTS

I would like to thank my advisor Associate Professor G. Miraç Bayhan for her support, encouragement, and valuable guidance. I am grateful to her for letting me share her professional knowledge and experience.

I would like to acknowledge the contributions of all my lecturers during my M. Sc. studies.

I wish thank to Research Assistants Ayfer Avşar, Gökalp Yıldız, İlkay Saraçoğlu and Şeyda Topaloğlu for their continuos encouragement, suggestions and help.

Finally I am grateful to my family for all their support.

Özcan KILINÇCI

ABSTRACT

Autoregressive moving average (ARMA) models have found wide applications in various signal processing applications like system identification, speech processing, spectrum estimation, etc. In this study, a general class of ARMA models, AR model, is identified for the data from the neutralization process of crude oil to forecast the amount of sodium oleate in an oil factory. Therefore first order of AR model is determined by using six order selection criteria, then parameters of AR model are estimated by using seven signal processing algorithms. The best model is determined according to the minimum mean square errors (MSE) computed by using estimated model parameters for each algorithm and for each selected order. The best model parameters are used to forecast the amount of sodium oleate in the nutr oil. Then MSEs of all algorithms are compared MSEs found by Box&Jenkins method. Each of the fourty alternatives is better than the one produced by Box&Jenkins method. As a result, each of the six algorithms is superior to Box&Jenkins method in this study. Also, computer programs in MATLAB are written for six order selection criteria and seven signal processing algorithms.

ÖZET

ARMA modeller, sistemin tanımlanması, konuşma (ses) işleme gibi çeşitli sinyal işleme uygulamalarında yaygın olarak kullanılmaktadır. Bu çalışmada, bir yağ fabrikasındaki ham yağın nötralizasyon işleminden elde edilen sabun miktarını tahmin etmek üzere, bu işlemden alınan data için ARMA modellerden AR modelin uygun olduğuna karar verildi. AR modelin mertebesi altı tane kriter yardımıyla belirlendikten sonra, yaygın olarak kullanılan yedi tane sinyal işleme algoritmasının herbiriyle modelin parametreleri tahmin edildi. Enküçük ortalama hata kareye (EOHK) sahip olan model eniyi model olarak belirlendi. Eniyi modelin parametreleri kullanılarak nötr yağdaki sabun miktarı tahmin edildi. Sinyal işleme algoritmaları ile elde edilen EOHK'ler, geleneksel tahmin tekniklerinden Box&Jenkins metodunun kullanılmasıyla bulunan EOHK'lerle kıyaslandığında önceki alternatiflerin daha iyi sonuçlar verdiği gözlendi. Bu çalışmada kullanılan herbir mertebe belirleme kriteri ve parametre tahmin yöntemi için gerekli bilgisayar programları MATLAB'da yazıldı.

CONTENTS

Page

Contents.....VII
List of Tables.....X
List of Figures.....XI

Chapter One
INTRODUCTION

1. INTRODUCTION.....1

Chapter Two
FORECASTING

2. FORECASTING.....2

2.1The Need for and Role of Forecasting.....2
2.2 Forecasting Process.....5
2.3 Autoregressive Process.....11
2.3.1 Autocorrelation and Partial Autocorrelation Functions.....12
2.3.2 Box&Jenkins Methodology.....14

Page

Chapter Three

AUTOREGRESSIVE MODELLING WITH SIGNAL PROCESSING

3. AUTOREGRESSIVE MODELLING WITH SIGNAL PROCESSING.....16

3.1 Statistical Signal Processing.....16

3.2 Autoregressive Modelling.....17

3.2.1 Least Squares Optimal Filtering.....18

3.2.2 AR Modelling Linear Prediction.....21

3.2.3 Order Selection Criteria.....24

3.2.4 AR Modeling Algorithms.....29

3.2.4.1 Autocorrelation Method.....29

3.2.4.2 Covariance Method and Prewindowed Covariance Method.....30

3.2.4.3 Modified Covariance Method.....31

3.2.4.4 Burg's Method.....33

3.2.4.5 Modified Iteratively Reweighted Least Squares Algorithm.....37

3.2.4.6 Predictive Least Squares Order Selection Criterion.....41

Chapter Four

APPLICATION

4. APPLICATION.....42

4.1 Forecasting the Amount of Sodium Oleate in the Neutralization Process of Crude Oil.....42

4.2 Autocorrelation Analysis of Data to Identify the Model.....42

4.3 Selection of AR Model Order.....45

4.4 Estimation of AR Model Parameters.....49

4.5 Comparison of MSEs of All Algorithms.....53

Page

Chapter Five
CONCLUSION

5. CONCLUSION.....57

REFERENCES

REFERENCES.....61

APPENDICES

APPENDICES.....62
Appendix 1. Singular Value Decomposition.....62
Appendix 2. Levinson Recursion.....64
Appendix 3. Lattice Representation.....69
Appendix 4. MATLAB Computer Programs.....71
4.1 AIC, CAT, FPE, MDL.....71
4.2 Effective Rank.....72
4.3 PLS.....73
4.4 Autocorrelation Method.....76
4.5 Recursive Autocorrelation Method.....78
4.6 Covariance Method.....81
4.7 Recursive Covariance Method.....83
4.8 Prewindowed Covariance Method.....85
4.9 Recursive Prewindowed Covariance Method.....87
4.10 Modified Covariance Method.....89
4.11 Recursive Modified Covariance Method.....91
4.12 Burg' Algorithm.....93
4.13 MIRLS Algorithm.....95

LIST OF TABLES

Page

Table 2.1 Forecasting Needs for Management....2
Table 2.2 Characteristics of Forecasting Models....7
Table 2.3 Behaviour of Theoretical Autocorrelation and Partial Autocorrelation Functions for AR Models....13
Table 3.1 Criteria for AR Model Order Selection....25
Table 4.1 Observations....43
Table 4.2 Values of Effective Rank....49
Table 4.3 MSEs for Autocorrelation Method....50
Table 4.4 MSEs for Covariance Method....50
Table 4.5 MSEs for Prewindowed Covariance Method....51
Table 4.6 MSEs for Modified Covariance Method....51
Table 4.7 MSEs of Performance Approaches....52
Table 4.8 MSEs for Burg's Algorithm....52
Table 4.9 MSEs for MIRLS Algorithm....53
Table 4.10 MSEs for PLS Algorithm....53
Table 4.11 MSEs for Box&Jenkins Method....54
Table 4.12 MSEs of All Algorithms and Box&Jenkins Method....54

LIST OF FIGURES

Page

Figure 2.1 Forecasting Process.....6
Figure 2.2 Box&Jenkins Methodology.....5
Figure 3.1 Prediction Error Filter.....17
Figure 3.2 AR Model.....17
Figure 4.1 Autocorrelation Functions of Data (First 101 Values).....44
Figure 4.2 Partial Autocorrelation Functions of Data (First 101 Values).....45
Figure 4.3 Variance of Errors.....46
Figure 4.4 AIC Values.....46
Figure 4.5 CAT Values.....47
Figure 4.6 FPE Values.....47
Figure 4.7 MDL Values.....48
Figure 4.8 PLS Values.....49
Figure 4.9 The Best Model (Covariance Model p=12).....55
Figure 4.10 The Worst Model (MIRLS Algorithm p=13).....55
Figure 4.11 Future Predictions of the Best Model (First Approach).....56
Figure 4.12 Future Predictions of the Best Model(Recursive Approach).....56

CHAPTER ONE
INTRODUCTION

Forecasting is the cornerstone of planning. It is widely used in the management of production and inventory systems and has also found application areas in process control, financial planning, marketing, investment analysis and distribution planning.

Statistical signal processing is used in electrical engineering. Deterministic signals are provided from a signal generator, but voice, images, music, noise, data and target motion signals are not deterministic. These signals are represented as autoregressive moving average (ARMA) models using statistical signal processing algorithms, and these models could be used in forecasting.

In this thesis, statistical signals processing algorithms are employed for forecasting. Thesis consists of three sections. The first section introduces general forecasting and autoregressive processes (AR). In the second section, some fundamental concepts of AR model building, order of model selection criteria and signal processing algorithms to estimate the model parameters are presented. In the third section, an AR model is built in order to forecast the amount of sodium oleate in the neutralization process of a crude oil produced by an oil factory. Also computer programs and some mathematical knowledge are given in appendices.

CHAPTER TWO
FORECASTING

2.1 The Need for and Role of Forecasting

Forecasting is the prediction, projection, or estimation of the occurrences of uncertain future events or levels of activity. Since the future rarely is certain, some system of forecasting, implicit or explicit, is necessary. Forecasting offers an organization some foresight in the premeditation of appropriate courses of action. Its purpose is to make use of the best available present information to guide future activities toward organizational goals. This function is particularly important in the allocation and use of resources.

Forecasts are usually classified according to the future time horizon. Time horizon generally can be divided into immediate term (less than one month), short term (one to three months) and long term (two years or more) (Makridakis&Wheelwright, 1989, p.27). The needs for forecasting in the management fields are shown in Table 2.1 (Makridakis&Wheelwright, 1989, pp. 20-22) according to the time horizon.

Table 2.1 Forecasting Needs for Management

Organizational Unit	**Time Horizon**			
	Immediate Term (Less than 1 Month)	**Short Term (1 to 3 Months)**	**Medium Term (3 Months to 2 Years)**	**Long Term (2 Years or More)**
Marketing	Sales of each product type, sales by geographic area, sales by customer, competition, prices, inventory levels	Total sales, product categories, major products, product groups, prices	Total sales, product categories, prices, general economic conditions	Total sales, major product categories, new product introduction, saturation points of existing products, customer preferences and tastes

Table 2.1 (continued)

Organizational Unit	**Time Horizon**			
	Immediate Term (Less than 1 Month)	**Short Term (1 to 3 Months)**	**Medium Term (3 Months to 2 Years)**	**Long Term (2 Years or More)**
Production	Demand of each product, plant loading	Total demands, demand of product categories and product groups, scheduling, employment level, costs	Costs, budget allocations, buying or ordering equipment and machinery, employment level	Costs, facility investments, expansion of plant and equipment, ordering of heavy machinery and equipment, demand of production facilities, new technologies
Inventory	Demand for each product's production, demand for material, demand for semifinished products, weather conditions	Demand for material, demand for semifinished products, demand for products, possible strikes	Possible strikes in suppliers or transportation facilitates	Total sales, expansion of warehouses
Finance and accounting	Sales revenue, production costs, inventory costs, leading indicators, cash inflows, cash outflows	Total demand, inventory levels, cash flows, short-term borrowing, prices	Budget allocations, cash flows	Total sales, investment selections, capital expenditure, allocations of resources, capital programs, cash flows
Purchasing	Production, cash availability, purchasing of supplies and materials	Demand for products, demand for materials, lead time for purchasing	Demand for products, demand for raw and other materials	Contracts for buying raw materials, customer preferences and tastes
R&D			New product introduction, R&D selections	Total sales, technological, social, political, and economic conditions of future, new product development
Top management		Total sales, sales breakdown, pricing	Demand for sales, costs and other expenses, cash position, general economic conditions, control objectives	Total sales, costs and other expenses, social and economic trends, goals, objectives, and strategies, new products, pricing policies
Economic unit		Level of economic activity	General economic conditions, turning point in economy, level of economic activity	State and type of economy, level of economic activity, sales of industry

Table 2.1 (continued)

Organizational Unit	Time Horizon			
	Immediate Term (Less than 1 Month)	**Short Term (1 to 3 Months)**	**Medium Term (3 Months to 2 Years)**	**Long Term (2 Years or More)**
Environmental unit			Available technologies	Areas of technological innovations, R&D selections, available technological opportunities
			Social attitudes	Social trends, tastes, areas of social concern
		Availability of money, interest rates	Fiscal and monetary policies	Trends in the rate of taxation, depreciation, and concept of free market
	Prices, sale promotions	Prices, advertising selection, sales promotions, new product introduction	New product development	Capital investment, new technologies, R&D selections of competitors
	Weather conditions	Weather conditions	Crops	General environmental constraints (pollution level, availability of raw materials, etc.)

In marketing, a number of decisions can be improved significantly by basing them on reliable forecasts of market size and market characteristics. For example, a company that produces and sells major appliances (washing machines, television sets, and refrigerators) must be able to forecast the demand for each of its products by geographic region and type of consumer. The marketing department can use these forecasts to plan its advertising, direct sales, and other promotional efforts. Marketing also requires forecasts for market share, prices, and trends in new product development.

In production, a major need for forecasting is the area of product demand. This involves predicting both volumes and produces mixes so that the firm can plan its production schedule and inventories to meet market requirements at best. Other areas of the production that requires forecasting include material procurement, labor scheduling, equipment purchasing, maintenance and plant capacity planning.

Finance and accounting departments must forecast cash flows and the rates which various expenses and revenues will occur if they are to maintain company liquidity and operation efficiency. In making such projections, it is often useful to predict each of the elements that make up the firm's net cash flow so these can be used by various departments within the organization. The finance and accounting functions must also forecast interest rates to support the acquisition of new capital, the collection of accounts receivable to help in planning working capital needs, and capital equipment expenditure rates to help balance the flow of funds in the organization.

The personnel department also requires a number of forecasts in planning for human resources. Workers must be hired and trained, and benefits must be provided to be competitive with those available in labor market. Furthermore, trends that affect variables such as labor turnover, retirement age, absenteeism, and tardiness need to be forecasted for planning and decision making.

Since the general management function is central to the successful operation of the firm, forecasts that can be used as the basis for decision making at this level are perhaps more critical.

2.2 Forecasting Process

Predicting the future is the essence of the forecasting. Estimating when future events will occur, or the values of the operating characteristic or performance measures over some planning horizon are representative outputs of the forecasting process. Few individuals may develop accurate forecast for future in a consistent manner without utilizing a systematic procedure. A comprehensive forecasting process is shown in Figure 2.1 (Reid, 1992, p.23). Engineering managers who engage in this sequential procedure achieve benefits relative to their areas of responsibility. Steps of the forecasting process are as follows.

***Step 1*:** It is critical to begin with a specific statement of the purpose for the forecast. Organizations use forecast a variety of purposes. For example, forecasted annual revenues

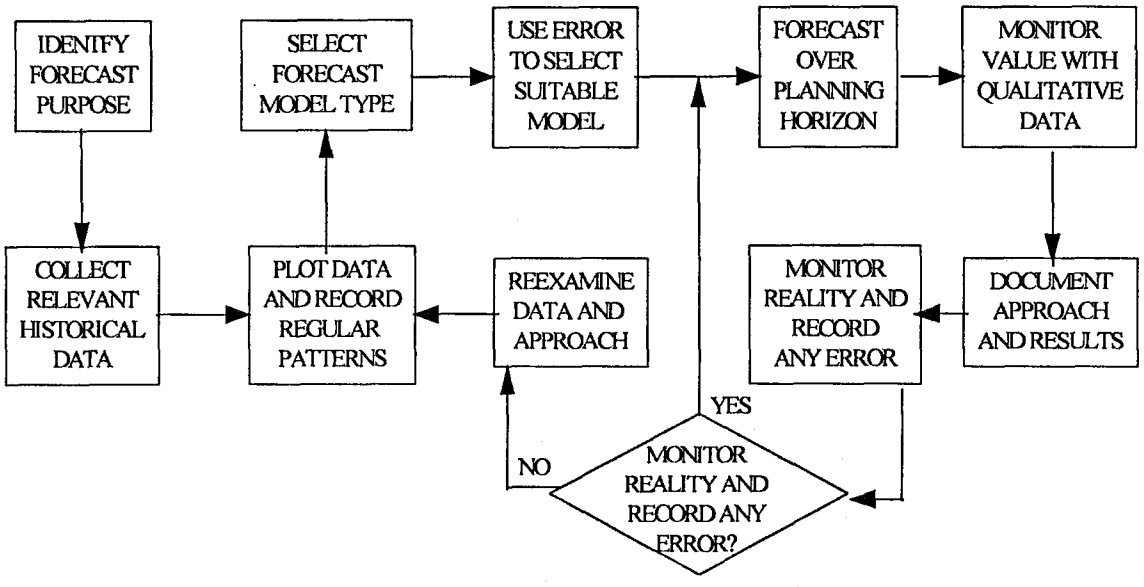

Figure 2.1 Forecasting Process

could be used by the marketing manager to establish quotas for sales representatives or by the engineering manager to make capacity decisions related to anticipated staffing needs or production levels. Most functional managers will want to develop their own set of forecasts that relate to their needs and managerial style. Thus engineering managers should be prepared to modify existing forecast that has been developed else where in the organization in order to reflect their needs and aid in making decisions to achieve their goals.

***Step 2*:** Collecting the historical data provides the opportunity to examine similar situations from an experiential perspective. Since the intent is to identify trends, seasons, or cycles associated with the past that could be projected in the future, historical performance data should be purged of any onetime occurrences that probably will not reoccur in the future. Data are commonly obtained from the organizational records, industrial trade association and government documents. If the organization is new or/and doesn't have any related historical data, the historical data for the forecasting should be obtained from similar services or organizations.

***Step 3*:** Displaying relevant historical data to determine if any patterns exist that would help in the predicting future values is the goal of the this step. Since graphical displays often reveal underlying regularities in past demands or performance levels, plotting historical data over time may produce a better understanding of previous behavior, which

should improve forecast accuracy. Most engineering managers realize that visual displays of past performance measures facilitate the identification of underlying time-dependent patterns of resource consumption.

Step 4: Careful consideration is given to using one or more following three different types of forecasting models:

- Time Series Models:
- Causal Models: or,
- Subjective Models.

Table 2.2 (Reid, 1992, p.24) summarizes the unique characteristics of each forecasting model, typical applications, costs, required data, technology considerations and relevant planning horizons. The engineering manager should employ those methods or techniques that are most suitable for addressing their current needs.

Table 2.2 Characteristics of Forecasting Models

Characteristics	**Time Series Models**	**Causal Models**	**Subjective Models**
Popular Methods	Moving Average Exponential Smoothing	Regression Econometrics	Delphi Historical Analogy
Typical Applications	Inventory Control	Workload Demands	Technological Capabilities
Relative Cost	Very Low	Moderate	High
Data Required	Twelve or more observations on the dependent variable	Twenty-four or more observations on both independent and dependent variables	General understanding of current state of affairs
Technology Needed	Calculator	Computer with statistical software	None
Planning Horizon	Short Term	Medium Term	Long Term

Time series models use an item's own historical data to predict the future demand values or performance levels. These methods examine past behavior of the dependent

variable over time and extrapolate those values into the future. Resultant forecasts are based on projecting any observed historical patterns. Time series models assume that time is surrogate measure for one or more independent variables that are to the dependent variable but may difficult to measure.

Causal models also use historical data on independent variables that typically represent the usage of related goods and services to develop a forecast for the value of interest, that is, the dependent variable. This type of forecasting model uses a variety of statistical methods to establish a functional relationship that depicts prevalent patterns in the data. The causal model assumes that as the system of relevant independent variables changes, a corresponding change will occur in the dependent variable. Thus, by knowing the values of the independent variables, an appropriate prediction of the dependent variable can be determined.

Specific curve-fitting methodologies within both types of models range from simple approaches such as an arithmetic average to very complex statistical models like multi-linear regression. Both time series and causal models use analytical procedures to determine underlying patterns, and thus, their validity rests on the assumption that the future will continue to behave like the past. In other words, the major effects governing past performance are assumed to remain unchanged over the current planning horizon. Typically, this assumption reflects a situation that is characterized statistically as a stationary process.

In contrast, subjective models are more qualitative and judgmental in nature. They bring together experts to develop agreements on forthcoming events or dependent variables through brainstorming, drawing analogies and writing scenarios that may relate to unusual, non-extractable, distant occurrences. These methodologies focus on predicting major happenings, turnings points or important changes and are considered to be more speculative than two previously mentioned models.

The engineering manager is responsible for identifying the type of the forecasting model that is most appropriate relative to the stated forecasting purpose. Since a variety of

specific forecasting techniques are available within each forecasting model type, the manager needs to select one or more of these techniques that will yield an acceptable level of forecasting error.

Step 5: Testing the accuracy of various forecasting techniques by having each predict recent actual performance or resources usage is the major focus of this step. Typically, the manager selects the technique that produces the smallest average forecasting error and uses it to forecast over the planning horizon. There are several common approaches to calculate the error associated with a forecast that is regularly used with time series or causal models. The popular measures of forecasting error are (Makridakis&Wheelwright, 1989, pp. 58-59):

- *Mean error*

$$ME = \frac{\sum_{i=1}^{n} \varepsilon_i}{n}$$

- *Standard deviation of errors*

$$SDE = \sqrt{\frac{\sum_{i=1}^{n} \varepsilon_i^2}{n-1}}$$

- *Mean absolute percentage error*

$$MAPE = \frac{\sum_{i=1}^{n} |PE_i|}{n}$$

- *Mean absolute deviation*

$$MAD = \frac{\sum_{i=1}^{n} |\varepsilon_i|}{n}$$

- *Percentage error*

$$PE_i = \frac{x_i - \hat{x}_i}{x_i} 100$$

Note:

$\varepsilon_j = x_j - \hat{x}_j$

$n =$ number of observations

- *Mean squared error*

$$MSE = \frac{\sum_{i=1}^{n} \varepsilon_i^2}{n}$$

- *Mean percentage error*

$$MPE = \frac{\sum_{i=1}^{n} PE_i}{n}$$

The appropriate measure of forecasting error depends upon the situation and the purpose underlying the forecast. It is important for engineering managers to realize that multiple measures of forecasting error exist. Often, it is meaningful to evaluate a forecasting technique by using several different error measures. A great deal of computer-assisted computation and analysis of different alternatives is involved in identifying a suitable forecasting technique needed to conclude this step.

