

# A Study on a Kalman Filter & Recursive Parameter Estimation Approach Applied to Stock Prediction

by

**Denis McGonigal**

A thesis submitted to the  
School of Graduate Studies and Research  
of the University of Ottawa in partial  
fulfillment of the requirements for the degree of

**Master's in Systems Science**

Department of Systems Science  
Faculty of Administration  
University of Ottawa  
Ottawa, Ontario  
Canada, K1N 6N5

©Denis McGonigal, Ottawa, Canada, 1996



National Library  
of Canada

Acquisitions and  
Bibliographic Services Branch

395 Wellington Street  
Ottawa, Ontario  
K1A 0N4

Bibliothèque nationale  
du Canada

Direction des acquisitions et  
des services bibliographiques

395, rue Wellington  
Ottawa (Ontario)  
K1A 0N4

*Your file* *Votre référence*

*Our file* *Notre référence*

**The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.**

**L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.**

**The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.**

**L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.**

ISBN 0-612-16425-X

**Canada**



**UNIVERSITÉ D'OTTAWA**  
**UNIVERSITY OF OTTAWA**

## ABSTRACT

This thesis describes a first experimental project using a recursive parameter estimation and Kalman filter approach to on-line modelling and prediction of stock market time-series. On-line (real-time) and daily closing price stock data are identified as Box-Jenkins ARIMA models. Differencing is performed to obtain a locally wide sense stationary process which is identified through spectral estimation methods. The initial model parameters are updated on-line via the Recursive Prediction Error algorithm and predictions are performed using the Kalman filter. This approach is studied and compared to the traditional Box-Jenkins SISO approach. The daily stock processes are also modeled as autoregressive processes embedded in white noise, which make an ideal investigation for the Kalman filter.

## ACKNOWLEDGEMENT

I wish to express my gratitude to my supervisor, Dr. Dan Ionescu, for his guidance throughout this project. The defence committee (Drs. Dan Lane, Tet H. Yeap, Jean-Michel Thizy, and Edward Pascal) should be thanked as well.

I wish to thank my colleagues and friends, Neil Keon, Phillippe Lavoie, and Mike Vukadinovic. Their assistance and friendship is well appreciated. A special thanks to Cris Lambiri for obtaining the real-time data required for this study.

Finally, I would like to express my sincere gratitude to all of my family to whom this work is dedicated. This opportunity could have never been realized without them.

# Contents

<b>Abstract</b> . . . . .	ii
<b>Acknowledgement</b> . . . . .	iii
<b>Table of Contents</b> . . . . .	iv
<b>Notation</b> . . . . .	viii
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation for Recursive Parameter Estimation . . . . .	2
1.2 Motivation for the Use of Kalman Filter in Prediction . . . . .	3
1.3 Thesis Objective & Contributions . . . . .	4
1.4 Time-Series Modelling Approach . . . . .	5
1.5 Organization . . . . .	6
<b>2 Time-Series Models, the Kalman Filter, &amp; Recursive Identification</b>	<b>7</b>
2.1 Time-Series Models . . . . .	8
2.1.1 ARMA Models . . . . .	8
2.1.2 ARIMA Models . . . . .	10
2.2 Spectral Estimation . . . . .	11
2.2.1 Overview of Spectral Methods . . . . .	12
2.3 The Kalman Filter . . . . .	17
2.3.1 State-Space System Model . . . . .	18

2.3.2	Kalman Filter Problem . . . . .	19
2.3.3	Kalman Filter Solution . . . . .	20
2.4	Real-time Recursive Parameter Estimation . . . . .	21
2.4.1	RPE Method . . . . .	22
<b>3</b>	<b>An Algorithm for Stock Process Prediction</b>	<b>24</b>
3.1	Initial Model Selection Strategy . . . . .	25
3.1.1	Differencing Strategy . . . . .	25
3.1.2	ARMA Modelling Methodology . . . . .	26
3.1.3	AR Processes in the Presence of White Noise . . . . .	27
3.2	Application of the Kalman Filter for ARMA Processes and Prediction . . . . .	28
3.2.1	State-Space ARMA Model . . . . .	29
3.2.2	Application to Prediction . . . . .	31
3.2.3	Obtaining Initial Conditions . . . . .	32
3.3	$n$ -step Prediction . . . . .	32
3.3.1	$n$ -step Prediction using the Kalman Filter . . . . .	33
3.3.2	$n$ -step SISO (Box-Jenkins) Prediction . . . . .	34
3.3.3	Constructing the ARIMA Prediction Through Undifferencing of the ARMA Model . . . . .	35
3.4	Implementation of the RPE Algorithm . . . . .	35
3.4.1	SISO RPE Algorithm for ARMA Models . . . . .	36
3.4.2	State-Space RPE Algorithm for ARMA Innovations Model . . . . .	37
3.4.3	RPE Algorithm for AR Model in White Noise . . . . .	38
3.4.4	Implementation Issues of the General RPE Method . . . . .	38
3.5	An Algorithm for Stock Process Prediction . . . . .	41

<b>4</b>	<b>Experiments Performed on On-Line Data</b>	<b>43</b>
4.1	Determination of Sampling Frequency . . . . .	43
4.2	Data Samples . . . . .	45
4.3	Determination of Needed Sample Size for Spectral Modelling . . . . .	46
4.4	Comparing the Performances of the Off-line Time-Invariant to the On-line State-Space and SISO RPE Approaches . . . . .	48
4.5	$n$ -step Prediction . . . . .	52
4.6	Higher Order Models . . . . .	54
<b>5</b>	<b>Experimentation on Daily Closing Price Processes</b>	<b>57</b>
5.1	Determination of Needed Sample Size for Spectral Modelling . . . . .	57
5.2	Comparing the Performances of the Off-line Time-Invariant to the On-line State-Space and SISO RPE Approaches . . . . .	59
5.3	$n$ -step Prediction . . . . .	64
5.4	Higher Order Models . . . . .	66
5.5	Fine Tuning the Forgetting Factor . . . . .	68
5.6	AR Modelling in White Noise . . . . .	71
<b>6</b>	<b>Summaries, Extensions &amp; Remarks</b>	<b>75</b>
6.1	Summary of Study on Prediction of On-line Processes . . . . .	75
6.2	Summary of Study on Prediction of Daily Processes . . . . .	76
6.3	Extensions . . . . .	77
6.4	<i>C++</i> Implementation . . . . .	79
6.4.1	Contribution to Public Programming Tools . . . . .	80
6.4.2	Class Implementation for ARMA Spectral Methods . . . . .	80
6.4.3	Class Implementation for Kalman Filter and RPE Algorithms . . . . .	81

6.5	Contributions . . . . .	83
6.6	Closing Remarks . . . . .	83
<b>A</b>	<b>Brief Review of Time Series and Stochastic Processes</b>	<b>86</b>
A.1	Discrete Time Series . . . . .	86
A.2	Discrete Stochastic Processes . . . . .	87
A.3	Wide Sense Stationary Processes . . . . .	87
A.4	Power Spectral Density . . . . .	88
A.5	Gaussian Processes . . . . .	88
A.6	White Noise Processes . . . . .	89
<b>B</b>	<b>Spectral Algorithms</b>	<b>90</b>
B.1	MYWE Algorithm . . . . .	90
B.2	Burg Algorithm . . . . .	91
<b>C</b>	<b>Results From Experimentation on On-line Data</b>	<b>92</b>
C.1	Higher Order Models . . . . .	92
<b>D</b>	<b>Results From Experimentation on Daily Data</b>	<b>93</b>
D.1	Determination of Sample Size for Modelling . . . . .	93
D.2	Fine Tuning the Forgetting Factor . . . . .	94
D.3	AR Modelling in White Noise . . . . .	96

# Notation

Various symbols, superscripts, subscripts, and abbreviations used frequently in this document are summarized below. All notation is fully defined where it first arises in the text.

## Symbols

- \* Denotes complex conjugate transpose.
- $T$  Denotes the transpose of a vector or matrix.
- $\hat{\cdot}$  Denotes prediction or estimate.
- $\sim$  Denotes a distribution of a random variable; *i.e.*  $X \sim N(0, 1)$ , denotes that  $X$  follows a Gaussian or Normal distribution.
- $a_i$   $i^{\text{th}}$  auto-regressive parameter of ARMA process.
- $b_i$   $i^{\text{th}}$  moving average parameter of ARMA process.
- $d$ 
  1. Degree of differencing in ARIMA modelling.
  2. Dimension of parameter vector in the RPE algorithm.
- $D_k$  Diagonal matrix used in UD-decomposition in the RPE algorithm.
- $e_k$  Prediction error or residual of process.
- $F_k$  Transition matrix of state-space model.
- $G_k$  State noise coupling matrix of state-space model.
- $H_k$  Observation matrix of state-space model.
- $K_k$ 
  1. Kalman filter gain and modified gain matrix.
  2. Reflection coefficient of AR process.
- $L_k$ 
  1. Output covariance of Kalman filter; a  $1 \times 1$  matrix.
  2. Parameter gain of RPE algorithm; a  $d \times 1$  matrix.
- $M_k$  Modified gain in the Kalman filter algorithm.
- $N(\mu, \sigma^2)$  Denotes a Gaussian or Normal distribution with mean  $\mu$  and variance  $\sigma^2$ .
- $P_k^{(+)}$  Associated *a posteriori* error covariance matrix of  $\hat{x}_k^{(+)}$  in Kalman filter.
- $P_k^{(-)}$  Associated *a priori* error covariance matrices of  $\hat{x}_k^{(-)}$  in Kalman filter.

- $P_{k+n}^{(-,n)}$  Associated  $n$ -step *a priori* error covariance matrix of  $\hat{x}_{k+n}^{(-,n)}$  in Kalman filter.
- $P_k$  Variable used to represent the matrix  $\gamma_k R_k^{-1}$  to facilitate the implementation of the RPE algorithm. A  $d \times d$  matrix.
- $Q_k$  Covariance matrix of driving noise  $w_k$  of state-space model.
- $R_k$  1. Covariance matrix of observation noise  $v_k$  of state-space model.  
2. Hessian matrix in RPE method.
- $S_k$  1. Correlation matrix of driving and observation noise of state-space model;  $\mathcal{E}\{w_k v_k\}$ .  
2. Also the denominator of the gain  $L_k$  in RPE algorithm; a  $1 \times 1$  matrix.
- $\{u_k\}$  Input driving sequence to ARMA process, usually a white noise process.
- $U_k$  Upper-triangular matrix used in U-D decomposition of the matrix  $P_k = U_k D_k U_k^T$  in the RPE algorithm.
- $\{v_k\}$  White noise process that corrupts the observation process  $\{z_k\}$  in state-space model. Termed observation noise.
- $\{x_k\}$  1. Output sequence of ARMA process.  
2. State vector of state-space model.
- $\hat{x}_k$  *A priori* estimate of state vector in RPE algorithm at time  $k$ . A one-step prediction state prediction.
- $\hat{x}_k^{(+)}$  *A posteriori* estimate of state vector of Kalman filter at time  $k$  using the measurement  $Z_k$ .
- $\hat{x}_k^{(-)}$  *A priori* estimate of state vector of Kalman filter at time  $k$  using the measurement  $Z_{k-1}$ . A one-step prediction state prediction.
- $\hat{x}_{k+n}^{(-,n)}$  *A priori*  $n$ -step state estimate in Kalman filter at time  $k$  using the measurement  $Z_{k-1}$ . An  $n$ -step prediction state prediction.
- $y_k$  System output in the RPE algorithm.
- $\hat{y}_k$  Prediction error of the output of the system in the RPE algorithm.
- $\hat{y}_k$  Prediction of the output of the system in the RPE algorithm.
- $\{z_k\}$  The observation process of state-space model.
- $\{\tilde{z}_k\}$  Innovations process in Kalman Filter.
- $Z_k$  Observation sequence  $z_0, z_1, \dots, z_k$ .

## Greek Letters and Symbols

$\delta_{kl}$	Kronecker delta function.
$\Delta$	Difference operator.
$\Delta^d z_k$	$d^{\text{th}}$ difference of observation vector $z_k$ .
$\varepsilon$	Prediction error in RPE algorithm. Equals $\hat{y}$ in RPE applied to state-space.
$\tilde{\varepsilon}$	Filtered signal in SISO RPE algorithm.
$\bar{\varepsilon}$	Residual in SISO RPE algorithm.
$\{\gamma_k\}$	Gain sequence in RPE method.
$\lambda$	Forgetting factor of the RPE method.
$\psi$	Gradient of predictions in RPE method. A $d \times 1$ matrix
$\rho$	Prediction error power; modelling error in SE.
$\sigma^2$	Variance of a process.
$\theta$	Parameter vector of RPE method. A vector of size $d = p + q$ for ARMA system.
$\hat{\theta}_k$	Estimate of parameter vector of RPE method at time $k$ .
$\tilde{\theta}_k$	Difference between parameter vectors $\hat{\theta}_{k+1}$ and $\hat{\theta}_k$ of RPE method.
$\mu$	1. Factor used in projection algorithm to ensure stable parameters in RPE algorithm. 2. Mean of a stochastic process.

## Caligraphic Letters and Symbols

$\mathcal{A}(z)$	$z$ -transform of the AR branch.
$\mathcal{B}(z)$	$z$ -transform of the MA branch.
$\mathcal{E}$	Expectation operator.
$\mathcal{H}(z)$	System transfer function for ARMA process; also the $z$ -transform of the impulse response function $h$ .
$\mathcal{M}_S$	Stable model set for ARMA model.
$\mathcal{Z}^{-1}$	Inverse $Z$ -transform operator.

## Special Symbols

$A(f)$	Denotes $\mathcal{A}(z)$ where $z$ is evaluated along the unit circle i.e. $z = \exp(j2\pi f)$ for $-0.5 \leq f \leq 0.5$ .
$B(f)$	Denotes $\mathcal{B}(z)$ where $z$ is evaluated along the unit circle i.e. $z = \exp(j2\pi f)$ for $-0.5 \leq f \leq 0.5$ .
AR( $p$ )	Auto-regressive process of order $p$ .
ARIMA( $p, d, q$ )	Auto-regressive Integrated Moving Average process of order $p, d, q$ .
ARMA( $p, q$ )	Auto-regressive Moving Average process of order $p, q$ .
MA( $q$ )	Moving Average process of order $q$ .
$O(\cdot)$	Order of an algorithm.
$P_{xx}(f)$	Power Spectral Density.
$r_{xx}[k]$	Autocorrelation function (ACF) at lag $k$ .

## Acronyms

ACF	Autocorrelation Function.
ADF	Adjusted Dickey-Fuller method.
AIC	Akaike Information Criterion.
AMLE	Akaike Maximum Likelihood Estimator.
AR	Auto-Regressive.
ARIMA	Auto-Regressive Integrated Moving Average.
ARMA	Auto-Regressive Moving Average.
DTS	Discrete Time Series.
EKF	Extended Kalman Filter.
MA	Moving Average.
MLE	Maximum Likelihood Estimator.
MYWE	Modified Yule-Walker Equations.
NaN	Not a Number; designates $\pm\infty$ .

SE	Spectral Estimation.
SISO	Single-Input Single-Output.
SS	State-Space.
PSD	Power Spectral Density.
RPE	Recursive Prediction Error.
SAS	Statistical Analysis System.
WSS	Wide Sense Stationary.

# Chapter 1

## Introduction

For many years stock market prediction has been a topic of interest in financial circles. Researchers in the area have used statistical models and the Box-Jenkins ARIMA approach for seasonal and non-seasonal time-series [1]. Other ongoing works use the neural net approach [2, 3, 4] and one other work has coupled neural networks with the statistical model approach [5]. These approaches to identification and prediction are performed using off-line methods to identify linear time-invariant models of daily closing price data which allowed ample time for analysis in application.

With the growing popularity of the Internet, on-line information of stock market processes are now available to virtually anybody with access. Computer programs can be written with reasonable programming effort to monitor this information which may be used for on-line prediction of stocks. For example, a communications program can be written using UNIX programming to query information sites that can reach a rate of one query for every minute. Information on histories of daily closing prices of stocks are also available from sites on the Internet <sup>1</sup>.

---

<sup>1</sup>The data of daily closing prices that were used in this project were obtained by FTP from the World

## 1.1 Motivation for Recursive Parameter Estimation

Although time-series models can yield good results in the short term, these methods may degrade if the processes are not remodeled as fresh data arrive. This degradation may happen due to a linear approximation of a likely non-linear system or due to time-variant properties that are ignored. A short time-series data set is assumed to be locally wide sense stationary and parameter estimates of the model are time-invariant. As new data arrive these assumptions may no longer be valid.

With limited resources, on-line techniques may be a necessity for real-time monitoring and prediction of many stock processes. The scarceness of resources may arise from a limited computing facility coupled with the monitoring of a large group of stocks in concert. As time continues, and further measurements of stock processes are available, it may be necessary to adapt models of these to the time-variant or non-linear nature of the system in order to obtain reasonable predictions. Off-line methods for remodelling processes may no longer be feasible for real-time needs.

On-line prediction and adaptive modelling of stock market processes appear to be absent from the literature. The advantage of on-line algorithms lies in the use of recursive formulae, so that state and parameter estimates can be updated without repeated computation of matrix solutions which can be quite time consuming. As well, with processing of data being made on-line, old data may be discarded and potential data storage problems are bypassed.

The recursive parameter estimation algorithm that is used in this project is the general Recursive Prediction Error (RPE) method [6]. This general algorithm has many implementations in identification. In this project it is applied to both SISO and state-space models<sup>2</sup>

---

Wide Web site <http://www.sccapl.com/cgi-bin/qs>.

<sup>2</sup>Throughout this report the term SISO refers to single-input single-output systems in *scalar* difference equation form. The term state-space refers to systems, that may be single-input single-output systems, but

and can be applied to time-variant or time-invariant modelling. The RFE method updates its parameter vector by using information obtained from its linear prediction errors of the process which is being modelled. The algorithm can be used to update system models when initial parameter estimates are uncertain or unknown in the time-variant and time-invariant cases. For time-variant parameter estimation, the algorithm's forgetting factor can be fine tuned or adjusted for better adaptation to the system being modeled.

## 1.2 Motivation for the Use of Kalman Filter in Prediction

The Box-Jenkins (SISO) prediction method is a well known approach for prediction of stationary stock processes. It uses linear combinations of past outputs measured from a process, and estimates of past inputs obtained from prediction errors of this process. This approach does not take into account or take advantage of the noise statistics of the model when computing its predictions.

The Kalman filter [7, 8, 9] is an alternative method than can be used for linear prediction for a wide range of processes. The distinctive feature of the Kalman filter is its mathematical formulation which is described in terms of stochastic state-space concepts and its solution for state estimation and prediction is computed recursively. The Kalman filter's variable gain may be an advantage in providing state estimates as the noise statistics of the state-space model are used to compute the filter gains which may produce enhanced predictions.

The state-space concept has only recently been popularized for time-series modelling <sup>3</sup>

---

the key difference is that the system is represented in state-space.

<sup>3</sup>Although the Kalman filter's main application was intended for state estimation, it is worth mentioning that it can also be used for estimation of model parameters [10], but this application is not investigated in this project.

[10, 11]. In control engineering, the Kalman filter has been a fundamental algorithm since its first appearances in [7, 12].

### 1.3 Thesis Objective & Contributions

This thesis describes a first experimental project using a recursive parameter estimation and state-space Kalman filter approach to on-line modelling and prediction of stock market processes using ARIMA modelling of time-series data. A study on these approaches is performed on on-line (real-time)<sup>4</sup>and daily closing price stock processes.

The study attempts to compare the traditional SISO (Box-Jenkins) approach to the state-space Kalman filter approach and looks at the importance of the use of on-line adaptive modelling. For the state-space approach, the processes are also modeled as auto-regressive processes embedded in noise which makes an ideal study for the Kalman filter. The ARIMA approach for time-series modelling of daily closing price time-series has been an approach used in the past but no studies appear to have been done using on-line modelling methods for on-line or daily closing price data.

#### Important Issues

It is important to mention a few things about the approaches that are taken in this project. The first is that a linear modelling approach is taken to identifying processes that can exhibit non-linear behaviour. This modelling approach may still be used since any non-linear behaviour is linearized around the nominal point through the recursive parameter estimation. The second is that the Kalman filter is a method that is based on Gaussian

---

<sup>4</sup>The terms on-line and real-time refer to the same data. They are time-series data that would be available on-line on a basis of every few minutes, a real-time application.

disturbances to the system. Stock markets processes are random processes that can exhibit non-Gaussian behaviour.

It should also be emphasized that a system using these approaches like any other forecasting system, can be a useful tool, but its suggestions (predictions) are not intended to be a main factor in a decision making process. The analysis provided by any system such as this may not be profitable unless it is coupled with the expertise that comes from studying the qualitative characteristics of the relevant economic environment.

## 1.4 Time-Series Modelling Approach

The Box-Jenkins Auto-Regressive Integrated Moving Average (ARIMA) models of time-series are used as the statistical models of the stock market processes. This approach has been used in the past for modelling time-series representing stock market processes. These models are identified by differencing an assumed non-stationary process to obtain a locally wide sense stationary (WSS) process. The resulting process is modeled as an Auto-Regressive Moving Average (ARMA) process. It is assumed that little knowledge or understanding of the mechanisms that drive stock market processes is known. This is the motivation for using this general class of models.

A nearly automated method for building initial ARMA models using spectral estimation methods is employed. These methods often provide efficient recursive algorithms to provide a set of models of different orders. An automated model selection criterion is used to choose a model from this set.

Upon obtaining the initial model using the off-line spectral techniques, the assumed time-variant parameters can be updated and the state and output of the system can be predicted on-line via the RPE algorithm. Traditionally, time-series modelling has mostly been used to

identify time-invariant models and predictions were mostly performed on a small range of data extending into the near future. For example, a process is usually modeled over 100 to 200 serial observations (depending on data availability) and prediction is performed on the next 5 to 10 units of time. The assumption of local wide sense stationarity is maintained over this short period. Since it is desired to perform on-line prediction as further new data are available, local wide sense stationarity is no longer valid and the time-variant nature of the system may be needed to be accounted for. This is the reason for using the RPE method. For systems showing time-invariant behaviour, a time-invariant implementation of the RPE algorithm can also be used to improve modelling of the system.

## **1.5 Organization**

The following chapter describes the general approaches that are used in this thesis such as the general class of ARIMA models, spectral estimation, Kalman filtering and the general algorithm for the RPE method. The purpose of Chapter 3 is to present the implementations of these approaches to the specific models used in this project. In Chapter 4, experiments are described and results of these experiments using on-line data are presented and discussed. Chapter 5 presents the results of experimentation on daily closing price data. In Chapter 6, the discussions of the results from the previous two Chapters 4 and 5 are presented as well as the conclusion and extensions. There is also an appendix that includes a review of time-series and stochastic processes, spectral algorithms, and results are deferred.

## Chapter 2

# Time-Series Models, the Kalman Filter, & Recursive Identification

This chapter introduces the tools that are used in this project. ARMA and ARIMA time-series models are presented along with an overview of the spectral methods that model ARMA processes. The general Kalman filter problem is defined and its solution presented. The general Recursive Prediction Error (RPE) algorithm used for real-time parameter estimation is also presented. In Appendix A, some basics on time-series and stochastic processes are reviewed. This revision could have been placed within this chapter but it was deferred to the appendix, as it was thought that most readers would be somewhat familiar with these concepts.

