INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

A STATE SPACE MODEL FOR INFLATION

Anthony Tippa

A Thesis in

The Department of Mathematics and Statistics

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Science at

Concordia University.

Montreal, Quebec, Canada.

May 2000

Abstract

A STATE SPACE MODEL FOR INFLATION

Anthony Tippa

Evidence has shown that the constant parameter assumption in Wilkie's Inflation model may be too restrictive. In this work we investigate the model proposed by Arsad (1999), where the mean level of inflation is assume to follow an AR(1) process. We compare the result obtained when prior knowledge of the initial parameter state is assumed known thus Bayesian analysis is applied and that obtained using classical approach.

Acknowledgements

I wish to acknowledge my indebtedness and sincere gratitude to my supervisor Doctor José Garrido for his comments, suggestions and immense support throughout the preparation of this work.

I will like to thank the Department of Mathematics and Statistics for their financial support during these two years.

I am also thankful to my friends Sabuj, Mamun, Sebastian and Sen for their objective criticism.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Outline of the thesis

The first chapter of the thesis reviews Wilkie's U.K. stochastic asset model. A very brief introduction to the model is made, for detailed descriptions of the model see Wilkie (1986, 1995). In Chapter 2 we introduce the state space model and Kalman filter technique. Methods used in estimating the parameters of the model are treated in Chapter 3, while Chapter 4 deals with the application of the state space model to inflation and a summary of the results is given.

1.2 Wilkie's Model

The Wilkie's Model was the first comprehensive stochastic investment model in the actuarial profession to be published. It was proposed in 1986 by the U.K. actuary David Wilkie, who developed work carried out by the Maturity Guarantees Working

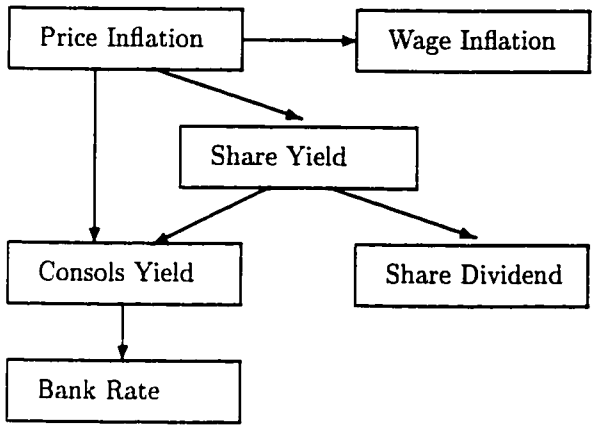

Figure 1.1: The "Cascade" structure of Wilkie's model

Party of the Institute of Actuaries. The original model is made up of four interconnected models, namely:

- price inflation model,
- share dividend yield model,
- share dividend model,
- Consols yield (long-term interest rate) model.

These models are essentially conventional ARIMA transfer function models. The model has been extended to include other interest rates and other asset returns [Wilkie (1995)]. Inflation is postulated as the driving force for other series. The relationships between the variables are based on a blend of statistical evidence and economic beliefs. The interdependence is shown pictorially in Figure 1.1, where the arrows indicate the direction of influence for times s, t such that $0 \leqslant s, t, s+t \leqslant 1$.

1.2.1 Price Inflation Model

In Wilkie's model, price inflation is assumed to be a factor influencing the returns on the assets in the model. The model uses a first-order autoregressive AR(1) process for the continuously compounded rate of inflation. Such a model can be written as [see Wilkie (1986)]:

$$i_t - \mu_Q = \alpha_Q[i_{t-1} - \mu_Q] + \sigma_Q \epsilon_{Qt} \,, \tag{1.2.1}$$

where $i_t = \log(Q_t/Q_{t-1})$, Q_t is the level of the inflation index, ϵ_{Qt} is an i.i.d. standard normal variable, and μ_Q, α_Q, and σ_Q are parameters to be estimated.