***Step 6*:** In this step, the selected technique is used to forecast the value of the dependent variable or the occurrence of an event during the desired planning time frame. It is often prudent to develop a range over which the forecast value may vary. If historical data are being used to predict to future, statistical techniques can be establish an analytical confidence level that the actual future value will reside within a prescribed interval. The use of a forecast range provides an overt indication of the uncertainty associated with forecasted future.

***Step 7*:** The purpose of this step is to assure that the impact of relevant interval and external factors on the results generated by specific forecasting techniques are incorporated. Quantitative adjustments to a specified forecast value are required for significant internal changes such as the introduction of new engineering services, production process redesigns, or facility/workstaff expansion plans. Additionally, expected changes in the business cycle, new government regulations, anticipated technological upgrades and other environmental activities must be evaluated to assess their impact on the forecast.

***Step 8*:** The importance of documenting all assumptions involved in developing the forecast is one of the important dimensions of this step. In addition, the procedures, which have been utilized to adjust the historical data, to test viable forecasting techniques and to analyze resulting forecasting errors need to be recorded. The written report will serve as a record of what was done and why certain choices were made. However, the real benefit from this activity occurs when the forecasting process is successful and documentation is readily available to replicate the forecasting procedures for the next planning period. Even if the process produced a poor forecast, management can learn from this documented experience and hopefully improve both the process and result in the future.

***Step 9*:** This step involves evaluating the actual performance relative to forecasted performance. Monitoring a relevant measure of forecasting error allows the effectiveness of the forecast to be continually evaluated. The manager must periodically decide whether the current forecasting process is producing a forecast that is acceptable for planning horizon. If the rate is tolerable, then the forecast should be updated for the next planning

period. If not, a new forecasting model or the same technique with different parameter values may be needed.

Step 10: If the forecasting results are not being used for their stated purpose, the corrective action needs to be initiated. Often this occurs when the forecasting error exceeds an unspecified tolerance limit. The engineering manager needs to be prepared to assess forecasting errors relative to acceptable tolerances and evaluate the costs and benefits of utilizing alternative forecasting models or techniques.

Personal involvement with the forecasting process does not guarantee improved forecast accuracy: however, it will provide additional insights into departmental activities and intraorganizational needs which will increase decision making effectiveness. The fact that a particular forecast "may be wrong" should not discourage the engineers from addressing these future uncertainties by becoming directly involved in selected aspects of the forecasting process. At a minimum, serious consideration should be given to (1) specifying the forecasting purpose, (2) reflecting on past data patterns, (3)modifying quantitatively determined forecast values with relevant qualitative information, (4) assuring that the forecasting approaches and results are documented for future use, and (5) determining if the resultant forecasting error is acceptable (Reid, 1992, pp.24).

2.3 Autoregressive Process

Consider a time series in which successive observations can be represented by a linear combination of independent random variables (ε_i) that are drawn from a stable probability distribution with mean 0 and variance σ_ε^2. The linear combination of the (ε_i) could be written as

$$x_t = \mu + \psi_0\varepsilon_t + \psi_1\varepsilon_{t-1} + \psi_2\varepsilon_{t-2} + \cdots$$

or

$$x_t = \mu + \sum_{j=0}^{\infty}\psi_j\varepsilon_{t-j} \qquad (2.1)$$

The linear combination in equation (2.1) would not be very useful time series model, since it contains an infinite number of unknown parameters (the ψ weights). Consider the special case of (2.1), in which only the first p of the weights are nonzero. This model may be written,

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

where $\phi_1, \phi_2, \ldots, \phi_p$ the finite set of weight parameters. Last equation is called an autoregressive process because the current observation x_t is regressed on previous realizations $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$ of the same time series. The process contains p unknown parameters $\phi_1, \phi_2, \ldots, \phi_p$, and as a result last equation (or model) is referred to as an autoregressive process of order p, abbreviated AR(p). In the following section, some fundamental concepts in autoregressive processes, that are autocorrelation and partial autocorrelation functions will be presented.

2.3.1 Autocorrelation and Partial Autocorrelation Functions

One of the basic assumptions in the time series model is that the value of the disturbance term in one period is independent of its value in any other period. In other words, error term in period t, is independent from error term in period s; $\text{cov}(\varepsilon_t, \varepsilon_s) = 0 \ , t \neq s$. But some of the time series don't have this assumption. There could be interdependence among successive values of time series. This case is called autocorrelation; $\text{cov}(\varepsilon_t, \varepsilon_s) \neq 0 \ , t \neq s$.

The degree of autocorrelation is measured by the correlation coefficient. The correlation coefficient varies among +1 and -1. If the correlation coefficient is +1, when the value of one variable increases the value of the other tends to increase also. This case is called positive autocorrelation. Conversely, if the coefficient is -1, this mean that there is a negative relation between variables. In the negative autocorrelation, when the value of one variable decreases, the value of the other tends to decrease.

The lag is delay between two observations. For example, if the autocorrelation between x_t and x_{t+k} is searched, then the lag is k. If N is the number of observations, k should be less than or equal to N/4 (Box-Jenkins, 1976, p.33, Montgomery&Lynwood,1976, p. 208).

The sample autocorrelation coefficient at lag k is

$$\hat{\rho}_k = \frac{\frac{1}{N-k}\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\frac{1}{N}\sum_{t=1}^{N}(x_t - \bar{x})}$$

The autocorrelation function is a graphical display of ρ_k versus the lag k. The value of the autocorrelation function is among +1 and -1.

Another useful concept in the description of the time series models is partial autocorrelation. It is convenient to think of the partial autocorrelation at lag k as the correlation between x_t and x_{t+k} with effects of the intervening the observations $(x_{t+1}, x_{t+2}, \ldots, x_{t+k-1})$ removed (Montgomery&Lynwood, 1976, p.192). If the k_{th} partial autocorrelation coefficient is ϕ_{kk}, a plot of ϕ_{kk} versus the lag k is called partial autocorrelation function. Note that $\phi_{00} = \rho_0 = 1$, and $\phi_{11} = \rho_1$. Other ϕ_{kk} are determined by using Yule-Walker equations.

The autocorrelation and partial autocorrelation functions may be helpful for determining autoregressive models that are autoregressive (AR), moving average (MA) and mixed (ARMA). They may be exhibited on a graph and a tentative model identified by comparing the observed patterns with the theoretical autocorrelation function patterns. These theoretical patterns are shown in Table 2.3 (Montgomery&Lynwood, 1976, p.208). The expression tails off in this table means that the function decays in an exponential, sinusoidal, or geometric fashion, approximately, with a relatively large number of nonzero values. Conversely, cuts off implies that the function truncates abruptly with only a few nonzero values (Montgomery&Lynwood, 1976, pp. 208-209).

Table 2.3 Behavior of Theoretical Autocorrelation and Partial Autocorrelation Functions for AR Models

Model	Autocorrelation function	Partial autocorrelation function
AR(p)	Tails off	Cuts off after lag p
MA(q)	Cuts off after lag q	Tails off
ARMA(p,q)	Tails off	Tails off

For example, if autocorrelation function decreases exponentially and partial autocorrelation function has only one large value at first lag, then the model is first order autoregressive model.

2.3.2 Box&Jenkins Methodology

Box and Jenkins present a general methodology for developing an appropriate ARIMA time series model, analyzing historical data. This methodology is shown in Figure 2.2 (Makridakis&Wheelwright,1989, p.133). Their approach consists of three-stage iterative procedure. In stage 1 tentative identification of an ARIMA time series model is done through analysis of actual historical data. In general, there must be at least 50 observations available to identify the appropriate model satisfactorily (Montgomery&Lynwood,1976, p. 236). An initial form for this model can be determined by examining the autocorrelation and partial correlation coefficients of data (See Table 2.3). In stage 2, the model is fitted to available historical data and running a check to determine whether it is adequate. If not, returning the stage 1 and alternative method of diagnostic checking is the compute the errors, and then estimate and examine their autocorrelation function. If the model is appropriate, the error sample autocorrelation function should have no structure to identify. When adequate model is accepted, in stage 3 forecast is developed for some future time period.

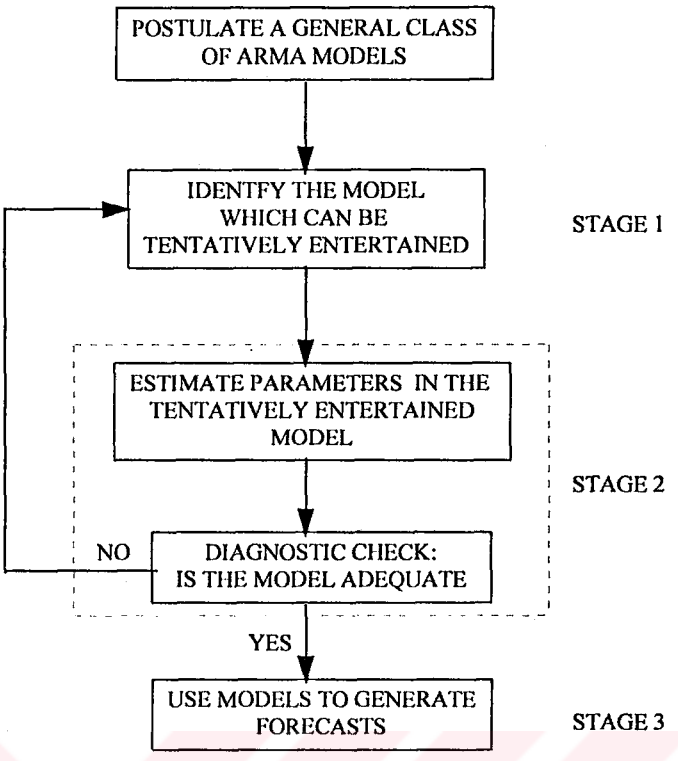

Figure 2.2 Box&Jenkins Methodology

Box&Jenkins models are especially well suited for time series in which the sampling interval is very small, so that relatively long history can be easily obtained(Montgomery&Lynwood,1976, p. 236). For this reason, they have been widely applied to series where hourly, daily, or weekly observations are of interest. For example, the time series of output characteristics of chemical processes, such as yield, purity, etc., are often of this general type. The Box&Jenkins models are most profitably employed in these situations where only a small number of time series is of interest, and management is willing to expend the resources necessary to achieve a high degree of forecast accuracy.

Estimation of model parameters is an important stage in ARIMA model building. Box&Jenkins method, that is briefly explained above, is often used to estimate model parameters. Also, statistical signal processing algorithms are used to build an AR model, some of these algorithms will be discussed in the following chapter.

CHAPTER THREE
AUTOREGRESSIVE MODELING WITH SIGNAL PROCESSING

3.1 Statistical Signal Processing

Although the study of signal processing usually begins with deterministic signals, few real-world signals are actually of that form. By deterministic it is meant that the signal or sequence can be expressed as some explicit mathematical formula. Signals that are deterministic in this sense are usually encountered as the output of a power supply, an oscillator, or some other type of signal generator. These signals do not well represent the common signals of speech, music, images, noise, data, target motion, and so on (that are called information-carrying signals), that are at the center of focus in many signal processing systems.

Modeling an information-carrying signal as a random process and analyzing it statistically is one to way achieve the goal of effective system design. However, there is a difference in point of view about whether a signal should truly be considered to be realization of a random process or simply to be observed data that is treated using statistical methods. In a sense it does not really matter if the effect of the processing is to produce the same results, and there are instances where both points of view are appropriate. The approach here begins with random process view point because it provides a fundamental framework in which to develop many needed statistical results.

3.2 Autoregressive Modeling

The basic goal in the signal processing is to obtain a model. The model has to fit to data. Model should be considered as a filter in Figure 3.1 (Therrien, 1992, p. 412), where $x[n]$ is

observation and input of the filter, $\varepsilon[n]$ is error term and output of the filter.

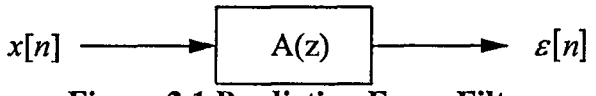

Figure 3.1 Prediction Error Filter

Transfer function of this filter is given like as (Therrien, 1992, p. 412),

$$A(z) = 1 + a_1^* z^{-1} + a_2^* z^{-2} + \cdots + a_P^* z^{-P}$$

The filter is inverted and driven with a white noise sequence of variance $\sigma_w^2 = \sigma_\varepsilon^2$ as shown Figure 3.2 (Therrien, 1992, p. 412),

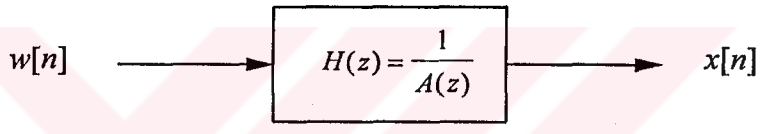

Figure 3.2 AR Model

This system will then produce a random sequence with the same statistical characteristics as those of the original sequence $x[n]$. A system of the form (Therrien, 1992, p. 412),

$$x[n] = -a_1 x[n-1] - a_2 x[n-2] - \cdots - a_P x[n-P] + w[n]$$

The model that is shown in the last equation is as a time series model, and unknown model parameters could be determined by using various methods, then the model could be used for forecasting. In the literature, there are a lot of methods to determine the model parameters. Some authors tried to identify a model by using various signals in their studies, for example, nonstationary signals (Ray, 1994), (Goto, Nakamura and Uosaki, 1995), periodic signals (Feder, 1993), (Chakroborty&Prasod, 1994), multisinusoidal signals, (Swingler, 1995), and cyclostationary signals (Giannakis&Zhou, 1994). Most of the new methods to determine the model parameters in the literature, are derived from developed methods or algorithms, for example, the a lot of estimation parameters methods have been derived from the maximum likelihood estimation method (Rasmussen, 1994),

(Armour&Margera, 1991), (Vis&Scharf, 1994), (Pham&Tong, 1994). Some new methods have been developed from TLS method (DeGroaf&Dawling, 1993), TLS-Prony method (Steedly, Ying and Moses, 1994), generalized l_p norm estimation algorithm (Chen, Chen and Shern, 1994) and novel recursive estimation algorithm (Rao, Huang and Dasgupto 1990). In some studies, models have been estimated by using output cumulants (Swami&Mendel, 1990) and cyclicstatistics (Giannakis&Zhou,1995). In the literature, filters have been frequently used in the parameters estimation studies. Optimal filter (Kim&Shafai, 1995), lattice filter (Li&Dickinson, 1988) and linear prediction error filter (Chi, Chang and Feng, 1994) are preferred. A new method for simultaneously selecting the order and identifying the parameters of an AR model has been developed in some studies (Katsikas, Likothanassis and Loiniotis, 1990). Some authors interested in industrial applications of signal processing (Bayhan, Bayhan and Badem, 1995).

In this thesis seven algorithms are employed to estimate AR model parameters from the historical data. These algorithms will be explained in following sections, but at first, some fundamental concepts, that are least squares optimal filtering and AR modeling via linear prediction, will be given. Also order selection criteria will be discussed, since model order must be known before the model parameters are estimated.

3.2.1 Least Squares Optimal Filtering

Optimal filtering problem is to estimate a desired sequence $d[n]$ from a related sequence $x[n]$ using an optimal linear filter. It is assumed that a typical data sequence for both $d[n]$ and $x[n]$ has been measured and recorded, and these sequences can be used to design the filter. The sequences x and d may be represent the input and output of a linear system, respectively. Building a filter to estimate d from x then amounts to designing a filter that behaves like the linear system. This is the system identification problem. Alternatively, the goal of the filter may really to be estimate an unknown signal from a known signal. In filtering, forecasting should be short-term. If the forecasting term is longer success of forecasting would decrease.

It is assumed that large sequences (d and x) are recorded. By using them, a filter can be designed. Unknown signal (d) can be estimated from known signal (x) by using designed filter. $x[0], \ldots, x[N_s\text{-}1]$ and $d[0], \ldots, d[N_s\text{-}1]$ are our sequences (or data) where x is input, d is output and N_s is number observations that used to design the filter. If a filter of length P (order) is used, then the estimate for the given data sequence is

$$\hat{d}[n] = \sum_{k=0}^{P-1} h[k]x[n-k] \tag{3.1}$$

and the error can be defined as

$$\varepsilon = d[n] - \hat{d}[n] \tag{3.2}$$

where $h[k]$ is filter coefficients, x is known signal (or data) and $\hat{d}$ is estimated signal. The aim is to design the filter to minimize the sum of the squared errors. n_I and n_F are initial and final values of sequences. If equation (3.1) is written in matrix form

$$\hat{d} = Xh \tag{3.3}$$

where

$$\hat{d} = \begin{bmatrix} \hat{d}[n_I] \\ \hat{d}[n_I + 1] \\ \vdots \\ \\ \vdots \\ \hat{d}[n_F] \end{bmatrix} \tag{3.4}$$

$$X=\begin{bmatrix} x[n_I] & x[n_I-1] & \cdots & x[n_I-P-1] \\ x[n_I+1] & x[n_I] & \cdots & x[n_I-P] \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x[n_F] & x[n_F-1] & \cdots & x[n_F-P+1] \end{bmatrix} \tag{3.5}$$

and

$$h=\begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[P-1] \end{bmatrix} \tag{3.6}$$

The matrix X is called data matrix and has dimension K x P where $K = n_F - n_I + 1$. It will be assumed that $K > P$. In fact that, usually K is considerably larger than P. d and ε are K dimensional vectors formed from the corresponding samples of $d[n]$ and $\varepsilon[n]$, respectively. The error can be defined as

$$\varepsilon = d - \hat{d} \tag{3.7}$$

and the problem is minimize

$$S = \|\varepsilon\|^2 = \varepsilon^T \varepsilon \tag{3.8}$$

If equations (3.3) and (3.7) are substituted in (3.8)

$$\begin{aligned} S &= (d - Xh)^T (d - Xh) \\ &= d^T d - h^T X^T d - d^T Xh + h^T X^T Xh \end{aligned} \tag{3.9}$$

If $S = d^T d - d^T Xh$ (See equation (3.13))

$$(X^T X)h = X^T d \tag{3.10}$$

Equation (3.10) is the least squares Wiener-Hopf equation. $X^T X$ has the form of an estimated correlation matrix. The filter coefficients (h) are

$$h = \left(X^T X\right)^{-1} X^T d \tag{3.11}$$

The sum of squared errors for the optimal filter can be found by using (3.9)

$$\begin{aligned} S &= (d - Xh)^T (d - Xh) \\ &= d^T (d - Xh) - (Xh)^T (d - Xh) \\ &= d^T d - d^T Xh - h^T \left(X^T d - X^T Xh\right) \end{aligned} \tag{3.12}$$

The term in parentheses is zero. If equation (3.12) is rewritten

$$\boxed{S = d^T d - d^T Xh} \tag{3.13}$$

3.2.2 AR Modeling Linear Prediction

Linear prediction is a special case of optimal filtering where $d[n]=x[n]$ and the observations are $x[n\text{-}1]$, $x[n\text{-}2]$, … ,$x[n\text{-}P]$ (P is order). First of all the autoregressive (AR) model should be described. If $\hat{x}$ is estimated value and a_k's (k = 1,2, …,P) are filter coefficients (or model parameters), AR model can be written as

$$\hat{x}[n] = -a_1 x[n-1] - a_2 x[n-2] \;\cdots\; -a_P x[n-P] \tag{3.14}$$

and the error is

$$\varepsilon[n] = x[n] - \hat{x}[n] = \sum_{k=0}^{P} a_k x[n-k] \qquad a_0 = 1 \tag{3.15}$$

Let us use (3.14) to predict the given data from n = n_I to n = n_F and write the equations in matrix form as

$$\begin{bmatrix} \hat{x}[n_I] \\ \hat{x}[n_I+1] \\ \vdots \\ \vdots \\ \hat{x}[n_F] \end{bmatrix} = - \begin{bmatrix} x[n_I-1] & x[n_I-2] & \cdots & x[n_I-P] \\ x[n_I] & x[n_I-1] & \cdots & x[n_I-P+1] \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x[n_F-1] & x[n_F-2] & \cdots & x[n_F-P] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \tag{3.16}$$

Equation (3.16) can be written as

$$\hat{x}_0 = -X_1 a \tag{3.17}$$

where

$$X_1 = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_P \\ | & | & & | \end{bmatrix} \qquad x_i = \begin{bmatrix} x[n_I-i] \\ x[n_I+1-i] \\ \vdots \\ \vdots \\ x[n_F-i] \end{bmatrix} \quad i = 0,1,\cdots,P \tag{3.18}$$

and

$$a' = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{bmatrix} \tag{3.19}$$

If the error vector for filtering problem is defined as

$$\varepsilon = x_0 - \hat{x}_0 \tag{3.20}$$

and then

$$\begin{bmatrix} | & \\ x_0 & X_1 \\ | & \end{bmatrix} \begin{bmatrix} 1 \\ a' \end{bmatrix} = Xa = \varepsilon \tag{3.21}$$

X defined as

$$X = \begin{bmatrix} | & \\ x_0 & X_1 \\ | & \end{bmatrix} = \begin{bmatrix} | & | & & | \\ x_0 & x_1 & \cdots & x_P \\ | & | & & | \end{bmatrix} \tag{3.22}$$

and

$$a = \begin{bmatrix} 1 \\ a' \end{bmatrix} = \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} \tag{3.23}$$

If $S = \varepsilon^T \varepsilon$ is the minimum sum of squared errors, then the Orthogonality Theorem can be written as

$$X^T \varepsilon = \begin{bmatrix} -- & x_0^T & -- \\ & X_1^T & \end{bmatrix} \varepsilon = \begin{bmatrix} S \\ 0 \end{bmatrix} \tag{3.24}$$

Combining (3.21) and (3.24) then yields

$$(X^T X)a = \begin{bmatrix} S \\ 0 \end{bmatrix} \tag{3.25}$$

This is the least squares form of the Yule-Walker equations. $X^T X$ is the correlation (unreversed) matrix. Since S is the sum of the squared errors, an estimate for the prediction error variance can be computed from

$$\sigma_{\varepsilon_P}^2 = \frac{1}{n_F - n_I + 1} S \tag{3.26}$$

In equation (3.25) there are two unknown variables; S and a_k. First, S should be found. If equation (3.25) should be rewritten by using (3.23)

$$\begin{bmatrix} 1 \\ a' \end{bmatrix} = (X^T X)^{-1} \begin{bmatrix} S \\ 0 \end{bmatrix} \tag{3.27}$$

If c is first value on the diagonal of $(X^T X)^{-1}$ matrix,

where

$$(X^T X)^{-1} = \begin{bmatrix} c & \cdots \\ \cdots & \cdots \end{bmatrix}_{PxP} \tag{3.28}$$

First multiplication in equation (3.27) $1 = cS$, therefore

$$S = 1/c \tag{3.29}$$

After the determining S, it substitute in equation(3.27), and then a_k is determined. If prediction errors are written in matrix form by using equation (3.21)

$$\begin{bmatrix} x[n_I] & x[n_I-1] & \cdots & x[n_I-P] \\ x[n_I+1] & x[n_I] & \cdots & x[n_I-P+1] \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x[n_F] & x[n_F-1] & \cdots & x[n_F-P] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \varepsilon[n_I] \\ \varepsilon[n_I+1] \\ \vdots \\ \vdots \\ \varepsilon[n_F] \end{bmatrix} \tag{3.30}$$

3.2.3 Order Selection Criteria

Selecting the order is one of the most difficult problems in developing a linear model for data. In most realistic situations, the model order is unknown, but it must be estimated prior to solving the parameter estimation problem. According to the Proakis&Manolakis (1996, p. 931) as a general rule if we select a model with too low an order, we obtain a highly smoothed model. On the other hand, if P is selected to high, we run the risk of introducing spurious low-level peaks in the model. Therefore order selection is very important in performance of the model.