## 2.1 Time-Series Models

Many discrete-time stochastic processes<sup>1</sup> encountered in practice are closely approximated by time-series models. Two of the most important model classes for time-series are the *auto-regressive moving average* (ARMA) and *auto-regressive integrated moving average* (ARIMA) models (also known as the Box-Jenkins class of models) which have been popular for representing stock market and economic [1, 5, 10] time-series. A convenient aspect about using these model classes is that modelling is performed with little knowledge or understanding of the dynamic mechanisms of the process under study.

### 2.1.1 ARMA Models

*Auto-Regressive Moving Average* (ARMA) models are a general class of linear models used to describe stationary stochastic processes. In a scalar ARMA model, an input driving sequence  $\{u_n\}$  and the output sequence  $\{x_n\}$  that are used to model the data are related by the linear difference equation (2.1). Generally  $\{u_n\}$  is a non-measurable white noise process with variance  $\sigma_u^2$ , that may only be estimated.

$$x_n = - \sum_{k=1}^p a_k x_{n-k} + \sum_{k=0}^q b_k u_{n-k} \quad (2.1)$$

The ARMA process is denoted as an ARMA( $p, q$ ) process. The  $a_k$  and  $b_k$  coefficients are termed the *auto-regressive* and the *moving average* parameters<sup>2</sup> respectively. If  $q = 0$  the process is termed auto-regressive, AR( $p$ ) and if  $p = 0$ , the process is termed moving average,

---

<sup>1</sup>It is henceforth assumed that the term *stochastic process* or *random process* will be represented by the term *process*.

<sup>2</sup>Here  $a_0 = b_0 = 1$ .  $a_0$  implicitly multiplies  $x_n$  on the left hand side.

MA( $q$ ). For time-varying ARMA processes, which are no longer WSS, the parameters of the model are time indexed.

The input driving noise in equation (2.1) should not be mistaken for observation noise that may be added onto the ARMA signal [13, page 109]. This driving noise gives rise to the random nature of the observed process. If observation noise is present, one would obtain,

$$y_n = x_n + w_n = - \sum_{k=1}^p a_k x_{n-k} + \sum_{k=0}^q b_k u_{n-k} + w_n \quad (2.2)$$

where  $\{w_n\}$  is often assumed to be a white-noise process independent of  $u_n$ , with variance  $\sigma_w^2$ .

ARMA models are often referred to as *transfer function* models. The system transfer function  $\mathcal{H}(z)$  between the input  $u_n$  and the output  $x_n$  for the ARMA process of (2.1) is the rational function,

$$\mathcal{H}(z) = \frac{\mathcal{B}(z)}{\mathcal{A}(z)} \quad (2.3)$$

where  $\mathcal{A}(z) = \sum_{k=0}^p a_k z^{-k}$  is the  $z$ -transform of the AR branch and  $\mathcal{B}(z) = \sum_{k=0}^q b_k z^{-k}$  is the  $z$ -transform of the MA branch.

To ensure that  $x_n$  is a WSS process the zeros of  $\mathcal{A}(z)$  must be within the unit circle around the origin of the complex plane [13, page 109]. Some restrictions on  $\mathcal{B}(z)$  are necessary in model identification. The implementation of certain ARMA identification algorithms require that the stability of the MA parameter branch be monitored to avoid numerical situations that may cause the algorithm to breakdown and crash [6, page 483] [13, page 311].

## 2.1.2 ARIMA Models

There is a wider class of models called *Auto-Regressive Integrated Moving Average* (ARIMA) models which is capable of exhibiting ARMA model behaviour as well as non-stationary behaviour. Non-stationary processes behave as if they had no fixed mean or autocorrelation function (ACF). Some of these processes can be *differenced* an appropriate number of times to obtain a locally WSS process. What is meant by *differencing* is explained in the following.

Suppose the observed process is  $z_0, z_1, \dots, z_n, \dots, z_N$ . Consider the operation defined for  $n \geq 1$ , by

$$\Delta z_n \triangleq z_n - z_{n-1}. \quad (2.4)$$

The  $\Delta$  operator is called the difference operator. The operator  $\Delta^d$  applied to  $z_n$ , for  $n \geq d$ , is called the  $d^{\text{th}}$  difference of  $z_n$  and is defined for  $d \geq 1$ , by

$$\Delta^d z_n \triangleq \Delta^{d-1} z_n - \Delta^{d-1} z_{n-1}. \quad (2.5)$$

If the number of times that the time-series  $z_n$  has been differenced is correct, the resulting process  $\{\Delta^d z_n\}$  will be locally stationary and it may now be represented by an appropriate ARMA model of order  $(p, q)$ . Now that  $\Delta^d z_n$  is a locally stationary process, it is represented by  $x_n$ . It is said that the  $z_n$  process is an ARIMA process of order  $(p, d, q)$  and is described by equation (2.6).

$$\Delta^d z_n \triangleq x_n = - \sum_{k=1}^p a_k x_{n-k} + \sum_{k=0}^q b_k u_{n-k} \quad (2.6)$$

The strategy for differencing is discussed further on in the next chapter.

## Reconstructing the ARIMA process through Anti-Differencing of the ARMA Process

The reconstruction of the ARIMA process from the ARMA process is shown here. This will be useful further on for constructing ARIMA predictions. Recalling the equations (2.4) and (2.5), it can be shown that the following closed form for  $\Delta^d z_n$  is obtained.

$$\Delta^d z_n = \sum_{i=0}^d (-1)^i \binom{d}{i} z_{n-i} \quad (2.7)$$

With equation (2.7), we may isolate and solve for  $z_n$ .

$$z_n = \Delta^d z_n - \sum_{i=1}^d (-1)^i \binom{d}{i} z_{n-i} \quad (2.8)$$

where

$$\binom{d}{i} = \frac{d!}{(d-i)!i!} \quad (2.9)$$

## 2.2 Spectral Estimation

The general problem of *spectral estimation* (SE) is that of determining the spectral content of a WSS stochastic process based on a finite set of observations from that process [13]. What concerns us the most about SE is that it can be used to give estimates of the parameters of an ARMA process [13]. The most important result from SE is the relationship between AR, MA and driving noise variance parameters and the ACF of a WSS process  $x_n$  given by (2.10).

$$r_{xx}[k] = \begin{cases} -\sum_{i=1}^p a_i r_{xx}[k-i] + \sigma_u^2 \sum_{i=0}^{q-k} h_i^* b_{i+k} & \text{for } k = 0, 1, \dots, q \\ -\sum_{i=1}^p a_i r_{xx}[k-i] & \text{for } k \geq q+1 \end{cases} \quad (2.10)$$

Here  $\sigma^2$  denotes the variance of the driving noise of the process,  $r_{xx}[i]$  is the ACF, and  $h$  is the impulse response function. The superscripted  $*$  denotes the complex conjugate transpose.

The relationships between the ACF and the AR, MA and driving noise variance parameters of the ARMA model is a nonlinear one. Given the ACF, we must solve a set of nonlinear equations to find the model parameters.

Realistically one may only obtain an estimate of the ACF based on the  $N$  contiguous observations  $\{x_0, x_1, \dots, x_{N-1}\}$  of a single realization of a WSS random process. For practical applications the observation interval can be quite short. From this truncated estimate of the ACF, estimates of the ARMA parameters can be obtained through the relationship given in (2.10).

There is another important reason for the limitation on the number of points in the observation interval. It is that many processes are not WSS but can be characterized as *locally WSS* if the non-stationarity is not too severe. Locally WSS processes are processes such that the expectation  $\mathcal{E}(x^*[n]x[n+k])$  has a small variation with  $n$  over the duration of the observation interval. If the observation interval, which is  $N$  samples in length, is sufficiently small, the process may be considered to be locally WSS [13].

### 2.2.1 Overview of Spectral Methods

There exist many well known algorithms related to spectral estimation for the identification of AR, MA and ARMA models[13]. This section briefly describes the methods that are used

in this project<sup>3</sup>.

### Akaike Approximate MLE

The *Akaike Approximate Maximum Likelihood Estimator* (AMLE) algorithm [15] is the choice algorithm[16] of ARMA parameter estimation methods. It attempts to solve equations (2.10) by minimization of a highly non-linear function. This algorithm is a non-linear optimization scheme which is iterative in nature and is not guaranteed to converge. If convergence does occur, the local minimum that is found may not be global. It is important to begin the iteration with an initial estimate<sup>4</sup> that is close to the true parameter values, so that hopefully the global minimum will be found.

The Akaike MLE algorithm attempts to find estimates of the ARMA parameters that minimize

$$Q(a, b) = \frac{1}{N} \int_{-\frac{1}{2}}^{\frac{1}{2}} |X(f)|^2 \left| \frac{A(f)}{B(f)} \right|^2 df. \quad (2.11)$$

where  $N$  is the number observations in the time-series being modelled. Akaike proposed using the Newton-Raphson iteration to find a zero of the following equation

$$\begin{bmatrix} a_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} a_k \\ b_k \end{bmatrix} - H^{-1}(a_k, b_k) \begin{bmatrix} \frac{\partial Q}{\partial a} \\ \frac{\partial Q}{\partial b} \end{bmatrix} \Bigg|_{\substack{a = a_k \\ b = b_k}} \quad (2.12)$$

where  $a_k$  and  $b_k$  are the  $k^{th}$  iterates of the AR and MA filter parameter vectors, respectively.  $H(a, b)$ , the Hessian of  $Q$ , is defined as

---

<sup>3</sup>For complete development and discussion of the algorithms and their statistical properties of these methods, one should refer to the main references on this subject [13, 14].

<sup>4</sup>This initial estimate may be found using the Modified Yule-Walker Equations algorithm which follows this brief presentation of the AMLE.

$$H(a, b) = \begin{bmatrix} \frac{\partial^2 Q}{\partial a \partial a'} & \frac{\partial^2 Q}{\partial a \partial b'} \\ \frac{\partial^2 Q}{\partial b \partial a'} & \frac{\partial^2 Q}{\partial b \partial b'} \end{bmatrix} \quad (2.13)$$

Approximations of the partial derivatives are given in the following :

$$\begin{aligned} \frac{\partial Q}{\partial a[k]} &= 2 \sum_{i=0}^p a[i] \hat{r}_{yy}[k-i] \quad \text{for } k = 1, \dots, p \\ \frac{\partial Q}{\partial b[l]} &= -2 \sum_{i=0}^q b[i] \hat{r}_{zz}[l-i] \quad \text{for } l = 1, \dots, q \end{aligned} \quad (2.14)$$

$$\begin{aligned} \frac{\partial^2 Q}{\partial a[k] \partial a[l]} &= 2 \hat{r}_{yy}[k-l] \quad \text{for } k = 1, \dots, p \\ &\quad \text{for } l = 1, \dots, p \\ \frac{\partial^2 Q}{\partial b[k] \partial b[l]} &= 2 \hat{r}_{zz}[k-l] \quad \text{for } k = 1, \dots, q \\ &\quad \text{for } l = 1, \dots, q \\ \frac{\partial^2 Q}{\partial a[k] \partial b[l]} &= -2 \hat{r}_{yz}[k-l] \quad \text{for } k = 1, \dots, p \\ &\quad \text{for } l = 1, \dots, q \end{aligned} \quad (2.15)$$

where

$$\begin{aligned} \hat{r}_{yy}[k] &= \frac{1}{N} \sum_{n=0}^{N-|k|-1} y[n] y[n+|k|] \\ \hat{r}_{zz}[k] &= \frac{1}{N} \sum_{n=0}^{N-|k|-1} z[n] z[n+|k|] \end{aligned}$$

and

$$\hat{r}_{yz}[k] = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-k-1} y[n] z[n+k] & \text{for } k \geq 0 \\ \frac{1}{N} \sum_{n=-k}^{N-1} y[n] z[n+k] & \text{for } k < 0 \end{cases} \quad (2.16)$$

The sequences  $y[n]$  and  $z[n]$  are defined as

$$\begin{aligned} y[n] &= \mathcal{Z}^{-1} \left\{ \frac{\mathcal{H}(z)}{\mathcal{B}(z)} \right\} \\ z[n] &= \mathcal{Z}^{-1} \left\{ \frac{\mathcal{H}(z) \mathcal{A}(z)}{\mathcal{B}^2(z)} \right\} \end{aligned}$$

where  $\mathcal{H}(z) = \sum_{n=0}^{N-1} x[n]z^{-n}$ . It is mentioned that the Akaike estimator may not yield minimum-phase filter estimates during the course of the iteration. If any iterate of the MA parameters causes  $\mathcal{B}(z)$  to have a zero outside the unit circle, then due to the instability of  $1/\mathcal{B}(z)$ , the  $y[n]$  and  $z[n]$  sequences will grow large. It is therefore necessary to monitor the stability of the  $1/\mathcal{B}(z)$  filter.

To avoid the inversion of the Hessian in equation (2.12), we may multiply the left and right hand sides by the Hessian to obtain

$$H(a_k, b_k) \begin{bmatrix} a_{k+1} \\ b_{k+1} \end{bmatrix} = H(a_k, b_k) \begin{bmatrix} a_k \\ b_k \end{bmatrix} - \begin{bmatrix} \frac{\partial Q}{\partial a} \\ \frac{\partial Q}{\partial b} \end{bmatrix} \Bigg|_{\substack{a = a_k \\ b = b_k}} \quad (2.17)$$

and this equation can be solved using the Cholesky decomposition since the Hessian is ensured to be positive definite.

As mentioned, the method is iterative and attempts to improve on an initial model estimate in which its order has been predetermined. It is unlike other methods that model recursively.

### Modified Yule-Walker Equations

The *Modified Yule-Walker Equations* (MYWE) algorithm is an ad hoc method to estimate the AR parameters of an ARMA process. The MYWE algorithm arises from the difficulties associated with solving the nonlinear equations (2.10) to obtain estimates of the ARMA parameters. This recursive method is direct, non-iterative and suboptimal. It can however provide a decent estimate of the parameters and can be used to provide an initial model estimate for an optimal ARMA parameter estimation algorithm such as the AMLE.

The MYWE method solves the MYWE equations (2.18) recursively to obtain an estimate of the AR parameters. The method requires only  $O(p^2)$  operations. Afterwards the ARMA

process is filtered using the AR parameters and the resulting MA process is identified via Durbin's method.

$$\begin{bmatrix} r_{xx}[q] & r_{xx}[q-1] & \dots & r_{xx}[q-p+1] \\ r_{xx}[q+1] & r_{xx}[q] & \dots & r_{xx}[q-p+2] \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}[q+p-1] & r_{xx}[q+p-2] & \dots & r_{xx}[q] \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_{xx}[q+1] \\ r_{xx}[q+2] \\ \vdots \\ r_{xx}[q+p] \end{bmatrix}. \quad (2.18)$$

The MYWE algorithm is presented in Appendix B.

### Durbin's Method

*Durbin's method* is used to find the parameters of MA processes. The straightforward algorithm has two main steps. The first step is to replace the MA( $q$ ) process by a large order AR process. That is for the time-series  $\{x[0], \dots, x[N-1]\}$ , some  $P$  is chosen<sup>5</sup> where  $q \ll P \ll N$ , and model this time-series as AR( $P$ ) using for example the Levinson or Burg algorithms.

The second step involves using the AR parameter estimates obtained in the first step as a new time-series and this one is modeled as an AR process as well. The resulting parameters of this second AR( $q$ ) process are the parameters for the MA( $q$ ) process. To obtain the variance of the process, the data is filtered to obtain an estimate of the driving white noise process and the power of this is taken.

### Burg Method

The most popular approach to AR parameter estimation is the *Burg* method. Based on the Levinson recursion [13, page 161], this method is another efficient algorithm in obtaining

---

<sup>5</sup>Typically we have  $q < N/10$  and  $P \sim N/5$ .

parameter estimates of an AR process. This method differs from the Levinson recursion<sup>6</sup> in that the AR parameters are estimated so that the method yields estimated poles that are guaranteed to be on or inside the unit circle.

This minimum phase property is not only useful for determining AR parameters but also for indirectly obtaining MA parameters which result from Durbin's method. (Recall that Durbin's method uses an AR method to obtain the MA parameters.) Durbin's method can now provide minimum phase MA parameter set to go along with the AR parameters obtained from the MYWE. The two parameter branches can be an initial parameter set to the AMLE which monitors the stability of the MA branch.

## 2.3 The Kalman Filter

The *Kalman filter* is a linear estimator for systems in state-space representation. In general, it is used to estimate the state of a linear dynamic system excited by white noise, by using observations of the system corrupted by additive white noise. The Kalman filter may also be used for predicting the state and output of a linear system.

The Kalman filter has the very important property that it is the optimal estimator for linear systems described above [9]. What is meant by this is that it is a computational algorithm that uses measurements of a process to deduce a minimum error, unbiased estimate of the state of the system by utilizing knowledge of the dynamics of the state and measurements and the assumed statistics of the system noises, measurement errors, and initial conditions [17]. By using the measurements from a process, the state estimate is thus a conditional

---

<sup>6</sup>The Levinson recursion estimates the AR parameters by recursively solving the MYWE (2.18) for  $p = 0, 1, \dots, p_{max}$  and  $q = 0$ .

estimate.

When the Gaussian assumption is dropped, there is no longer any guarantee that the Kalman filter will give the conditional mean of the state vector. However, it is still an optimal estimator in the sense that it minimizes the mean square error within the class of all linear estimators [10, page 105].

In the remainder of this section, the state-space model of the general system from which measurements are obtained is described. The discrete Kalman filtering problem and its solution are presented. The application of the Kalman filter to ARMA processes will be presented in the next chapter.

### 2.3.1 State-Space System Model

The state and observation equations are given respectively by the *state-space* model (2.19)(a,b). Here  $\{x_k\}$  is the state vector process in  $R^n$ , which in general is not observable.  $\{z_k\}$  is the observation vector process in  $R^p$ .

$$x_{k+1} = F_k x_k + G_k u_k \quad (2.19a)$$

$$z_k = H_k x_k + v_k \quad (2.19b)$$

The state equation is subject to initial conditions where  $x_0$  is a Gaussian random variable with mean and covariance (2.20).

$$x_0 \sim N(\bar{x}_0, P_0) \quad (2.20)$$

$\{u_k\}$  for  $k \geq 0$  is a Gaussian white noise vector process which disturbs the state process while the correlated and Gaussian  $\{v_k\}$  corrupts the observation process.  $\{u_k\}$  and  $\{v_k\}$  have mean 0 and covariances given by (2.21) and are not necessarily independent.

$$\mathcal{E} \left\{ \begin{bmatrix} u_k \\ v_k \end{bmatrix} \begin{bmatrix} u_l & v_l \end{bmatrix} \right\} = \begin{bmatrix} Q_k & S_k \\ S_k^T & R_k \end{bmatrix} \delta_{kl} \quad (2.21)$$

### 2.3.2 Kalman Filter Problem

The Kalman filtering problem is the conditional *state estimation* of the state  $x_k$ , using a linear function of the observations  $z_k$ . State estimates are denoted  $\hat{x}_k^{(+)}$  and  $\hat{x}_k^{(-)}$  and are called the *a posteriori* and *a priori* estimates of  $x_k$  respectively. Explained otherwise  $\hat{x}_k^{(+)}$  and  $\hat{x}_k^{(-)}$  are state estimates at time  $k$  using information of the observations up until time  $k$  and  $k - 1$  respectively. Formally these estimates are given by :

$$\hat{x}_k^{(-)} \triangleq \mathcal{E}[x_k | Z_{k-1}] \quad (2.22a)$$

$$\hat{x}_k^{(+)} \triangleq \mathcal{E}[x_k | Z_k] \quad (2.22b)$$

as well as the associated *a priori* and *a posteriori* conditional error covariance matrices :

$$P_k^{(-)} \triangleq \mathcal{E}\{[x_k - \hat{x}_k^{(-)}][x_k - \hat{x}_k^{(-)}]^T | Z_{k-1}\} \quad (2.23a)$$

$$P_k^{(+)} \triangleq \mathcal{E}\{[x_k - \hat{x}_k^{(+)}][x_k - \hat{x}_k^{(+)}]^T | Z_k\} \quad (2.23b)$$

where  $Z_k = \{z_1, z_2, \dots, z_k\}$ .

### 2.3.3 Kalman Filter Solution

The Kalman filter is described as in [8, page 122] by equations (2.25) and (2.26), subject to the initial conditions (2.24).

$$\hat{x}_0^{(-)} = \bar{x}_0 \quad (2.24a)$$

$$P_0^{(-)} = P_0 \quad (2.24b)$$

First  $\tilde{z}_k$  is denoted the innovations. The innovations  $\tilde{z}_k$  contains the new information that the observation brings that is not carried in  $Z_{k-1}$ . The output error covariance  $L_k$  represents the uncertainty of the output. The matrix gain  $K_k$  gives the value of the new information, the innovations  $\tilde{z}_k$ .

$$\tilde{z}_k = z_k - H_k \hat{x}_k^{(-)} \quad (2.25a)$$

$$L_k = H_k P_k^{(-)} H_k^T + R_k \quad (2.25b)$$

$$K_k = P_k^{(-)} H_k^T L_k^{-1} \quad (2.25c)$$

Note that  $P_k^{(-)}$  and  $R_k$  are symmetric positive semi-definite matrices and if  $R_k$  is positive definite then the inverse of  $L_k$  is guaranteed to exist. The core of the algorithm is the following set of recursive equations.

$$\hat{x}_k^{(+)} = \hat{x}_k^{(-)} + K_k \tilde{z}_k \quad (2.26a)$$

$$P_k^{(+)} = P_k^{(-)} - K_k H_k P_k^{(-)} \quad (2.26b)$$

$$\hat{x}_{k+1}^{(-)} = F_k \hat{x}_k^{(+)} + G_k S_k \tilde{z}_k \quad (2.26c)$$

$$P_{k+1}^{(-)} = F_k P_k^{(+)} F_k^T - F_k K_k S_k G_k^T - G_k S_k^T K_k^T F_k^T - G_k S_k^T L_k S_k G_k^T + G_k Q_k G_k^T \quad (2.26d)$$

It is also worth mentioning that when  $S_k$  is zero then the driving noise of the state process and the observation process are independent. The equations of (2.26) simplify somewhat to what is commonly referred to as the *Conventional Kalman filter*.

## 2.4 Real-time Recursive Parameter Estimation

The need for on-line recursive parameter estimation arises when fresh experimental data are continuously in supply and it is desired to improve our parameter estimates by making use of this new information. With a recursive formula, the estimates can be updated without repeatedly computing matrix solutions (often involving matrix inversions) that can be quite time consuming.

Many on-line methods exist for estimating parameters of a time-invariant model. These methods give equal weighting for all the measured data as the process evolves since the more recent data has as much value as older data in providing information about the system that is being modeled. These methods hold when the parameters of the system are virtually constant throughout the period of estimation. When it is suspected that the system that is

being modeled has time-variant properties, a turn to real-time recursive parameter estimation is needed. These methods place more weight on the more recent data and discount older information. As a result the parameter tracking capability can be increased if the weighting policy is correctly applied.

### 2.4.1 RPE Method

The algorithm used to estimate time-variant parameters in this project is the *Recursive Prediction Error* (RPE) method with quadratic criterion. There are other alternatives to this algorithm such as the Extended Kalman filter (EKF) [9, page 193] and the Real-time Least-squares Algorithm [18, page 29]. However in [6], it is discussed that the RPE encompasses these approaches.