1.2.2 Wage Inflation Model

The wage inflation model can be represented by the following equation:

$$J_t = \mu_W + W_{W1} i_t + W_{W2} i_{t-1} + \alpha_W [J_{t-1} - \mu_W$$

$$- W_{W1} i_{t-1} - W_{W2} i_{t-1}] + W_Q [i_{t-1} \mu_Q] \tag{1.2.2}$$

$$+ \sigma_W [\rho \epsilon_{Qt} + \sqrt{1-\rho^2}\, \epsilon_{Wt}] \,,$$

where $J_t = \log(W_t/W_{t-1})$, W_t is the level of the wage (or earnings) index at time t, thus J_t is the rate (strictly force) of wage inflation over the year $(t-1, t)$, ϵ_{Wt} is an i.i.d. standard normal variable, and μ_W, α_W, W_{W1}, W_{W2}, W_Q and σ_W are parameters to be estimated.

1.2.3 ARCH Inflation Model

Engle (1982) uses auto-regressive conditional heteroscedasticity (ARCH) models to generalise the implausible assumption of a constant variance. ARCH models have non-constant variances conditional on the past, but constant unconditional variances. Essentially, the variance of the current stochastic disturbance is assumed to depend on the magnitudes of residuals; this can be extended so that the variance depends on the actual value of the series in the previous period, thereby capturing the notion that there is more variability at higher levels of inflation. Engle finds that the ARCH effect is significant for a series of quarterly U.K. inflation data.

Wilkie (1995) noted that the residuals $\epsilon_Q(t)$ may not be independent, and that allowing for heteroscedasticity may be necessary. To allow for the apparent non-stationary in the variance σ_Q^2, he endeavoured to fit an ARCH model of the form:

$$i_t = \mu_Q + \alpha_Q[i_{t-1} - \mu_Q] + \sigma_{Qt}\epsilon_{Qt} \,, \tag{1.2.3}$$

$$\sigma_{Qt}^2 = \sigma^2 + \beta_Q[i_{t-1} - \mu_Q]^2. \tag{1.2.4}$$

1.2.4 Share Dividend Yield Model

Wilkie's share dividend yield model is of the form

$$\ln Y_t = Y_W i_t + Y_{Nt} \,, \tag{1.2.5}$$

where

$$Y_{Nt} = \ln \mu_Y + \alpha_Y[Y_{N\,t-1} - \ln \mu_Y] + \sigma_Y \epsilon_{Yt} \,, \tag{1.2.6}$$

and Y_t is the dividend yield on ordinary shares at time t, Y_N is an AR(1) process and μ_Y, α_Y, σ_Y, and Y_W are parameters to be estimated.

1.2.5 Share Dividend Model

If we define K_t as the logarithm of the increase in the share dividends index from year $t-1$ to year t, Wilkie's share dividend model is of the form:

$$K_t = D_W \left(\frac{D_D}{1-(1-D_D)B} \right) i_t + D_X i_t + \mu_D \qquad (1.2.7)$$

$$+ D_Y Y_{E\,t-1} + D_B D_{E\,t-1} + D_{E\,t}\ ,$$

where the backwards step operator B is defined by $BX_t = X_{t-1}$, $K_t = \ln D_t - \ln D_{t-1}$, the dividend residuals $D_{E\,t}$ can be written as $D_{E\,t} = \sigma_D \epsilon_{D\,t}$, $\epsilon_{D\,t}$ is a sequence of independent identically distributed unit normal variate, and D_W, D_D, D_X and μ_D are parameters to be estimated. The term in parentheses (1.2.7) involving D_D represents an infinite series of lag effects, with exponentially declining coefficients D_D, $D_D(1-D_D)$, $D_D(1-D_D)^2$, and so on.

The sum of the coefficients is unity, so that part of the formula represents the lagged effect of inflation with unit gain. This means that if retail prices rise by one percent this term will also rise by one percent. It can alternatively be described as the carried forward effect of inflation, $D_{M\,t}$, where

$$D_{M\,t} = D_D\ i_t + (1-D_D)\ D_{M\,t-1}\ ,$$

from which it can be seen that the amount that enters the dividend model each

year is D_D times the current inflation rate, plus $(1 - D_D)$ times the amount brought forward to the next year.