In the literature there are a lot of suggestions and algorithms for order selection. Most of suggestions are based on prediction error variance of the model. The prediction error variance decreases when the order increases (Proakis&Manolakis, 1996, p.931),

(Therrien, 1992, p.548). The rate of decrease could be monitored. When the rate of decrease becomes relatively slow the order is chosen. This procedure may not work well in practice everytime. The prediction error variance may not decrease monotonically.

In the literature, there are four well-known algorithms (Therrien, 1992, pp. 548-549). These are Akaike's information-theoretic criteria (AIC), Parzen's criterion autoregressive transfer (CAT), Akaike's final prediction error (FPE) and Schwartz&Rissanen's minimum description length (MDL). These are computed from the prediction error variance $\left(\sigma_{\varepsilon_P}^2\right)$, and typically are applied in the following way. First the range of AR model orders to be considered is selected (e.g., $0 \leq p \leq p_{\max}$). Next, for each p model parameters are estimated under the assumption that p is the correct model order ($p = P$). Prediction error variance for this model is then calculated. From this error variance the selected criterion (one of AIC, CAT, FPE, MDL) value for this p may be obtained. These criteria are listed in Table 3.1 (Therrien, 1992, p. 549). Finally p yielding the lowest value of the selected criterion is chosen as the best estimate of the true model. Each of these has a distinct minimum at the optimal order. For each of the criteria, when the prediction error variance decreases, value of the criterion increases monotonically. If these values decrease when the prediction error variance decreases, then optimal p can be chosen in stage p.

Table 3.1 Criteria for AR Model Order Selection

AIC	$\mathrm{AIC}(P) = N_s \ln \sigma_{\varepsilon_P}^2 + 2P$
CAT	$\mathrm{CAT}(P) = \left(\frac{1}{N_s}\sum_{p=1}^{P}\frac{N_s - p}{N_s \sigma_{\varepsilon_P}^2}\right) - \frac{N_s - P}{N_s \sigma_{\varepsilon_P}^2}$
FPE	$\mathrm{FPE}(P) = \sigma_{\varepsilon_P}^2 \left(\frac{N_s + P + 1}{N_s - P - 1}\right)$
MDL	$\mathrm{MDL}(P) = N_s \ln \sigma_{\varepsilon_P}^2 + P \ln N_s$

Ulrych and Bishop (1975), Jones (1976), and Berryman (1978), found that the FPE(P) criterion tends to underestimate the model order. Kashyap (1980) showed that the AIC criterion is statistically inconsistent as $N \rightarrow \infty$. On the other hand, the MDL information

criterion proposed by Rissanen is statistically consistent. (Proakis&Manolakis, 1996, p. 932).

Another procedure is effective rank for selection order. The rank of the data matrix X is denoted by r and considers a rank $r' < r$ approximation to X in the form

$$X_{r'} = \sum_{k=1}^{r'} \sigma_k u_k v_k^T$$

where σ_k are the singular values, u_k are the left singular vectors, and v_k the right singular vectors of X (See Appendix 1). If the singular values obey the standard convention $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r'}$, then it can be shown that $X_{r'}$ is an optimal approximation to X in the sense that Euclidean norm

$$\|X - X_{r'}\|$$

or the Frobenius norm

$$\|X - X_{r'}\|_F$$

of the difference matrix is minimal over all possible approximations to X that have rank r'. A procedure for selecting the order based on the Frobenius norm is as follows. First, dimension of X is chosen to be somewhat larger that the suspected order of the model. Next find the singular values of X and define the reduced rank r' such that

$$\left(\frac{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_{r'}^2}{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}\right)^{\frac{1}{2}} \approx 1$$

The quantity on the left is the ratio of the Frobenius norm of the approximation to the Frobenius norm of the original matrix. The smallest value of r' that satisfies this condition will be called the *effective rank*. Choose the order P of the model to be equal to r'-1.

Another algorithm is proposed by Mati Wax (Wax, 1988, pp. 581-588). The central notation in the predictive least squares (PLS) principle is that of the true prediction error. In

AR models, the true prediction error is defined as the error in predicting x_t from the past observations (x_0,..., x_{t-1}). Thus, considering an AR model of order k, the true prediction error is given by

$$\hat{e}_{k,t} = x_t - \sum_{i=1}^{k} \hat{a}_{k,i,t} x_{t-i} \tag{3.31}$$

where $\hat{a}_{k,i,t}$ are the model parameters determined from the past data (x_0,..., x_{t-1}) by the conventional least squares criterion

$$\min_{a_1,\cdots,a_k} \sum_{i=1}^{t-1} \bar{e}_{k,i}^2 \tag{3.32}$$

with

$$\bar{e}_{k,i} = x_i - \sum_{j=1}^{k} a_j x_{i-j} \tag{3.33}$$

Notice that $\bar{e}_{k,i}$ is the conventional residual of x_i obtained by a predictor whose coefficients are computed from the data (x_0,..., x_{t-1}), in contrast to the true prediction error $\hat{e}_{k,t}$ where only the past data (x_0,..., x_{t-1}) are used.

The PLS criterion asserts that the order should be selected as that for which the sum of the true prediction errors along the sequence is minimized. Restricting the maximum order to $p_{\max}$, this gives the procedure,

$$\min_{k \le p_{\max}} \mathrm{PLS}(k) \tag{3.34}$$

where

$$\mathrm{PLS}(k) = \frac{1}{n} \sum_{t=1}^{n} \hat{e}_{k,t}^2 \tag{3.35}$$

To compute the PLS criterion, the true prediction errors must be recomputed for every order and for every sample point along the sequence. Thus, for order k and sample point t, first, model parameters $\hat{a}_{k,i,t}$ are computed from (3.32) and (3.33), and then compute the prediction error $\hat{e}_{k,t}$ from (3.31). The solution of (3.32) and (3.33) is given by the well-known formula

$$\hat{a}_{k,t} = \left(\sum_{i=1}^{t} x_{k,i} x_{k,i}^T \right)^{-1} \sum_{i=1}^{t} x_{k,i} x_i \tag{3.36}$$

where $\hat{a}_{k,t}$ is the $k \times 1$ vector

$$\hat{a}_{k,t} = \left[\hat{a}_{k,1,t}, \cdots, \hat{a}_{k,k,t} \right] \tag{3.37}$$

and $x_{k,t}$ is the is the $k \times 1$ vector

$$x_{k,i} = \left[x_{i-1}, \cdots, x_{i-1-k} \right] \tag{3.38}$$

Performing the operations implied by (3.34), (3.35), (3.36), (3.37) and (3.38) for every order and every sample point along the sequence is computationally expensive. If PLS criterion is summarized;

Initialization:

$e_{0,t} = r_{0,t} = x_t$

$R^e_{0,t} = R^r_{0,t} = R^e_{0,t-1} + x_t^2$

$\gamma_{-1,t} = 1$

Main loop, to be done for $k = 0$ *to* $min\{p_{max}, t\}$ -1*:*

$\theta_{k+1,t} = \theta_{k+1,t-1} + e_{k,t} r_{k,t-1} / \gamma_{k-1,t-1}$

$\gamma_{k,t} = \gamma_{k-1,t} - r_{k,t}^2 / R^r_{k,t}$

$e_{k+1,t} = e_{k,t} - \theta_{k+1,t} r_{k,t-1} / R^r_{k,t-1}$

$R^e_{k+1,t} = R^e_{k,t} - \theta^2_{k+1,t} / R^r_{k,t-1}$

$r_{k+1,t} = r_{k,t-1} - \theta_{k+1,t} e_{k,t} / R^e_{k,t}$

$R^r_{k+1,t} = R^r_{k,t-1} - \theta^2_{k+1,t} / R^e_{k,t}$

$\hat{e}_{k+1,t} = e_{k+1,t} / \gamma_{k,t-1}$

The determined $\hat{e}$'s are used in (3.35) for computing PLS. Then order is selected by using (3.34).

Also, Makano&Tagami (1987) developed a method by modifying FPE. Liang, Wilkes and Cadzow (1993) proposed a method that model order estimation based on eigenvalues of covariance matrix, Konstantinides (1991) used singular value decomposition for

determining order, Choi (1993) developed an algorithm by using chi-square statistics. Goto, Nakamra and Uosaki (1995) used non stationary data in their researches, Broressen (1985) developed an order selection algorithm for small. Katsikas, Likohanassis and Lainiatis (1990) proposed a new method for simultaneously selecting the order and identifying model parameters of an AR model.

Some experimental results indicate that for small data lengths, the order of the AR model should be selected to be in the range N/3 to N/2 for good results (Proakis&Manolakis, 1996, p. 932). We mention six methods most well known, which are AIC, CAT, FPE, MDL, effective rank and an algorithm based on Predictive Least Squares (PLS).

3.2.4 AR Modeling Algorithms

3.2.4.1 Autocorrelation Method

In this method initial and final values of X (data matrix) are $n_I = 0$ and $n_F = N_s + P - 1$. Values that early from the initial value $\left(x[n_I - 1] \cdots x[n_I - P + 1]\right)$ and late from the final value $\left(x[N_s] \cdots x[N_s + P - 1]\right)$ are zero. Some predicting values are estimated from zero values. For this method the data matrix has the specific structure

$$X = \begin{bmatrix} x[0] & 0 & \cdots & 0 \\ x[1] & x[0] & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ x[P] & x[P-1] & \cdots & x[0] \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x[N_s - 1] & x[N_s - 2] & \cdots & x[N_s - P - 1] \\ 0 & x[N_s - 1] & \cdots & x[N_s - P] \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x[N_s - 1] \end{bmatrix} \tag{3.39}$$

$$X=\begin{bmatrix} x[0] & 0 & \cdots & 0 \\ x[1] & x[0] & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ x[P] & x[P-1] & \cdots & x[0] \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ x[N_s-1] & x[N_s-2] & \cdots & x[N_s-P-1] \\ 0 & x[N_s-1] & \cdots & x[N_s-P] \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x[N_s-1] \end{bmatrix} \tag{3.39}$$

X data matrix is substituted in equation (3.25) and S and a_k are determined with way that explained in equations (3.27), (3.28), (3.29). $X^T X$ is a Toeplitz correlation matrix.

This method has a disadvantage. It is assumed that unmeasured data are zero. The filter coefficients (a_k) are biassed. Therefore some predictions, that are predicted zeros, may be incorrect.

3.2.4.2 Covariance Method and Prewindoved Covariance Method

In covariance method $n_I = P$ and $n_F = N_s - 1$ are chosen. There isn't any zero in data matrix, therefore zeros aren't used in the prediction. In other words, all values in data matrix are measured. The data matrix has specific form

$$X=\begin{bmatrix} x[P] & x[P-1] & \cdots & x[0] \\ \vdots & \vdots & \cdots & \vdots \\ & & & \\ \vdots & \vdots & \cdots & \vdots \\ x[N_s-1] & x[N_s-2] & \cdots & x[N_s-P-1] \end{bmatrix} \tag{3.40}$$

After determining X (data matrix), S and a_k are found by using equations (3.27), (3.28), (3.29). A variation of covariance method called prewindowed covariance method. Unless covariance method, $n_I = 0$ is chosen in this method. $n_F = N_s - 1$ is same the covariance method. The data matrix is

In these methods $X^T X$ is not a Toeplitz correlation matrix. The covariance method is often preferred, because it makes use of only measured data. This avoids the bias of the filter coefficients that is caused in the autucorrelation method.

3.2.4.3 Modified Covariance Method

Autocorrelation, covariance and prewindowed covariance methods are based on forward least squares linear prediction. Modified covariance method based on both of the forward and backward least squares linear prediction. Before the explaining modified covariance method, backward least squares linear prediction will be examined, briefly.

First change will be in data matrix structure. If $\tilde{X}$ is data matrix for the backward least squares linear prediction, P is order, n_I is initial value, n_F is final value then $\tilde{X}$ structure is

$$\tilde{X} = \begin{bmatrix} x[n_F - P] & x[n_F - P - 1] & \cdots & x[n_F] \\ x[n_F - P + 1] & x[n_F - P] & \cdots & x[n_F - 1] \\ \vdots & \vdots & & \vdots \\ & & & \\ \vdots & \vdots & & \vdots \\ x[n_I - P] & x[n_I - P + 1] & \cdots & x[n_I] \end{bmatrix} \tag{3.42}$$

If $\hat{\tilde{x}}$ is estimated values, and backward model parameters are defined a

$$\hat{\tilde{x}} = -\tilde{X}a \tag{3.43}$$

and the backward prediction errors can be written

$$\tilde{\varepsilon}^b = \tilde{X}a \tag{3.44}$$

Least squares Yule-Walker equations for the backward problem have the form

$$\left(\tilde{X}^T\tilde{X}\right)a=\begin{bmatrix}S^b\\0\end{bmatrix} \tag{3.45}$$

S^b is the backward sum of squared errors. Determining the model parameters (a_k) like as in the forward linear prediction.

In this least squares formulation of the linear prediction problem there is a question. Is sum of squared errors S^b in equation (3.45) equal to the sum of squared errors S in the forward Yule-Walker equations (3.25)? The answer depends on the reversed and unreversed correlation matrix ($\tilde{X}^T\tilde{X}$ and X^TX respectively). If the correlation matrix is Toeplitz, as in the autocorrelation method, the solution to (3.25) and (3.45) are identical and $S^b = S$. If the correlation matrix is not a Toeplitz, as in the covariance method, two sets of parameters will be different.

After the brief knowledge about backward prediction, explaining the modified covariance method can be started. The sum of squared errors for the modified covariance method is,

$$S^{fb}=\sum_{n=P}^{N_x-1}\left(\left|\varepsilon[n]\right|^2+\left|\varepsilon^b[n]\right|^2\right)=\left\|\varepsilon\right\|^2+\left\|\varepsilon^b\right\|^2 \tag{3.46}$$

the error terms

$$\left\|\varepsilon\right\|^2=\varepsilon^T\varepsilon=a^TX^TXa \tag{3.47}$$

and

$$\left\|\varepsilon^b\right\|^2=\left\|\tilde{\varepsilon}^b\right\|^2=\left(\tilde{\varepsilon}^b\right)^T\tilde{\varepsilon}^b=a^T\tilde{X}^T\tilde{X}a \tag{3.48}$$

Substituting last expressions in (3.46)

$$S^{fb}=a^T\left(X^TX+\tilde{X}^T\tilde{X}\right)a \tag{3.49}$$

This quantity is to be minimized subject to the constraint that first component of a is 1,

$$a^T\imath=1 \tag{3.50}$$

where ι is the vector [1 0 ... 0]^T. The problem can be solved formally using a single Lagrange multiplier to incorporate the real constraint.

$$\nabla_a\left[a^T\left(X^TX+\tilde{X}^T\tilde{X}\right)a+\lambda\left(1-a^T\iota\right)\right]=\left(X^TX+\tilde{X}^T\tilde{X}\right)a-\lambda\iota=0 \tag{3.51}$$

where λ is the Lagrange multiplier. Multiplying through by a^T yields

$$a^T\left(X^TX+\tilde{X}^T\tilde{X}\right)a-\lambda=0 \tag{3.52}$$

and comparison with (3.49) shows that

$$\lambda=S^{fb}$$

Finally, substituting this result in (3.51) and rearranging yields the modified covariance Yule-Walker equations

$$\boxed{\left(X^TX+\tilde{X}^T\tilde{X}\right)a=\begin{bmatrix}S^{fb}\\0\end{bmatrix}} \tag{3.53}$$

It is assumed that R is the correlation matrix. If $R = X^TX$, then the second term $\left(\tilde{X}^T\tilde{X}\right)$ is just $\tilde{R}$. The resulting correlation matrix is not a Toeplitz but does have a special structure. This procedure, that another name is forward-backward method, has two advantages. First it eliminates the dilemma of choosing between two different but equally valid estimates for the parameters. Second, it effectively provides twice as much data upon which to base the estimates and so reduces to the variances (Therrien, 1992, p. 542).

3.2.4.4 Burg's Method

In the early 1960's, John Parker Burg developed a method for spectral estimation based on AR modeling. The method is maximum entropy method. As a part of this method Burg developed an approach to estimate the AR model parameters directly from the data without the intermediate step of computing a correlation matrix and solving Yule-Walker equations

(Therrien, 1992, p 545). Burg's method based on the Levinson recursion (See Appendix 2.) and the lattice structure (See Appendix 3.).