The objective of the RPE algorithm is to minimize the criterion function  $l(k, \theta, \varepsilon)$ . Where  $k$  is the time index,  $\theta$  is the model's parameter vector and  $\varepsilon$  is the prediction error of the output of the system. Most often, as is the case in this project, the objective function is quadratic and expressed as :

$$l(k, \theta, \varepsilon) = \frac{1}{2} \varepsilon^T \Lambda^{-1} \varepsilon. \quad (2.27)$$

Here  $\Lambda$  is a positive definite weighting matrix for penalizing prediction errors. In practice, it is desired to minimize the expectation of this :

$$V(\theta) = \frac{1}{2} \mathcal{E} \{ \varepsilon^T \Lambda^{-1} \varepsilon \} \quad (2.28)$$

The following algorithm is the general RPE method for quadratic criteria[6, page 94].

## General RPE Method

$$\varepsilon_k = y_k - \hat{y}_k \quad (2.29a)$$

$$\hat{\Lambda}_k = \hat{\Lambda}_{k-1} + \gamma_k [\varepsilon_k \varepsilon_k^T - \hat{\Lambda}_{k-1}] \quad (2.29b)$$

$$R_k = R_{k-1} + \gamma_k [\psi_k \hat{\Lambda}_{k-1}^{-1} \psi_k^T - R_{k-1}] \quad (2.29c)$$

$$\hat{\theta}_k := [\hat{\theta}_{k-1} + \gamma_k R_k^{-1} \psi_k \hat{\Lambda}_{k-1}^{-1} \varepsilon_k]_{\mathcal{M}_S} \quad (2.29d)$$

$$\xi_{k+1} = A(\hat{\theta}_k) \xi_k + B(\hat{\theta}_k) z_k \quad (2.29e)$$

$$\begin{pmatrix} \hat{y}_{k+1} \\ \text{col} \Psi_{k+1} \end{pmatrix} = C(\hat{\theta}_k) \xi_{k+1} \quad (2.29f)$$

In the algorithm,  $y_k$  is the output of the system which is being modeled and  $\hat{y}_k$  is the prediction of this output. If the output of the system is scalar, as is the case for ARMA models, the matrix  $\Lambda$  becomes a scalar and can be taken as 1. The scalar  $\gamma$  is a gain between 0 and 1 for incorporating new information.

$R_k$  is the second derivative of  $V(\theta)$  with respect to  $\theta$  termed the Hessian<sup>7</sup>.  $\psi$  is the gradient of the predictions.  $\xi_{k+1}$  is the state of the system that is being modeled.  $\hat{\theta}_k$  is the estimate of the parameter vector of the system model. The subscripting the right hand side of the recursive update of  $\hat{\theta}_k$  denotes that the parameter estimate must fall within a stability region denoted  $\mathcal{M}_S$ .  $\begin{pmatrix} \hat{y}_{k+1} \\ \text{col} \psi_{k+1} \end{pmatrix}^T$  is the output vector of the algorithm.  $z_k$  is termed the input output data. The matrices A, B, and C are dependent on the system model chosen (i.e. SISO or state-space).

The applications of algorithm (2.29) to SISO and state-space models are presented in the following chapter along with implementation issues of the general RPE algorithm.

---

<sup>7</sup>The Hessian  $R_k$  should not be confused with the output noise covariance  $R_k$  of the Kalman filter algorithm.

## Chapter 3

# An Algorithm for Stock Process Prediction

This chapter discusses and presents the implementation of the tools presented in the previous chapter. The off-line modelling strategy for obtaining initial ARIMA models is discussed. The state-space ARMA representation and the Kalman filter applied to this model as one-step prediction are presented. Applications of the general RPE algorithm to ARMA models in SISO and state-space forms are given.  $n$ -step predictions are also presented using both SISO and state-space forms. In the final Section 3.5, these methods and strategies are brought together to describe an algorithm for ARIMA process prediction that will be applied to stock process prediction in experimentation described in the following chapters.

## 3.1 Initial Model Selection Strategy

The approach used in the selection for a model of a time-series process is a three stage process. The first step involves determining the degree of differencing needed to produce a stationary process. The second step involves the modelling of the resulting locally WSS process using the spectral methods. Many models of different order are identified in this stage. In the final step, one of these models is chosen via a model selection criterion.

### 3.1.1 Differencing Strategy

In Section 2.1.1, it was mentioned that (for an ARMA process) the zeros of  $\mathcal{A}(z)$  must lie within the unit circle around the origin of the complex plane for  $x_n$  to be WSS. In [1, page 174], it is said that if none of the zeros of  $\mathcal{A}(z)$  lie close to boundary of the unit circle,  $r_{xx}[k]$  (ACF) shows a tendency to “die away” towards zero as  $k$  increases. If the ACF does not show a tendency to die out quickly, then this suggests that a zero of  $\mathcal{A}(z)$  close to unity may exist and perhaps the underlying stochastic process is not stationary and differencing should be carried out.

**Method** The method [1, page 175] used to identify the degree of differencing is given in the following. Differencing is performed until the estimate of the ACF of  $w_k = \delta^d z_k$  dies out fairly quickly. At each step the mean of the differenced process is subtracted before taking the ACF [1, page 9]. In practice  $d$  is normally 0, 1 or 2 and it is sufficient to inspect the first 20 or so estimates of the ACF of the time-series and its first few differences.

No algorithms that automatically determine whether the ACF is “dying away” have yet to

be made public. At least one algorithm has been developed for this purpose. The *Adjusted Dickey-Fuller test* is implemented in the Forecast Pro time-series package, but the exact procedure is proprietary. From the literature it would appear that the differencing aspect of time-series modelling is more of an art than it is a science.

### **3.1.2 ARMA Modelling Methodology**

Although Box-Jenkins ARMA models are used in this project, the Box-Jenkins methodology[1] (identification, estimation and diagnostic checking) to modelling the stationary processes is not used. This method of model building demands the intervention of an experienced analyst to study the time-series, which can be excessively time consuming and costly. We opt for an automated model building process using the spectral estimation methods previously discussed, with an objective model selection criteria. The SE methods often provide efficient recursive algorithms to provide many options of models of different orders. A study resumed in [16], showed that the model selection criteria provide model selection that is often equal or superior to the time-consuming Box-Jenkins method. These results coupled with the difficulty of the Box-Jenkins methodology justify the use of the objective model selection criteria.

Upon obtaining an initial parameter set using the off-line SE techniques, the time-variant models and state of the system are then updated and predicted on-line in real-time via the RPE method previously introduced.

#### **Identification of ARMA Model Set**

This second step of the model selection strategy involves the identification of an initial model through spectral methods. This off-line identification uses recursion and iteration in the estimation of the ARMA model parameters.

The strategy is to use the ad hoc MYWE algorithm (Section 2.2.1) to estimate the AR parameters recursively for a given value of  $q$ . Since the value of  $q$  is fixed the method must be used to iterate over different values of  $q$  to obtain a complete set of models to be considered. Durbin's method is used to estimate the MA parameters for each model order. Once the model set is identified, each model's parameter estimate is refined using the AMLE.

### **Model Selection Criteria : Akaike Information Criterion**

After the generation of the model set, one model is chosen via a selection criteria. A wealth of model selection criteria are available [16]. These methods couple parsimony and the goodness of fit measure to select a preferred model from a set of prospective models of different orders. The *Akaike Information Criterion* (AIC), defined by equation (3.1), appears to be a method of choice for model selection in ARMA time-series modelling [13, page 235].

$$AIC(i, j) = N \cdot \ln \cdot \hat{\sigma}_{ij}^2 + 2 \cdot (i + j) \quad (3.1)$$

Here  $i$  and  $j$  are the assumed AR and MA model orders respectively.  $\hat{\sigma}^2$  is an estimate of the variance of the parameterized process. The selected model order is the one that minimizes the AIC. .

### **3.1.3 AR Processes in the Presence of White Noise**

In some cases AR processes are embedded in white noise such as in equation (2.2). When this is the case the resolution of the spectral estimator is diminished. If the white noise is powerful enough, the true AR signal is drowned out and when attempting to model this signal the auto-regressive parameter estimates are less accurate or a lower order model can

be selected.

Fortunately there exist techniques for modelling AR processes in white noise. The method that is used [13, page 238] in this project is described in the following paragraph. It is henceforth assumed that this method will be referred to as *Parzen's method*. Methods for modelling ARMA models in white noise are not known.

**Parzen's Method** It is known that the model for an AR( $p$ ) process in white noise can be given by an ARMA( $p,p$ ) model, where the auto-regressive parameters of the ARMA model are identical to those of the AR process[19]. Estimates of the independent driving white noise and observation white noise statistics ( $\hat{\sigma}_u^2$  and  $\hat{\sigma}_w^2$ ) can be obtained using a least squares estimator proposed by Parzen [13, 20]. The least-squares equation that must be solved is given in equation (3.2), where  $\star$  is the convolution operator. The MYWE method is used to obtain estimates for the auto-regressive parameters.

$$\begin{bmatrix} 1 & \sum_{k=0}^p |\hat{a}_k|^2 \\ \sum_{k=0}^p |\hat{a}_k|^2 & \sum_{k=0}^{2p} |\hat{a}_k \star \hat{a}_k|^2 \end{bmatrix} \begin{bmatrix} \hat{\sigma}_u^2 \\ \hat{\sigma}_w^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{N-p} \sum_{n=p}^{N-1} |x_n \star a_n|^2 \\ \frac{1}{N-2p} \sum_{n=2p}^{N-1} |x_n \star a_n \star a_n|^2 \end{bmatrix} \quad (3.2)$$

In an experiment described in Chapter 5, Parzen's method will be investigated and its performance will be compared to the method previously described.

## 3.2 Application of the Kalman Filter for ARMA Processes and Prediction

It was previously stated that the Kalman filter is based on the state-space approach. Therefore it is necessary to introduce this representation for the ARMA process. In this

project, the Kalman filter is used more so for prediction of the output of a stock process, than for state estimation. In this section, the algorithm is presented as a tailored version for one-step prediction. It is also shown how to obtain initial conditions for the algorithm.

### 3.2.1 State-Space ARMA Model

Various state-space models exist for ARMA processes. It is important to choose one that is simple and that conforms to the state-space model that the Kalman filter solution is based on. For example, a state-space model with non-white (time-correlated) noise should be avoided since it would complicate the solution (2.26) which is based on white noise.

The concept of detectability plays an important role in state estimation. A detectable system is preferred, as it is needed for the existence of solutions of the matrix Riccati equation which determines the error covariance matrix.

A discussion on choosing minimal dimensional models is presented in [11, page 23]. Algorithms for filtering, estimation and control are usually stated for models in minimal dimensional representation. These algorithms may break down, or require special handling if applied to non-minimal dimensional representations, because certain matrices may become singular or explode.

The minimal dimensional state-space model for the ARMA process is given by (3.3) for  $p \geq q$ . Recall that  $b_{0,k} = 1$  and that  $u_k \sim N(0, Q_k)$  where  $Q_k = \sigma_u^2$ . A demonstration for the model is found in [9, page 237].

$$\begin{aligned}
x_{k+1} &= \underbrace{\begin{bmatrix} -a_{1,k+1} & 1 & \cdots & 0 & 0 \\ -a_{2,k+2} & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdots & 1 & 0 \\ -a_{p-1,k+p-1} & 0 & \cdots & 0 & 1 \\ -a_{p,k+p} & 0 & \cdots & 0 & 0 \end{bmatrix}}_{F_k} x_k + \underbrace{\begin{bmatrix} b_{1,k+1} - a_{1,k+1} \\ b_{2,k+2} - a_{2,k+2} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}}_{G_k} u_k \quad (3.3a) \\
z_k &= \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}}_H x_k + u_k \quad (3.3b)
\end{aligned}$$

When  $q < p$ , the redundant MA parameters  $b_{q,k+1}, \dots, b_{p,k+p}$  are equal to 0. If  $q > p$ , the order of the auto-regressive branch is augmented with  $a_{p+1,k+p+1} = \dots = a_{q,k+q} = 0$ .

It is noted here that the noise covariance matrices  $Q_k$ ,  $R_k$ , and  $S_k$  are equal since the process noise is identical to the observation noise. When this is the case, the state-space model is often referred to as the *innovations* representation. As well the parameter in (3.3) are time indexed, and the state-space model requires future parameters. Since future values of these parameters are unavailable, current estimates have to be used. If the system does not have strong time-variant properties and if the order of the system is small, the predictions from the model may not be greatly affected by the unavailability.

In some cases, the ARMA signal may be perturbed by further observation noise. If this is the case then the output signal may be written as

$$\begin{aligned}
z_k &= Hx_k + v_k \\
&= Hx_k + u_k + w_k \quad (3.4a)
\end{aligned}$$

where  $w_k \sim N(0, \sigma_w^2)$  is a white noise process uncorrelated with  $u_k$ . So then  $v_k$  has the

statistics  $v_k \sim N(0, R_k)$  where  $R_k = Q_k + \sigma_w^2$ . The state equation remains unchanged and  $S_k = Q_k$ .

### 3.2.2 Application to Prediction

Since prediction of ARMA processes is the main use of the Kalman filter in this project, the computation of the *a posteriori* state estimate may be by-passed. The Kalman filter solution presented here is strictly in terms of one-step prediction.

We first denote the output error covariance as  $L_k$ , the *modified* Kalman gain as  $M_k$ , and the innovation as  $\tilde{z}_k$ . The Kalman filter one-step prediction algorithm given in (3.5) is simplified by skipping the *a posteriori* computations.

$$L_k^{(-)} = H_k P_k^{(-)} H_k^T + R_k \quad (3.5a)$$

$$M_k = (F_k P_k^{(-)} H_k^T + G_k S_k) (L_k^{(-)})^{-1} \quad (3.5b)$$

$$\tilde{z}_k = z_k - H_k \hat{x}_k^{(-)} \quad (3.5c)$$

$$\hat{x}_{k+1}^{(-)} = F_k \hat{x}_k^{(-)} + M_k \tilde{z}_k \quad (3.5d)$$

$$P_{k+1}^{(-)} = F_k P_k^{(-)} F_k^T - M_k L_k^{(-)} M_k^T + G_k Q_k G_k^T \quad (3.5e)$$

Quite often with the innovations representation (3.3), the state of the system is fully known, that is  $P_k^{(-)} = 0$ . In this case, the output covariance  $L_k$  equals  $R_k$  and the gain  $M_k$  equals  $G_k$ . We may take advantage of the fact that  $Q_k = R_k = S_k$ . When all of these are the case the algorithm simplifies somewhat to :

$$\hat{x}_{k+1}^{(-)} = F_k \hat{x}_k^{(-)} + G_k \tilde{z}_k \quad (3.6a)$$

$$\tilde{z}_k = z_k - H_k \hat{x}_k^{(-)} \quad (3.6b)$$

which is equivalent to (3.3).

### 3.2.3 Obtaining Initial Conditions

One may ask how to choose the initial conditions for the Kalman filter if they are not known. Since the ARMA process is WSS with mean 0, one can assume that the initial conditions are zero for the state *i.e.*  $\hat{x}_0^{(-)} = 0$ .

For the error covariance matrix  $P_0^{(-)}$ , the exact initial condition would be the solution of  $P = APA' + BQB'$ . A simpler choice would be to choose the covariance of the output,  $L_0$ , along the diagonal of  $P_0$ . This will make the initial uncertainty larger than it really is, but this is commonly done in Kalman filter applications.

One can also take advantage of the fact that in order to obtain the model for the system, a time-series history is required for input into the modelling algorithm. Since prediction is done on a continuation of this time-series, one may run the Kalman filter through this history to obtain a better initial estimate of the covariance when the predictions begin.

## 3.3 $n$ -step Prediction

In this project  $n$ -step prediction is performed since most people interested in prediction would like to be able to get projections of a process a few hours or days ahead. In this section the SISO and state-space  $n$ -step prediction of ARMA processes is presented. The undifferencing of this, to obtain ARIMA prediction, is also shown.

### 3.3.1 $n$ -step Prediction using the Kalman Filter

When  $n$ -step prediction is performed, it is assumed that no new information of the process will be acquired at the present time for the future. This would be the same as receiving observations of the process  $z_{k+1}, \dots, z_{k+n}$  of the future but with the observation noise statistics  $R_{k+1}, \dots, R_{k+n}$  being infinitely large. When this is the case, the value of this speculative information is zero. This statement is supported by the computation of the modified Kalman gains. Since  $R_{k+i} = \dots = R_{k+n} = \infty$ , then  $L_{k+i} = \dots = L_{k+n} = \infty$ , so then the modified Kalman gains  $M_{k+i} = \dots = M_{k+n} = 0$ . The  $1, 2, \dots, n$ -step prediction is given by equation (3.7).

$$\hat{x}_{k+1}^{(-)} = F_k \hat{x}_k^{(-)} + M_k \tilde{z}_k \quad (3.7a)$$

$$\hat{x}_{k+2}^{(-,2)} = F_{k+1} \hat{x}_{k+1}^{(-)} \quad (3.7b)$$

$\vdots$

$$\hat{x}_{k+n}^{(-,n)} = F_{k+n-1} \hat{x}_{k+n-1}^{(-,n-1)} \quad (3.7c)$$

The associated covariances are given by :

$$P_{k+1}^{(-)} = F_k P_k^{(-)} F_k^T - M_k L_k^{(-)} M_k^T + G_k Q_k G_k^T \quad (3.8a)$$

$$P_{k+2}^{(-,2)} = F_k P_{k+1}^{(-)} F_k^T + G_k Q_k G_k^T \quad (3.8b)$$

$\vdots$

$$P_{k+n}^{(-,n)} = F_{k+n-1} P_{k+n-1}^{(-,n-1)} F_{k+n-1}^T + G_{k+n-1} Q_{k+n-1} G_{k+n-1}^T. \quad (3.8c)$$

For predicting the output of a process, we have :

$$\hat{y}_{k+i} = H\hat{x}_{k+i}^{(-,i)} \quad (3.9a)$$

$$L_{k+i}^{(-,i)} = HP_{k+i}^{(-,i)}H^T + R_{k+i} \quad (3.9b)$$

for  $i = 1, \dots, n$ .

### 3.3.2 $n$ -step SISO (Box-Jenkins) Prediction

Traditionally prediction of ARMA processes has used the Box-Jenkins (SISO) linear method. Initially the prediction uses the past  $p$  outputs of the process for the variables,  $x_i$ , and estimates of noise variables,  $u_i$ , are obtained from past  $q$  prediction errors. As prediction is performed, state variables are shifted a position and replaced by the prediction while the noise variables are shifted and replaced by 0 since  $\mathcal{E}\{u_i\} = 0$ .

For the practical reasons mentioned in the previous section, one may only use the most current estimate of the ARMA model parameters<sup>1</sup>.

$$\hat{x}_{k+1} = -\sum_{i=1}^p \hat{a}_i x_{k-i+1} + \sum_{i=1}^q \hat{b}_i \hat{u}_{k-i+1} \quad (3.10a)$$

$$\hat{x}_{k+2} = -\hat{a}_1 \hat{x}_{k+1} - \sum_{i=2}^p \hat{a}_i x_{k-i+2} + \sum_{i=2}^q \hat{b}_i \hat{u}_{k-i+2} \quad (3.10b)$$

⋮

$$\hat{x}_{k+n} = -\sum_{i=1}^{\min(n,p)} \hat{a}_i \hat{x}_{k+n-i} - \sum_{i=n+1}^p \hat{a}_i x_{k+n-i} + \sum_{i=n+1}^q \hat{b}_i \hat{u}_{k+n-i} \quad (3.10c)$$

---

<sup>1</sup>It should be noted here that  $\hat{\cdot}$  denotes an estimate for parameters whereas  $\hat{\cdot}$  denotes a prediction for the process variables.

One may have noticed that for the 1-step SISO prediction that the noise statistics do not play a role in the prediction. For the Kalman filter 1-step prediction noise statistics do play a role (through the Kalman gain) in the prediction if the state error covariance matrix  $P$  is not zero or if  $Q_k = R_k = S_k$  is not true.

### 3.3.3 Constructing the ARIMA Prediction Through Undifferencing of the ARMA Model

Once the  $n$ -step prediction of the output  $\Delta^d \hat{z}_{n+k}$  in the underlying ARMA process has been done (via the Kalman filter or SISO method), it is used to predict the value of the ARIMA variable  $z_n$ . Recalling equation (2.9), the ARIMA prediction is :

$$\hat{z}_{k+n} = \Delta^d \hat{z}_{k+n} - \sum_{i=1}^{\min(n,d)} (-1)^i \binom{d}{i} \hat{z}_{k+n-i} - \sum_{i=n+1}^d (-1)^i \binom{d}{i} z_{k+n-i} \quad (3.11)$$

## 3.4 Implementation of the RPE Algorithm

In this section, three implementations of the general RPE algorithm are presented. The first implementation is applied to ARMA models in SISO form and the second is applied in state-space innovations representation. The third implementation is for the AR process in white noise. Recall the definitions of the general algorithm's variables in Section 2.4.1<sup>2</sup>.

---

<sup>2</sup>For complete development and discussion of the algorithms, one should refer to the main reference on this subject [6].

### 3.4.1 SISO RPE Algorithm for ARMA Models

In this algorithm,  $y_k$  is the output from the ARMA process,  $\hat{y}_k$  is the prediction of this output, and  $\varepsilon_k$  is the prediction error.  $\bar{\varepsilon}_k$  is termed the residual, and  $\hat{x}_{k+1}$  is the state consisting of the past  $p$  outputs of the system and  $q$  residuals. The variables  $\tilde{y}_k$  and  $\tilde{\varepsilon}_k$  are filtered signals and  $\psi_{k+1}$  is the filtered state.

$$\varepsilon_k := y_k - \hat{y}_k \quad (3.12a)$$

$$R_k := R_{k-1} + \gamma_k [\psi_k \psi_k^T - R_{k-1}] \quad (3.12b)$$

$$\hat{\theta}_k := [\hat{\theta}_{k-1} + \gamma_k R_k^{-1} \psi_k \varepsilon_k]_{\mathcal{M}_S} \quad (3.12c)$$

$$\bar{\varepsilon}_k := \hat{\theta}_k^T \hat{x}_k \quad (3.12d)$$

$$\hat{x}_{k+1}^T := ( -y_k \quad \dots \quad -y_{k-p+1} \quad \bar{\varepsilon}_k \quad \dots \quad \bar{\varepsilon}_{k-q+1} ) \quad (3.12e)$$

$$\hat{y}_{k+1} := \hat{\theta}_k^T \hat{x}_{k+1} \quad (3.12f)$$

$$\tilde{y}_k := y_k - \hat{b}_{1,k} \tilde{y}_{k-1} - \dots - \hat{b}_{q,k} \tilde{y}_{k-q} \quad (3.12g)$$

$$\tilde{\varepsilon}_k := \bar{\varepsilon}_k - \hat{b}_{1,k} \tilde{\varepsilon}_{k-1} - \dots - \hat{b}_{q,k} \tilde{\varepsilon}_{k-q} \quad (3.12h)$$

$$\psi_{k+1}^T := ( -\tilde{y}_k \quad \dots \quad -\tilde{y}_{k-p+1} \quad \tilde{\varepsilon}_k \quad \dots \quad \tilde{\varepsilon}_{k-q+1} ) \quad (3.12i)$$

where

$$\hat{\theta}_k^T = [ \hat{a}_{1,k} \quad \dots \quad \hat{a}_{p,k} \quad \hat{b}_{1,k} \quad \dots \quad \hat{b}_{q,k} ]. \quad (3.13)$$

The subscripting the right hand side of the recursive update of  $\hat{\theta}_k$  denotes that the parameter estimate must lie within a stability region denoted  $\mathcal{M}_S$ . For the ARMA model this means that both the AR and MA branches must have their roots within the unit circle.  $\hat{y}_{k+1}$  is the 1-step SISO predictor incorporated into the algorithm to obtain predictions of

the output of the system.