1.2.6 Long-term Interest Rate (Consol Yield) Model

The long-term interest rate model is made up of a real component $C_{N\,t}$ and a component representing investors' inflationary expectations. The model is of the form:

$$C_t = C_{M\,t} + C_{N\,t}\,, \tag{1.2.8}$$

where

$$C_{M\,t} = C_W \left(\frac{C_D}{1 - (1 - C_D)B}\right) i_t \tag{1.2.9}$$

and

$$\ln C_{N\,t} = \ln \mu_C + (C_{A1}\,B + C_{A2}\,B^2 + C_{A3}\,B^3)(\ln C_{N\,t} - \ln \mu_C) + C_Y\,\dot{Y}_{E\,t} + \sigma_C \epsilon_{C\,t}.$$

C_t is the long-term interest rate and $\epsilon_{C\,t}$ is a sequence of independent identically distributed unit normal variates, and μ_C, C_{A1}, C_{A2}, C_{A3},C_W,C_D, C_Y, and σ_C are parameters to be estimated. The term in parentheses in C_D has a similar form to the D_D term in the dividend model, though the parameter value is different. It represents the current value of expected future inflation as an exponentially weighted moving average of past rates of inflation.

1.2.7 Short-term Interest Rate (Bank Rate) Model

The short-term interest rate model is defined by the following equation:

$$\ln B_t = \ln C_t - \mu_B - \alpha_B[\ln C_{t-1} - \ln B_{t-1} - \mu_B] - \sigma_B \epsilon_{B\,t} \tag{1.2.10}$$

where B_t is the short-term interest rate, $\epsilon_{B\,t}$ is a sequence of independent identically distributed unit normal variates and μ_B, σ_B, α_B are parameters to be estimated.

1.2.8 Index-linked Yield Model

The index-linked yield model is defined by the following equation:

$$\ln R_t = \ln \mu_R + \alpha_R[\ln R_{t-1} - \ln \mu_R] + R_{BC}\, \sigma_C \epsilon_{C\,t} + \sigma_R \epsilon_{R\,t} \tag{1.2.11}$$

where R_t is the real yield on index-linked securities, and $\epsilon_{R\,t}$ is a sequence of independently distributed unit normal random variables and μ_R, σ_R, σ_C, α_R, and R_{BC} are parameters to be estimated.

1.2.9 Property Yield Model

The property yield model is defined by the following equation:

$$\ln Z_t = \mu_Z + \alpha_Z[\ln Z_{t-1} - \mu_Z] + \sigma_Z \epsilon_{Z\,t}\,, \tag{1.2.12}$$

where Z_t is the property yield, $\epsilon_{Z\,t}$ is a sequence of independently distributed unit normal random variables and μ_Z, σ_Z, and α_Z are parameters to be estimated.

1.2.10 Property Income Model

If we define L_t as the logarithm of the increase in the property income index from year $t-1$ to year t. The property income model is defined by the following equations:

$$L_t = E_{M\,t} + E_X\, i_t + \mu_E + E_{BZ}\sigma_Z \epsilon_{Z\,t} + \sigma_E \epsilon_{Z\,t}\,, \tag{1.2.13}$$

where

$$E_{M\,t} = E_W \left(\frac{E_D}{1-(1-E_D)B}\right) i_t \tag{1.2.14}$$

and $E_{M\,t}$ is the property income index, while $\epsilon_{E\,t}$ is a sequence of independently distributed unit normal random variables and μ_E, σ_E, E_X, E_W, E_D, and E_{BZ} are parameters to be estimated.

Chapter 2

State Space Model and the Kalman filter

In this chapter we shall define the state space model and consider three fundamental problems associated with it. These are all concerned with finding best (in the sense of minimum mean-square error) linear estimates of the state vector θ_t in terms of the observations $i_1, i_2, ...$, and a random vector i_0.