The Burg procedure is a recursive procedure, and a single reflection coefficient is estimated at each step in the recursion. The p^{th} reflection is chosen to minimize the sum of the forward and backward squared prediction errors

$$S_p^{fb} = \sum_{n=p}^{N_s-1} \left(\left| \varepsilon_p[n] \right|^2 + \left| \varepsilon_p^b[n] \right|^2 \right) \tag{3.54}$$

According to the Burg, the forward and backward prediction problems are statistically identical. Therefore, both ε_p and ε_p^b should be included in the minimization criterion. Burg's method can be explained as follows. According to the lattice equation (See Appendix 3. equation (2)), p^{th} order forward and backward error terms satisfy the order recursions

$$\begin{bmatrix} \varepsilon_p[p] \\ \varepsilon_p[p+1] \\ \vdots \\ \varepsilon_p[N_s-1] \end{bmatrix} = \begin{bmatrix} \varepsilon_{p-1}[p] \\ \varepsilon_{p-1}[p+1] \\ \vdots \\ \varepsilon_{p-1}[N_s-1] \end{bmatrix} - \gamma_p \begin{bmatrix} \varepsilon_{p-1}^b[p-1] \\ \varepsilon_{p-1}^b[p] \\ \vdots \\ \varepsilon_{p-1}^b[N_s-2] \end{bmatrix} \tag{3.55}$$

and

$$\begin{bmatrix} \varepsilon_p^b[p] \\ \varepsilon_p^b[p+1] \\ \vdots \\ \varepsilon_p^b[N_s-1] \end{bmatrix} = \begin{bmatrix} \varepsilon_{p-1}^b[p-1] \\ \varepsilon_{p-1}^b[p] \\ \vdots \\ \varepsilon_{p-1}^b[N_s-2] \end{bmatrix} - \gamma_p' \begin{bmatrix} \varepsilon_{p-1}[p] \\ \varepsilon_{p-1}[p+1] \\ \vdots \\ \varepsilon_{p-1}[N_s-1] \end{bmatrix} \tag{3.56}$$

These equations can be written by defining the vectors

$$\varepsilon_p = \begin{bmatrix} \varepsilon_p[p] \\ -- \\ \\ e_p^f \end{bmatrix} \quad ; \quad \varepsilon_p^b = \begin{bmatrix} e_p^b \\ \\ ---- \\ \varepsilon_p^b[N_s-1] \end{bmatrix} \tag{3.57}$$

where

$$e_p^f = \begin{bmatrix} \varepsilon_p[p+1] \\ \varepsilon_p[p+2] \\ \vdots \\ \varepsilon_p[N_s-1] \end{bmatrix} \quad ; \quad e_p^b = \begin{bmatrix} \varepsilon_p^b[p] \\ \varepsilon_p^b[p+1] \\ \vdots \\ \varepsilon_p^b[N_s-2] \end{bmatrix} \tag{3.58}$$

Since $\gamma_p' = \gamma_p$, the lattice relations (3.55) and (3.56) can be written as

$$\begin{aligned} \varepsilon_p &= e_{p-1}^f - \gamma_p e_{p-1}^b \\ \varepsilon_p^b &= e_{p-1}^b - \gamma_p e_{p-1}^f \end{aligned} \tag{3.59}$$

With these definitions and relations equation (3.54) can be expressed as

$$\begin{aligned} S_p^{fb} &= \left\|\varepsilon_p\right\|^2 + \left\|\varepsilon_p^b\right\|^2 \\ &= \left(e_{p-1}^f - \gamma_p e_{p-1}^b\right)^T \left(e_{p-1}^f - \gamma_p e_{p-1}^b\right) + \left(e_{p-1}^b - \gamma_p e_{p-1}^f\right)^T \left(e_{p-1}^b - \gamma_p e_{p-1}^f\right) \\ &= \left(1+\gamma_p\gamma_p\right)\left(\left\|e_{p-1}^f\right\|^2 + \left\|e_{p-1}^b\right\|^2\right) - 2\gamma_p\left(e_{p-1}^f\right)^T e_{p-1}^b - 2\gamma_p\left(e_{p-1}^b\right)^T e_{p-1}^f \end{aligned} \tag{3.60}$$

Then using the expressions for the scalar form of the complex gradient provides the necessary condition(Therrien, 1992, p. 546)

$$\nabla_{\gamma_p} S_p^{fb} = \gamma_p\left(\left\|e_{p-1}^f\right\|^2 + \left\|e_{p-1}^b\right\|^2\right) - 2\left(e_{p-1}^f\right)^T e_{p-1}^b = 0 \tag{3.61}$$

or

$$\gamma_p = \frac{2\left(e_{p-1}^f\right)^T e_{p-1}^b}{\left\|e_{p-1}^f\right\|^2 + \left\|e_{p-1}^b\right\|^2} \tag{3.62}$$

The form of equation (3.62) guarantees that estimated reflection coefficient is $\left|\gamma_p\right| \le 1$. Equations (3.57), (3.59), (3.62) are summarized below:

$$\gamma_p = \frac{2\left(e^f_{p-1}\right)^T e^b_{p-1}}{\left\|e^f_{p-1}\right\|^2 + \left\|e^b_{p-1}\right\|^2} \quad (a)$$

$$\begin{bmatrix} \times \\ -- \\ \\ e^f_p \end{bmatrix} = e^f_{p-1} - \gamma_p e^b_{p-1} \quad (b) \qquad (3.63)$$

$$\begin{bmatrix} e^b_p \\ \\ -- \\ \times \end{bmatrix} = e^b_{p-1} - \gamma_p e^f_{p-1} \quad (c)$$

Iterations for $p = 1,2,\ldots,P$ start from the initial conditions

$$\begin{bmatrix} \times \\ -- \\ \\ e^f_0 \end{bmatrix} = \begin{bmatrix} e^b_0 \\ \\ -- \\ \times \end{bmatrix} = \begin{bmatrix} x[0] \\ x1 \\ \vdots \\ x[N_s - 1] \end{bmatrix} \qquad (3.64)$$

The algorithm is very simple in this form. At each stage of the iteration first of all reflection coefficient is computed from (3.63)(a). The vectors e^f_p and e^b_p are combined as in (b) and (c). The elements that denoted × are ignored and dropped, thus the vectors e^f_p and e^b_p decrease in size at each iteration, the steps are repeated. At each iteration the size of the prediction error filter grows (thus the number of the filter parameters increases), the original data remains in fixed size and fewer error terms are computed. An estimate for the prediction error variance can be computed by the additional recursion

$$\sigma^2_{\varepsilon_p} = \left(1 - |\gamma_p|^2\right)\sigma^2_{\varepsilon_{p-1}} \qquad (3.65)$$

The filter parameters can be computed from the recursion

$$a_p = \begin{bmatrix} a_{p-1} \\ -- \\ 0 \end{bmatrix} - \gamma_p \begin{bmatrix} 0 \\ -- \\ \tilde{a}_{p-1} \end{bmatrix} \tag{3.66}$$

$\tilde{a}$ is the reverse of the a vector.

3.2.4.5 Modified Iteratively Reweighted Least Squares Algorithm

Modified iteratively reweighted least squares algorithm is an l_p norm estimation algorithm. Since the l_p norm method provides a good approach to the parameter estimation problem, the l_p norm estimation method is discussed (Chen, Chen and Shern, 1994, p.1063-1072). Consider the general linear system model as follows:

$$y = Xa + \varepsilon \tag{3.67}$$

where

y n-vector of the observable random variables

X n-by-k matrix of known regressor variables

a k-vector of unknown parameters

ε n-vector of unobservable errors

The l_p norm estimation problem is then defined in the following: Find the parameter vector $\hat{a} = [\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_k]^T$ that minimizes

$$S(\hat{a}) = \sum_{i=1}^{n} |y_i - X_i \hat{a}|^p = \sum_{i=1}^{n} |\hat{\varepsilon}_i|^p \tag{3.68}$$

where X_i is the ith row vector of matrix X, y_i is the ith element of y. Let $\hat{\varepsilon} = y - X\hat{a}$ be the corresponding-vector of residuals (the estimate errors) and $\hat{\varepsilon}_i$ be the ith element of $\hat{\varepsilon}$. $\hat{a}$ is the l_p norm estimate of $a = [a_1, a_2, \cdots, a_k]^T$.

The iteratively reweighted least squares (IRLS) algorithm is applicable to a variety of optimization problems in which some function of residuals $\hat{\varepsilon}$ is minimized. For norm minimization, the IRLS solves the problem represented in (3.68) by iteratively computing

$$\hat{a}_{IRLS}^{k+1} = \left(X^T W^{(k+1)} X\right)^{-1} X^T W^{(k+1)} y \tag{3.69}$$

where the matrix $W^{(k+1)}$ is a diagonal matrix whose iith entry $\left(W^{(k+1)}\right)_{ii} = W_i^{(k+1)}$ is calculated from the residual $\hat{\varepsilon}_i^{(k)} = y_i X_i \hat{a}_{IRLS}^{(k)}$ ($k \equiv$ iteration index)

$$W_i^{(k+1)} = \left|\hat{\varepsilon}_i^{(k)}\right|^{p-2} \tag{3.70}$$

for $1 < p < 2$. $W_i^{(k+1)}$ can be chosen as follows

$$W_i^{(k+1)} = \begin{cases} \left|\hat{\varepsilon}_i^{(k)}\right|^{p-2} & \left|\hat{\varepsilon}_i^{(k)}\right| \geq \varepsilon \\ \varepsilon^{p-2} & \left|\hat{\varepsilon}_i^{(k)}\right| < \varepsilon \end{cases} \tag{3.71}$$

ε is a small positive number.

In fact that the minimum of $S(\hat{a})$ in (3.68) may be achieved finding the roots of

$$\frac{\partial S(\hat{a})}{\partial(\hat{a}_i)} = 0 \qquad i = 1,2,\cdots,k \tag{3.72}$$

whereas (3.72) can be solved iteratively by finding the solution of

$$\frac{\partial S\left(\hat{a}^{(k-1)}\right)}{\partial \hat{a}_i} + \sum_{j=1}^{n} \frac{\partial^2 S\left(\hat{a}^{(k-1)}\right)}{\partial \hat{a}_i \partial \hat{a}_j}\left(\hat{a}_j^{(k)} - \hat{a}_j^{(k-1)}\right) = 0 \tag{3.73}$$

which is the Newton-Raphson method. From (3.68), we find that (at kth iteration)

$$\begin{aligned}\frac{\partial S(\hat{a}^{(k-1)})}{\partial \hat{a}_i} &= -p\sum_{l=1}^{n}\left|y_l - X_l\hat{a}^{(k-1)}\right|^{p-1} X_{li}\,sign\left(y_l - X_l\hat{a}^{(k-1)}\right) \\ &= -p\sum_{l=1}^{n} W_l^{(k)}\left(y_l - X_l\hat{a}^{(k-1)}\right)X_{li} \\ \frac{\partial S(\hat{a}^{(k-1)})}{\partial \hat{a}_i \partial \hat{a}_j} &= p(p-1)\sum_{l=1}^{n}\left|y_l - X_l\hat{a}^{(k-1)}\right|^{p-2} X_{li}X_{lj} \\ &= p(p-1)\sum_{l=1}^{n} W_l^{(k)} X_{li}X_{lj}\end{aligned} \tag{3.74}$$

where

$$sign(m) = \begin{cases} 1 & \text{if } m > 0 \\ -1 & \text{if } m < 0 \\ 0 & \text{if } m = 0 \end{cases}$$

$W_l^{(k)} = \left|y_l - X_l\hat{a}^{(k-1)}\right|^{p-2}$, and X_{li} is the lith entry of matrix X.

Substituting (3.74) into (3.73)

$$(p-1)H^{(k)}\left(\hat{a}^{(k)} - \hat{a}^{(k-1)}\right) = D^{(k)} - H^{(k)}\hat{a}^{(k-1)}$$

where $H^{(k)}$ is a k-by-k matrix with $\sum_{l=1}^{n} W^{(k)} X_{li}X_{lj}$ in iith entry and

$$D^{(k)} = \left[\sum_{l=1}^{n} W_l^{(k)} y_l X_{l1}, \cdots, \sum_{l=1}^{n} W_l^{(k)} y_l X_{lk}\right]^T$$

Note that $H^{(k)} = X^T W^{(k)} X$ and $D^{(k)} = X^T W^{(k)} y$; then, by (3.69), we get $H^{(k)}\hat{a}_{IRLS}^{(K)} = D^{(k)}$. Therefore

$$(p-1)H^{(k)}\left(\hat{a}^{(k)} - \hat{a}^{(k-1)}\right) = H^{(k)} - H^{(k)}\hat{a}^{(k-1)}$$

It is noted that the matrix $H^{(k)}$ is non-singular. Thus, algorithm is

$$\hat{a}^{(k)} = \frac{(p-2)\hat{a}^{(k-1)} + \hat{a}^{(k)}_{IRLS}}{p-1} \tag{3.75}$$

The above algorithm is a Newton's type method, and it is a rearranged form of Newton-Raphson method. Since the Newton's method is theoretically the most desirable, the iteration scheme of (3.75) plays a major role in proposed l_p norm estimation algorithm for finding the roots of (3.72).

If the MIRLS algorithm is summarized;

Step 0: Give a small positive number ε and initial estimate $\hat{a}^{(j)}$ of a at $j = 1$.

Step 1: Calculate the residuals

$$\hat{\varepsilon}_i^{(j)} = y_i - X_i\hat{a}^{(j)}$$

Step 2: Set

$$W_i^{(j+1)} = \begin{cases} \left|\hat{\varepsilon}_i^{(j)}\right|^{p-2} & 2 \le p < \infty \\ \min\left(\left|\hat{\varepsilon}_i^{(j)}\right|^{p-2}, \varepsilon^{p-2}\right) & 1 < p < 2 \end{cases}$$

Step 3: Find the IRLS estimate of a

$$\hat{a}_{IRLS}^{j+1} = \left(X^T W^{(j+1)} X\right)^{-1} X^T W^{(j+1)} y$$

Step 4: Modify the IRLS estimation $\hat{a}_{IRLS}^{(j+1)}$ by the following formula

$$\hat{a}^{(j+1)} = \begin{cases} \hat{a}_{IRLS}^{(j+1)} & \text{if } 1 < p < \dfrac{3}{2} \\ \dfrac{\left((p-2)\hat{a}^{(j)} + \hat{a}_{IRLS}^{(j+1)}\right)}{p-1} & \text{if } \dfrac{3}{2} < p < \infty \end{cases}$$

Step 5: Let j = j + 1. If $\hat{a}$ converges then STOP, else go the *Step 1*

3.2.4.6 Predictive Least Squares Order Selection Criterion

In section 3.2.3.3, optimal order is chosen by predicting errors. In the end of the each iteration predicted errors ($\hat{e}_{k,i}$) were computed. If errors (real or predicted) are known model parameters would be determined by using equation (3.31).

$$\hat{e}_{k,t} = x_t - \sum_{i=1}^{k} \hat{a}_{k,i,t} x_{t-i} \tag{3.31}$$

If (3.31) is rearranged in matrix form for determining model parameters

$$\hat{a} = \left(X^T X\right)^{-1} X^T \left(y - \hat{e}\right) \tag{3.76}$$

where

X is (n x p) data matrix (historical data is from n_I to n_F - 1)

y is (n x 1) data vector (historical data is from n_I+ 1 to n_F)

$\hat{e}$ is (n x 1) predicted errors (from n_I+ 1 to n_F)

Statistical signal processing is widely used in AR model building. In this chapter, seven statistical signal processing algorithms were explained to estimate AR model parameters. These algorithms will be employed to build an AR model for forecasting of amount of sodium oleate in the neutrealization process of a crude oil.

CHAPTER FOUR
APPLICATION

4.1 Forecasting the Amount of Sodium Oleate in the Neutralization Process of Crude Oil

In an oil factory, crude oil is treated through a chemical process until it becomes edible. At the first step of the process, the crude oil is neutralized, and free fatty acid and sodium oleate in the oil are separated. In Table 4.1, the observations taken from the neutralization procedure in every twenty minutes are given. The quality engineers of the factory want to design a model to predict the amount of sodium oleate at any given time of the process. For this reason, 113 observations are taken from the process, and the first 101 observations are used to estimate model parameters by using seven signal processing algorithms. Last 12 observations are compared with 12 predicted values found by the each algorithm. Comparisons between predicted and real 12 values are used to determine the performance of an algorithm.

Analysis (model construction process) consists of three steps. The first step is autocorrelation analysis of data to identify the model, the second is the selection of the model order, and the third is to estimate the model parameters.

4.2 Autocorrelation Analysis of Data to Identify the Model

We accept that, an autocorrelation or partial autocorrelation coefficient is zero if the absolute value of its estimate is less than twice its standard error (Montgomery&Lynwood, 1976, p.209). It is useful to plot the limits $\pm 2S(\hat{\rho}_k)$ and $\pm 2S(\hat{\phi}_{kk})$ directly on the graphs of autocorrelation or partial autocorrelation functions. Results of the autocorrelation analysis of data are shown in Figure 4.1. In this figure, sample autocorrelation coefficient

Table 4.1 Observations

n	*x*[*n*]
1	650
2	685
3	855
4	795
5	580
6	570
7	608
8	638
9	616
10	684
11	545
12	786
13	758
14	866
15	748
16	730
17	646
18	948
19	904
20	813
21	711
22	357
23	456
24	418
25	380
26	418
27	380
28	471
29	494
30	608
31	616
32	740
33	594
34	354
35	354
36	494
37	418
38	699
39	532
40	740

n	*x*[*n*]
41	380
42	418
43	494
44	342
45	403
46	509
47	380
48	403
49	456
50	380
51	410
52	304
53	456
54	403
55	456
56	342
57	258
58	304
59	555
60	342
61	304
62	312
63	425
64	365
65	620
66	418
67	532
68	456
69	418
70	430
71	470
72	584
73	599
74	509
75	354
76	486
77	537
78	714
79	732
80	765

n	*x*[*n*]
81	576
82	855
83	773
84	478
85	620
86	532
87	354
88	502
89	555
90	608
91	580
92	438
93	570
94	608
95	412
96	675
97	616
98	354
99	418
100	608
101	634
102	516
103	412
104	570
105	516
106	328
107	354
108	546
109	686
110	516
111	624
112	580
113	546

at lag 1, 2 and 3 are found different from zero. So we accept that these observations are autocorrelated, and can be modeled by an ARMA model.

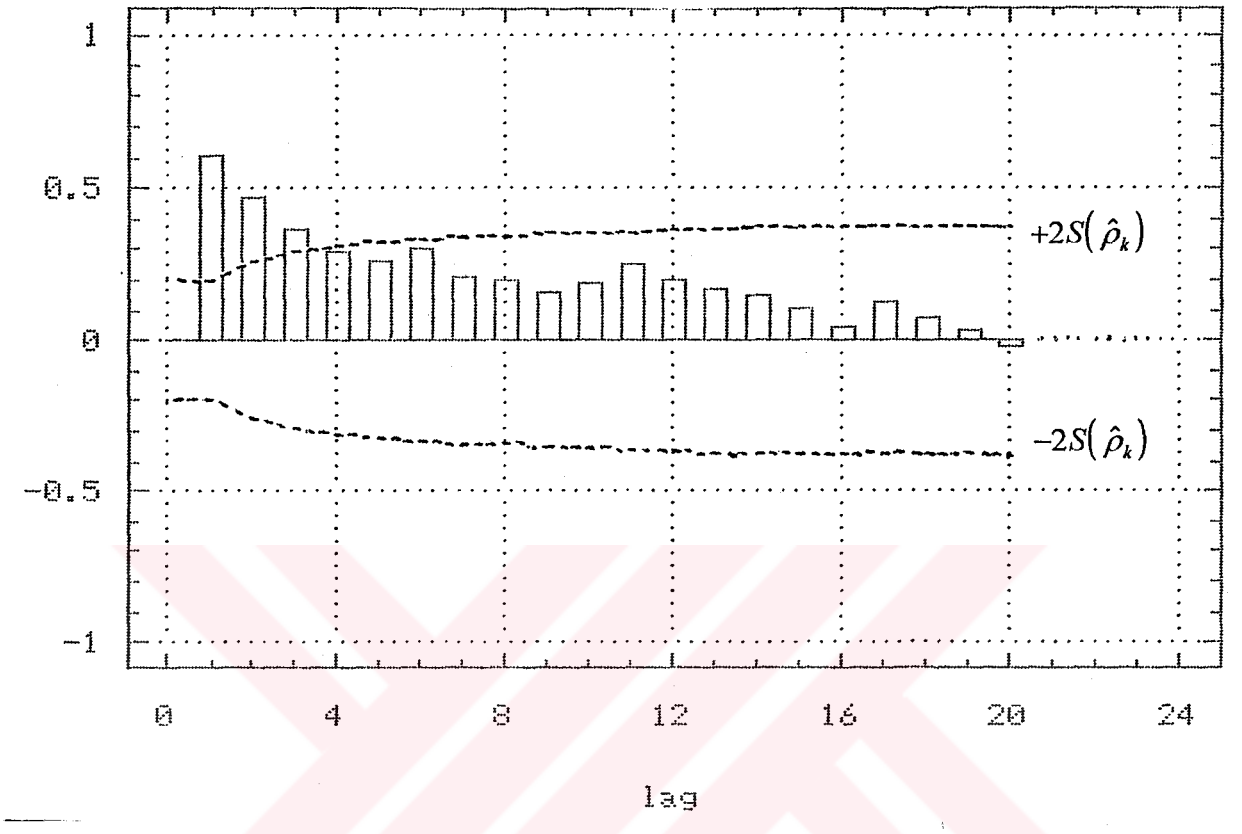

Figure 4.1 Autocorrelation Functions of Data (First 101 Observations)

In order to identify the class of ARMA model, partial autocorrelation function of data (Figure 4.2) is necessary. In section 2.3.2, selection of the class of ARMA model was explained. According to the Figure 4.2 and 4.3, autocorrelation function decreases during the lag, and partial autocorrelation function has only one coefficient different from zero at lag one. If the results are analyzed according to section 2.3.2 and Table 2.3, class of ARMA model for data is determined as AR. In both of the functions, the number of lag is selected as 20 (See section 2.3.2)

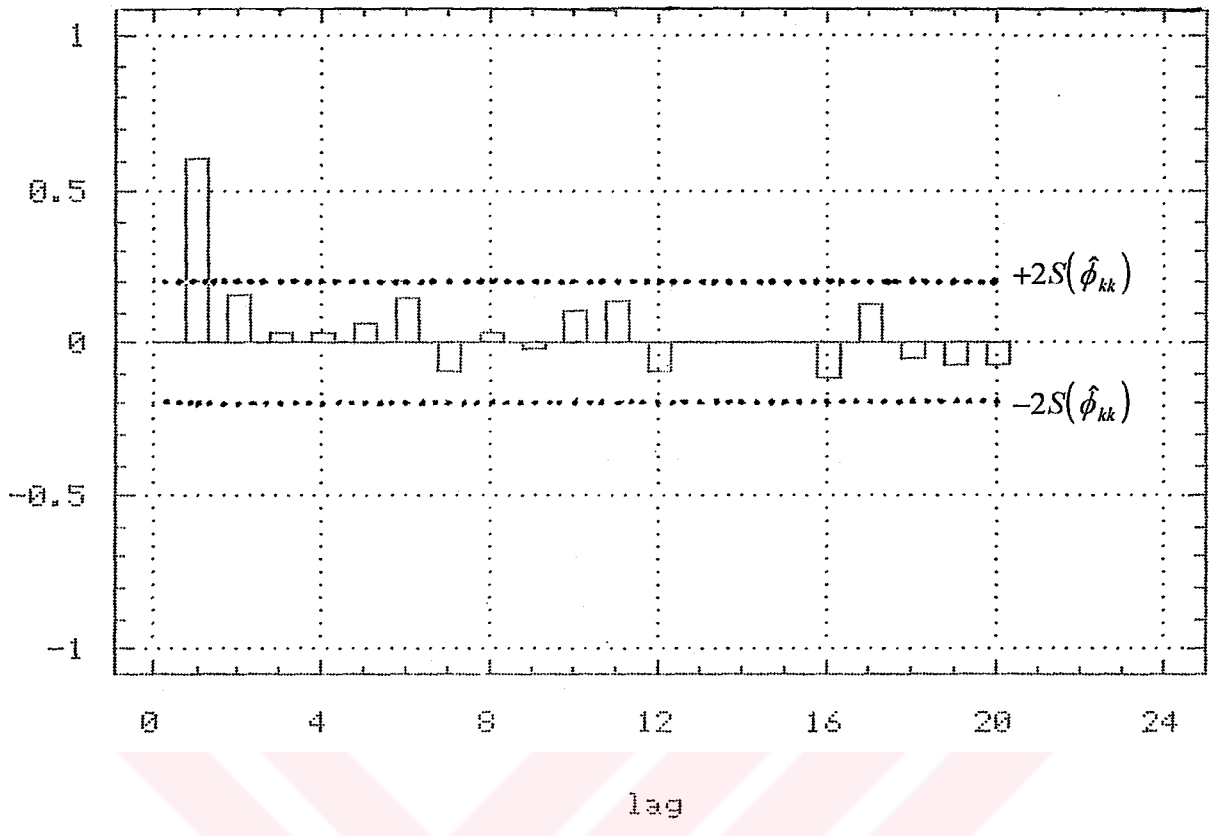

Figure 4.2 Partial Autocorrelation Functions of Data (First 101 Observations)

4.3 Selection of AR Model Order

There isn't any preknowledge about order of model, thus optimal order (or orders) is selected by using some criteria explained in chapter 3. In analyses maximum order is selected as 15.