### 3.4.2 State-Space RPE Algorithm for ARMA Innovations Model

The Kalman filter (3.6) is integrated into this state-space implementation as the predictor of the system output. Here  $F_k$ ,  $G_k$  and  $H_k$  are the matrices from the ARMA state-space model (3.3). Recall that the Kalman gain equals  $G_k$ .

$$\varepsilon_k := y_k - \hat{y}_k \quad (3.14a)$$

$$R_k := R_{k-1} + \gamma_k [\psi_k \psi_k^T - R_{k-1}] \quad (3.14b)$$

$$\hat{\theta}_k := [\hat{\theta}_{k-1} + \gamma_k R_k^{-1} \psi_k \varepsilon_k]_{\mathcal{M}_S} \quad (3.14c)$$

$$\hat{x}_{k+1}^{(-)} := F_k \hat{x}_k^{(-)} + G_k \varepsilon_k \quad (3.14d)$$

$$\hat{y}_{k+1} := H \hat{x}_{k+1}^{(-)} \quad (3.14e)$$

$$W_{k+1} := [F_k - G_k H_k] W_k + \bar{M}_k \quad (3.14f)$$

$$\psi_{k+1} := W_{k+1}^T H_k^T \quad (3.14g)$$

Here the vector  $\hat{\theta}_k$  is the same as in (3.13) and does not include the parameters for the noise covariances. In fact, for the RPE innovations model the covariances are not even present in the Kalman filter.

The variable  $W_k$  is the derivative of the state estimate  $\hat{x}_k^{(-)}$  with respect to  $\theta$ , that is  $W_k = \frac{d}{d\theta} \hat{x}_k^{(-)}$ , and  $\bar{M}_k = \frac{\partial}{\partial \theta} [F(\theta)x + G(\theta)\varepsilon] |_{\theta=\hat{\theta}_k}$ . The gradient of the prediction is also written as  $\psi_k = \left[ \frac{d}{d\theta} \hat{y}_k(\theta) \right]^T$ .

### 3.4.3 RPE Algorithm for AR Model in White Noise

This implementation of the RPE algorithm for AR processes in white noise was derived following the recipe of [6, pp.427-428] for general state-space models. The Kalman filter (3.5) is integrated into this implementation.  $M_k^{kf}$  is the Kalman gain.

$$\varepsilon_k := y_k - \hat{y}_k \quad (3.15a)$$

$$R_k := R_{k-1} + \gamma_k[\psi_k \psi_k^T - R_{k-1}] \quad (3.15b)$$

$$\hat{\theta}_k := [\hat{\theta}_{k-1} + \gamma_k R_k^{-1} \psi_k \varepsilon_k]_{\mathcal{M}_S} \quad (3.15c)$$

$$\hat{x}_{k+1}^{(-)} := F_k \hat{x}_k^{(-)} + M_k^{kf} \varepsilon_k \quad (3.15d)$$

$$\hat{y}_{k+1} := H \hat{x}_{k+1}^{(-)} \quad (3.15e)$$

$$M_k^* := M(\theta) + \mathcal{K}_k \varepsilon_k \quad (3.15f)$$

$$W_{k+1} := [F_k - M_k^{kf} H_k] W_k + M_k^* \quad (3.15g)$$

$$\psi_{k+1} := W_{k+1}^T H_k^T \quad (3.15h)$$

Here  $\mathcal{K}_k^{kf}$  is the gradient of the Kalman gain  $M_k^{kf}$ . The columns of this gradient are defined as  $\mathcal{K}_k^{(i)} = \frac{d}{d\theta_i} M_k^{kf}(\theta)$ . In this AR case, the components of  $\hat{\theta}$  are auto-regressive parameters.  $M(\theta)$  is defined as  $M(\theta) = \frac{\partial}{\partial \theta} F(\theta)x$ . For the implementation in this project, no on-line updates of the noise covariances are performed, they are assumed to be constant.

### 3.4.4 Implementation Issues of the General RPE Method

The general RPE algorithm has a few implementation problems related to numerical memory, computation time, accuracy and stability. All of these problems and their solutions are discussed fully in [6]. A brief description of these are given in the following.

**Inversion of the Hessian** The first problem is the inversion of the Hessian matrix  $R_k$  during update of the parameter vector  $\hat{\theta}_k$ . This is solved by using the *Matrix Inversion Lemma* [18, page 23] which takes advantage of algebraically equivalent representations of equations to reduce the number of operations in computation. The result is a scaled inverse Hessian defined by (3.16). The dynamics of this matrix are given in (3.17).

$$P_k = \gamma_k R_k^{-1} \quad (3.16)$$

$$\lambda_k := \frac{\gamma_{k-1}}{\gamma_k} [1 - \gamma_k] \quad (3.17a)$$

$$S_k := \psi_k^T P_{k-1} \psi_k + \lambda_k \quad (3.17b)$$

$$L_k := P_{k-1} \psi_k S_k^{-1} \quad (3.17c)$$

$$P_k := [I - L_k \psi_k^T] P_{k-1} / \lambda_k \quad (3.17d)$$

The scalar  $\lambda_k$  is termed the *forgetting factor*. The matrix  $S$  is a  $1 \times 1$  matrix making inversion much more efficient. One may notice that the dynamics and structures of (3.17) are analogous to those of the conventional Kalman filter algorithm.

**Matrix Factorization and Regularization** The conventional Kalman filter and its analogous algorithm (3.17) are known to have poor numerical properties. The  $P$  matrices of both algorithms are theoretically symmetric and positive (semi-)definite but in practice suffer a non-symmetric and non-positive definite phenomenon caused from round-off errors. This problem can be relieved in Kalman filtering by applying *Square-root filtering* [21, page 236] which uses a matrix factorization scheme for the dynamics of the state error covariance matrix. The matrix  $P$  is factored as  $P = UDU^T$ , where  $U$  and  $D$  are an upper-triangular

and diagonal matrix respectively. For the RPE algorithm the *Biermann's UD-factorization* algorithm[6, page 329] is used. As well, a method called *Regularization* is also applied within this factorization algorithm to ensure that the Hessian remains positive definite or equivalently that the matrices  $P$  and  $D$  do not take infinite values.

**Stability and Projection** The parameter update equation of the general RPE algorithm contains a monitoring and projection of the parameters into a set of stable predictors. For the RPE's ARMA implementation, this means that the AR and MA branches are monitored for having roots lying within the unit circle. Without this monitoring for stability, the algorithm may explode (in the sense that certain variables may become infinitely large). This problem is solved by projecting the parameter branch estimate into the unit circle. The algorithms used for testing stability and projection can be found in [22, page 153] and [6] respectively.

**Choice of Forgetting Factor** In the previous section a new scalar variable, related to the gain sequence  $\gamma$ , called the forgetting factor  $\lambda_k$  was introduced for the modified algorithm. For time-varying systems the choice of  $\lambda_k$  can influence the RPE algorithm's adaptive performance. There is always a tradeoff between tracking ability and noise sensitivity. A common choice for time-varying systems is to use  $\lambda_k$  constant :

$$\lambda_k \equiv \bar{\lambda} < 1. \quad (3.18)$$

A popular value is  $\bar{\lambda} = 0.99$ . By increasing the value of  $\lambda_k$ , the algorithm is less sensitive to noise but it can be slower in adaptation. A smaller  $\lambda_k$  can result in quicker adaptation but the algorithm will be more sensitive to noise. The forgetting factor can be fine tuned for the process being modeled to obtain improved adaptive parameter estimation.

For time-invariant identification, it is recommended to use

$$\lambda_k = 0.99\lambda_{k-1} + 0.01 \quad \text{with } 0.95 \leq \lambda_0 \leq 1.0. \quad (3.19)$$

### 3.5 An Algorithm for Stock Process Prediction

In Chapter 2, the general concepts used in this project were introduced. In the previous sections of this third chapter, the specific applications of these concepts to the ARMA problem were presented. These concepts can now be put together to construct a complete algorithm for our approach to prediction. The algorithm is broken down into two parts. The first is the off-line modelling of the process and the second is the on-line prediction and adaptive modelling of the process<sup>3</sup>.

#### Off-line Modelling

1. Obtain NS data (history)  $\{z_{-m}, \dots, z_{-1}\}$ .
2. Determine differencing order  $d$  needed to obtain a WSS process.
  - (a) for  $i = 0, \dots, d_{max}$ 
    - i. Perform differencing  $\Delta^i z_n = \Delta^{i-1} z_n - \Delta^{i-1} z_{n-1}$
    - ii. Compute process mean,  $\overline{\Delta^i z_n}$ .
    - iii. Compute  $ACF_i(\Delta^i z_n - \overline{\Delta^i z_n})$ .
  - (b) Analyse ACF's : use differencing strategy to choose  $d$ .
3. Obtain WSS history  $\{x_{-m+d}, \dots, x_{-1}\}$ .
  - (a) Difference NS history  $d$  times.

---

<sup>3</sup>In the algorithm, the term WSS is meant to mean *locally WSS*. The term NS represents *Non-Stationary*. The term history denotes the *history* of a process.

- (b) Subtract WSS history mean.
- 4. Use spectral methods to model WSS process.
  - (a) for  $q = 0, \dots, Q_{max}$ , for  $p = 0, \dots, P_{max}$ 
    - i. Apply MYWE to obtain ARMA( $p, q$ ) model.
    - ii. Apply AMLE to refine ARMA( $p, q$ ) parameters.
  - (b) Use AIC to select model.

### **On-line Prediction and Modelling**

1. Input model into SISO/state-space RPE algorithm.
2. Run RPE through WSS history to initialize algorithm.
3. Perform prediction and update parameters.
  - (a) Obtain observation  $z_n$ , and difference it.
  - (b) Compute prediction error  $\tilde{x}_n$  of ARMA process.
  - (c) Use  $\tilde{x}_n$  to update parameters.
  - (d) Use  $\tilde{x}_n$  to predict ARMA process.
  - (e) Undifference ARMA prediction to obtain ARIMA prediction.
  - (f) Repeat prediction and update.

# Chapter 4

## Experiments Performed on On-Line Data

The contents of this chapter present experimentation performed on on-line (real-time) data. The first experiments attempt to determine sampling frequencies and minimum samples sizes for reliable predictions. Other experiments measure and compare prediction performance of the SISO and state-space approaches to the time-invariant approach. An experiment on  $n$ -step prediction is performed and the final experiment compares the performance of the AIC to selecting higher order models.

### 4.1 Determination of Sampling Frequency

In order to obtain full information from on-line stock processes, it is necessary to determine the sampling frequency needed to obtain full spectral information from the process. This can be done by sampling a process and then finding its power spectral density (PSD) via the periodogram (sample spectrum) or Blackman-Tuckey methods[13]. The cut-off frequency

is found from the PSD, and by using Shannon's sampling theorem, a minimum sampling frequency is determined.

Data was obtained at a frequency of one sample per minute for the IBM stock found on the NYSE on 27 and 30 October, 1995, using a program that queried a server on the Internet. The periodograms of these processes were taken and it was determined that sampling should be performed at a frequency of one sample every 2 minutes or better to obtain full spectral information for this actively traded stock. An example of a PSD from the IBM process is given below.

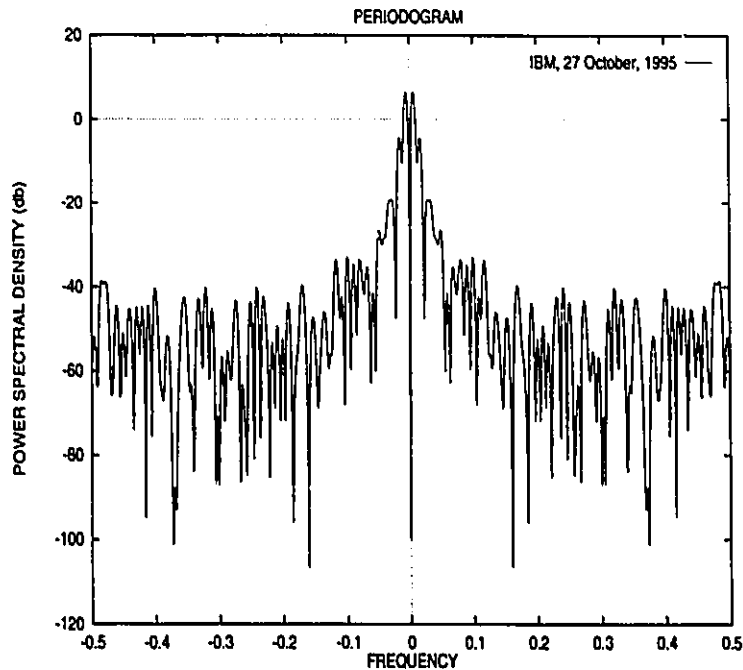


Figure 4.1: PSD of the IBM stock process.

A problem was noticed in the data acquisition program in that it could not obtain a regular sampling rate for many of the stocks when the frequency was at one sample for every two minutes. Many of the data files containing stock samples did not exhibit a regular time difference between samples. The sampling irregularity (delayed response to a query)

was most likely due to a loaded server or large volumes of traffic on the network. When performing time-series analysis, regularly equidistant successive samples for discrete time-series are a necessity. A compromise was made to alleviate this problem. The data files were filtered to obtain time-series with an approximate 5 minute interval between samples<sup>1</sup>.

A problem arises from this compromise, the number of samples that one can obtain in a period becomes more limited. For example, in one day about 70 samples can be made available since it appears that stocks are only traded during the time period from 10AM to 4PM on the NYSE<sup>2</sup>.

## 4.2 Data Samples

The time-series that are used in the remaining experiments of this chapter were obtained using a real-time program that queried a *World Wide Web* (WWW) site on the Internet. The names of the processes that were sampled are listed in Table 4.1.

Stock Name	Period (1995)	Weekdays	Sample Size
IBM	Nov. 7 & 8	Tue & Wed	138
Micron	Oct. 27 & 30	Fri & Mon	114
Microsoft	Oct. 27 & 30	Fri & Mon	160
Netscape	Oct. 27 & 30	Fri & Mon	110
Oracle	Nov. 7 & 8	Tue & Wed	120

Table 4.1: On-line stock processes.

---

<sup>1</sup>The unit of time for on-line data will be one unit per 5 minutes. This is implicitly understood in the graphical results of this section.

<sup>2</sup>It was discovered during the completion of this project that real-time stock data of a 30 second frequency are available from a provider for a fee of 200 US\$ per month.

### 4.3 Determination of Needed Sample Size for Spectral Modelling

In order to obtain reliable predictions, the model representing the process must be accurate. One would expect that from a short history of data, that the identified model would give less accurate predictions than one with more data. However as a longer history becomes available, the model may be influenced by behaviours which should have been discounted. Old data reflect the behaviour of its era (or time period) and not the present era where the WSS assumption holds true.

The goal of this experiment is to find a rule of thumb<sup>3</sup> value representing the length of the history needed in order to properly identify a model for a stock process. No objective methods were found for determining what size a history should be for modelling these types of processes. The criterion used here for determining the history length is to minimize or stabilize the prediction error of the immediate one-step predictions of the process. The RPE algorithm is not used since it is assumed that the prediction interval still lies within the local WSS region, hence adaptive modelling is not needed. This rule or thumb modelling policy will allow us to continue modelling with the other experiments where a proper initial model is needed for on-line adaptive modelling.

The experiment consists of modelling the ARMA process<sup>4</sup> with varying lengths of history from a given base point. For example, for the sample based at 12:00 hrs on a given day, histories of 30, 40, . . . , 80 (previous) observation points are modeled respectively. For each model, 10 one-step predictions are performed. The length of the history, the ARMA model

---

<sup>3</sup>A general principal regarded as roughly correct but not intended to be scientifically accurate.

<sup>4</sup>Off line study on the ACFs of the processes under study suggest that they were already locally WSS, hence differencing was not required. Seasonalities were not noticed.

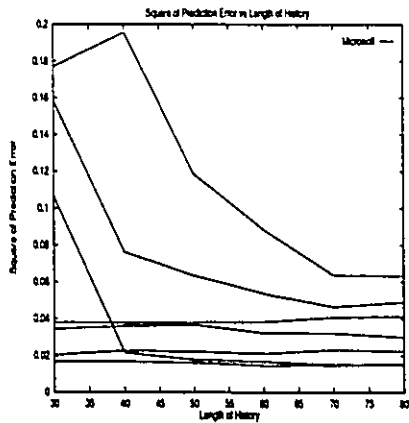
orders, the estimated modelling error, and the average of the squared prediction error are output for each model. Afterwards the process is repeated for samples based at 12:30, 13:00, 13:30, 14:00, etc. until near closing.

A problem encountered in this experiment was the lack of available on-line data. The quantity of sampled processes is limited to a few stocks. As well, for each of these processes the sample sizes are also limited mostly to around 120 samples over 2 days. In many identification procedures, hundreds or even thousands of observation samples are used for modelling. The maximum length of the identification intervals were constrained due to limited sample sizes of the processes. It should be noted that each of these modelling intervals overlap, so the results are not independent within a stock.

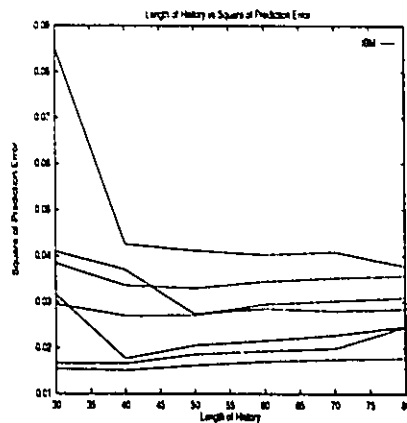
## Results

Experiments were performed on the IBM, Microsoft, Oracle stock value data histories. It was felt that the sample sizes of the other 2 were too short for study. For the Microsoft and IBM stocks, results suggest that as the history gets longer, the modelling process gives better immediate predictions, Figure 4.2. The results from the Oracle stock do not strongly support nor discredit this suggestion. From these results, one may subjectively conclude that as a rule of thumb, a history of about 70 observations is needed for proper modelling.

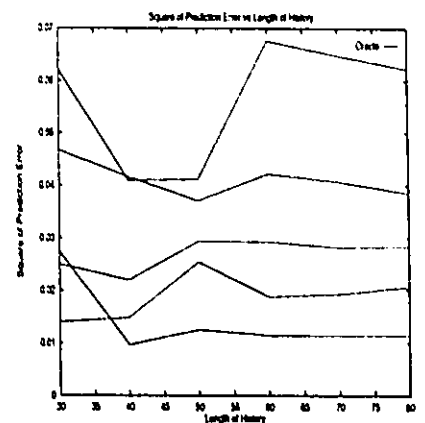
**Note** Information such as modelling errors and model orders  $(p, q)$  chosen by the AIC, were collected during experimentation. These results suggested uncertainty in the choice of models since as modeled history was lengthened, the chosen model order  $(p, q)$  (from the AIC) varied with the histories length. For example, the Microsoft stock was modeled from one given base point. The ARMA models that were identified were of order  $(p, q) = (0, 2), (1, 1), (2, 0)$  with history lengths of 30, 40 and 70. This may just be a problem related to modelling short



(a) Appears to stabilize after a history length of 70 of observations for Microsoft stock.



(b) Appears to stabilize after a history length of 60 of observations for IBM stock.



(c) Does not seem to stabilize for Oracle stock.

Figure 4.2: Mean  $|PE|^2$  vs. History Length.

histories. With longer histories, the model order chosen by the AIC may (or may not) stabilize.

Inspection of the modelling errors also suggest uncertainty in the choice of models. One would expect that as the history of modeled data is lengthened, then the modelling error would be reduced. This was supported for only one of the five stocks being studied.

#### 4.4 Comparing the Performances of the Off-line Time-Invariant to the On-line State-Space and SISO RPE Approaches

In this experiment the performances of the off-line time-invariant approach are compared to the state-space and SISO approaches to adaptive modelling and prediction. The state-space

and SISO approaches or also compared between themselves.

Model identification is performed using a history of 70 observations. One-step predictions were performed throughout the interval immediately following the history used for identification<sup>5</sup>. Performances are evaluated by computing the mean of the absolute value of the prediction errors and the mean of the square of the prediction errors. The experiments were performed on each of the on-line stock data of Table 4.1, p.45. The state-space model (3.3) was used for the state-space RPE algorithm. The value used for the forgetting factor in the RPE algorithm<sup>6</sup> was  $\lambda = 0.99$ . For the predictions using the time-invariant models, the identification that was performed was done entirely off-line. This is different from on-line modelling of time-invariant models using RPE method mentioned in Section 3.4.4, Equation (3.19).

## Results

The results, presented in Table 4.2, suggest that the SISO RPE approach performs slightly better than the state-space RPE approach, although the performances of the two approaches are essentially equal. The performance of the time-invariant SISO and state-space predictors are exactly equal and are thus presented under the same column. The reason for this is explained further on. In 4 of the 5 processes, the RPE predictors have an improved performance over the time-invariant predictors.

The plots for the Microsoft and Netscape stocks' actual value and the RPE predictions are given in Figures 4.3 and 4.5. The two sets of one-step RPE predictions are barely distinguishable. The plots of the parameters are given alongside. The histories of these two

---

<sup>5</sup>The length of this interval ranged between 30 and 70. This length depended upon the availability of data for the stock in question.

<sup>6</sup>It is henceforth assumed that  $\lambda = 0.99$  unless otherwise stated.

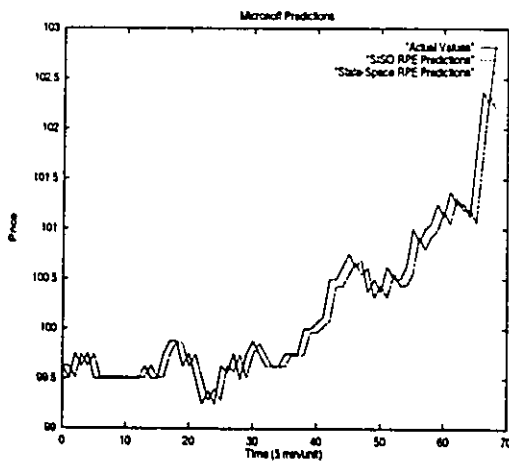
Stock Name	SISO	State-Space	Time-Invariant	
& Model Order	Mean of $PE^2$	Mean of $PE^2$	Mean of $PE^2$	$\hat{\sigma}_x^2$
Mean Price	Mean of $ PE $	Mean of $ PE $	Mean of $ PE $	$\hat{\sigma}_u^2$
IBM (1,0)	0.0409424	0.0411927	0.0431887	0.914362
99.92	0.157879	0.158151	0.162413	0.121421
Micron (1,0)	0.031598	0.0321328	0.0493228	1.25161
70.35	0.132057	0.134005	0.17386	0.101469
Microsoft (1,0)	0.0444518	0.0459948	0.0574223	0.11479
99.6417	0.153007	0.153525	0.164735	0.0190158
Netscape (1,1)	0.474373	0.482817	0.47329	0.28066
83.5734	0.390934	0.392343	0.396194	0.0745232
Oracle (1,0)	0.0456912	0.0470247	0.0531538	0.193072
44.92	0.168585	0.171539	0.186784	0.0557176

Table 4.2: Comparing the off-line time-invariant and on-line time-variant approaches.  $\hat{\sigma}_x^2$  and  $\hat{\sigma}_u^2$  represent estimates of the process and noise variance respectively.