Estimation of θ_t in terms of

1. $i_0, ..., i_{t-1}$ defines the *prediction* problem
2. $i_0, ..., i_t$ defines the *filtering* problem, and
3. $i_0, ..., i_\tau$ defines the *smoothing* problem (in which it is assumed that $\tau > t$).

Each of these problems can be solved recursively using an appropriate set of Kalman recursions which will be established in this chapter.

2.1 The Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [Kalman (1960)]. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application. The Kalman filter is commonly employed by control engineers and other physical and social scientists in such diverse areas as the processing of signals in aerospace tracking and underwater sonar, and the optimal control of economic system. Detailed discussion of the Kalman filter theory can be found in Harvey (1989).

The Kalman filter is based on a state space model. The unknown state of the system is denoted by θ_t, and is referred to as the *state vector*. The measurements i_t consist of linear combinations of the state variables corrupted by a sequence of uncorrelated random errors η_t. In the state space form, we write the system equation and the measurement equation, respectively, as

$$\theta_t = a_t + \Omega_t \theta_{t-1} + \epsilon_t \,, \tag{2.1.1}$$

and

$$i_t = b_t + A_t \theta_t + \eta_t \,. \tag{2.1.2}$$

Here

i_t is an $N_i \times 1$ observed vector ,

b_t is an $N_i \times 1$ vector ,

θ_t is an $N_\theta \times 1$ vector representing the state of the process at time t ,

a_t is an $N_\theta \times 1$ vector

Ω_t is an $N_i \times N_\theta$ state transition matrix that relates θ_{t-1} to θ_t,

ϵ_t represents a vector of serially uncorrelated disturbances with mean zero and covariance matrix C_t, i.e $\mathrm{E}(\epsilon_t) = 0$ and $\mathrm{Var}(\epsilon_t) = C_t$,

η_t is a vector of serially uncorrelated disturbances with mean zero and covariance matrix R_t, i.e $\mathrm{E}(\eta_t) = 0$ and $\mathrm{Var}(\eta_t) = R_t$,

A_t is an $N_\theta \times N_\theta$ linear connection matrix between the output i_t and state θ_t.

The specification of the state space system is completed by the assumption that the disturbances η_t and ϵ_t are uncorrelated with each other in all time periods, and uncorrelated with the initial state, that is

$$\mathrm{E}(\eta_t \epsilon_s) = 0 \text{ for all } s,t = 1, ...T,$$

and

$$\mathrm{E}(\eta_t \theta_0) = 0 \text{ for all } t = 1, ...T.$$

Once a model has been put in a state space form, the way is opened for the application of the Kalman filter algorithms. The Kalman filter is a recursive procedure for computing the estimate of the state vector at time t, based on the information available at time t. This information consists of the observations up to and including i_t. This estimate may be improved as additional observations of the process become available.

In certain engineering applications the Kalman filter is important because of on-line estimation. The current value of the state vector is of prime interest (for example, it may represent the co-ordinates of a rocket in space) and the Kalman filter enables the estimate of the state vector to be continually updated as new observations become available. At first sight, the value of such a procedure in economic applications would appear to be limited. New observations tend to appear at rather less frequent intervals and the emphasis is on making predictions of future observations based on a given sample. The state vector does not always have an economic interpretation and, in cases where it does, it is more appropriate to estimate its value at a particular point in time using all the information in the sample, not just part of it. These two problems are known as prediction and smoothing respectively. It turns out that the Kalman filter provides the basis for the solution of both of them.

Another reason for the central role of the Kalman filter is that when the disturbances and the initial state vector are normally distributed, it enables the likelihood function to be calculated via what is known as the prediction error decomposition. This opens the way for the estimation of any unknown parameter in the model.