According to the prediction error variance of model, and values of AIC, CAT, FPE, MDL criteria, optimal order is determined. Figures 4.3, 4.4, 4.5, 4.6 and 4.7 show the results. Computer program of these criteria is given in Appendix 4.1.

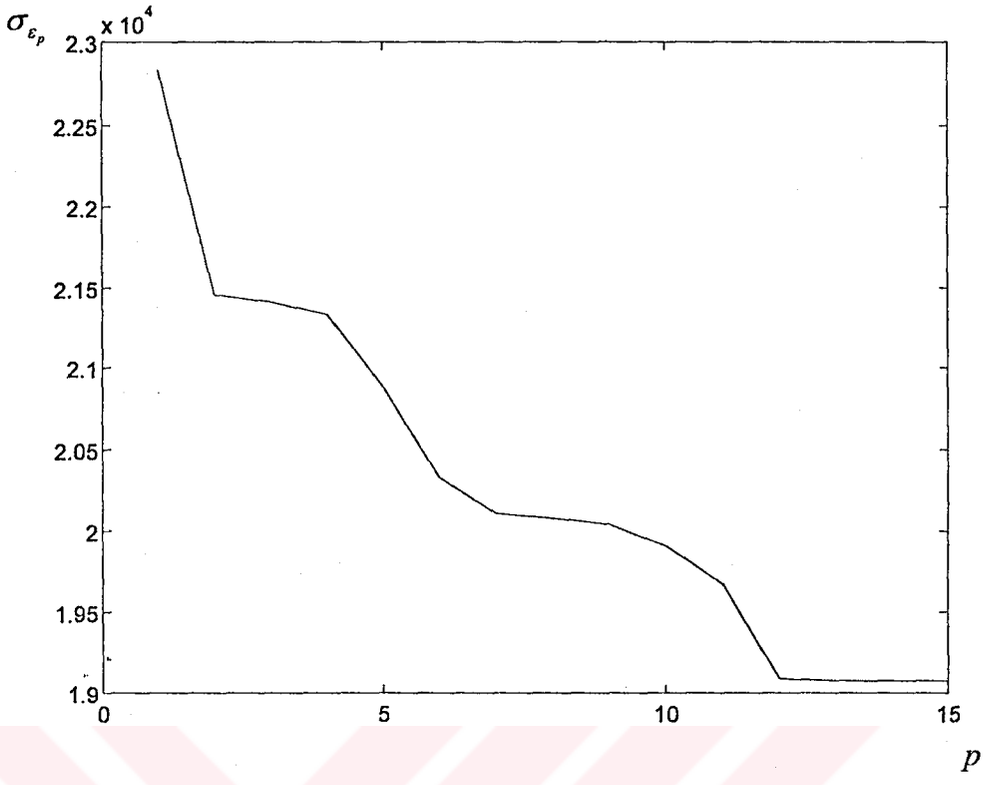

Figure 4.3 Variance of Errors

Optimal orders can be selected as 2,7 and 12, since variance of errors slows down at these orders.

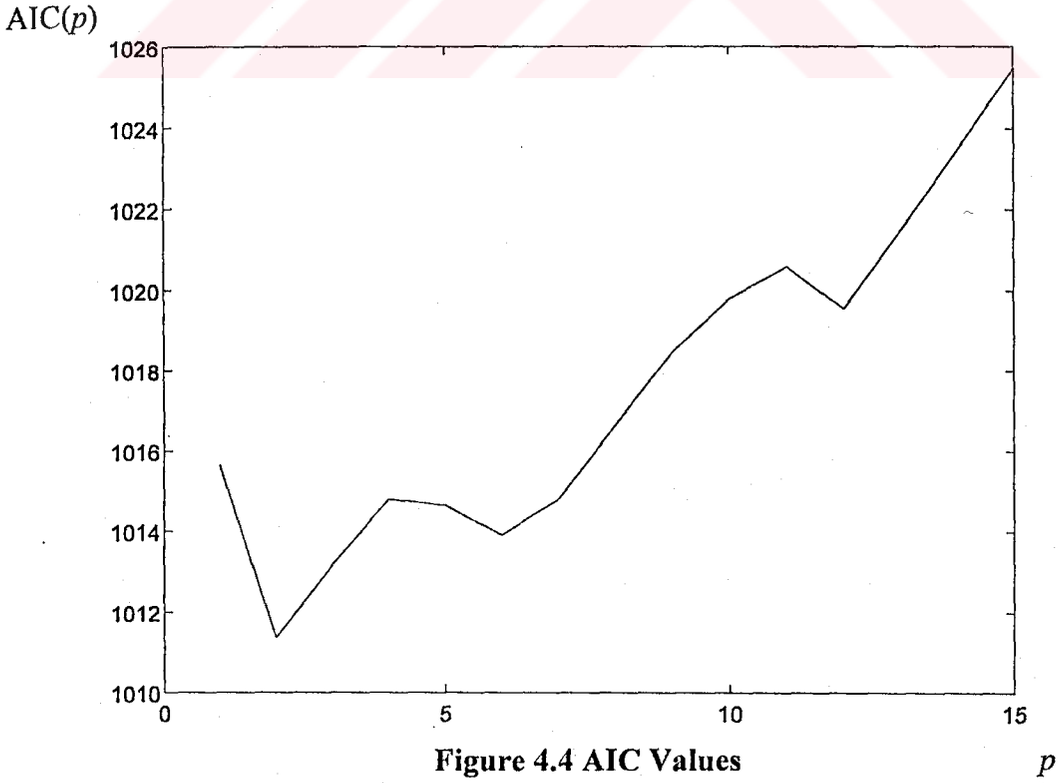

Figure 4.4 AIC Values

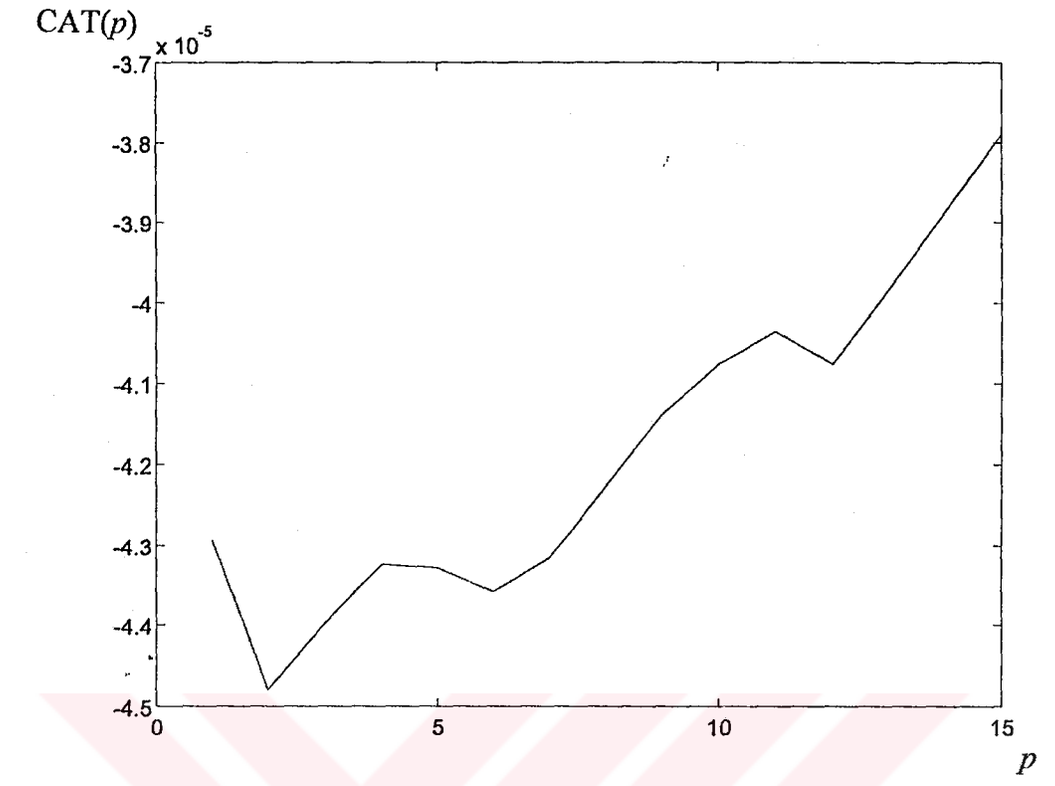

Figure 4.5 CAT Values

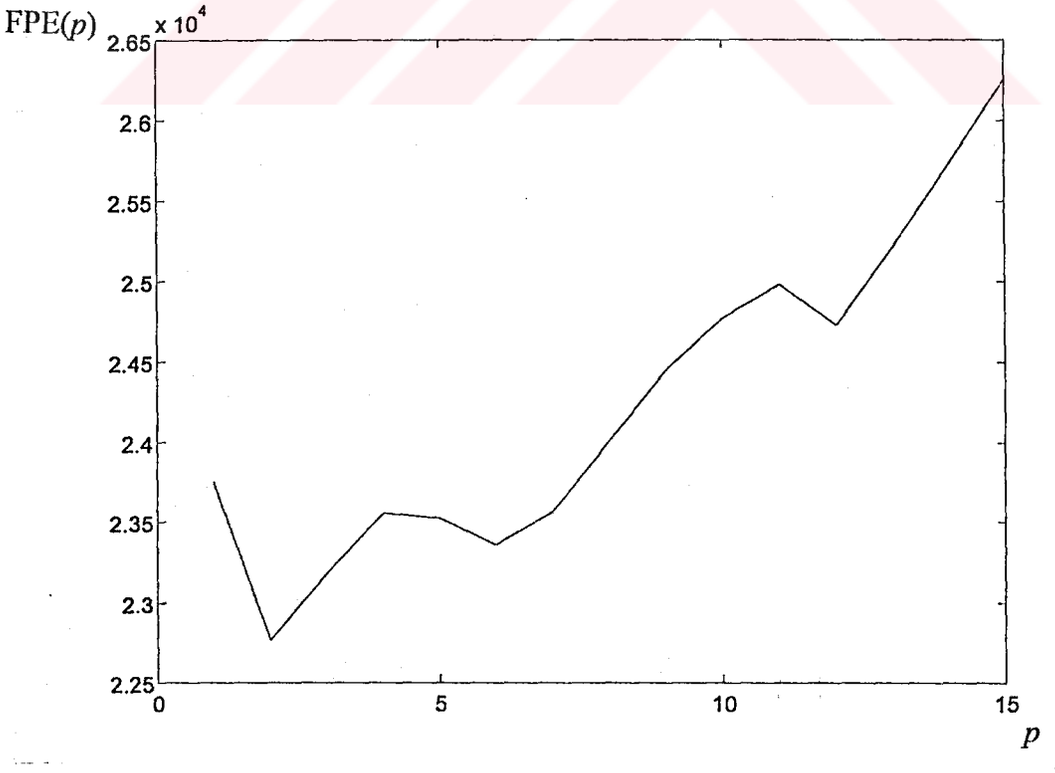

Figure 4.6 FPE Values

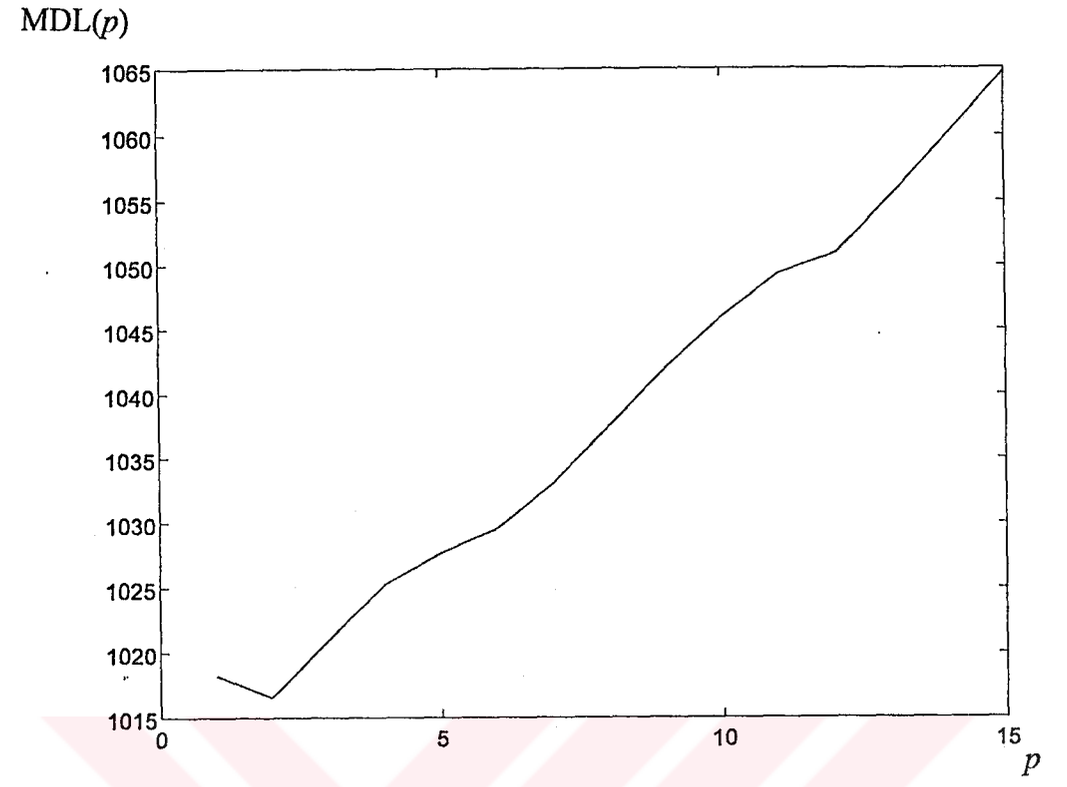

Figure 4.7 MDL Values

When variance of errors decreases, decreasing in value of criteria can be seen from the figures. For AIC, CAT and FPE criteria, orders may be selected as 2, 5, 6 and 12, since values of criteria decrease at these orders. But according to the MDL criterion, order is determined as 2, since values of criterion decrease at only this order.

Another order selection method is effective rank method, and its computer program is given in Appendix 4.2. In this method, first, an arbitrary value that is close to 1 is determined. In this study, 0.99 was selected. Results of the computer program are shown in Table 4.2, 0.99 is compared with values in this table. Effective rank can be selected 14, since 0.9904 is the nearest value to arbitrary value. Since the optimal order is equal to effective rank minus one (r'-1), order should be selected as 13.

Last order selection criterion in this study is PLS, the computer program is given Appendix 4.3. Results of computer program are shown in Figure 4.8. According to the results, minimum value of PLS is at order 2, thus order can be selected as 2.

Table 4.2 Values of Effective Rank

Effec. Rank(r')	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	0.7086	0.7728	0.8127	0.8436	0.8680	0.8899	0.9062	0.9214
Effec. Rank(r')	**9**	**10**	**11**	**12**	**13**	**14***	**15**	
	0.9346	0.9472	0.9594	0.9702	0.9806	0.9904	1.0000	

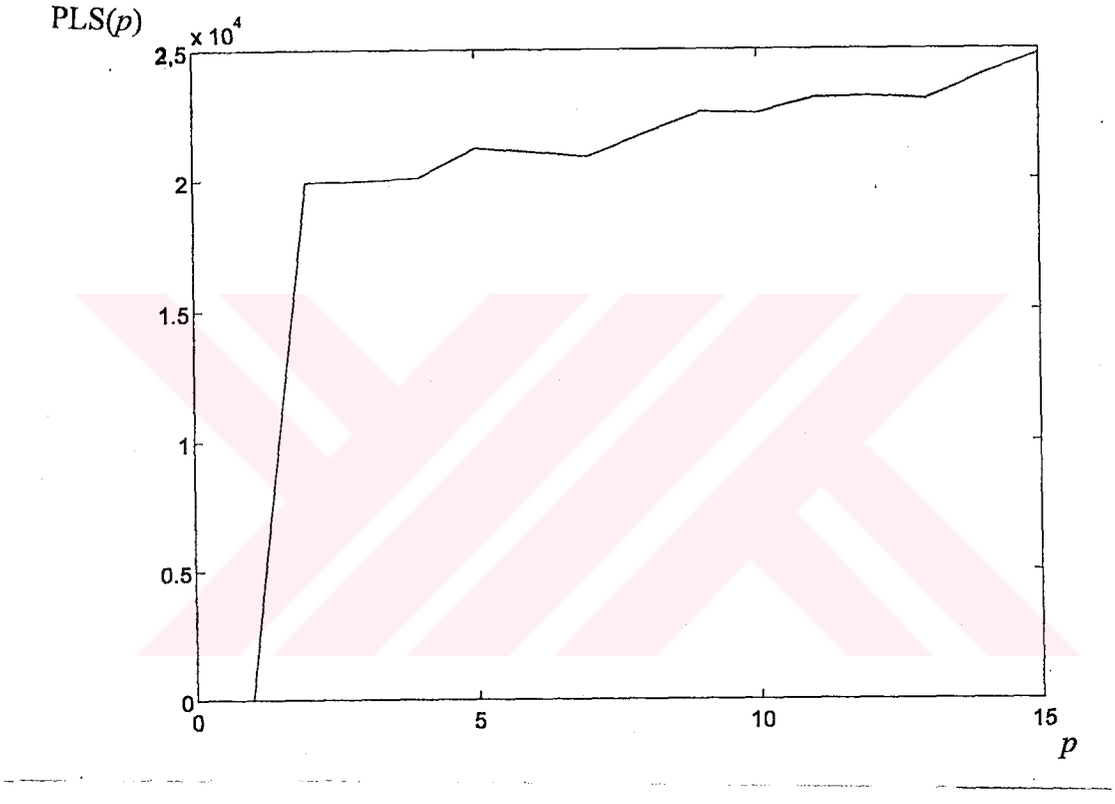

Figure 4.8 PLS Values

If all results are combined, orders, that will be used in parameter estimation, are 2, 5, 6, 7, 12 and 13.

4.4 Estimation of AR Model Parameters

After the order selection, seven AR modelling algorithms are used to estimate model parameters. In each algorithm, same steps are employed. These steps are below:

1. Model parameters are estimated for each order of 2,5,6,7,12 and 13.

2. According to the orders, MSEs are computed by using estimated model parameters, and MSEs are given in a table.

3. The best order for each algorithm is determined according to the minimum MSE, then the best AR model having minimum MSE is given.

In the following algorithms, results will be given in above sequence.

Autocorrelation Method

Computer program is given in Appendix 4.4.

Table 4.3 MSEs for Autocorrelation Method

Order	*p*=2	*p*=5	*p*=6	*p*=7	*p*=12	*p*=13*
MSE	17526	17056	16714	16603	15624	15609

The best order is 13, and the best AR model is,

$$\hat{x}[n] = 0.7309x[n-1] + 0.1488x[n-2] - 0.0037x[n-3] + 0.0138x[n-4] - 0.0059x[n-5] + 0.1225x[n-6] - 0.0406x[n-7] + 0.0095x[n-8] - 0.0781x[n-9] + 0.0479x[n-10] + 0.182x[n-11] - 0.1338x[n-12] - 0.019x[n-13]$$

Covariance Method

Computer program is given in Appendix 4.6.