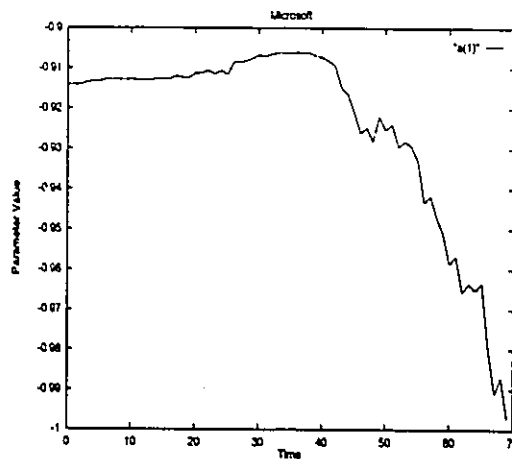
processes are given in Figure 4.4. The prediction interval of the two stocks begin at the corresponding time 70 in the histories. The results should be considered with caution as the prediction intervals are quite short.

The Microsoft predictions show the typical behaviour in predictions of the other AR(1) models. This behaviour being that the predictions are essentially a one step delay of the process or a near random walk. This is explained by the fact that the time-variant parameter  $a_1$  remains inside the interval  $(-1.0, -0.9)$ . From looking at the graph, one can see that the predictions do not give much valuable insight into the movement of the stock.

During experimentation, monitoring of the Kalman filter's state error covariance  $P_k^{(-)}$  was performed and it was noticed that this matrix quickly stabilized to near 0. The reasons for this are likely related to the stability of the error covariance equation. Recall that when the model (3.3) is used with  $P_k^{(-)} = 0$ , then the Kalman filter equations reduce to (3.6) since  $Q_k = R_k = S_k$ . (In this case the RPE algorithm applied to the innovations process (3.14)

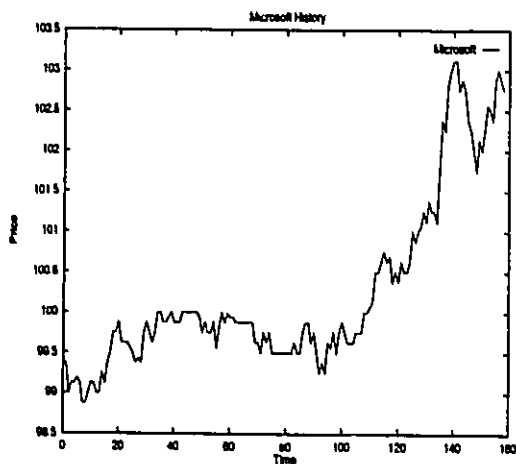


(a) 1-step predictions.

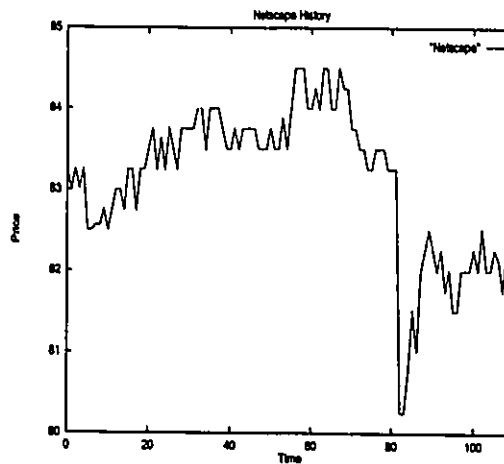


(b)  $\hat{a}_1$  parameter adapting to rising price.

Figure 4.3: Plot of Microsoft 1-step predictions and parameter.



(a) Microsoft history.



(b) Netscape history.

Figure 4.4: Histories of Microsoft and Netscape stocks.

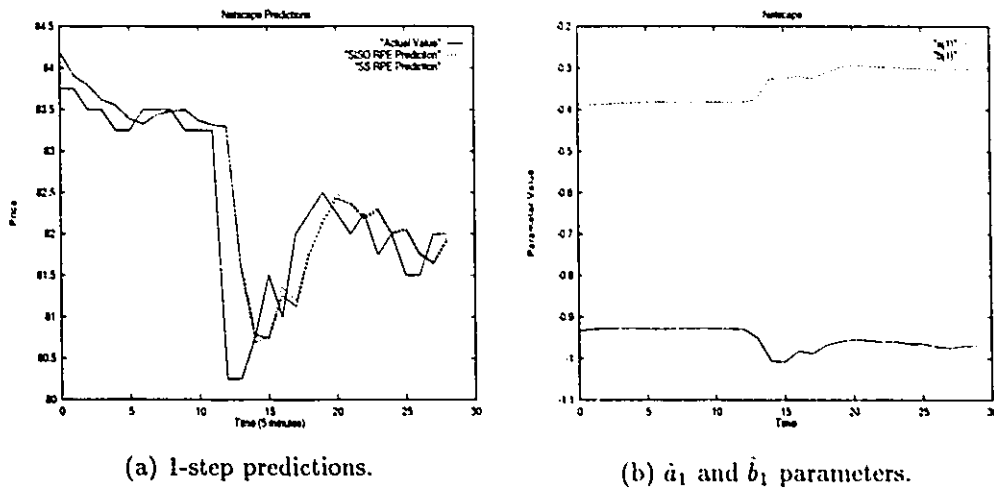


Figure 4.5: Plot of Netscape 1-step predictions and parameters.

can be used and was used.)

When this happens, the Kalman filter solution is equal to equation (3.3) where the Kalman innovations and state estimate replace the noise process and state processes respectively. The noise covariance matrices no longer influence the Kalman gain, so the predictions are not influenced by the noise statistics, hence there should be no advantage in using the Kalman filter over the Box-Jenkins approach. The marginally decreased performance of the RPE's Kalman filter approach is most likely due to the unavailability of the future parameters. In the case of a time-invariant model the SISO and state-space predictors are exactly the same.

## 4.5 $n$ -step Prediction

Another interest in on-line forecasting lies in short term  $n$ -step prediction. Realistically, people interested in trading stocks are not always concerned with predicting a stock 5 or 10 minutes from the present. Short term interests lie mostly in what the stock value will be an

hour or two from the present. The experiments that have been run to date perform one-step predictions only.

In this experiment,  $n$ -step predictions were performed for  $n$  ranging between 1 and 10. A 10-step prediction represents a prediction 50 minutes into the future. The identification was performed using the same parameters and stock histories as in Section 4.4 so the same low order models were identified (see Table 4.2).

## Results

Two graphical results are presented for 10-step predictions of the Microsoft and Netscape stocks in Figure 4.6. With these models the effect of an  $n$ -step prediction is again generally to shift the present value of the stock  $n$  units to the right. The resulting predictions are not able to adapt to the changing state giving no valuable insight into the future of the process.

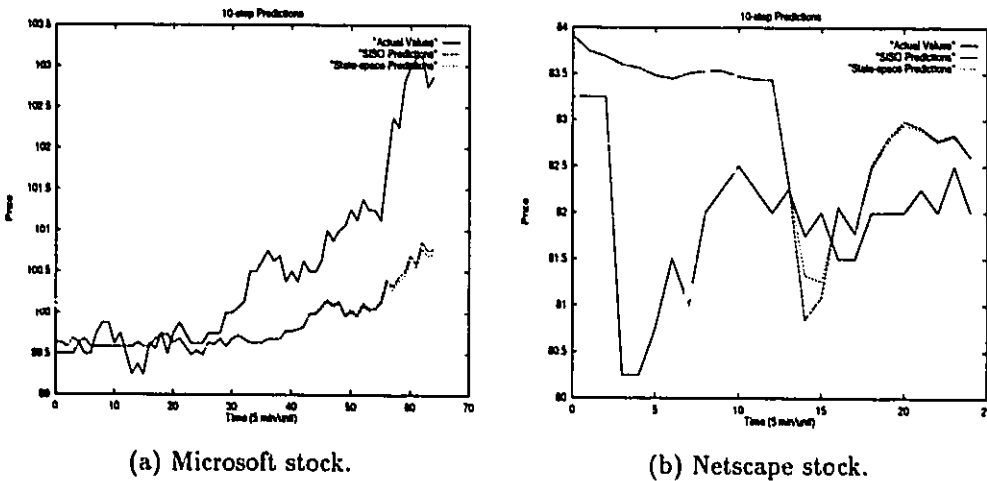


Figure 4.6: 10-step predictions using the SISO and state-space predictors.

## 4.6 Higher Order Models

In the previous experiments, low order models (*i.e.* AR(1), ARMA(1,1) ) were chosen by the AIC as the best candidates for models to represent the stock processes. The effect on predictions of these processes was mainly to shift the general pattern of the history into the future. It was suspected that perhaps the AIC was underparametrizing the models and that higher order models should have been chosen. This experiment attempts to see if higher order models give better prediction results than those chosen by the AIC. One-step SISO predictions are performed using higher order models.

### Results

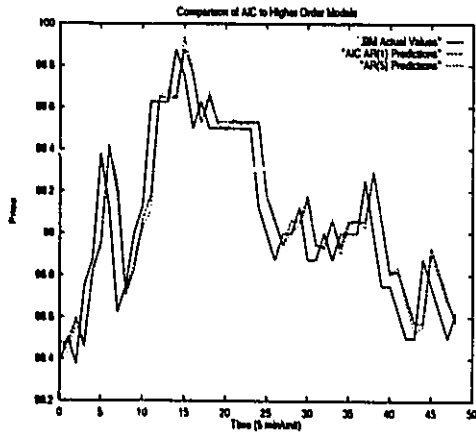
The results from the experiments showed that in general the models chosen by the AIC were appropriate although improvements were noticed for some of the higher order models such as the Microsoft and Micron processes. The performances of higher order AR models were essentially at the same level as the models chosen by the AIC, whereas the performances of high order ARMA and MA models ranged between slightly lower to much lower than the models chosen by the AIC. Examples of results are given for the IBM and Microsoft processes in Tables 4.3 and 4.4. Graphical results displaying SISO predictions for AR(1) and AR(5) models are given in Figure 4.7. Tabular results for the other stocks are deferred to Appendix C.1.

Model Order	AIC (1,0)	(7,0)	(5,0)	(7,2)	(5,2)	(6,3)	(5,5)	(0,7)	(0,5)
Mean $PE^2$	0.040924	0.043358	0.043063	0.061503	0.043524	0.043193	0.092682	0.063032	0.080614
Mean $ PE $	0.15879	0.158029	0.159245	0.197889	0.161045	0.159763	0.243838	0.197031	0.23449

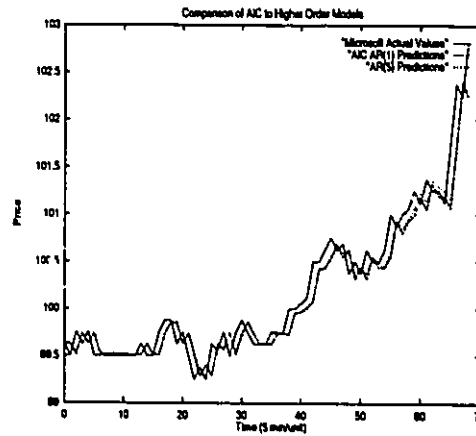
Table 4.3: Results of one-step prediction of IBM process using higher order models.

Model Order	AIC (1,0)	(7,0)	(5,0)	(7,2)	(5,2)	(6,3)	(5,5)	(0,7)	(0,5)
Mean $PE^2$	0.045066	0.041500	0.039016	0.062979	0.063978	0.043193	0.092628	0.063032	0.080614
Mean $ PE $	0.153805	0.151666	0.146155	0.18105	0.199446	0.159763	0.243838	0.197031	0.23449

Table 4.4: Results of one-step prediction of Microsoft process using higher order models.



(a) IBM process.



(b) Microsoft process.

Figure 4.7: One-step SISO predictions of AR(1) and AR(5) models.

**AR Model Poles** The poles of the AR branch were computed for the IBM AR(1) and AR(5) models that were identified using the AMLE spectral algorithm. For the AR(1) model, the pole was found at  $z = 0.933471$ . For the AR(5) model, the 5 poles were found to be at  $z = 0.9344, 0.0037 \pm 0.5995i, -0.6308, 0.6091$ . Both models had near equal dominant poles. For the AR(5) model, the modulus of the largest poles is  $|z_1| = 0.9344$ , whereas as the modulus of the second largest pole is  $|z_2| = 0.6308 < |z_1|^5 = 0.7123$ . From this it is clear that the additional poles of the higher order are insignificant, so a higher order model does not give more insight into the system.

The poles of the Microsoft AR branch were computed as well and the same result came about that there was only one significant pole, although the higher order model produced a better result. Although the higher model seems to perform better from the statistics, by inspecting the graphical results, Figure 4.7(b), one can see that there is no qualitative improvement on the predictions.

## Chapter 5

# Experimentation on Daily Closing Price Processes

In this chapter, experiments applied to prediction of daily closing price stock processes<sup>1</sup> are presented along with results. The experiments from the previous Chapter 4 have been repeated for the daily data. New experiments involving fine tuning of the RPE algorithms and the modelling of the stock processes embedded in noise are also presented.

### 5.1 Determination of Needed Sample Size for Spectral Modelling

This section presents the results of the experiment described in Section 4.3. The goal of this experiment is to find a rule of thumb value representing the length of the history needed to properly identify a model for a stock process in order to obtain reliable predictions. The criteria for evaluation is to find an approximate minimum of the mean of the absolute value

---

<sup>1</sup>From here on in, the term *daily closing price stock process* will be referred to as *daily process*.

of the prediction errors for the 10 immediate predictions following the identification interval. (Beyond this interval, the on-line RPE algorithm is supposed to handle the identification.)

The daily processes that were used in experimentation are presented in Table 5.1 along with the degree of differencing ( $d$ ) required to obtain stationarity, the available sample size, and the resulting approximate sample size needed for proper identification to obtain reliable predictions.

For most of the processes, the observation intervals roughly spanned from 30 August, 1993 to 2 October, 1995<sup>2</sup>. Holidays and trade suspensions were filtered out. The history lengths used for modelling ranged between 30 and 200 observations by increments of 10. The reference points varied by increments of 20. It should be noted that each of these modelling intervals overlap, so these results are not independent within a stock.

paragraphDifferencing For each of the data sets off-line analysis of one interval of data was performed to determine the degree of differencing that was required. It was decided that most of the processes needed one degree of differencing with the exception of the Standard & Poor's index (S&P). The ACFs for only one interval of data were analyzed since work involved in analyzing the ACF for each interval of data used in identification would be extremely time consuming. With an available objective method, such as the Adujusted Dickey-Fuller test (ADF) (page 26), proper ACF analysis would be feasible.

## Results

Results suggest that a history of at least 130 points is necessary to obtain models that provide decent forecasts. When the histories are less than 100 points long, the performances of the immediate forecasts are quite variable. Histories of 130 and longer give generally

---

<sup>2</sup>These data will be reused in the remaining experiments of this chapter.

stable forecasting performances. The performances seemed to degrade only slightly as the identification approached 200 points. Graphical results for the IBM and Standard & Poor's stock process are given in Figure 5.1. Other graphical results are deferred to Appendix D.1.

It was mentioned in the previous chapter for this experiment applied to on-line (real-time) data, that it was noticed that the results suggested uncertainty in the choice of model order. A similar phenomena happened with the experiment on daily data. As the modeled history was lengthened, the AIC's chosen model order  $(p, q)$  varied, but seemed to stabilize (in the sense that the same model order was almost consistently chosen) after a certain length of history, usually around 130 observations. More importantly though, as the base point varied the stabilized model order varied as well. This may suggest that the model orders vary with time.

Stock Name	Apple	Borland	IBM	NewBridge	S&P's	Xerox
$d$	1	1	1	1	0	1
Available	450	440	440	420	400	300
Needed	120	160	120	130	140	130

Table 5.1: *Stock names, differencing order  $d$ , available sample size and approximate sample size needed for identification.*

## 5.2 Comparing the Performances of the Off-line Time-Invariant to the On-line State-Space and SISO RPE Approaches

The experiment presented in Section 4.4 was repeated for the daily closing price data. A new process, the NASDAQ index, was considered in this experiment. It was not considered

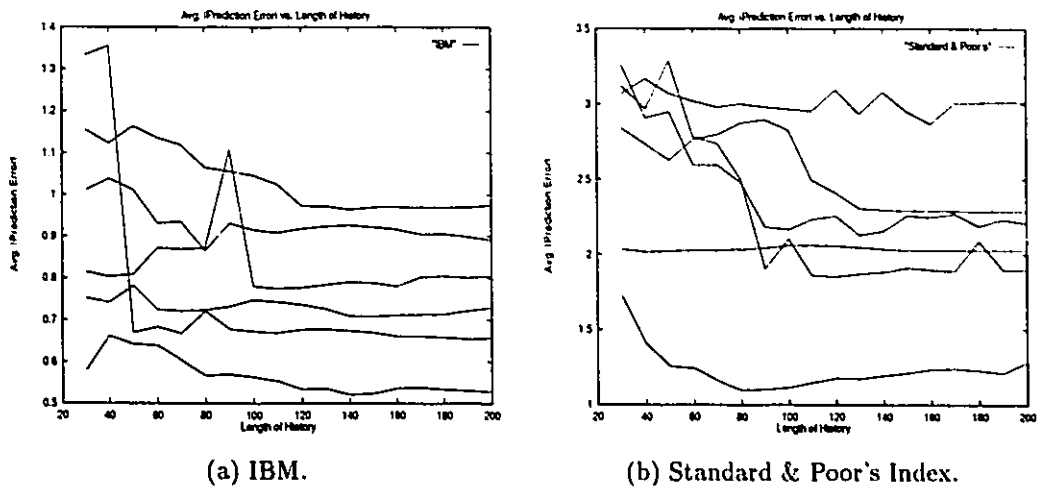


Figure 5.1: Mean  $|PE|$  vs. History length.

in the previous experiment (Section 5.1) as the available history was too short. The history lengths used for identification of initial models were each 130 observations<sup>3</sup>. Predictions were performed on the following interval of 270 points with the exception of the NASDAQ process in which only 70 1-step predictions were performed due to the availability of data. The default value that was used for the forgetting factor of the RPE algorithm was  $\lambda = 0.99$ .

## Results

The results from the experiments are summarized in Table 5.2.  $\hat{\sigma}_x^2$  represents the history process variance,  $\hat{\sigma}_u^2$  represents an estimate of the white noise variance obtained from modelling.

For 4 of 7 stocks, the adaptive modelling approaches had better performance than the time-invariant approach. In particular, the Standard & Poor's, Figure 5.2 and NASDAQ (Figure 5.3) indices appear to perform much better using the RPE based on the results of

<sup>3</sup>From here on, 130 observations will be the default length for the history used for identification, unless otherwise stated.

the prediction error statistics. The NewBridge stock appears as well to have a noticeable improvement using the RPE. For the 4 other stocks, the differences between the time-invariant and the time-variant adaptive approaches appear to be minor. In comparing the SISO and state-space RPE algorithms, one can see that the two approaches were again essentially equal.

In Figure 5.3(a), the predictions of the state-space RPE and time-invariant predictions are compared for the NASDAQ process. One can see that the RPE predictions has the better tracking performance. In Figure 5.3(c), the parameters of the NASDAQ process are plotted. One can see how the AR parameter tends towards -1, as it adapts to the new non-stationary behaviour of the process. In Figure 5.3(b), the history of the process is plotted. One may notice the stationary behaviour of the first 130 observations used in the initial identification. Afterwards the index climbed and the AR parameter adapted to it. Hindsight suggests that the process should have been differenced.

Figure 5.4 plots the state-space predictions for the NewBridge process using the RPE and not using the RPE. Although the results from the table show the RPE modelling performs better, the qualitative information does not tell much more.

Stock Name	Model Order (p,d,q)	SISO RPE	State-Space RPE	Time-Invariant
		Mean of $PE^2$	Mean of $PE^2$	Mean of $PE^2$
$\hat{\sigma}_x^2$	$\hat{\sigma}_u^2$	Mean of $ PE $	Mean of $ PE $	Mean of $ PE $
Apple	(0,1,1)	0.869235	0.869961	0.871526
0.76077	0.74767	0.666991	0.667387	0.668579
Borland	(0,1,1)	0.20985	0.214042	0.208973
0.214184	0.213545	0.282349	0.283489	0.283004
IBM	(0,1,1)	1.35325	1.35477	1.33727
1.02037	1.01235	0.835886	0.83592	0.831389
NewBridge	(0,1,5)	2.1887	2.1865	2.28298
3.73005	3.36162	0.825405	0.827097	0.893631
NASDAQ	(1,0,1)	31.8439	31.0619	52.2551
487.221	27.2962	4.64061	4.6004	6.04612
S&P's	(1,0,0)	8.44748	8.52624	15.22
67.7452	7.54823	2.11042	2.10726	2.87412
Xerox	(1,1,0)	3.25091	3.16477	3.16507
1.91078	1.91062	1.25169	1.23308	1.23313

Table 5.2: Results of experiment comparing the performance of the SISO and state-space approaches.

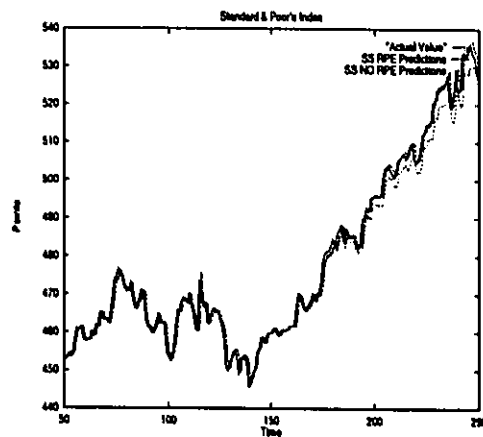
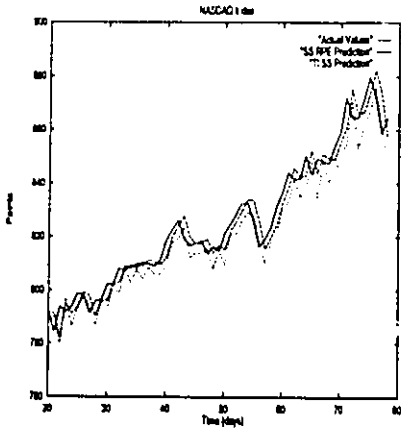
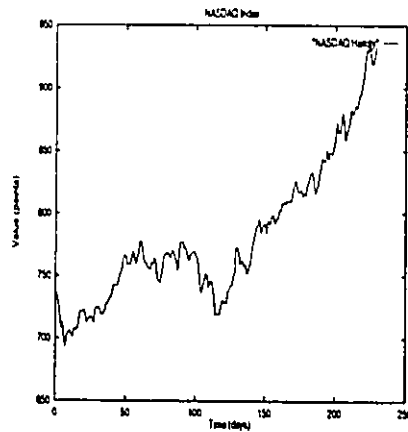


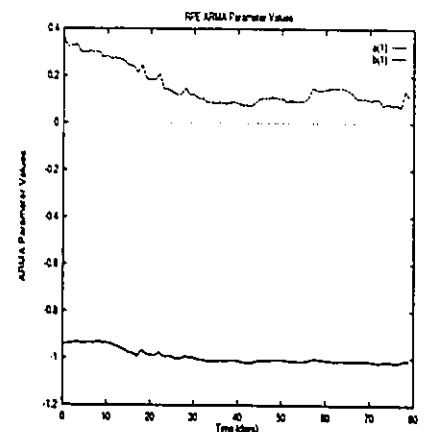
Figure 5.2: A section of the RPE and time-invariant Kalman filter predictions for the Standard & Poor's index.