Let $I_t = \{i_1, i_2, ... i_t\}$ and $\hat{\theta}_{t-1|t-1}$ denote the optimal estimator of θ_{t-1} based on the information available up to and including i_{t-1}. Let $P_{t-1|t-1}$ denote the covariance matrix of the estimation error, that is

$$P_{t-1|t-1} = \mathrm{E}[(\theta_{t-1} - \hat{\theta}_{t-1|t-1})(\theta_{t-1} - \hat{\theta}_{t-1|t-1})^T] \,. \tag{2.1.3}$$

Under the normality assumption, the initial state vector θ_0, has a multivariate normal distribution with mean $\hat{\theta}_{0|0}$ and covariance matrix $P_{0|0}$. Th disturbances η_t and ϵ_t also have multivariate normal distributions for $t = 1, ..., T$ and are distributed independently of each other and of θ_0. In many cases, we begin the estimation problem with no prior measurements. Thus, in this case, if the process mean is zero, the initial estimate is zero and the associated error covariance matrix is just the covariance matrix of θ itself. The system matrices together with $\hat{\theta}_{0|0}$ and $P_{0|0}$ are assumed to be known in all time periods and so do not need to be explicitly included in the information set.

Assuming initial values $\hat{\theta}_{0|0}$ and $P_{0|0}$ are available, the optimal estimator of θ_t and the corresponding covariance matrix of the estimation error are

$$\hat{\theta}_{t|t-1} = a_t + \Omega_t \hat{\theta}_{t-1|t-1} \ , \tag{2.1.4}$$

and

$$P_{t|t-1} = \Omega_t P_{t-1|t-1} \Omega_t^T + R_t \ . \tag{2.1.5}$$

These two equations are known as the *prediction equations.* With the assumption of a prior estimate $\hat{\theta}_{t|t-1}$, we now seek to to use the measurement i_t to improve the prior estimate. We choose a linear blending of the noisy measurement and the prior estimate to obtain the *updating equations*, which are

$$\hat{\theta}_{t|t} = \hat{\theta}_{t|t-1} + K_t(i_t - A_t \hat{\theta}_{t|t-1} - b_t) \ , \tag{2.1.6}$$