Table 4.4 MSEs for Covariance Method

Order	*p*=2	*p*=5	*p*=6	*p*=7	*p*=12*	*p*=13
MSE	17084	15780	15138	15252	14298	14393

The best order is 12, and the best AR model is,

$$\hat{x}[n] = 0.5364x[n-1] + 0.1624x[n-2] + 0.0447x[n-3] - 0.0389x[n-4] - 0.029x[n-5] + 0.246x[n-6] - 0.068x[n-7] + 0.0216x[n-8] - 0.1368x[n-9] + 0.0801x[n-10] + 0.2357x[n-11] - 0.0715x[n-12]$$

Prewindowed Covariance Method

Computer program is given in Appendix 4.8.

Table 4.5 MSEs for Prewindowed Covariance Method

Order	*p*=2	*p*=5	*p*=6	*p*=7	*p*=12*	*p*=13
MSE	17623	17572	17166	17108	16916	17095

The best order is 12, and the best AR model is,

$$\hat{x}[n] = 0.7143x[n-1] + 0.2144x[n-2] - 0.0414x[n-3] - 0.0904x[n-4] + 0.0105x[n-5] + 0.2706x[n-6] - 0.1117x[n-7] - 0.0158x[n-8] - 0.072x[n-9] + 0.0536x[n-10] + 0.2455x[n-11] - 0.1842x[n-12]$$

Modified Covariance Method

Computer program is given in Appendix 4.10.

Table 4.6 MSEs for Modified Covariance Method

Order	*p*=2	*p*=5	*p*=6	*p*=7	*p*=12	*p*=13*
MSE	17465	16894	16354	16115	15129	15115

The best order is 13, and the best AR model is,

$$\hat{x}[n] = 0.7158x[n-1] + 0.2453x[n-2] - 0.086x[n-3] - 0.089x[n-4] + 0.0358x[n-5] + 0.2496x[n-6] - 0.1194x[n-7] - 0.0087x[n-8] - 0.0532x[n-9] + 0.0397x[n-10] + 0.2331x[n-11] - 0.1547x[n-12] - 0.0261x[n-13]$$

Recursive Approach

In the first four algorithms, model parameters were estimated using first 101 observations, and 12 future predictions were made using these parameters. In recursive approach, model parameters are updated for each prediction. For example, for 1st future prediction the model parameters are estimated from first 101 observations, for 2nd the model parameters are estimated from first 102 observations. New performances of the model parameters are computed by using this approach. Related computer programs are in Appendix 4.5, 4.7, 4.9 and 4.11. MSEs of two approaches are shown in Table 4.7.

Table 4.7 MSEs of Performance Approaches

Algorithms	*Approaches*	*Orders*					
		p=2	*p=5*	*p=6*	*p=7*	*p=12*	*p=13*
Autocorrel.	*First*	15173	14213	14847	15777	15838	15693
	Recursive	15384	15824	16786	18983	19739	20419
Covariance	*First*	15726	13823	16778	18233	17438	17338
	Recursive	15888	13667	15985	16520	15245	15305
Prew. Cov.	*First*	15388	16541	18825	22746	23303	22824
	Recursive	15558	16018	17863	20145	19563	19375
Mod. Cov.	*First*	15321	16653	18803	23298	23823	23373
	Recursive	15374	15948	17453	20258	20831	21607

According to the Table 4.7, MSEs found by recursive approach are worse than MSEs found by first approach for autocorrelation method. Also for order 2, MSEs found by recursive approach are worse than MSEs found by first approach for other methods. Recursive approach is better in other results. In this study, it is said that recursive approach is better for covariance, prewindowed covariance and modified covariance methods except for order 2, and first approach is better for autocorrelation method.

Burg's Algorithm

Computer program is given in Appendix 4.12.

Table 4.8 MSEs for Burg's Algorithm

Order	*p=2*	*p=5*	*p=6*	*p=7*	*p=12**	*p=13*
MSE	17563	17698	17397	17193	17011	17164

The best order is 12, and the best AR model is,

$$\hat{x}[n] = 0.5309x[n-1] + 0.151x[n-2] + 0.0355x[n-3] - 0.0045x[n-4] - 0.0283x[n-5] + 0.2411x[n-6] - 0.1034x[n-7] + 0.025x[n-8] - 0.0682x[n-9] + 0.0639x[n-10] + 0.2173x[n-11] - 0.0674x[n-12]$$

MIRLS Algorithm

Computer program is given in Appendix 4.13.

Table 4.9 MSEs for MIRLS Algorithm

Order	*p*=2*	*p*=5	*p*=6	*p*=7	*p*=12	*p*=13
MSE	17623	17717	17686	17824	22075	22307

The best order is 2, and the best AR model is,

$$\hat{x}[n] = 0.7326x[n-1] + 0.2472x[n-2]$$

PLS Algorithm

Computer program is given in Appendix 4.13.

Table 4.10 MSEs for PLS Algorithm

Order	*p*=2	*p*=5	*p*=6	*p*=7*	*p*=12	*p*=13
MSE	18070	18304	17629	17494	17664	17811

The best order is 7, and the best AR model is,

$$\hat{x}[n] = 0.8903x[n-1] + 0.2433x[n-2] - 0.2637x[n-3] - 0.0736x[n-4] + 0.1779x[n-5] + 0.2253x[n-6] - 0.2115x[n-7]$$

4.5 Comparison of MSEs of All Algorithms

In this section, MSEs of all algorithms are compared with each other and MSEs found by Box&Jenkins method. At first, model parameters are estimated for each order of 2,5,6 and 7 by using Box&Jenkins method. They aren't estimated for orders 12 and 13, since in the literature maximum model order of Box&Jenkins method is 9.

Box&Jenkins Method

Table 4.11 MSEs for Box&Jenkins Algorithm

Order	*p*=2	*p*=5	*p*=6	*p*=7*	*p*=12	*p*=13
MSE	21461	20886	20330	20107		

The best order is 7, and the best AR model is,

$$\hat{x}[n] = 0.70278x[n-1] + 0.21646x[n-2] - 0.03336x[n-3] - 0.08706x[n-4] + 0.05652x[n-5] + 0.24056x[n-6] - 0.10522x[n-7]$$

MSEs of all algorithms and Box&Jenkins method are shown in Table 4.12. All algorithms except MIRLS algorithm for orders 12 and 13 yielded less MSEs than that of produced by Box&Jenkins method.

Table 4.12 MSEs of All Algorithms and Box&Jenkins Method

Alg. *Order*	*p*=2	*p*=5	*p*=6	*p*=7	*p*=12	*p*=13
Autocor. M.	17526	17056	16714	16603	15624	15609
Cov. M.	17084	15780	15138	15252	14298	14393
Pre. Cov. M.	17623	17572	17166	17108	16916	17095
Mod. Cov. M.	17465	16894	16354	16115	15129	15115
Burg's Alg.	17563	17698	17397	17193	17011	17164
MIRLS Alg.	17623	17717	17686	17824	22075	22307
PLS Alg.	18070	18304	17629	17494	17664	17811
Box&Jenkins	21461	20886	20330	20107		

The best model parameters are estimated by using covariance method for order 12. The worst model parameters are estimated by using MIRLS algorithm for order 13. Predicted and real data for these algorithms are given in Figures 4.9 and 4.10

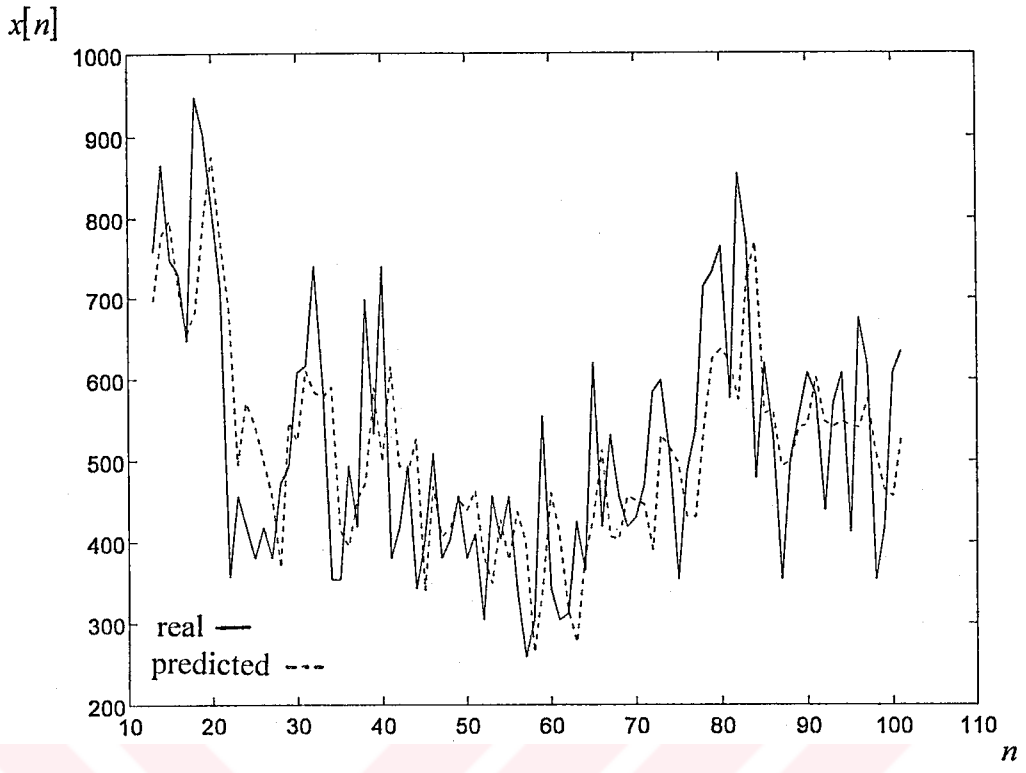

Figure 4.9 The Best Model (Covariance Method p=12)

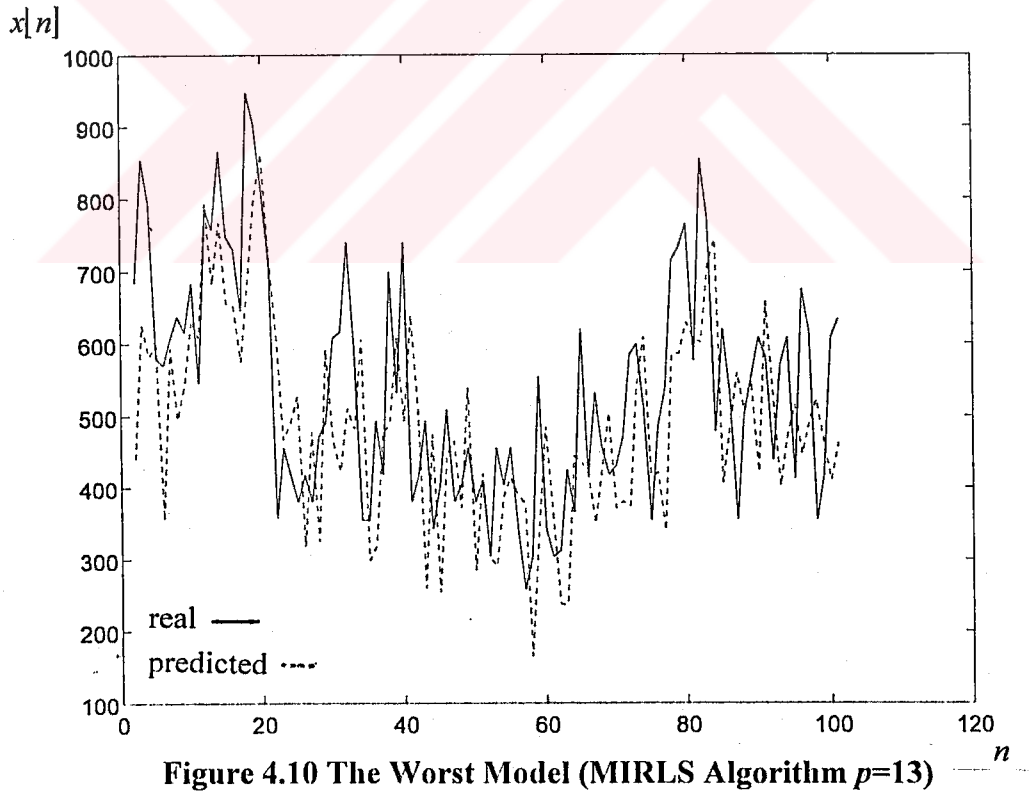

Figure 4.10 The Worst Model (MIRLS Algorithm p=13)

Also, future predictions of the best model are shown in Figure 4.11 and 4.12 using first and recursive approaches.

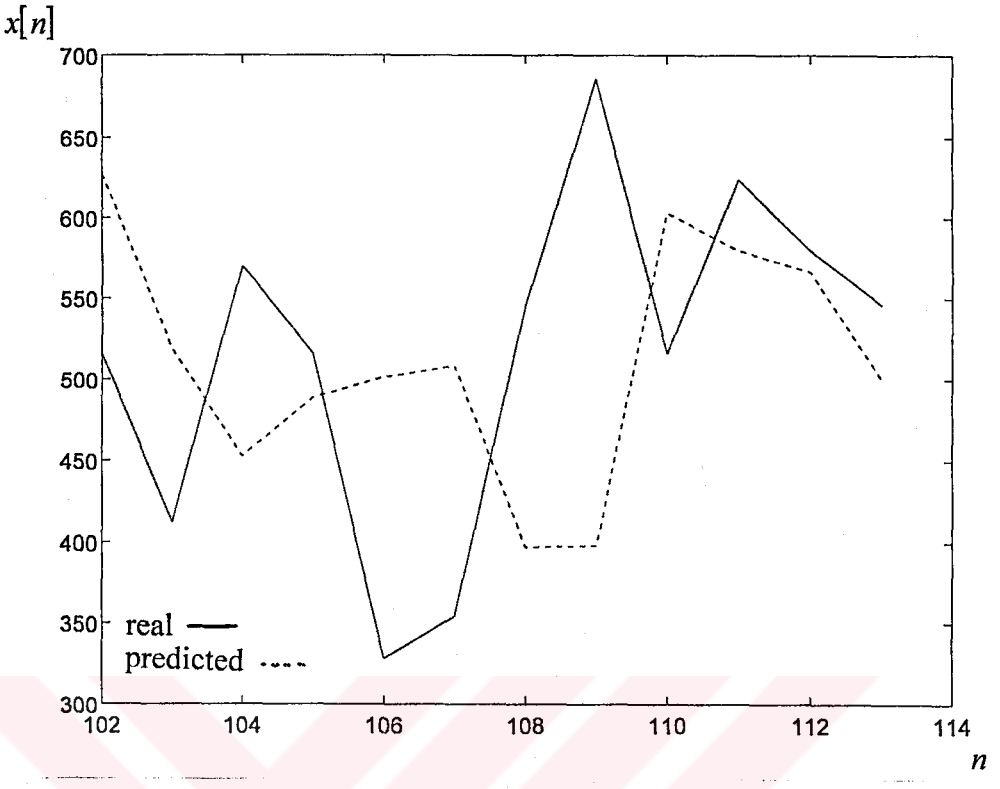

Figure 4.11 Future Predictions of the Best Model (First Approach)

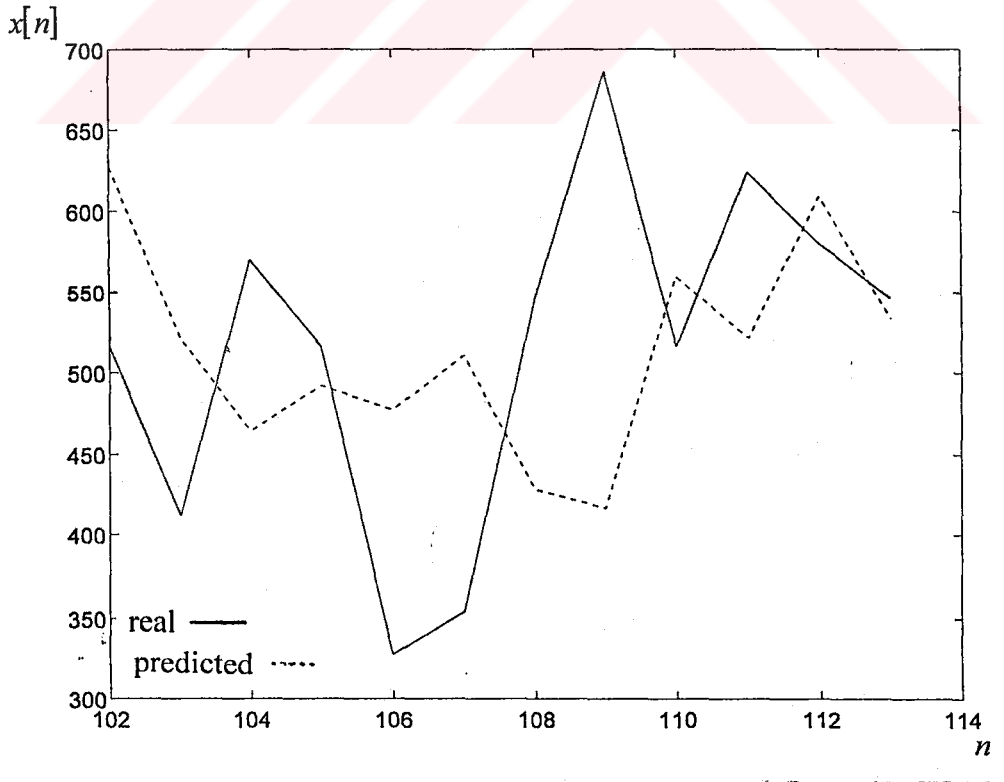

Figure 4.12 Future Predictions of the Best Model (Recursive Approach)

CHAPTER FIVE
CONCLUSION

Statistical signal processing algorithms are used for representation of voice, image, music, data, noise, target motion signals in electrical engineering. In this thesis, seven signal processing algorithms were used to estimate the amount of sodium oleate in the neutralization process of a crude oil produced by an oil company.

In the first step of this study, a general class of ARMA models, AR model, was identified. In the second step, six optimal model orders for data were determined by using six order selection criteria, the seven signal processing algorithms were used for model building purposes. Model parameters were estimated for each algorithm and for each optimal order. As a result, 42 alternatives of AR model were constructed. In the third step, mean square errors (MSE) of alternatives were computed using estimated parameters. Since the model of which parameters were estimated using covariance method for order 12 has minimum MSE, it was determined as the best model. In the fourth step, forecasts were generated using the best and the worst models, and their forecasts were compared with each other.

After these steps, all MSEs found by statistical signal processing algorithms were compared with MSEs found by Box&Jenkins method. Each of the 40 alternatives is better than one produced by Box&Jenkins method. As a result, each of the algorithms used in this study to forecast is superior to Box&Jenkins method.

REFERENCES

Box, G. E. P.,&Jenkins, G. M. (1976). Time series analysis: forecasting and control.(Revised ed.) San Fransisco: Holden-Day Inc.

Dilworth, J. B. (1993). Production and operations management manufacturing and services. (International ed.). McGraw-Hill Inc.

Heizer, J.,&Render, B. (1993). Production and operations management strategies and tactics. Simon&Schuster Inc.

Makridakis, S.,&Wheelwright, S. C. (1989). Forecasting methods for management. (5th ed.). John Willey&Sons Inc.

Montgomery, D. C.,&Lynwood, A. J. (1976). Forecasting and time series analysis. McGraw-Hill Inc.

Proakis, J. G.,&Manolakis, D. G. (1996). Digital signal processing principles, algorithms, and applications. (3rd ed.). New Jersey: Prentice-Hall Inc.

Sandy, R. (1990). Statistics for business and economics. McGraw-Hill Inc.

Therrien, C. W. (1992). Discrete random signals and statistical signal processing. New Jersey: Prentice-Hall Inc.

Bayhan, G. M., Bayhan, M., Badem, N. (1995). Modeling of field returns in industrial quality service by signal processing. Proceedings of 7th Eueropean Simulation Symposium, 334-338

Armour, B.,&Morgera, S. D. (1991). An exact forward-backward maximum likelihood autoregressive parameter estimation method. IEEE Transactions on Signal Processing, 39, 1985-1993.

Broersen, P. M. T. (1985). Selecting the order of autoregressive models from small samples. IEEE Transactions on Acoustics, Speech, and Signal Processing, 33, 874-879.

Chakraborty, M.,&Prasad, S. (1994). Computation of a useful cramer-rao bound for multichannel ARMA parameter estimation. IEEE Transactions on Signal Processing, 42, 466-469.

Chen, B. S., Chen, J. M., Shern, S. C. (1994). An ARMA robust system identification using a generalzed l_p norm estimation algorithm. IEEE Transactions on Signal Processing, 42, 1063-1072.

Chi, C. Y., Chang, W. J., Feng, C. C. (1994). A new algorithm for design of linear prediction error filters using cumulant-based MSE criteria. IEEE Transactions on Signal Processing, 42, 2876-2880

Choi, B. S. (1993). Two chi-square statistics for determining the orders p and q of an ARMA(p,q) process. IEEE Transactions on Signal Processing, 41, 2165-2176.