(a) Section RPE and time-invariant SS predictions.

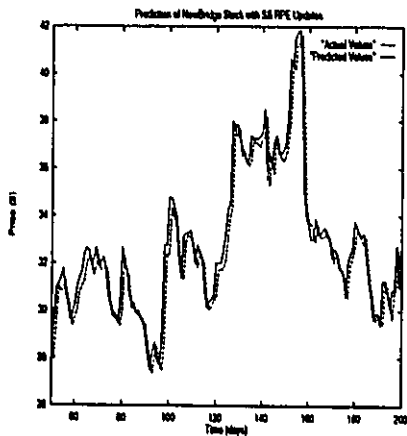


(b) NASDAQ History.

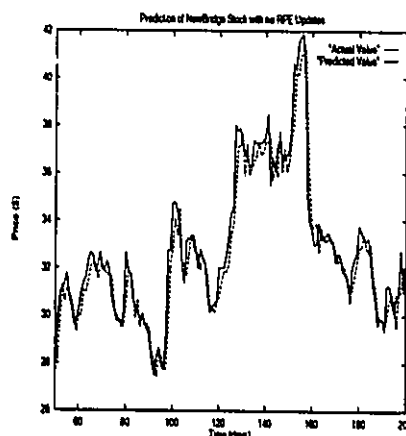


(c) ARMA(1,1) Parameters.

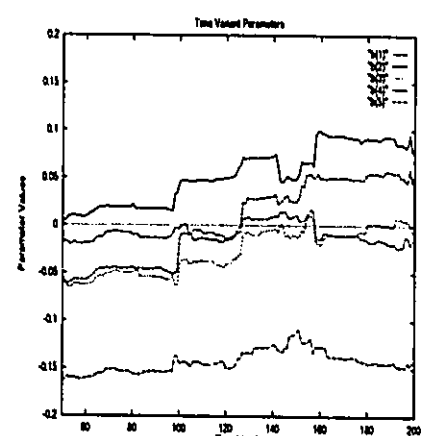
Figure 5.3: NASDAQ ARMA(1,1) predictions, history and RPE parameter estimates.



(a) RPE predictions.



(b) Time-invariant predictions.



(c) RPE time-variant parameters.

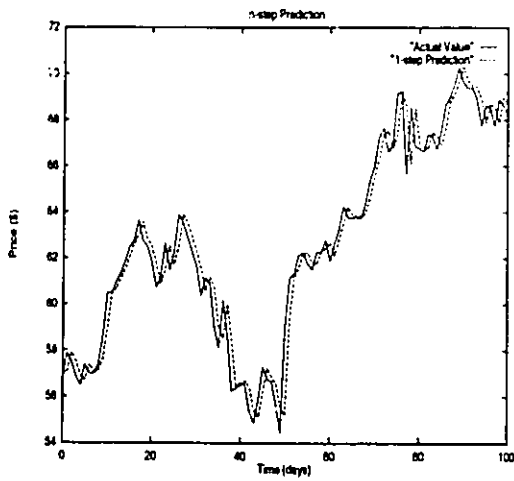
Figure 5.4: NewBridge ARIMA(0,1,5) state-space RPE and time-invariant predictions and estimated parameters.

### 5.3 $n$ -step Prediction

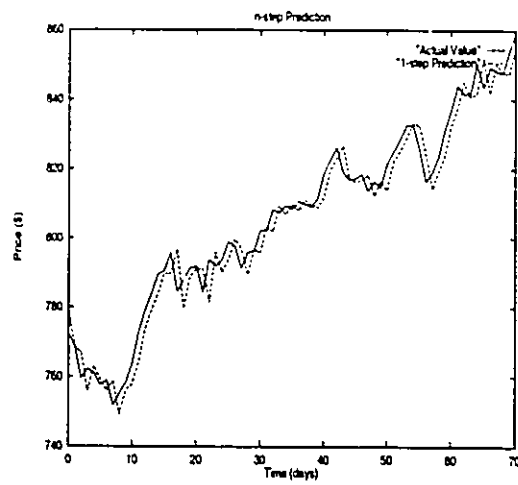
This experiment repeats the one of Section 4.5 for closing price data.  $n$ -step predictions were performed for  $n = 1, \dots, 6$  days. The identification used the same 130 points as the other experiments and on-line adaptive modelling was performed using the RPE.

#### Results

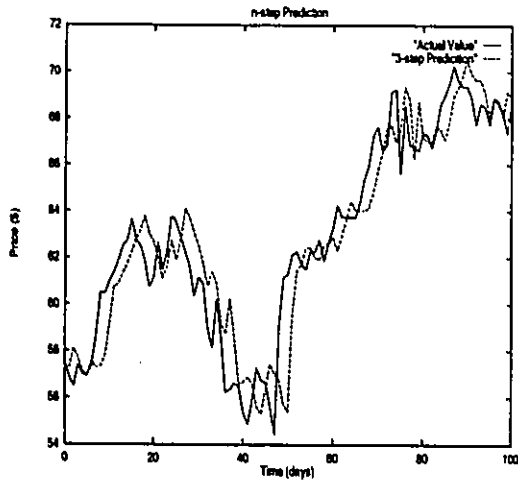
The graphical results of the 1-, 3- and 6-step predictions for the IBM and NASDAQ stock processes are given in Figure 5.3. The predictions are again essentially  $n$ -unit shifts to the right for the process. The resulting predictions are not able to adapt to the changing state, giving no valuable insight into the future of the process.



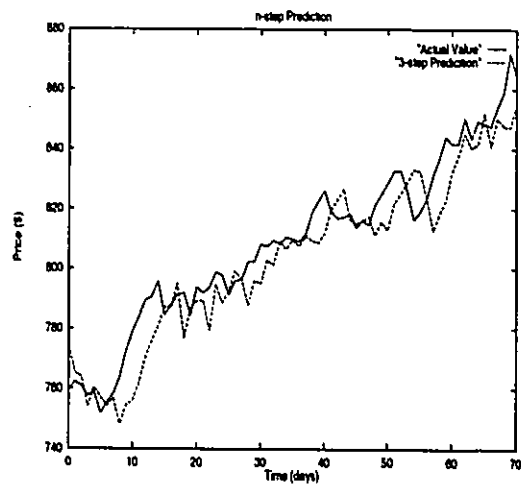
(a) IBM 1-step. ARIMA(0,1,1)



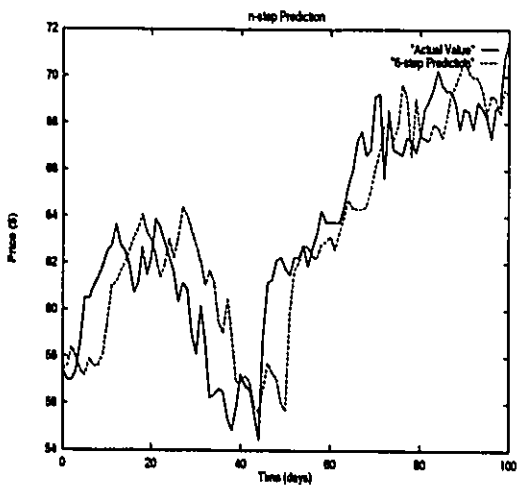
(b) NASDAQ 1-step. ARIMA(1,0,1)



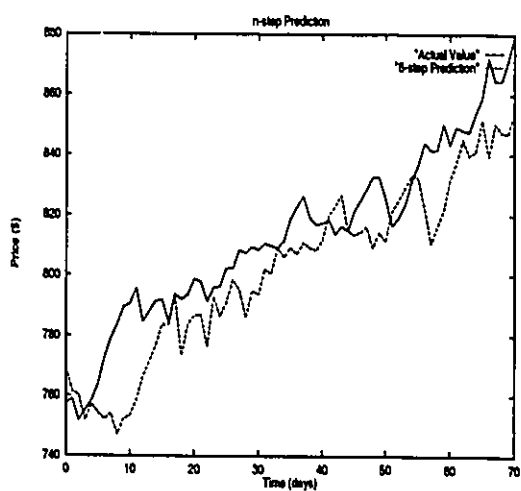
(c) IBM 3-step.



(d) NASDAQ 3-step.



(e) IBM 6-step.



65 (f) NASDAQ 6-step.

Figure 5.5:  $n$ -step predictions for IBM and NASDAQ processes.

## 5.4 Higher Order Models

An experiment similar to that of Section 4.6 was performed. It was suspected that perhaps the AIC is under-parameterizing models, so in this experiment higher order models are chosen to see if they would give a better prediction performance.

### Results

The Apple, Borland and Standard & Poor's processes were studied. Results in tabular form are given in Tables 5.3, 5.4 and 5.5. The higher order models used are listed in these tables.

The predictions from these higher models were not as good as the predictions obtained from models selected by the AIC. Although the higher order models give better modelling errors ( $\sigma_u^2$ ), the prediction error functions (Mean  $PE^2$  and Mean  $|PE|$ ) are generally larger for the higher order models. These results show that the AIC chooses appropriate models.

An example of the graphical results for the Apple process are given Figure 5.6. Predictions of higher order models are barely distinguishable to those of the AIC.

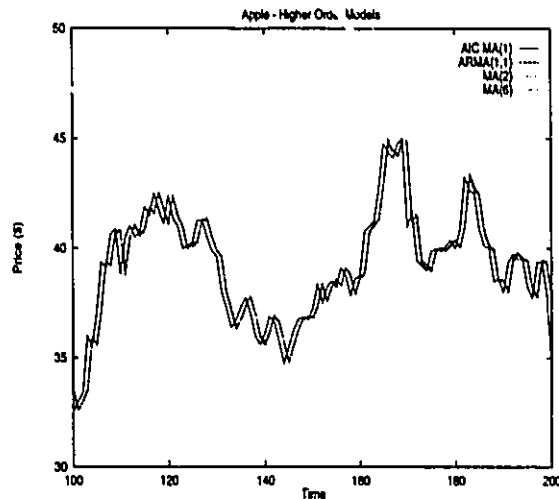


Figure 5.6: Apple. Higher Order Predictions.

(p,d,q)	AIC (0,1,1)	(1,1,1)	(0,1,2)	(0,1,6)
$\sigma_u^2$	0.74767	0.747671	0.747327	0.7391
Mean $PE^2$	0.869235	0.869554	0.878188	0.918815
Mean $ PE $	0.666991	0.66649	0.669958	0.687163

Table 5.3: Higher order models for the Apple process.

(p,d,q)	AIC (0,1,1)	(0,1,2)	(0,1,6)	(0,1,7)
$\sigma_u^2$	0.214184	0.213369	0.210842	0.0.210308
Mean $PE^2$	0.20985	0.211622	0.21477	0.214529
Mean $ PE $	0.282349	0.284194	0.287446	0.285868

Table 5.4: Higher order models for the Borland process.

(p,d,q)	AIC (1,0,0)	(2,0,0)	(6,0,0)	(7,0,0)
$\sigma_u^2$	7.54823	7.54344	7.45797	7.43253
Mean $PE^2$	8.44748	8.69711	8.79885	8.81771
Mean $ PE $	2.11042	2.12534	2.1472	2.13484

Table 5.5: Higher order models for the Standard & Poor's process.

## 5.5 Fine Tuning the Forgetting Factor

So far in all of the experiments that have been performed, the only value of the RPE method's forgetting factor that has been considered was  $\lambda = 0.99$ . This is a common value that is used throughout the reference [6]. Depending on the system that is being studied, this value may be fine tuned to find a value that best suits the system.

In this experiment, fine tuning of the RPE algorithm is performed by adjusting the forgetting factor. The values of  $\lambda$  that were considered varied between 0.940 and 1.000 with increments of 0.005. 130 points were used for off-line identification and the remaining observations for on-line adaptive modelling.

### Results

The results from Table 5.6 show that for most of the processes the optimal value from the set of forgetting factors is 1.0 or very near 1.0. The Standard & Poor's and Xerox processes have minima around 0.98 but from inspection of the graphical results the differences in performance over the values of  $\lambda$  are minimal. Graphical results are found in the following pages and in the Appendix D.2.

	Stock Name	Apple	Borland	IBM	NewBridge	S& P's	Xerox
SISO	Min. $PE^2$ at $\lambda=$	1.0	1.0	1.0	0.995	0.985	1.0
SISO	Min. $ PE $ at $\lambda=$	1.0	0.995	1.0	0.99	0.975	1.0
SS	Min. $PE^2$ at $\lambda=$	1.0	1.0	1.0	0.995	0.985	0.975
SS	Min. $ PE $ at $\lambda=$	1.0	0.99	1.0	0.99	0.98	0.985

Table 5.6: Fine tuning. Optimal  $\lambda$  yielding minima.

A policy of  $\lambda = 1$  can effectively correspond to on-line time-invariant policy of Equation (3.19) for the proper choice of  $\lambda_0$ . Recall that 130 points were already used in the off-line

	Stock Name	Apple	Borland	IBM	NewBridge	S& P's	Xerox
SISO	Avg. $PE^2$	0.86755	0.209107	1.3434	2.19661	8.72352	3.20091
SISO	Avg. $ PE $	0.665641	0.282441	0.832608	0.840137	2.13901	1.24159
SS	Avg. $PE^2$	0.867542	0.209105	1.34329	2.19049	8.77946	3.16485
SS	Avg. $ PE $	0.665643	0.2829	0.832587	0.835917	2.15204	1.23309

Table 5.7: Prediction error statistics at  $\lambda = 1$ .

identification, so then one would have that  $\lambda_{130+} \approx 1$ . These results suggest that these stocks are best modeled by the RPE as time-invariant systems or as systems with mildly time-variant characteristics. Another suggestion from these results is that the RPE algorithm is quite sensitive to the noise of these processes.

Graphical results are presented for the IBM and NewBridge processes in Figures 5.7, 5.8, 5.9, and 5.10. For the IBM process, one may notice that when  $\lambda$  gets closer to 1, the parameters change very little. As  $\lambda$  is furthered from 1, the parameters are more sensitive to noise, Figure 5.8(a). Although the parameters change with  $\lambda$ , the predictions change very little, Figure 5.8(b).

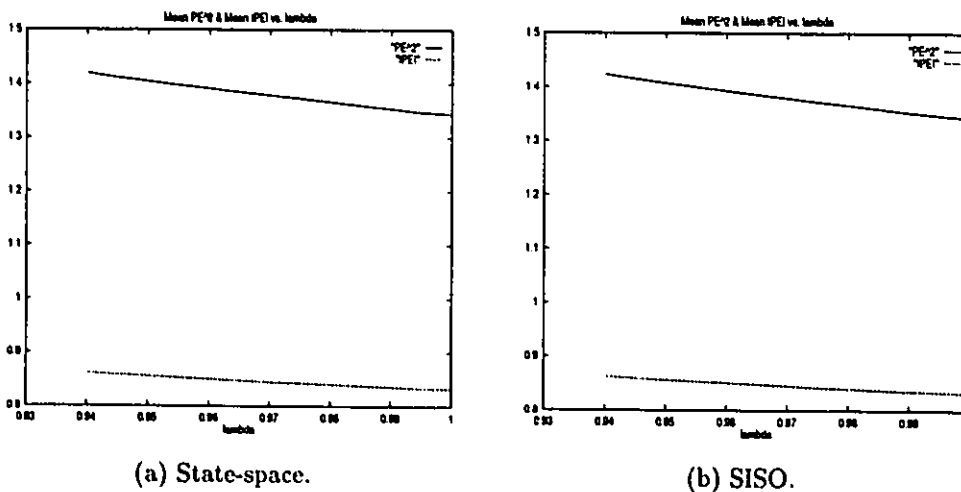
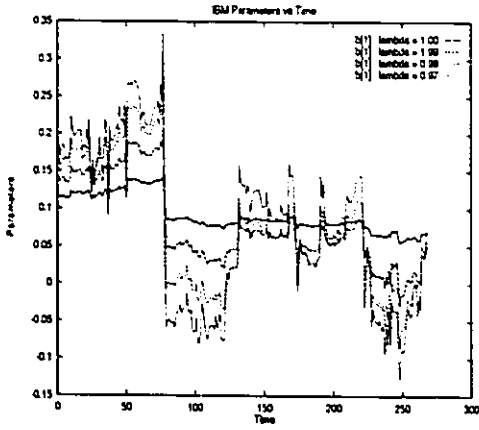
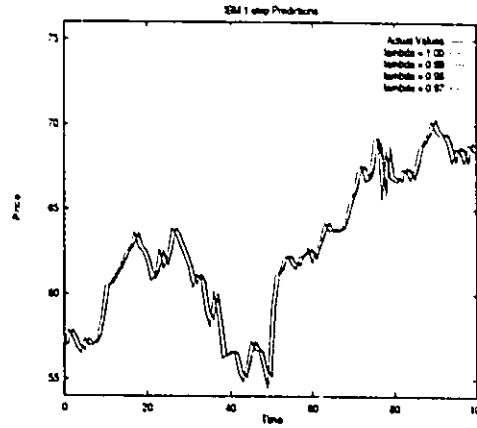


Figure 5.7: Prediction error statistics vs.  $\lambda$  for IBM. Minima at  $\lambda = 1.0$ .

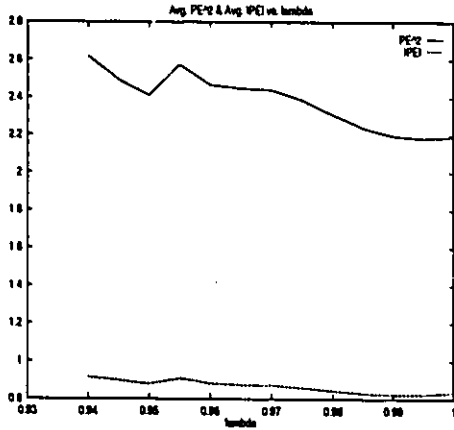


(a) IBM  $b_1$  parameter.

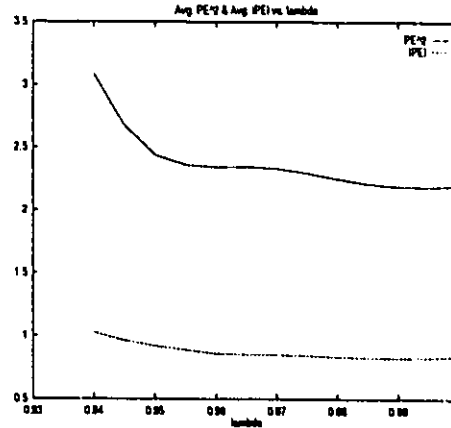


(b) Section of predictions for IBM stock.

Figure 5.8: IBM ARIMA(0,1,1) parameter and predictions with varying  $\lambda$ .



(a) State-space.



(b) SISO.

Figure 5.9: Prediction error statistics vs.  $\lambda$  for NewBridge. Minima at  $\lambda = 0.99$ .

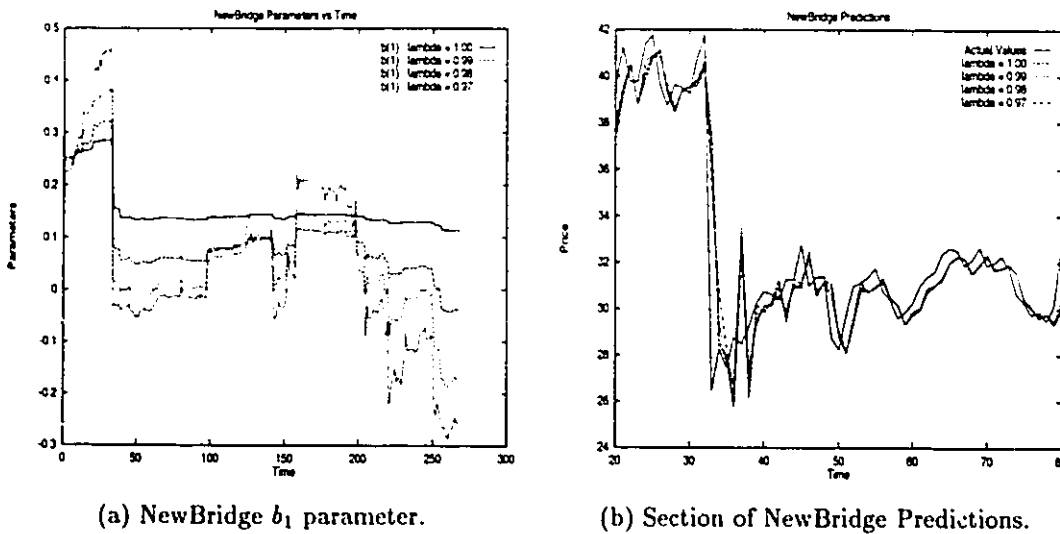


Figure 5.10: NewBridge ARIMA(0,1,5) SISO Parameters and predictions.

## 5.6 AR Modelling in White Noise

Most of the models that were identified for the stocks studied in this project were low order models. (Refer to Table 5.2 on page 62.) For many of the models, the input noise variance estimate,  $\hat{\sigma}_u^2$ , is nearly as large as the process variance,  $\sigma_x^2$ . These large relative values suggest that perhaps the processes are white noise processes<sup>4</sup> or that the processes have observation noise added to them causing a bias towards choosing a lower order model with poor parameter estimates.

It was suggested that perhaps daily closing prices are noisy measurements of the true state (value) of stocks. One can argue that a better measurement of the true value of a stock could be the mean or median trading price for a given day. This argument suggests that perhaps daily closing price processes are embedded in noise.

In this experiment, it is assumed that (after differencing) the stock processes are auto-

---

<sup>4</sup>White noise models were eliminated from consideration since the RPE would be useless in adaptive parameter estimation.

regressive signals in the presence of observation white noise. The MYWE and Parzen's method of page 27 are used for estimating the AR parameters and noise statistics. It is hoped that by using this method, a better model would be identified, and hence the state of the system would be better estimated and the outputs more accurately predicted by the RPE's Kalman filter.

The model orders for  $P$  which are investigated range between 1 and 10. An objective model order selection criteria similar to the AIC exists [23] but it was not well understood, and thus not used.

## Results

The results from the experiment modelling the processes as AR in noise were mostly poorer than the traditional ARMA modelling previously performed. The results for the NewBridge process are presented in Table 5.8 of this section. The NewBridge stock was the only process to yield better predictions. However the reasons for this may lie with the response to the impulse on the system just before Time = 40 in Figure 5.11. This impulse is seen as noise by the MA(5) model and since the model is higher order, the response to this impulse will last for 5 units of time.

It was noticed that Parzen's method can give some odd results. In some cases, one of the two variance estimates was negative, which is a theoretical impossibility. It is not known to this author why this happens, but it may be related to the occurrence of the MYWE not yielding stable auto-regressive branch estimates.

The model orders in boldface represent the best predictions for the stock. The model orders with the '\*' symbol denote these erroneous results<sup>5</sup> in Table 5.8. One may notice that

---

<sup>5</sup>For the other tables included in the appendix, the erroneous results have been removed to economize on space.

in some cases the variance estimates for the noise are larger than the actual variance of the process. One can compare this with Table 5.2 of page 62. As well one may notice that for some of the better performances, that the sum of the variances nearly match the process variance. Other tabular results are presented in Appendix D.3.

P	NORMAL	1*	2	3	4	5*
$\hat{\sigma}_v^2$	3.36162	-4.30768	1.2922	0.192922	3.51079	-3.05547
$\hat{\sigma}_w^2$	0.0	8.02378	1.95951	3.72499	3.18344	5.52271
Mean $PE^2$	2.1956	521.873	2.05615	2.1276	4.64766	106394
Mean $ PE $	0.623072	4.89196	0.796015	0.835764	1.30319	53.3551
P	-	6*	7	8*	9*	10
$\hat{\sigma}_v^2$	-	-3.57448	3.0734	-1.44445	-14.4172	14.6117
$\hat{\sigma}_w^2$	-	5.46104	1.10805	4.34187	5.09878	3.67079
Mean $PE^2$	-	4372.81	3.59751	9856.83	7.2079e+6	36.5182
Mean $ PE $	-	15.6188	1.14628	33.0567	345.072	3.95917

Table 5.8: NewBridge.  $\sigma_x^2 = 3.73005$ . Best result at  $P = 2$ .

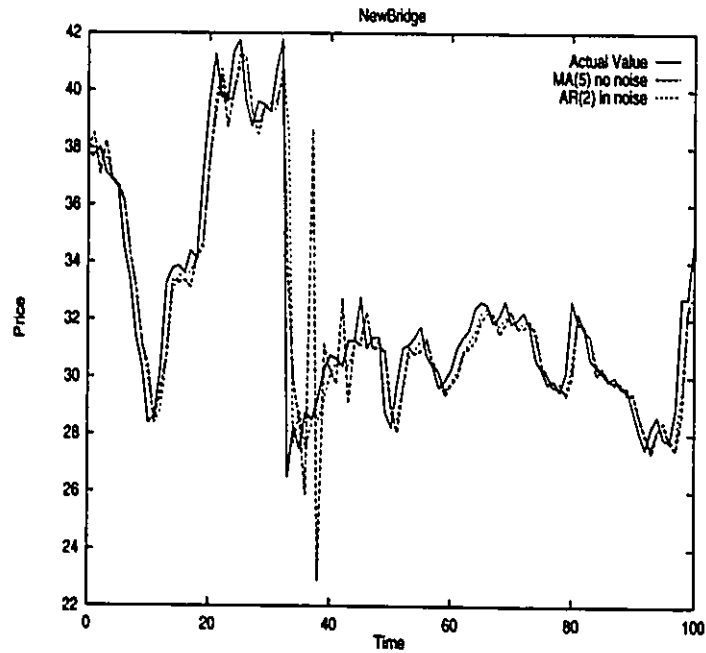


Figure 5.11: Section of the NewBridge predictions. AR(2) in noise, MA(5) no noise,  $d = 1$ .

Although the results from the other tables suggest that the quantitative results are worse,

the qualitative results deserve attention. Many of the best performing AR models in noise were larger order models (compared to modelling ARMA in no noise). The structures of some of the AR predictors in white noise were often different from those of the modelling method absent of observation noise where a shift prediction occurred. The NASDAQ predictions are shown in Figure 5.12.

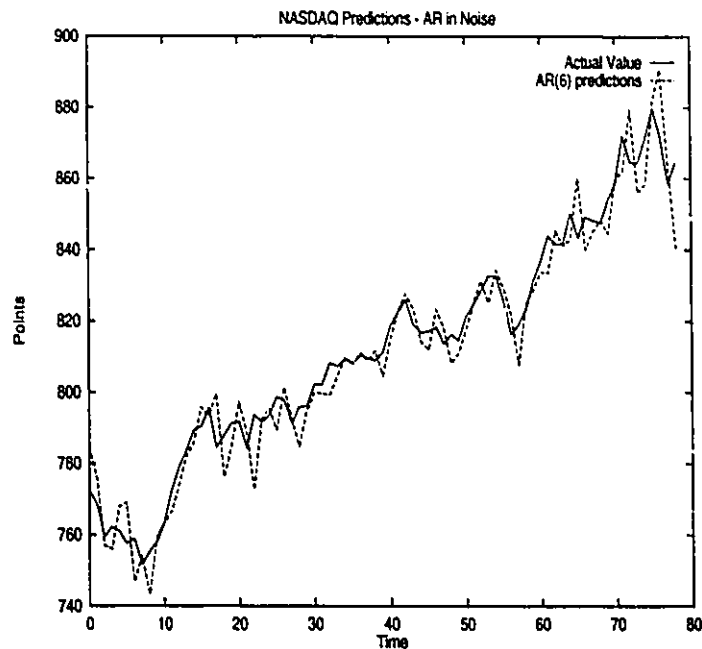


Figure 5.12: Section of the NASDAQ predictions. AR(6) in noise.

# Chapter 6

## Summaries, Extensions & Remarks

In this final chapter, summaries of results, extensions and remarks are discussed.

### 6.1 Summary of Study on Prediction of On-line Processes

In the first experiment, it was found that the sampling frequency for on-line processes should be one sample every two minutes or better. However, problems occurred with the reliability of such a “high” frequency. In the second experiment, it was found that history length of 70 observations was the rule of thumb to use for proper off-line identification of real-time data. In the third experiment, it was found from the tabular results that the RPE method for adaptive modelling appeared to give better results than the experiments with no on-line modelling. While comparing the two RPE predictors the SISO predictor performed marginally better than the Kalman predictor. This was most likely due to the unavailability of future parameters in the state-space representation. However, when inspecting the graphical results, one can conclude that the methods are qualitatively the same. The predictions for each method and model yielded a near random walk phenomena, that is the

predictions were simply a 1-unit shift of the actual value. In the third experiment,  $n$ -step predictions were performed but these were found to be incapable of giving insight into the future of system as the predictions were essentially  $n$ -unit shifts of the actual value to the right. In the fifth experiment, predictions of higher order stock models failed to improve on qualitative performance although it was found that some of these can give marginally better performance measures. The results from this experiment confirmed that the AIC chose proper models.

The results from all of the ARMA predictors were somewhat unsatisfactory. Although the RPE algorithm appeared to adapt the ARMA parameters well, it appears that it is the ARMA models themselves which seem to lack the ability to give insight into the evolution of the processes. Perhaps this is due to the limited data that were used for off-line modelling. Interested parties wishing to continue research on real-time prediction of stocks should obtain generous on-line histories for proper modelling. A general conclusion that can be made would be that the ARMA modelling is not a good approach to model on-line data. One may ask if there exist any models that would give any insight into on-line processes.

## **6.2 Summary of Study on Prediction of Daily Processes**

In the first experiment, it was found that history length of 130 observations was the rule of thumb to use for proper off-line identification of daily closing time processes. In the second experiment which tested the RPE, only 4 of the 7 processes had better performance using the RPE algorithm. Two of these processes (NASDAQ and Standard & Poor's indices) showed noticeable improvements both quantitatively and qualitatively. For the other processes no noticeable differences were noticed. In comparing the two RPE predictors, the SISO and

Kalman predictors were again qualitatively the same. The experiment performing  $n$ -step prediction was repeated and same behaviour of the on-line processes occurred.

The experiment that considered higher order models showed that modelling errors were improved but the predictions from these models were no better and often worse. The results from this experiment again confirmed that the AIC chose proper models.

Fine tuning of the RPE's forgetting factor was performed and the best value for  $\lambda$  was found to be at 1 or very near to 1 for most of the stocks. The results suggested that the processes were best modeled as time-invariant or mildly time-variant or that the RPE was quite sensitive to the noise from the processes.

The results from all of the ARMA predictors were again unsatisfactory. It was noticed that the input noise covariances were often nearly as large as the process variance. To compensate for this, the processes were modeled as auto-regressive signals in the presence of observation noise, but this approach also failed to give improved results. Perhaps if the RPE had incorporated adaptive modelling of the noise covariances, the prediction performances may have been better.

## 6.3 Extensions

This section discusses possible extensions to the approaches used in this thesis.

### AR Signal Modelling in the Presence of Noise

**Other Modelling Methods** The first extension to modelling the AR signal in the presence of noise would be to investigate the other modelling methods described in [13, page 239]. One approach filters the data with a Wiener filter to enhance the AR signal from the noise. The other method, called *noise compensation*, attempts to subtract the extra white

noise variance from the  $0^{th}$  lag of the ACF. To better explain this, suppose  $\{x_k\}$  is the AR process, and  $\{y_k\}$  is the AR process embedded in the noise  $\{w_k\}$  with variance  $\sigma_w^2$ , then  $\{y_k = x_k + w_k\}$ . So then  $r_{yy}[0] = r_{xx}[0] + \sigma_w^2$ . When using an AR estimator on the ACF,  $r_{yy}$ , the parameters will be biased because of the observation noise variance.

**On-line Estimation of Noise Statistics** The second extension to this approach is to perform on-line parameter estimation on the noise covariance statistics of the state-space model. This however demands a larger programming effort since the computations in the RPE algorithm (3.4.3) will become more complex.

**Classical Time-Series Study** A last extension to this approach is to perform a classical off-line time-series study where the process is modeled over a history of observations excluding the final 5 to 10 measurements. Short term predictions are then performed over the immediate 5 to 10 observations following the modeled interval. The method could be tested against other methodologies, such as the Box-Jenkins method or the ARMA model building method proposed in this thesis. The time-series chosen for study should be those chosen from books proposing the Box-Jenkins methodology. A list of these time-series is given in [16, pp. 431-2]. These time-series were not considered in this thesis for on-line modelling because it was felt that they were not lengthy enough for the type of study performed in this project.

### **RPE Algorithm**

In Section 5.5, a suggestion was made that the systems were best modeled as time-invariant or mildly time-variant. One could test this claim with on-line time-invariant modelling using the policy of equation (3.19) for  $\lambda$ .

It was mentioned in the previous section that the RPE algorithm for modelling AR in noise could be improved for on-line estimation of the noise covariance statistics. It could also be improved to be more robust in handling outliers [6, page 270] such as in the NewBridge process in Figure 5.11 of page 73 where it took time for the response to the impulse to die out.

### Performance Measures

In this project the performance measures (average squared prediction error and average absolute prediction error) are rather subjective. The averages over intervals did not discount outliers or did not account for performance on subintervals where abrupt changes occurred. Development of objective criteria are needed for this type of study.

## 6.4 C++ Implementation

The algorithms that were discussed in Chapters 2 and 3 were assembled into a prediction system implemented in the popular *object oriented* programming language C++. This object oriented implementation may become an important contribution to free and open (public and reusable) programming tools similar to *MatClass*<sup>1</sup> and *Kalman*<sup>2</sup> which are available by FTP via Internet sites. C++ is a popular programming language for public code, since the language is very portable in that code written solely in C++, involves no machine code or dependencies and compiles correctly with most PC and workstation compilers.

---

<sup>1</sup> Available from *uts.mcc.ac.uk* (130.88.200.3)

<sup>2</sup> Available from *//usc.edu/pub/C-numanal/kalman.tar.gz*

### 6.4.1 Contribution to Public Programming Tools

In this thesis project, classes defining and implementing the Kalman filter, general RPE algorithm and spectral estimation methods were established. The code is extensible in the sense that other spectral estimation algorithms may be implemented rather easily by inheritance of this base class. As well class structures have been implemented for the RPE algorithms seen in this report. The Kalman filter class was initially borrowed from an FTP site on the Internet.

Upon being placed in public domain (*i.e.* when made available on the Internet), the code will be subject to peer review and open to third party improvement and extension, unlike commercial software. Along with other time-series modelling code, this base class could potentially become a contribution to a free and available time-series modelling package similar to the commercially available statistical package SAS (Statistical Analysis System).

### 6.4.2 Class Implementation for ARMA Spectral Methods

Classes defining and implementing the spectral estimation methods were established with complete structures as autocorrelation and cross-correlation functions and matrices, the reflection (partial correlation) coefficients for AR processes, objective model selection criteria, prediction error power and parameter branches and parameter sets. The following paragraphs describe the class definitions of the spectral algorithms.

**Class Implementation for AIC** A class for the Akaike Information Criterion was defined. Structures such as the maximum and optimal AR and MA orders and sets of variance estimates and AIC function values are defined. The AIC function is also implemented in this class for AR, MA and ARMA.

**Class Implementation for Cross-Correlation Function** A class defining and implementing the cross-correlation function for a time-series was developed. This class has cross-correlation matrices and vectors as structures. The default mode accepts a single time-series and implements the auto-correlation function.

**Class Implementation for AR Spectral Methods** A general class for AR methods was developed. The default algorithm for estimating AR parameters is the Levinson algorithm. Structures such as a parameter estimate set, input time-series and size, optimal model order, reflection coefficients are defined. Other AR parameter estimation algorithms such as Burg's method, Marple's least-square and the Recursive MLE method were implemented by inheriting this class. This class inherits the AIC and uses it to select the optimal model order. The class also inherits the cross(auto)-correlation function class as the ACF is often used by spectral algorithms. A similar MA class was defined for the Durbin algorithm.

**Class Implementation for ARMA Spectral Methods** A general class for ARMA methods was also developed. This class inherited the AR and MA classes and used their structures. The Akaike MLE, MYWE and Least-Squares MYWE algorithms were implemented by inheriting this class.

### **6.4.3 Class Implementation for Kalman Filter and RPE Algorithms**

Classes defining and implementing the RPE algorithms were established. These include the general RPE algorithm, its UD-decomposition, and state-space implementations. The following paragraphs describe the classes that were defined to create a general state-space implementation of the RPE algorithm.

**General RPE Algorithm** A class for the general RPE algorithm (2.29) was defined and established using mainly the error, vector and matrix classes obtained from the internet. All of the structures and equations of this algorithm are defined or implemented.

**UD-Decomposition of RPE Algorithm** A class implementing the UD-decomposition (Section 3.4.4) was created by inheriting the general RPE algorithm and by redefining the evolution of the  $P$  matrix of the algorithm via the UD-decomposition.

**State-Space Class** A state-space class was written implementing the state-space model (vectors and, coupling and covariance matrix) described by equations (2.19) and (2.21). These were defined and established using mainly the error, vector and matrix classes obtained from the internet.

**Kalman Filter Class** A Kalman filter class originally obtained from the internet was remodelled for one-step prediction (Algorithm (3.5)). The class was implemented by inheriting the state-space class and by defining the new Kalman filter structures such as the innovations, output estimates, Kalman gains and state-covariance matrices. The state ( $x$ ) from the state-space class was implicitly used as the state estimate ( $\hat{x}$ ).

**State-Space RPE Class** A class for the state-space RPE algorithm was implemented by first defining classes of Kalman structure derivatives w.r.t to the parameter set  $\theta$ . This new class along with the Kalman and UD-RPE classes were used to create the state-space RPE algorithm class.

## 6.5 Contributions

This thesis has performed a first experimental study using a recursive parameter estimation and state-space Kalman filter approach to modelling of on-line data and prediction of on-line (real-time) and daily closing price stock market processes.

The traditional SISO (Box-Jenkins) and the state-space Kalman filter approaches were compared and it was found that for the conventional ARMA models that these two predictors were qualitatively and essentially the same. For the state-space approach, the processes are also modeled as auto-regressive processes embedded in noise. This was an ideal Kalman filter application but it was found that this approach did not give any general improvement in prediction.

Recursive parameter estimation was used via the RPE algorithm. It was shown that the on-line parameter estimation was useful in that improvements in predictions were obtained while using the algorithm. Fine-tuning of the RPE algorithm was also performed and it was found that prediction performance was better as the forgetting factor  $\lambda$  approached 1.

## 6.6 Closing Remarks

**White Noise and Random Walk Models of Stock Processes.** The stocks that were modeled in this project showed two main classes of behaviour. The on-line stocks were modeled mostly as AR(1) model and showed a random walk behaviour. Many of the daily stocks (excluding the NASDAQ and Standard & Poor's indices) were modeled as MA processes with driving noise variance estimates close to the process variance. This suggests that these were nearly white noise models. In [4], it was stated the several mathematical models have been developed for stock market prediction but the results were unsatisfactory. This was part the

authors' motivation in [4] for pursuing the neural network approach and their statement is now better understood.

**Kalman Filter** The state-space Kalman filter approach to prediction was one of the two main issues of this thesis. The results obtained from the Kalman filter were disappointing. In particular using the AR model embedded in noise was disappointing since this was an ideal study for the Kalman filter.

The Kalman filter is not to blame as it has had hundreds of successful applications in various fields. I believe that failure in predicting the stock processes in this project lies with the linear and Gaussian modelling assumptions of these processes. It is quite clear, from inspection of the histories that the on-line processes are not Gaussian, simply by the common plateaus that occur. This occurrence however is not so bad with the daily processes. As for the linear ARMA models, it was quite apparent from the shift predictions that these models failed to give any insight into the processes that were studied in this project. The ARMA stock models are quite simplified. For example only the stock price was considered as a variable, whereas volume and measures can influence this price. Research efforts should be considered in replacing the Kalman filter by its lesser known analog for non-linear processes, the Zakai equation problem.

The references [10, 11], lay down the framework for the Kalman filter to show its potential with economic time-series in the prediction of vector time-series (as opposed to the scalar approach performed here) and the state estimation where the components of the system are not directly observable or if its measurements are uncertain or noisy. For example, the Kalman filter should be successful at predicting economic indices which are correlated with other indices (when the structural models are well known). A second example is the prediction and estimation of indices such as unemployment where measurements are often

uncertain or unavailable.

# Appendix A

## Brief Review of Time Series and Stochastic Processes

The purpose of this section, is to introduce or review some basic definitions from time series and stochastic processes.

### A.1 Discrete Time Series

A *discrete time series* (DTS) is a discrete set of observations generated sequentially in time. The observations of a DTS made at times  $\tau_0, \tau_1, \dots, \tau_n, \dots, \tau_N$  may be denoted by  $z(\tau_0), z(\tau_1), \dots, z(\tau_n), \dots, z(\tau_N) \in R^p$ . Often  $p = 1$ . We will consider only DTS where the observations are made at some fixed interval  $h$ . When  $N + 1$  successive values of a DTS are available for analysis, we write  $z_0, z_1, \dots, z_n, \dots, z_N$  to denote observations made at equidistant time intervals  $\tau_0, \tau_0 + h, \dots, \tau_0 + nh, \dots, \tau_0 + Nh$ . We denote  $Z_k$  as the observation sequence  $z_0, z_1, \dots, z_k$ . We denote  $\{z_k\}$  as the observation process which may be a vector. For many purposes the values of  $\tau_0$  and  $h$  are unimportant, but if the observation times need

to be defined exactly, these two values can be specified. If we adopt  $\tau_0$  as the origin and  $h$  as the unit of time, we can regard  $z_n$  as the observation at time  $n$  [1, page 23]. From here on in the term *discrete time-series* will be referred as *time series*.

## A.2 Discrete Stochastic Processes

A statistical phenomenon that evolves in time according to probabilistic laws is called a *stochastic process*, commonly referred to simply as a *process*. A *discrete* stochastic process  $\{x_n\}$  is a sequence of random variables (from a time series), defined for all integers  $n$  [13, page 51]. Here  $x_n \in R^p$ . Stochastic processes are often referred to as *random processes*.

The time series to be analysed may then be thought of as a particular *realization*, produced by the underlying probability mechanism, of the system under study. In other words, in analyzing a time series we regard it as a realization of a stochastic process.

## A.3 Wide Sense Stationary Processes

A stochastic process is called *wide sense stationary* (WSS) if, throughout the process, its mean is constant and its autocorrelation depends on the lag within the sample. That is

$$\mathcal{E}[x_n] = \mu_x \quad (\text{A.1a})$$

$$r_{xx}[k] = \mathcal{E}[x_n^* x_{n+k}]. \quad (\text{A.1b})$$

Here  $\mathcal{E}$  is the expectation operator and  $x_n^*$  is the conjugate transpose of  $x_n$ . If  $x_n$  is real

then  $x_n^* = x_n^{T1}$ . The *autocorrelation function* (ACF)  $r_{xx}$ , defined by equation (A.1), does not depend on the time of the process [24, page 298].

It should be noted here that when working with limited sampled data, an approximation to the ACF may only be obtained, which we denote as  $\hat{r}_{xx}[k]$ . An algorithm to compute an estimate of the ACF, is given in [13, page 102] and [25].

## A.4 Power Spectral Density

The *Power Spectral Density* (PSD) of a WSS random process  $\{x_n\}$  is given in terms of its ACF (A.1) and is given in the following :

$$P_{xx}(f) \triangleq \sum_{k=-\infty}^{\infty} r_{xx}[k] \exp(j2\pi fk), \quad -0.5 \leq f \leq 0.5. \quad (\text{A.2})$$

The frequency  $f$  may either be thought of as the fraction of the sampling frequency used in obtaining the data samples from a continuous random process or as the number of cycles/sample [13]. The PSD function describes the distribution of power with frequency of the WSS random process.

Relationships between the PSD and the parameters of an ARMA model can be drawn via the ACF of the process. The relationships are presented in Section 2.2.

## A.5 Gaussian Processes

A stochastic process  $\{x_n\}$  is termed *Gaussian*, if for the times  $k_1, \dots, k_n$ , the random variables  $x_{k_1}, \dots, x_{k_n}$  are jointly Gaussian, that is :

---

<sup>1</sup>Throughout this report, it is assumed that all processes are real.

$$p(x) = \frac{1}{(2\pi)^{n/2} |\Sigma_x|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_x)^T \Sigma_x^{-1} (x - \mu_x)\right] \quad (\text{A.3})$$

where the vector  $X = [x_k, \dots, x_{k_n}]^T$  has mean and covariance

$$\mathcal{E}[X] = \mu_x \quad (\text{A.4a})$$

$$\mathcal{E}\{(X - \mu_x)[X - \mu_x]^T\} = \Sigma_x. \quad (\text{A.4b})$$

The notation  $X \sim N(\mu_x, \Sigma_x)$  is commonly used to denote that  $X$  has a Gaussian distribution with mean  $\mu_x$  and covariance  $\Sigma_x$ .

## A.6 White Noise Processes

*White noise* processes  $\{v_k\}$  are usually zero mean, Gaussian processes with covariance  $\mathcal{E}[v_k v_l] = C_k \delta_{kl}$  [9, page 320], where the discrete time Kronecker delta function is defined as in equation (A.5).

$$\delta_{kl} \triangleq \begin{cases} 0 & k \neq l \\ 1 & k = l \end{cases} \quad (\text{A.5})$$

# Appendix B

## Spectral Algorithms

### B.1 MYWE Algorithm

The MYWE algorithm for solving AR parameters is given below. Recall that  $a_k[0] = 1$  for all  $k$ . The  $K$  parameters represent the *reflection coefficients* [13] of the ARMA process which are sometimes referred to as the *partial auto-correlation function*.  $\rho$  represents a variance estimate of the driving noise or the process which is better estimated once the MA parameters are estimated from Durbin's method.