DeGroat, R. D.,&Dowling E. M. (1993). The data least sqaures problem and channel equalization. IEEE Transactions on Signal Processing, 41, 407-411.

Feder M. (1993). Parameter estimation and extraction of helicopter signals observed with a wide-band interference. IEEE Transactions on Signal Processing, 41, 232-243

Giannakis, G. B.,&Zhou, G. (1994). Parameter estimation of cyclostationary AM time series with application to missing observations. IEEE Transactions on Signal Processing, 42, 2408-2418.

Giannakis, G. B.,&Zhou, G. (1995). Harmonics in multiplicative and additive noise: parameter estimation using cyclic statistics. IEEE Transactions on Signal Processing, 43, 2217-2221.

Goto, S., Nakamura, M., Uoasaki, K. (1995). On-line spectral estimation of nonstationary time series based on AR model parameter estimation and order selection with a forgetting factor. IEEE Transactions on Signal Processing, 43, 1519-1522.

Katsikas,S. K., Likothanassis, S. D., Lainiotis, D. G. (1990). Ar model identification with unknown process order. IEEE Transactions on Signal Processing, 38, 872-875.

Kim, K.,&Shafai, B. (1995). Finite impulse response estimator (FIRE). IEEE Transactions on Signal Processing, 43, 2186-2189.

Konstantinides, K. (1991). Threshold bounds in SVD and a new iterative algorithm for order selection in AR models. IEEE Transactions on Signal Processing, 39, 1218-1221.

Li, S.,&Dickinson, B. W. (1988). Application of the lattice filter to robust estimation of an AR and ARMA models. IEEE Transactions on Signal Processing, 36, 502-512.

Liang, G., Wilkes, D. M., Cadzow, J. A. (1993). ARMA model order estimation based on the eigenvalues of the covariance matrix. IEEE Transactions on Signal Processing, 41, 3003-3009.

Liu, Z. S. (1995). QR methods of $O(N)$ complexity in adaptive parameter estimation. IEEE Transactions on Signal Processing, 43, 720-728.

Nakano, J.,&Tagami, S. (1987). Order selection of a mutivariate autoregressive model by a modification of the FPE criterion. International Journal of Control, 45, 589-596.

Pham, D. T.,&Tong, D. Q. (1994). Maximum likelihood estimation for a multivariate autoregressive model. IEEE Transactions on Signal Processing, 42, 3061-3072.

Rao, A. K., Huang, Y. F., Dasgupta, S. (1990). ARMA parameter estimation using a novel recursive estimation algorithm with selective updating. IEEE Transactions on Signal Processing, 38, 447-457.

Rasmussen, K. B. (1994). Maximum likelihood estimation of the parameters of nonminimum phase and noncausal ARMA models. IEEE Transactions on Signal Processing, 42, 209-211.

Ray, G. C. (1994). An algorithm to separate nonstationary part of a signal using mid-prediction filter. IEEE Transactions on Signal Processing, 42, 2276-2279.

Reid, R. A. (1992). The forecasting process: guidelines for the industrial engineer. Industrial Management, 22-24.

Steedly, W. M., Ying, C. H. J., Moses, R. L. (1994). A modified TLS-prony method using data decimation. IEEE Transactions on Signal Processing, 42, 2292-2302.

Swami A.,&Mendel, J. M. (1990). ARMA parameter estimation using only output cumulants. IEEE Transactions on Signal Processing, 38, 1257-1264.

Swingler, D. N. (1995). On the application of the jacknife to the estimation of the parameters of short multisinusoidal signals using data-matrix formulation. IEEE Transactions on Signal Processing, 43, 2135-2142.

Vis, M. L.,&Scharf, L. L. (1994). A note on recursive maximum likelihood for autoregressive modeling. IEEE Transactions on Signal Processing, 42, 2881-2883.

Wax, M. (1988). Order selection for the AR models by predictive least squares. IEEE Transactions on Signal Processing, 36, 581-588.

MATLAB for Windows. 1984-1993 The MathWorks Inc. Version 4.0

STATGRAPHICS Version 6.1. Copyright 1985-1992 by Statistical Graphics Corporation Portion Copyright 1992 Manugistic, Inc.

APPENDICES

Appendix 1. Singular Value Decomposition

If X is data matrix and its dimension is K x P then X could be written as

$$X = U\Sigma V^T$$

where U is the K x K unitary matrix of left singular vectors

$$U = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_K \\ | & | & & | \end{bmatrix}$$

and V is the P x P matrix of right singular vectors

$$V = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_P \\ | & | & & | \end{bmatrix}$$

The matrix $\sum$ is the KxP matrix of nonnegative real singular values,

$$\Sigma = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix}$$

where S_1 is a diagonal matrix of non zero singular values

$$S_1 = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}$$

and r is the rank of the X.

$X = U\Sigma V^T$ could be written in the alternative form

$$X = \sum_{k=1}^{r} \sigma_k u_k v_k^T$$

Appendix 2. Levinson Recursion

$$\tilde{R}_x = E\{\tilde{x}\tilde{x}^T\}$$

where

$$\tilde{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-P] \end{bmatrix}$$

The forward normal equations are written

$$\tilde{R}_x^{(p)} a_p = \begin{bmatrix} \sigma_p^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{1}$$

where

$$\tilde{R}_x^{(p)} = \begin{bmatrix} R_x[0] & R_x[1] & \cdots & R_x[p] \\ R_x[-1] & R_x[0] & \cdots & R_x[p-1] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[-p] & R_x[-p+1] & \cdots & R_x[0] \end{bmatrix}$$

and

$$a_p = \begin{bmatrix} 1 \\ a_1^{(p)} \\ \vdots \\ a_p^{(p)} \end{bmatrix}$$

The backward normal equations have the form

$$R_x^{(p)} b_p = \begin{bmatrix} \sigma'^2_p \\ b_1^{(p)} \\ \vdots \\ b_p^{(p)} \end{bmatrix}$$

where

$$R_x^{(p)} = \begin{bmatrix} R_x[0] & R_x[-1] & \cdots & R_x[-p] \\ R_x[1] & R_x[0] & \cdots & R_x[1-p] \\ \vdots & \vdots & \vdots & \vdots \\ R_x[p] & R_x[p-1] & \cdots & R_x[0] \end{bmatrix}$$

and

$$b_p = \begin{bmatrix} 1 \\ b_1^{(p)} \\ \vdots \\ b_p^{(p)} \end{bmatrix}$$

r_p is defined as

$$r_p = \begin{bmatrix} R_x[1] \\ R_x[2] \\ \vdots \\ R_x[p+1] \end{bmatrix}$$

Then the reversed correlation matrix of order p can be partitioned as

$$\tilde{R}_x^{(p)} = \left[\begin{array}{ccc|c} & \tilde{R}_x^{(p-1)} & & \tilde{r}_{p-1} \\ & & & \\ \hline & & & \\ \tilde{r}_{p-1}^T & & & R_x[0] \end{array}\right]$$

then

$$R_x^{(p)} = \left[\begin{array}{ccc|c} & R_x^{(p-1)} & & \tilde{r}_{p-1} \\ & & & \\ \hline & & & \\ \tilde{r}_{p-1}^T & & & R_x[0] \end{array}\right]$$

The forward normal equations have the form

$$\tilde{R}_x^{(p)} \begin{bmatrix} a_{p-1} \\ -- \\ 0 \end{bmatrix} = \left[\begin{array}{ccc|c} & \tilde{R}_x^{(p-1)} & & \tilde{r}_{p-1} \\ & & & \\ \hline & & & \\ \tilde{r}_{p-1}^T & & & R_x[0] \end{array}\right] \begin{bmatrix} a_{p-1} \\ -- \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_{p-1}^2 \\ 0 \\ \vdots \\ \Delta_p \end{bmatrix} \tag{2}$$

where

$$\Delta_p = \tilde{r}_{p-1}^T a_{p-1} = r_{p-1}^T \tilde{a}_{p-1}$$

Also backward normal equations

$$R_x^{(p)} \begin{bmatrix} b_{p-1} \\ -- \\ 0 \end{bmatrix} = \left[\begin{array}{ccc|c} & R_x^{(p-1)} & & \tilde{r}_{p-1} \\ & & & \\ \hline & & & \\ \tilde{r}_{p-1}^T & & & R_x[0] \end{array}\right] \begin{bmatrix} b_{p-1} \\ -- \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma'^2_{p-1} \\ 0 \\ \vdots \\ \Delta'_p \end{bmatrix} \tag{3}$$

where

$$\Delta'_p = \tilde{r}_{p-1}^T b_{p-1} = r_{p-1}^T \tilde{b}_{p-1}$$

If (3) is reversed

$$\tilde{R}_x^p \begin{bmatrix} 0 \\ -- \\ \tilde{b}_{p-1} \end{bmatrix} = \begin{bmatrix} \Delta'_p \\ 0 \\ \vdots \\ \sigma'^2_{p-1} \end{bmatrix}$$

If this equation is multiplied by a constant c_1 and added to (2)

$$\tilde{R}_x^{(p)} = \left[\begin{bmatrix} a_{p-1} \\ -- \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ -- \\ \tilde{b}_{p-1} \end{bmatrix} \right] = \begin{bmatrix} \sigma^2_{p-1} \\ 0 \\ \vdots \\ \Delta_p \end{bmatrix} + c_1 \begin{bmatrix} \Delta'_p \\ 0 \\ \vdots \\ \sigma'^2_{p-1} \end{bmatrix}$$

If this result is compared to (1)

$$\begin{bmatrix} \sigma^2_{p-1} \\ 0 \\ \vdots \\ \Delta_p \end{bmatrix} + c_1 \begin{bmatrix} \Delta'_p \\ 0 \\ \vdots \\ \sigma'^2_{p-1} \end{bmatrix} = \begin{bmatrix} \sigma^2_p \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4}$$

then

$$\begin{bmatrix} a_{p-1} \\ -- \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ -- \\ b_{p-1} \end{bmatrix} = a_p$$

from (4)

$$\sigma^2_{p-1} + c_1 \Delta'_p = \sigma^2_p$$

and

$$\Delta_p + c_1 \sigma'^2_{p-1} = 0$$

Now this procedure is repeated for the backward equations. If (2) is reversed

$$R_x^p \begin{bmatrix} 0 \\ -- \\ \tilde{a}_{p-1} \end{bmatrix} = \begin{bmatrix} \Delta_p \\ 0 \\ \vdots \\ \sigma^2_{p-1} \end{bmatrix}$$

If this equation is multiplied by a constant c_2 and added to (3)

$$R_x^{(p)} = \left[\begin{bmatrix} b_{p-1} \\ -- \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ -- \\ \tilde{a}_{p-1} \end{bmatrix} \right] = \begin{bmatrix} \sigma'^2_{p-1} \\ 0 \\ \vdots \\ \Delta'_p \end{bmatrix} + c_2 \begin{bmatrix} \Delta_p \\ 0 \\ \vdots \\ \sigma^2_{p-1} \end{bmatrix}$$

then

$$\sigma'^2_{p-1} + c_2 \Delta_p = \sigma'^2_p$$

and

$$\Delta'_p + c_2 \sigma^2_{p-1} = 0$$

and

$$c_1 = -\Delta_p / \sigma'^2_{p-1}$$
$$c_2 = -\Delta'_p / \sigma^2_{p-1}$$

if $\gamma_p = -c_1$, $\gamma'_p = -c_2$ $b_{p-1} = a_{p-1}$, $\sigma'^2_{p-1} = \sigma^2_{p-1}$

$$\Delta'_p = r^T_{p-1} \tilde{b}_{p-1} = r^T_{p-1} \tilde{a}_{p-1} = \Delta_p$$
$$\gamma'_p = \frac{\Delta'_p}{\sigma^2_{p-1}} = \frac{\Delta_p}{\sigma'^2_{p-1}} = \gamma_p$$

By using these additional relations backward and forward equations can be written as single-thread recursion,

$$\gamma_p = \frac{r_{p-1}^T \tilde{a}_{p-1}}{\sigma_{p-1}^2}$$

$$a_p = \begin{bmatrix} a_{p-1} \\ -- \\ 0 \end{bmatrix} - \gamma_p \begin{bmatrix} 0 \\ -- \\ \tilde{a}_{p-1} \end{bmatrix}$$

$$\sigma_p^2 = \left(1 - |\gamma_p|^2\right)\sigma_{p-1}^2$$

Initial conditions are,

$$a_0 = [1] \;\; ; \quad r_0 = R_x[1] \;\; ; \quad \sigma_0^2 = R_x[0]$$

Appendix 3. Lattice Representation

$$x_p[n] = \begin{bmatrix} x[n-p] \\ x[n-p+1] \\ \\ x[n-1] \\ x[n] \end{bmatrix}$$

The prediction error is

$$\varepsilon_p[n] = \sum_{i=0}^{p} a_i^{(p)} x[n-i] = a_p^T \tilde{x}_p[n] = \tilde{a}_p^T x_p[n]$$

backward prediction error is

$$\varepsilon_p^b[n] = \sum_{i=0}^{p} b_i^{(p)} x[n-p+i] = b_p^T x_p[n] = \tilde{b}_p^T \tilde{x}_p[n]$$

The vector x_p can be partitioned in two ways,

$$x_p[n] = \begin{bmatrix} x[n-p] \\ ---- \\ \\ x_{p-1}[n] \end{bmatrix} = \begin{bmatrix} x_{p-1}[n-1] \\ \\ ---- \\ x[n] \end{bmatrix} \qquad (1)$$

where

$$x_{p-1}[n] = \begin{bmatrix} x[n-p+1] \\ \vdots \\ x[n-1] \\ x[n] \end{bmatrix}$$

The (1) is reversed

$$\tilde{x}_p[n] = \begin{bmatrix} \tilde{x}_{p-1}[n] \\ \\ ---- \\ x[n-p] \end{bmatrix} = \begin{bmatrix} x[n] \\ ---- \\ \\ \tilde{x}_{p-1}[n-1] \end{bmatrix}$$

A recursion for the forward and backward prediction errors can be developed. The recursion for the forward filter parameter,

$$\varepsilon_p[n] = \varepsilon_{p-1}[n] - \gamma_p \varepsilon^b_{p-1}[n-1]$$

The recursion of the backward filter,

$$\varepsilon^b_p[n] = \varepsilon^b_{p-1}[n-1] - \gamma'_p \varepsilon_{p-1}[n]$$

These last two equations can be combined to write,

$$\boxed{\begin{bmatrix} \varepsilon_p[n] \\ \varepsilon^b_p[n] \end{bmatrix} = \begin{bmatrix} 1 & -\gamma_p \\ -\gamma'_p & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_{p-1}[n] \\ \varepsilon^b_{p-1}[n-1] \end{bmatrix}} \quad (2)$$

Appendix 4. MATLAB Computer Programs

4.1. AIC, CAT, FPE, MDL

```
clear
disp('algorithms (AIC,FPE,MDL,CAT) for determining order')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634]
z=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
% model is obtained by using first nF observation
% nI is first observation, M is maximum order
nF=101;nI=1;;M=15;sum=0;Ns=nF
  for p=1:1:M
   % model is obtained for determining the error variance(E)
   r=z(1:p+1)
   x=toeplitz(c,r)
   m=x'*x
   i=inv(m)
   y=i(1:1,1:1)
   S(p)=inv(y)
   E(p)=S(p)/(nF-nI+1)
   % algorithms are started
   AIC(p)=Ns*log(E(p))+2*p
   sum=sum+(Ns-p)/(Ns*E(p))
   CAT(p)=sum/Ns-(Ns-p)/(Ns*E(p))
   FPE(p)=E(p)*((Ns+p+1)/(Ns-p-1))
   MDL(p)=Ns*log(E(p))+p*log(Ns)
  end
% plotting results
plot(E);xlabel('order');ylabel('E(p)')
pause
plot(AIC);xlabel('order');ylabel('AIC(p)')
pause
plot(CAT);xlabel('order');ylabel('CAT(p)')
pause
plot(FPE);xlabel('order');ylabel('FPE(p)')
pause
plot(MDL);xlabel('order');ylabel('MDL(p)')
pause
close
```

4.2 Effective Rank

```
clear
disp('effective rank method for determining order')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
sumg=0
x=toeplitz(c,r)
[u,s,v]=svd(x)
for p=1:1:15
g(p)=s(p:p,p:p)
sumg=sumg+g(p)
end
sumge=0;k=0
  for p=15:-1:1
   % computing values
   R(p)=sqrt((sumg-sumge)/sumg)
   sumge=sumge+g(p)
  end
```

4.3 PLS

```
clear
disp('predictive least squares method for detemining order
and model parameter')
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
T=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nI=1;nF=101;Ns=113;M=15;v=2;y=[0,c]
% Initial values of Re,Rr,e,r,gamma
Re(0+v,0-1+v)=0;Rr(0+v,0-1+v)=0
  for t=0:nF-1
   e(0+v,t+v)=y(t+v)
   r(0+v,t+v)=y(t+v)
   Re(0+v,t+v)=Re(0+v,t-1+v)+y(t+v)^2
   Rr(0+v,t+v)=Re(0+v,t-1+v)+y(t+v)^2
   gamma(-1+v,t+v)=1
  end
  % PLS iterations
  for k=0:M-1
   n0(k+1+v)=0
    for t=0:nF-1
      if e(k+v,t+v)==0&r(k+v,t-1+v)==0|gamma(k-1+v,t-1+v)==0
       teta(k+1+v,-1+v)=0
       teta(k+1+v,t+v)=teta(k+1+v,t-1+v)
      else
       teta(k+1+v,t+v)=teta(k+1+v,t-1+v)+e(k+v,t+v)*r(k+v,t-
1+v)/gamma(k-1+v,t-1+v)
      end
       if Rr(k+v,t+v)==0
       gamma(k+v,t+v)=gamma(k-1+v,t+v)
      else
       gamma(k+v,t+v)=gamma(k-1+v,t+v)-
r(k+v,t+v)^2/Rr(k+v,t+v)
      end
      if teta(k+1+v,t+v)==0&r(k+v,t-1+v)==0|Rr(k+v,t-1+v)==0
       e(k+1+v,t+v)=e(k+v,t+v)
      else
       e(k+1+v,t+v)=e(k+v,t+v)-teta(k+1+v,t+v)*r(k+v,t-
1+v)/Rr(k+v,t-1+v)
      end
      if Rr(k+v,t-1+v)==0
       Re(k+1+v,t+v)=Re(k+v,t+v)
```

```
       else
        Re(k+1+v,t+v)=Re(k+v,t+v)-teta(k+1+v,t+v)^2/Rr(k+v,t-
1+v)
       end
       if teta(k+1+v,t+v)==0&e(k+v,t+v)==0|Re(k+v,t+v)==0
        r(k+1+v,t+v)=r(k+v,t-1+v)
       else
        r(k+1+v,t+v)=r(k+v,t-1+v)-
teta(k+1+v,t+v)*e(k+v,t+v)/Re(k+v,t+v)
       end
       if Re(k+v,t+v)==0
        Rr(k+1+v,t+v)=Rr(k+v,t-1+v)
       else
        Rr(k+1+v,t+v)=Rr(k+v,t-1+v)-
teta(k+1+v,t+v)^2/Re(k+v,t+v)
       end
       if gamma(k+v,t-1+v)==0
        et(k+1+v,t+v)=0
        n0(k+1+v)=n0(k+1+v)+1
       else
        et(k+1+v,t+v)=e(k+1+v,t+v)/gamma(k+v,t-1+v)
       end
       %
       if t=nF
         wt=et(k+v:k+v,:)
         SA(k+1,t+1)=wt*wt'
         SA1=SA(k+1:k+1,t+1:t+1)
         n1=n0(k+1+v)
         PLS(k+1,t+1)=SA1/(t+1-n1)
       end
     end
  end
% plotting result
plot(PLS)
pause
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     x=T(nI:nF,1:p)
     y1=c(nI+1:nF+1);y1=y1'
     e1=et(p+v:p+v,nI+v:nF+v);e1=e1'
     i=inv(x'*x)
     % a is a vector of the model parameters
     a=(i*x')*(y1-e1)
     % model is being used for forecasting
     l=T(nI:Ns-1,1:p)
     xt=l*a
     de=xt'
     % forecasting is done for observations that are used in
obtaining model
     % f is real observation, d is predicted observation and
h is error
     % SS is sum of the squared errors for these
```

```
      d=de(1:nF-nI)
      f=c(nI+1:nF)
      h=f'-d'
      SS=h'*h;MSE=SS/(nF-(p+1)+1)
      % forecasting is done from nF+1 to Ns
      % ft is real, dt is predicted, th is error, sumth is sum
 of the squared error
      dt=de(nF+1-nI:Ns-nI)
      ft=c(nF+1:Ns)
      th=ft'-dt'
      sumth=th'*th
      z=nI+1:nF
      ze=nF+1:Ns
      B=[z',f',d',h]
      A=[ze',ft',dt',th]
      % plotting and displaying results
      plot(z,f,'-',z,d,':')
      pause
      plot(z,h)
      pause
      close
      disp('    z        f         d         t')
      disp(B)
      disp('SS');disp(SS)
      disp('MSE');disp(MSE)
      disp('a');disp(a)
      pause
      plot(ze,ft,'-',ze,dt,':')
      pause
      plot(ze,th)
      pause
      close
      disp('  ze         ft        dt        th')
      disp(A)
      disp('sumth');disp(sumth)
      pause
    else
    end
  end
```