#### Algorithm

$$\rho_0 = r_{xx}[0]$$

for  $k = 1, \dots, p$

$$a_k[k] := -\sum_{i=0}^{k-1} r_{xx}[q+k-i]a_{k-1}[i]/\rho_{k-1}$$

$$a_k[i] := a_{k-1}[i] + a_k[k]K_{k-1}[k-i] \quad (\text{for } i = 1, \dots, k-1)$$

$$K_k[k] := -\sum_{i=0}^{k-1} r_{xx}[q-k+i]K_{k-1}[i]/\rho_{k-1}$$

$$K_k[i] := K_{k-1}[i] + K_k[k]a_{k-1}[k-i] \quad (\text{for } i = 1, \dots, k-1)$$

$$\rho_k := (1 - a_k[k]K_k[k])\rho_{k-1}$$

When  $q = 0$ , the MYWE algorithm reduces to the Levinson algorithm, a popular method for parameter estimation in AR models.

## B.2 Burg Algorithm

The processes  $\hat{e}_k^f[n]$  and  $\hat{e}_k^b[n]$  are termed the forward and backward predictions respectively. This constrained least-squares minimization method [14, page 1392] does not yield a global minimum for the prediction error power which is estimated as the average of the forward and backward prediction powers,  $\hat{\rho} = (\hat{\rho}^f + \hat{\rho}^b)/2$ .

### Algorithm

$$\rho_0 := r_{xx}[0]$$

$$\hat{e}_0^f[n] := x[n] \quad (\text{for } n = 1, \dots, N-1)$$

$$\hat{e}_0^b[n] := x[n] \quad (\text{for } n = 0, \dots, N-2)$$

for  $k = 1, \dots, p$

$$K_k := \{-2 \sum_{n=k}^{N-1} \hat{e}_{k-1}^f[n] \hat{e}_{k-1}^b[n-1]\} / \{\sum_{n=k}^{N-1} \hat{e}_{k-1}^f[n]^2 + \hat{e}_{k-1}^b[n-1]^2\}$$

$$a_k[k] := K_k$$

$$\rho_k := (1 - |K_k|^2) \rho_{k-1}$$

$$a_k[i] := a_{k-1}[i] + K_k a_{k-1}[k-i] \quad (\text{for } i = 1, \dots, k-1)$$

$$\hat{e}_k^f[n] = \hat{e}_{k-1}^f[n] + \hat{K}_k \hat{e}_{k-1}^b[n-1] \quad (\text{for } n = k+1, \dots, N-1)$$

$$\hat{e}_k^b[n] = \hat{e}_{k-1}^b[n-1] + \hat{K}_k \hat{e}_{k-1}^f[n] \quad (\text{for } n = k, \dots, N-2)$$

# Appendix C

## Results From Experimentation on On-line Data

### C.1 Higher Order Models

Model Order	(1,0)(AIC)	(7,0)	(5,0)	(7,2)	(5,2)	(6,3)	(5,5)	(0,7)	(0,5)
Avg. $PE^2$	0.031598	0.0286183	0.0290221	0.0805264	0.0299685	0.0307505	0.113205	0.802206	0.17090941
Avg. $ PE $	0.132057	0.119944	0.121349	0.235699	0.127064	0.132256	0.278074	0.877174	0.357318

Table C.1: 1-step prediction of Micron process using higher order models.

Model Order	(1,0)(AIC)	(7,0)	(5,0)	(7,2)	(5,2)	(6,3)	(5,5)	(0,7)	(0,5)
Avg. $PE^2$	0.474356	0.522662	0.50189	0.545339	0.980285	NaN	4.8482	0.586593	0.512886
Avg. $ PE $	0.390934	0.411341	0.412928	0.43967	0.647408	NaN	1.71025	0.477282	0.443479

Table C.2: 1-step prediction of Netscape process using higher order models.

# Appendix D

## Results From Experimentation on Daily Data

### D.1 Determination of Sample Size for Modelling

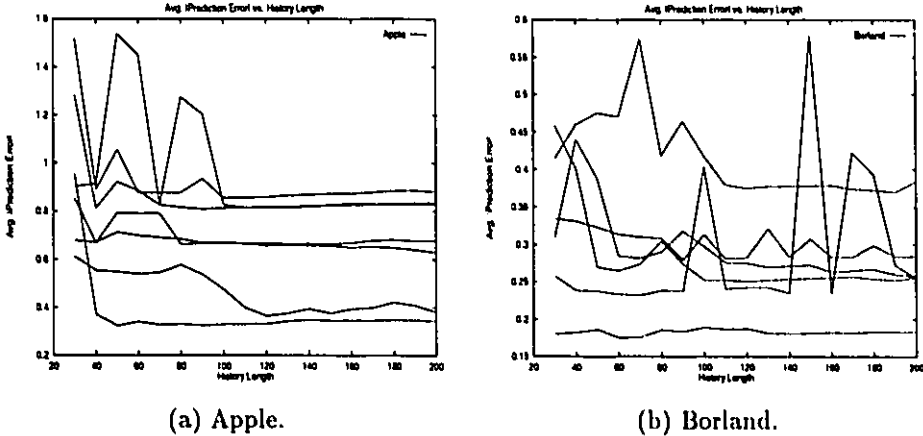
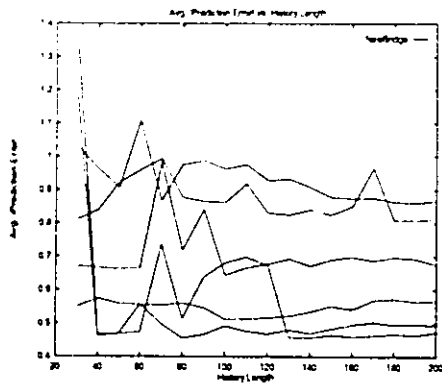
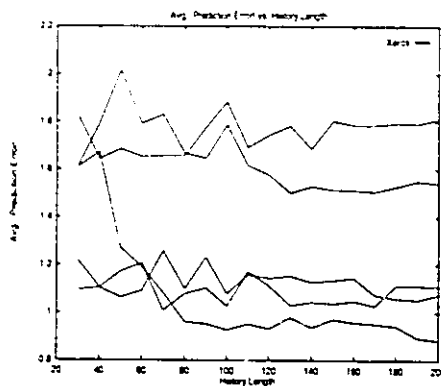


Figure D.1: Mean  $|PE|$  vs. History Length.



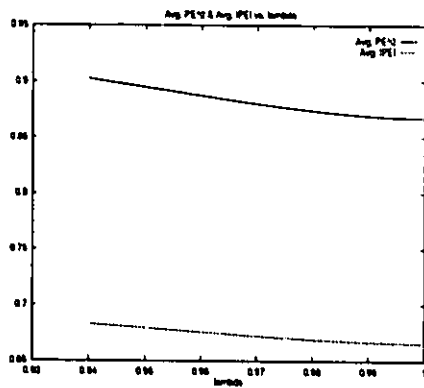
(a) NewBridge.



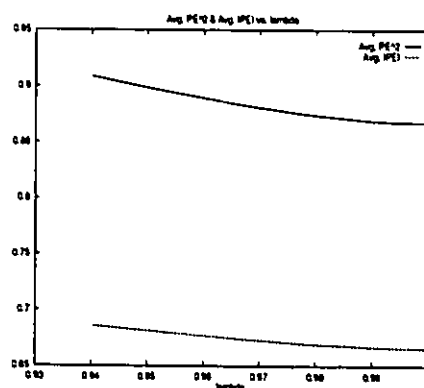
(b) Xerox.

Figure D.2: Mean  $|PE|$  vs. History Length.

## D.2 Fine Tuning the Forgetting Factor

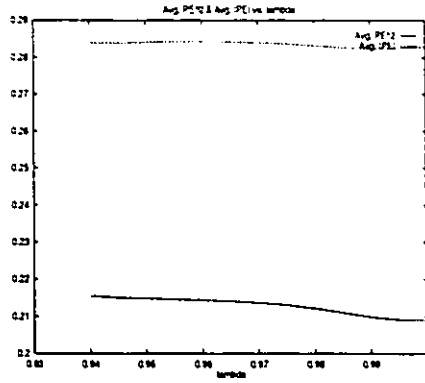


(a) State-space

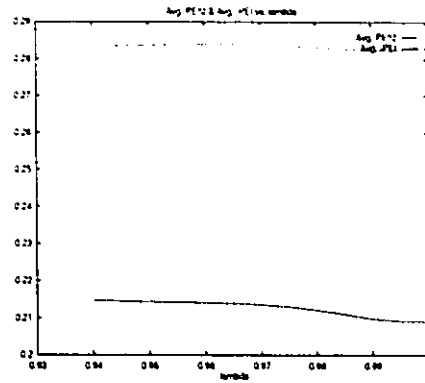


(b) SISO

Figure D.3: Apple. Minima at  $\lambda = 1.0$ .

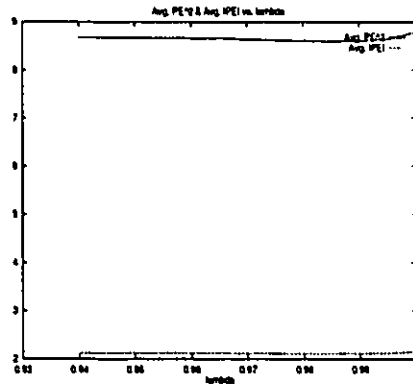


(a) State-space

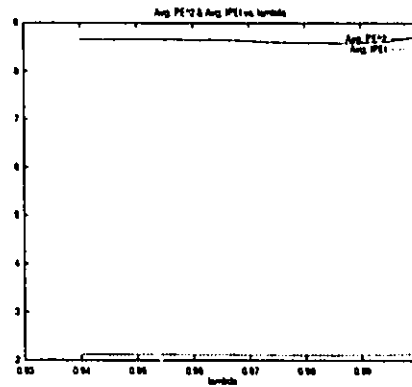


(b) SISO

Figure D.4: Borland. Minima near  $\lambda = 0.995$ .



(a) State-space



(b) SISO

Figure D.5: Standard & Poor's. Minima at  $\lambda = 0.99$ .

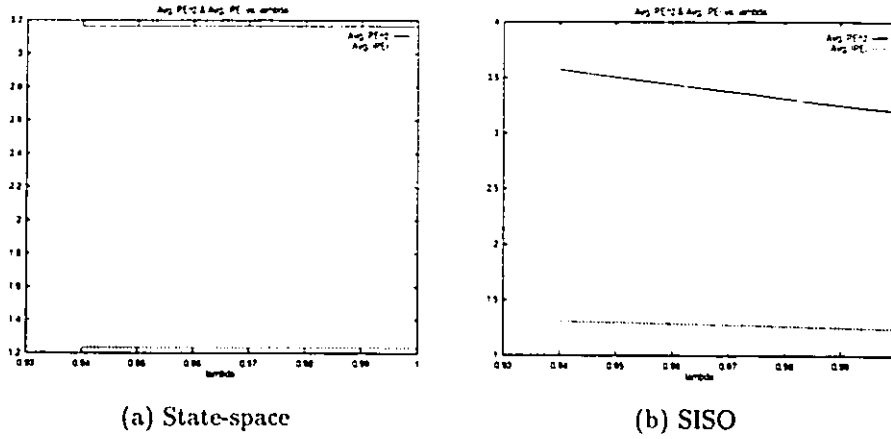


Figure D.6: Xerox. Minima at  $\lambda = 1.0$ .

### D.3 AR Modelling in White Noise

Results from experiment of AR process in white noise. The results with the theoretically impossible (negative) variances have been removed to economize space.

P	Normal	5	6	7
$\hat{\sigma}_x^2$	0.74767	0.145543	0.039935	0.158128
$\hat{\sigma}_w^2$	0.0	0.583273	0.568528	0.695139
Mean $PE^2$	0.869283	1.10559	1.68516	1.55419
Mean $ PE $	0.667001	0.799178	1.02393	0.974219

Table D.1: Apple.  $\sigma_x^2 = 0.76077$ . Best results at  $P = 5$ .

P	Normal	3	4	5	6	7	8	9	10
$\hat{\sigma}_x^2$	0.213545	0.413997	0.623201	0.0453932	0.476834	0.019858	0.187716	0.211285	0.316427
$\hat{\sigma}_w^2$	0.0	0.0417786	0.163401	0.215804	0.192908	0.192911	0.162818	0.159128	0.170266
Mean $PE^2$	0.209835	0.432475	0.869075	0.328334	1.2399	0.308284	0.441277	0.497505	1.03076
Mean $ PE $	0.282293	0.461883	0.665511	0.381803	0.758262	0.380422	0.454852	0.480828	0.696364

Table D.2: Borland.  $\sigma_x^2 = 0.214184$ . Best results at  $P = 7$ .

P	Normal	5	6	7	8	9	10
$\delta_u^2$	1.01235	0.062615	0.20314	0.075716	2.61823	2.5779	0.128346
$\delta_w^2$	0.0	1.13256	0.82669	0.952776	0.796529	0.75276	0.770618
Mean $PE^2$	1.35297	2.84501	1.77777	1.94955	6.23007	9.42096	2.38438
Mean $ PE $	0.835924	1.26436	1.01791	1.06194	1.88231	2.36345	1.15511

Table D 3: IBM.  $\sigma_x^2 = 1.02037$ . Best results at  $P = 6$ .

P	Normal	3	4	6	7	8	9	10
$\delta_u^2$	27.29	41.147	45.3801	12.5187	132.709	325.475	564.239	856.266
$\delta_w^2$	0.0	0.93967	0.164195	6.05244	19.6512	11.2551	7.65737	9.96706
Mean $PE^2$	31.3039	65.8729	103.076	58.76	265.73	466.137	760.869	1135.46
Mean $ PE $	4.6439	6.83895	8.97638	5.74424	13.0306	16.2516	20.3871	25.1165

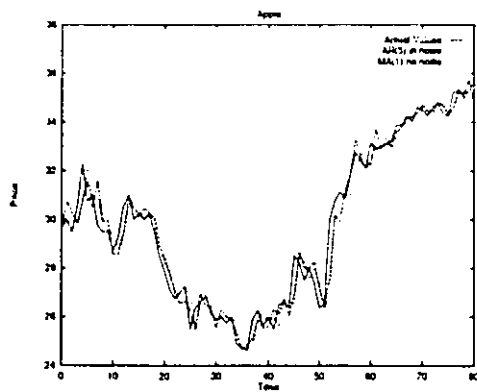
Table D.4: NASDAQ.  $\sigma_x^2 = 487.221$ . Best results at  $P = 6$ .

P	NORMAL	2	4	5	6	7
$\delta_u^2$	7.54823	14.3926	7.81035	2.4954	30.2868	239.686
$\delta_w^2$	0.0	0.534161	1.5179	3.70892	6.21711	5.93808
Mean $PE^2$	8.52824	185.133	13.3084	14.2681	53.7549	395.056
Mean $ PE $	2.10726	9.48629	2.80961	2.86611	5.78293	15.6201

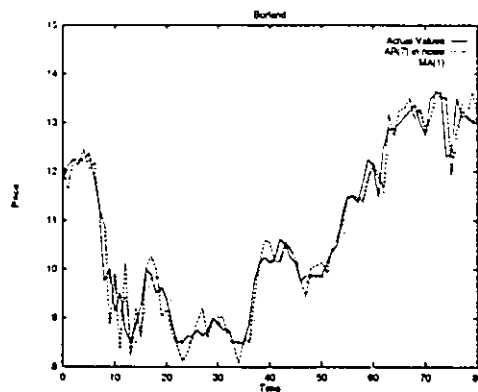
Table D.5: Standard & Poor's.  $\sigma_x^2 = 67.7452$ . Best results at  $P = 4$ .

P	NORMAL	1	2	3	6	7	8	9	10
$\delta_u^2$	1.91062	0.059196	0.19480	0.188913	1.30779	1.66227	9.34367	3.21153	5.26904
$\delta_w^2$	0.0	1.86974	1.85966	1.91362	1.78053	1.87288	1.03250	1.50237	1.8538
Mean $PE^2$	3.16472	3.2636	3.55695	3.79137	8.85419	13.5464	62.669	13.356	41.2806
Mean $ PE $	1.23308	1.26069	1.32375	1.3753	2.18998	2.71282	5.81904	2.72466	4.63205

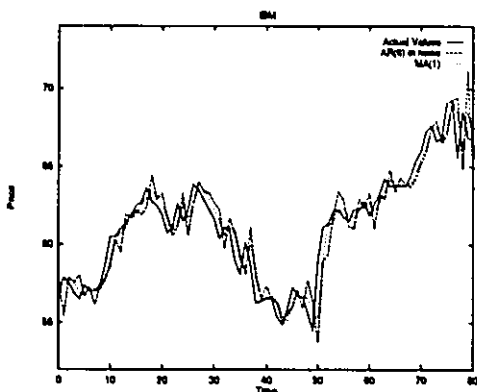
Table D.6: Xerox.  $\sigma_x^2 = 1.91078$ . Best results at  $P = 1$ .



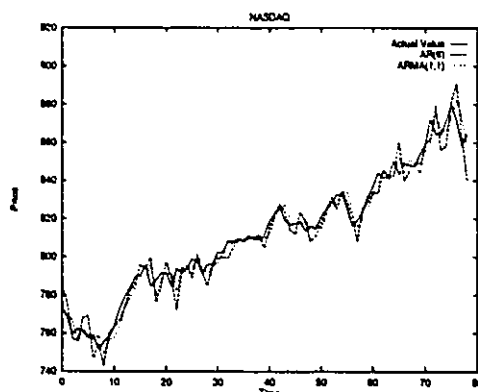
(a) Section of Apple predictions. AR(5) in noise, MA(1) no noise,  $d = 1$ .



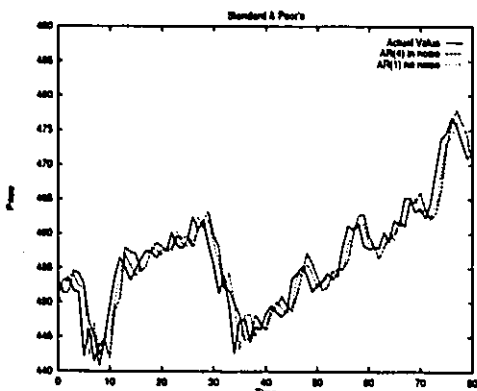
(b) Section of Borland predictions. AR(7) in noise, MA(1) no noise,  $d = 1$ .



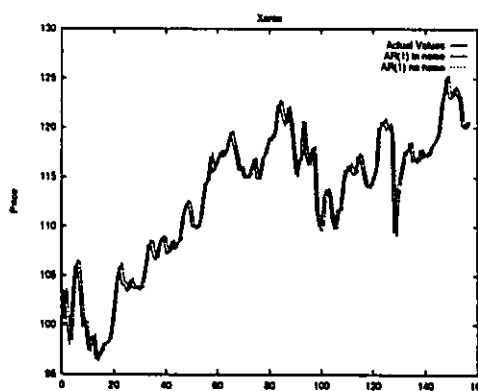
(c) Section of IBM predictions. AR(6) in noise, MA(1) no noise,  $d = 1$ .



(d) NASDAQ predictions. AR(5) in noise, ARMA(1,1) no noise,  $d = 0$ .



(e) Section of Standard & Poor's predictions. AR(4) in noise, ARMA(1,1) no noise,  $d = 0$ .



(f) Section of Xerox predictions. AR(1) in noise, AR(1) no noise,  $d = 1$ .

# Bibliography

- [1] G.E.P. Box & G.M. Jenkins. *Time Series Analysis Forecasting and Control*. Holden-Day, 500 Sansome Street, San Francisco, California, 1970.
- [2] Ken ichi Kamijo & Tetsuji Tanigawa. Stock price pattern recognition - a recurrent neural network approach. *IJCNN - International Joint Conference on Neural Networks*, 1 of 3:I-215 – I-221, 17-21 June, 1990.
- [3] James E. Bowen. Using neural nets to predict future values from time series data: Stock market. *Canadian Conference on Electrical and Computer Engineering, Ottawa, Ontario, Canada*, pages pp. 6.1.1-6.1.4, September 3-6 1990.
- [4] Morio Yoda & Masakazu Takeoka Takashi Kimoto, Kazuo Asakawa. Stock market prediction system with modular neural networks. *IJCNN - International Joint Conference on Neural Networks*, 1 of 3:I-1 – I-6, 17-21 June, 1990.
- [5] Shaun inn Wu & Ruey-Pyng Lu. Combining artificial neural networks and statistics for stock-market forecasting. *ACM*, pages 257-264, May 1993.
- [6] Lennart Ljung & Torsten Soderstrom. *Theory and Practice of Recursive Identification*. The MIT Press, 1983.

- [7] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82:35–45, March 1960.
- [8] Frank L. Lewis. *Optimal Estimation with an Introduction to Stochastic Control Theory*. John Wiley & Sons, 1986.
- [9] Brian D.O. Anderson & John B. Moore. *Optimal Filtering*. Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, New Jersey, 07632, 1979.
- [10] Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge, CB2 1RP, 1989.
- [11] Masanao Aoki. *State Space Modeling of Time Series, Second, Revised and Enlarged Edition*. Springer-Verlag, 1991.
- [12] R. E. Kalman & R. S. Bucy. New results in linear filtering and prediction theory. *ASME Journal of Basic Engineering*, 83:95–107, March 1961. Series D.
- [13] Steven M. Kay. *Modern Spectral Estimation: Theory and Application*. Prentice Hall Signal Processing Series. Prentice Hall, Inc., 1988.
- [14] Steven M. Kay & Stanley Lawrence Marple Jr. Spectrum analysis - a modern perspective. *Proceedings of the IEEE*, 69(11):1380–1419, November 1981.
- [15] Hirotugu Akaike. Maximum likelihood identification of gaussian autoregressive moving average models. *Biometrika*, 2(60):255–265, 1973.
- [16] Steve Beveridge & Cyril Oickle. A comparison of box-jenkins and objective methods for determining the order of a non-seasonal arma model. *Journal of Forecasting*, 13(5):419–434, 1994.

- [17] Saleh Radeideh & N.U. Ahmed. Course notes: Elg 7113, kalman filtering for engineers and scientists. Winter 1995, University of Ottawa, Department of Electrical Engineering.
- [18] T.C. Hsia. *System Identification*. Lexington Books, 1977.
- [19] M. Pagano. Estimation of models of autoregressive signal plus white noise. *Ann. Statist.*, 2:99-108, 1974.
- [20] M. Parzen. An approach to time series analysis. *Ann. Math. Statist.*, 32:951-989, 1961.
- [21] Mohinder S. Grewal & Angus P. Andrews. *Kalman Filtering: Theory and Practice*. Prentice-Hall, 1993.
- [22] Vladimir Kučera. *Discrete Linear Control*. John Wiley & Sons, 1979.
- [23] Howell Tong. Autoregressive model fitting with noisy data by akaike's information criterion. *IEEE Transactions on Information Theory*, IT21:475-480, July 1975.
- [24] Athanasios Papoulis. *Probability, Random Variable, and Stochastic Processes*. McGraw-Hill, Inc., New York, 1991.
- [25] Flannery Press & Teukolsky, Vetterfling. *Numerical Recipes in C*. Cambridge University Press, 1992.