4.4 Autocorrelation Method

```
clear
disp('autocorrelation method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
v=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
c1=c(1:nF)
c2=[c1';v']
t=toeplitz(c2',r)
tt=toeplitz(c,r)
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     x=t(nI:nF+p,1:p+1)
     m=x'*x
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     E(p)=S(p)/(nF-nI+1)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=tt(nI:Ns-1,1:p)
     xt=-l*a
     de=xt'
     % forecasting is done for observations that are used in
obtaining model
     % f is real observation, d is predicted observation and
h is error
     % SS is sum of the squared errors for these
     d=de(nI:nF-nI)
     f=c(nI+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(nI+1)+1)
     % forecasting is done from nF+1 to Ns
```

```
    % ft is real, dt is predicted, th is error, sumth is sum
of the squared error
    dt=de(nF+1-nI:Ns-nI)
    ft=c(nF+1:Ns)
    th=ft'-dt'
    sumth=th'*th
    z=nI+1:nF
    B=[z',f',d',h]
    ze=nF+1:Ns
    A=[ze',ft',dt',th]
    % plotting and displaying results
    plot(z,f,'-',z,d,':')
    pause
    plot(z,h)
    pause
    close
    disp('    z        f          d          t')
    disp(B)
    disp('SS');disp(SS)
    disp('MSE');disp(MSE)
    disp('a');disp(a)
    pause
    plot(ze,ft,'-',ze,dt,':')
    pause
    plot(ze,th)
    pause
    close
    disp('  ze         ft         dt         th')
    disp(A)
    disp('sumth');disp(sumth)
    pause
   else
   end
 end
```

4.5 Recursive Autocorrelation Method

```
clear
disp('recursive autocorrelation method')
pause
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
v=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     c1=c(1:nF)
     c2=[c1';v']
     t=toeplitz(c2',r)
     x=t(nI:nF+p,1:p+1)
     m=x'*x
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     E(p)=S(p)/(nF-nI+1)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=t(nI:nF,1:p)
     xt=-l*a
     de=xt'
     % f is real, d is predicted, h is error, SS is sm of the
squared errors
     d=de(1:nF-nI)
     f=c(nI+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(nI+1)+1)
     z=nI+1:nF
     B=[z',f',d',h]
     ze=nF+1:Ns
     % plotting and displaying results
     plot(z,f,'-',z,d,':')
     pause
```

```
plot(z,h)
pause
close
disp('    z       f         d         h')
disp(B)
pause
disp('SS');disp(SS)
disp('MSE');disp(MSE)
disp('a');disp(a)
pause
% recursive forecasting from nF+1 to Ns
sumth=0
n2=Ns-nF
  for n=1:n2
   nF=101
   nF=nF+n-1
   % model is updated
   c1=c(1:nF)
   c2=[c1';v']
   t=toeplitz(c2',r)
   x=t(nI:nF+p,1:p+1)
   m=x'*x
   i=inv(m)
   y=i(1:1,1:1)
   S(p)=inv(y)
   k(1)=S(p)
     for j=2:p+1,k(j)=0, end
   b=i*k'
   a=b(2:p+1)
   l=t(nI:nF,1:p)
   xt=-l*a
   % ft is real, dt is predicted, th is error, sumth is
sum of the squared error
   dt(n)=xt(nF+1-nI:nF+1-nI)
   ft(n)=c(nF+1:nF+1)
   th(n)=ft(n)-dt(n)
   thn=th(n:n)
   sumth=sumth+thn^2
   disp('a');disp(a)
   ap=[ap;a']
   pause
  end
nF=101
A=[ze',ft',dt',th']
% plotting and displaying results
plot(ze,ft,'-',ze,dt,':')
pause
plot(ze,th)
pause
close
disp('  ze        ft        dt        th')
disp(A)
disp('sumth');disp(sumth)
disp('ap');disp(ap)
```

```
    pause
  else
  ap=[]
  end
end
```

4.6 Covariance Method

```
clear
disp('covariance method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     x=t(p+nI:nF,1:p+1)
     m=x'*x
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=t(p:Ns-1,1:p)
     xt=-l*a
     de=xt'
     % forecasting is done for observations that are used in
obtaining model
     % f is real observation, d is predicted observation and
h is error
     % SS is sum of the squared errors for these
     d=de(1:nF-p)
     f=c(p+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(p+1)+1)
     % forecasting is done from nF+1 to Ns
     % ft is real, dt is predicted, th is error, sumth is sum
of the squared error
     dt=de(nF+1-p:Ns-p)
     ft=c(nF+1:Ns)
     th=ft'-dt'
     sumth=th'*th
```

```
    z=p+1:nF
    ze=nF+1:Ns
    B=[z',f',d',h]
    A=[ze',ft',dt',th]
    % plotting and displaying results
    plot(z,f,'-',z,d,':')
    pause
    plot(z,h)
    pause
    close
    disp('    z        f          d          t')
    disp(B)
    disp('SS');disp(SS)
    disp('MSE');disp(MSE)
    disp('a');disp(a)
    pause
    plot(ze,ft,'-',ze,dt,':')
    pause
    plot(ze,th)
    pause
    close
    disp('  ze         ft         dt         th')
    disp(A)
    disp('sumth');disp(sumth)
    pause
  else
  end
end
```

4.7 Recursive Covariance Method

```
clear
disp('recursive covariance method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model isbeing obtained
     x=t(p+nI:nF,1:p+1)
     m=x'*x
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=t(p:nF,1:p)
     xt=-l*a
     de=xt'
     % f is real, d is predicted, h is error, SS is sm of the
squared errors
     d=de(1:nF-p)
     f=c(p+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(p+1)+1)
     z=p+1:nF
     B=[z',f',d',h]
     % plotting and displaying results
     plot(z,f,'-',z,d,':')
     pause
     plot(z,h)
     pause
     close
     disp('    z       f         d         t')
     disp(B)
```

```
      pause
      disp('SS');disp(SS)
      disp('MSE');disp(MSE)
      disp('a');disp(a)
      pause
      % recursive forecasting from nF+1 to Ns
      sumth=0
      n2=Ns-nF
        for n=1:1:n2
         nF=101
         nF=nF+n-1
         % model is updated
         x=t(p+nI:nF,1:p+1)
         m=x'*x
         i=inv(m)
         y=i(1:1,1:1)
         S(p)=inv(y)
         k(1)=S(p)
           for j=2:p+1,k(j)=0,end
         b=i*k'
         a=b(2:p+1)
         l=t(p:nF,1:p)
         xt=-l*a
         % ft is real, dt is predicted, th is error, sumth is
sum of the squared error
         dt(n)=xt(nF+1-p:nF+1-p)
         ft(n)=c(nF+1:nF+1)
         th(n)=ft(n)-dt(n)
         thn=th(n:n)
         sumth=sumth+thn^2
         disp('a');disp(a)
         ap=[ap;a']
         pause
        end
      nF=101
      ze=nF+1:Ns
      A=[ze',ft',dt',th']
      % plotting and displaying results
      plot(ze,ft,'-',ze,dt,':')
      pause
      plot(ze,th)
      pause
      close
      disp('  ze         ft         dt         th')
      disp(A)
      disp('sumth');disp(sumth)
      disp('ap');disp(ap')
      pause
     else
     ap=[]
     end
   end
```

4.8 Prewindowed Covariance Method

```
clear
disp('prewindowed covariance method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     x=t(nI:nF,1:p+1)
     m=x'*x
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=t(nI:Ns-1,1:p)
     xt=-l*a
     de=xt'
     % forecasting is done for observations that are used in
obtaining model
     % f is real observation, d is predicted observation and
h is error
     % SS is sum of the squared errors for these
     d=de(1:nF-nI)
     f=c(nI+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(p+1)+1)
     % forecasting is done from nF+1 to Ns
     % ft is real, dt is predicted, th is error, sumth is sum
of the squared error
     dt=de(nF+1-nI:Ns-nI)
     ft=c(nF+1:Ns)
     th=ft'-dt'
     sumth=th'*th
```

```
    z=nI+1:nF
    ze=nF+1:Ns
    B=[z',f',d',h]
    A=[ze',ft',dt',th]
    % plotting and displaying results
    plot(z,f,'-',z,d,':')
    pause
    plot(z,h)
    pause
    close
    disp('    z        f          d          t')
    disp(B)
    disp('SS');disp(SS)
    disp('MSE');disp(MSE)
    disp('a');disp(a)
    pause
    plot(ze,ft,'-',ze,dt,':')
    pause
    plot(ze,th)
    pause
    close
    disp('  ze         ft         dt         th')
    disp(A)
    disp('sumth');disp(sumth)
    pause
  else
  end
end
```

4.9 Recursive Prewindowed Covariancce Method

```
clear
disp('recursive prewindowed covariance method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
  for p=1:1:15
  % model is obtained for only determined orders
   if  p==2|p==5|p==6|p==7|p==12|p==13
   % model is being obtained
    x=t(nI:nF,1:p+1)
    m=x'*x
    i=inv(m)
    y=i(1:1,1:1)
    S(p)=inv(y)
    E(p)=S(p)/(nF-nI+1)
    k(1)=S(p)
      for j=2:p+1,k(j)=0,end
    b=i*k'
    % a is a vector of the model parameters
    a=b(2:p+1)
    % model is being used for forecasting
    l=t(nI:nF,1:p)
    xt=-l*a
    de=xt'
    % f is real, d is predicted, h is error, SS is sm of the
squared errors
    d=de(1:nF-nI)
    f=c(nI+1:nF)
    h=f'-d'
    SS=h'*h;MSE=SS/(nF-(nI+1)+1)
    z=nI+1:nF
    ze=nF+1:Ns
    B=[z',f',d',h]
    % plotting and displaying results
    plot(z,f,'-',z,d,':')
    pause
    plot(z,h)
    pause
    close
```

```
    disp('    z        f          d          h')
    disp(B)
    pause
    disp('SS');disp(SS)
    disp('MSE');disp(MSE)
    disp('a');disp(a)
    pause
    % recursive forecasting from nF+1 to Ns
    sumth=0
    n2=Ns-nF
      for n=1:n2
       nF=101
       nF=nF+n-1
       % model is updated
       x=t(nI:nF,1:p+1)
       m=x'*x
       i=inv(m)
       y=i(1:1,1:1)
       S(p)=inv(y)
       k(1)=S(p)
         for j=2:p+1,k(j)=0, end
       b=i*k'
       a=b(2:p+1)
       l=t(nI:nF,1:p)
       xt=-l*a
       % ft is real, dt is predicted, th is error, sumth is
sum of the squared error
       dt(n)=xt(nF+1-nI:nF+1-nI)
       ft(n)=c(nF+1:nF+1)
       th(n)=ft(n)-dt(n)
       thn=th(n:n)
       sumth=sumth+thn^2
       disp('a');disp(a)
       ap=[ap;a']
       pause
      end
    nF=101
    A=[ze',ft',dt',th']
    % plotting and displaying results
    plot(ze,ft,'-',ze,dt,':')
    pause
    plot(ze,th)
    pause
    close
    disp('  ze         ft         dt         th')
    disp(A)
    disp('sumth');disp(sumth)
    disp('ap');disp(ap')
    pause
   else
   ap=[]
   end
  end
```

4.10 Modified Covariance Method

```
clear
disp('modified covariance method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
  for p=1:1:M
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     t1=t(nI:nF,1:p+1)
     t2=t1(nF:-1:nI,p+1:-1:1)
     m=(t1'*t1)+(t2'*t2)
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=t(nI:Ns-1,1:p)
     xt=-l*a
     de=xt'
     % forecasting is done for observations that are used in
obtaining model
     % f is real observation, d is predicted observation and
h is error
     % SS is sum of the squared errors for these
     d=de(1:nF-nI)
     f=c(nI+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(nI+1)+1)
     % forecasting is done from nF+1 to Ns
     % ft is real, dt is predicted, th is error, sumth is sum
of the squared error
     dt=de(nF+1-nI:Ns-nI)
     ft=c(nF+1:Ns)
     th=ft'-dt'
```

```
    sumth=th'*th
    z=nI+1:nF
    ze=nF+1:Ns
    B=[z',f',d',h]
    A=[ze',ft',dt',th]
    % plotting and displaying results
    plot(z,f,'-',z,d,':')
    pause
    plot(z,h)
    pause
    close
    disp('    z        f          d          t')
    disp(B)
    disp('SS');disp(SS)
    disp('MSE');disp(MSE)
    disp('a');disp(a)
    pause
    plot(ze,ft,'-',ze,dt,':')
    pause
    plot(ze,th)
    pause
    close
    disp('  ze         ft         dt         th')
    disp(A)
    disp('sumth');disp(sumth)
    pause
  else
  end
end
```

4.11 Recursive Modified Covariance Method

```
clear
disp('recursive modified covariance method')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
  for p=1:1:15
    % model is obtained for only determined orders
    if p==2|p==5|p==6|p==7|p==12|p==13
     % model is being obtained
     t1=t(nI:nF,1:p+1)
     t2=t1(nF:-1:nI,p+1:-1:1)
     m=(t1'*t1)+(t2'*t2)
     i=inv(m)
     y=i(1:1,1:1)
     S(p)=inv(y)
     E(p)=S(p)/(nF-nI+1)
     k(1)=S(p)
       for j=2:p+1,k(j)=0,end
     b=i*k'
     % a is a vector of the model parameters
     a=b(2:p+1)
     % model is being used for forecasting
     l=t1(nI:nF,1:p)
     xt=-l*a
     de=xt'
     % f is real, d is predicted, h is error, SS is sm of the
squared errors
     d=de(1:nF-nI)
     f=c(nI+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(nI+1)+1)
     z=nI+1:nF
     ze=nF+1:Ns
     B=[z',f',d',h]
     % plotting and displaying results
     plot(z,f,'-',z,d,':')
     pause
     plot(z,h)
     pause
```

```
      close
      disp('    z        f          d          t')
      disp(B)
      disp('SS');disp(SS)
      disp('MSE');disp(MSE)
      disp('a');disp(a)
      pause
      % recursive forecasting from nF+1 to Ns
      sumth=0
      n2=Ns-nF
        for n=1:n2
         nF=101
         nF=nF+n-1
         % model is updated
         t1=t(nI:nF,1:p+1)
         t2=t1(nF:-1:nI,p+1:-1:1)
         m=(t1'*t1)+(t2'*t2)
         i=inv(m)
         y=i(1:1,1:1)
         S(p)=inv(y)
         k(1)=S(p)
           for j=2:p+1,k(j)=0,end
         b=i*k'
         a=b(2:p+1)
         l=t1(nI:nF,1:p)
         xt=-l*a
         % ft is real, dt is predicted, th is error, sumth is
sum of the squared error
         dt(n)=xt(nF+1-nI:nF+1-nI)
         ft(n)=c(nF+1:nF+1)
         th(n)=ft(n)-dt(n)
         thn=th(n:n)
         sumth=sumth+thn^2
         disp(a)
         ap=[ap;a']
         pause
        end
      nF=101
      A=[ze',ft',dt',th']
      % plotting and displaying results
      plot(ze,ft,'-',ze,dt,':')
      pause
      plot(ze,th)
      pause
      close
      disp('  ze        ft        dt        th')
      disp(A)
      disp('sumth');disp(sumth)
      disp('ap');disp(a')
      pause
    else
    ap=[]
    end
  end
```

4.12 Burg's Algorithm

```
clear
disp('BURGs algorithm')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
t=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number of observations, M is
maximum order
nF=101;nI=1;Ns=113;M=15
% Initial conditions of ef, eb, L, a
ef=c(2:nF)
eb=c(1:nF-1)
L=length(ef)
ef=ef';eb=eb'
a=[1]
  for p=1:M
   % algorithm iterations
   gamma(p)=(2*ef'*eb)/(ef'*ef+eb'*eb)
    if p<M+1
     tmp1=ef-conj(gamma(p))*eb
     tmp2=eb-gamma(p)*ef
     ef=tmp1(2:L)
     eb=tmp2(1:L-1)
     a=a'
     a=[a,0];a=a';ar=a(p+1:-1:1)
     a1=a-conj(gamma(p))*ar
     % a or model parameter is determined
     a=a1
     x=t(nI:Ns-1,1:p)
     a2=a(2:p+1)
     de=-x*a2;de=de'
     % forecasting is done for observations that are used in
obtaining model
     % f is real observation, d is predicted observation and
h is error
     % SS is sum of the squared errors for these
     d=de(nI:nF-nI)
     f=c(nI+1:nF)
     h=f'-d'
     SS=h'*h;MSE=SS/(nF-(nI+1)+1)
     dt=de(nF:Ns-nI)
     ft=c(nF+1:Ns)
     th=ft'-dt'
```

```
      sumth=th'*th
      z=nI+1:nF
      ze=nF+1:Ns
      Sp=[Sp;SS]
      B=[z',f',d',h]
      ST=[ST;sumth]
        if p==2|p==5|p==6|p==7|p==12|p==13
         % plotting and displaying results for determined
orders
         A=[ze',ft',dt',th]
     plot(z,f,'-',z,d,':')
     pause
     plot(z,h)
     pause
     close
     disp('    z        f          d          t')
     disp(B)
        disp('SS');disp(SS)
        disp('MSE');disp(MSE)
        disp('a');disp(a)
        pause
        plot(ze,ft,'-',ze,dt,':')
        pause
        plot(ze,th)
        pause
     close
     disp('  ze         ft         dt         th')
        disp(A)
     disp('sumth');disp(sumth)
     pause
       end
    end
   L=L-1
  end
pp=1:M
plot(pp,Sp)
pause
plot(pp,ST)
pause
close
```

4.13 MIRLS Algorithm

```
clear
disp('MIRLS algorithm')
pause
% c is all observations
c=[650 685 855 795 580 570 608 638 616 684 545 786 758 866
748 730 646 948 904 813 711 357 456 418 380 418 380 471 494
608 616 740 594 354 354 494 418 699 532 740 380 418 494 342
403 509 380 403 456 380 410 304 456 403 456 342 258 304 555
342 304 312 425 365 620 418 532 456 418 430 470 584 599 509
354 486 537 714 732 765 576 855 773 478 620 532 354 502 555
608 580 438 570 608 412 675 616 354 418 608 634 516 412 570
516 328 354 546 686 516 624 580 546]
r=[650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
T=toeplitz(c,r)
% model is obtained by using first nF observation
% nI is first observation, Ns number o fobservations, M is
maximum order
nF=101;nI=1;Ns=113;M=15;y=c(nI+1:nF);y=y'
for p=1:M
  % model is obtained for only determined orders
  if p==2|p==5|p==6|p==7|p==12|p==13
   % model is being obtained
   X=T(nI:nF-1,1:p)
    for i=1:p
     % Initial conditions
     aI(i,1)=1
    end
    % algorithm is started
    for j=1:50
     ab=aI(:,j)
     xt=X*ab
     e=y-xt
     e1=abs(e)
      for i=nI:nF-1
       w1=e1(i:i)
       w(i)=w1^(p-2)
      end
     W=diag(w)
     abI=inv((X'*W)*X)*(X'*W)*y
     ab1=((p-2)*ab+abI)/(p-1)
     aI(:,j+1)=ab1
     aIRLS(:,j+1)=abI
    end
   % convergence analys
   for n=1:p
    a(p,n)=0
     for j=1:50
      div=abs(aI(n,j+1)/aI(n,j))
      con=abs(div-1)
       if a(p,n)==0
         if con==0
          a(p,n)=aI(n,j)
```

```
        end
      end
    end
      if a(p,n)==0
       a(p,n)=aI(n,j+1)
      end
   end
   % model is being used for forecasting
   l=T(nI:Ns-1,1:p)
   a=a(p:p,:);a=a'
   xt=l*a
   de=xt'
   % forecasting is done for observations that are used in
obtaining model
   % f is real observation, d is predicted observation and h
is error
   % SS is sum of the squared errors for these
   d=de(1:nF-nI)
   f=c(nI+1:nF)
   h=f'-d'
   SS=h'*h;MSE=SS/(nF-(p+1)+1)
   % forecasting is done from nF+1 to Ns
   % ft is real, dt is predicted, th is error, sumth is sum
of the squared error
   dt=de(nF+1-nI:Ns-nI)
   ft=c(nF+1:Ns)
   th=ft'-dt'
   sumth=th'*th
   z=nI+1:nF
   ze=nF+1:Ns
   B=[z',f',d',h]
   A=[ze',ft',dt',th]
   % plotting and displaying results
   plot(aI')
   pause
   plot(z,f,'-',z,d,':')
   pause
   plot(z,h)
   pause
   close
   disp('    z        f          d          t')
   disp(B)
   disp('SS');disp(SS)
   disp('MSE');disp(MSE)
   disp('a');disp(a)
   pause
   plot(ze,ft,'-',ze,dt,':')
   pause
   plot(ze,th)
   pause
   close
   disp('  ze         ft         dt         th')
   disp(A)
   disp('sumth');disp(sumth)
```

ÖĞRETİM KURU
DOKÜMANTASYON MERKEZİ

```
    pause
    aIRLS=[]
   end
end
```