and

$$P_{t|t} = P_{t|t-1} - K_t A_t P_{t|t-1} \ , \tag{2.1.7}$$

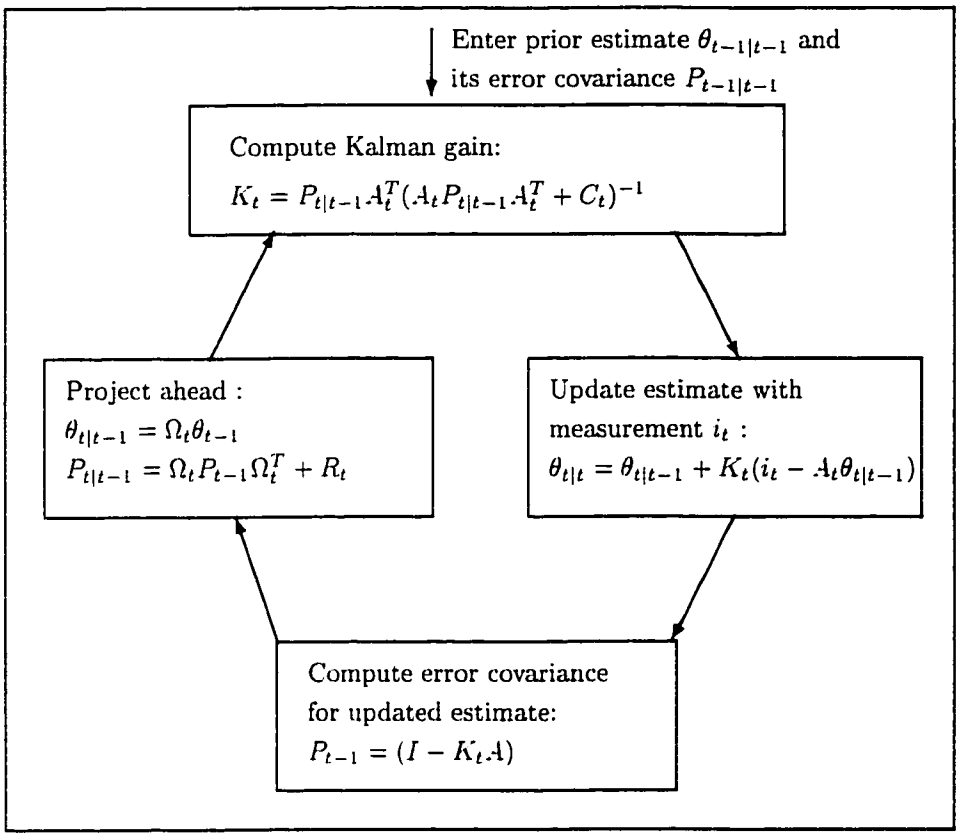

Figure 2.1: A complete picture of the operation of the Kalman filter.

where

$\hat{\theta}_{t|t}$ is the updated estimate ,

$K_t = P_{t|t-1}A_t^TF_t^{-1}$ is known as the Kalman gain ,

$F_t = A_tP_{t|t-1}A_t^T + C_t$.

We now have the needed quantities at time t, and the measurement i_t can be assimilated just as in the previous step. Equations (2.1.4) (2.1.5), (2.1.6) and (2.1.7) comprise the Kalman filter recursive equations. It should be clear that once the loop is entered, it can be continued ad infinitum. The pertinent equations and the sequence of computational steps are shown pictorially in Figure 2.1. This summarizes

what is now known as the *Kalman filter.*

2.1.1 Example

Let us introduce the example considered by Phadke (1981) in the context of statistical quality control. Here the observation i_t is a simple (approximately normal) transform of the defectives observed in a sample obtained at time t, while $\theta_{1,t}$ and $\theta_{2,t}$ represent, respectively, the true defective index of the process and the drift of this index. We then have as the observation equation

$$i_t = \theta_{1,t} + \eta_t$$

and as the system equations

$$\theta_{1,t} = \theta_{2,t} + \epsilon_{1,t}$$

$$\theta_{2,t} = \theta_{2,t-1} + \epsilon_{2,t}$$

In vector notation, this system of equations becomes

$$\theta_t = \Omega\, \theta_{t-1} + \epsilon_t$$

where

$$\theta_t = \begin{pmatrix} \theta_{1,t} \\ \\ \theta_{2,t} \end{pmatrix}, \ \Omega = \begin{pmatrix} 0 & 1 \\ \\ 0 & 1 \end{pmatrix}, \qquad \text{and} \qquad \epsilon_t = \begin{pmatrix} \epsilon_{1,t} + \epsilon_{2,t} \\ \\ \epsilon_{2,t} \end{pmatrix} \tag{2.1.8}$$

Ω does not change with time.

To further simplify the above model, we can remove the drift parameter. This yields

$$\theta_t = \theta_{t-1} + \epsilon_t$$

In this situation, $A_t \equiv \Omega_t \equiv 1$; if we further specify that $P_{0|0} = 1$, $R_t \equiv 1$, $C_t \equiv 2$, we can easily demonstrate inductively that $P_{t|t-1} = \Omega_t P_{t-1|t-1}\Omega_t^T + R_t \equiv 2$, and from (2.1.7), $P_{t|t} \equiv 1$. In (2.1.6), then, our recursive relationship becomes

$$\hat{\theta}_{t|t} = \hat{\theta}_{t-1|t-1} + \frac{1}{2}(i_t - \hat{\theta}_{t-1|t-1})$$

$$= \frac{1}{2}(i_t + \hat{\theta}_{t-1|t-1})$$

$$= \sum_{j=0}^{t-1} (\frac{1}{2})^{j+1} \, i_{t-j} + (\frac{1}{2})^t \hat{\theta}_{0|0} \, . \tag{2.1.9}$$

2.2 Kalman Prediction

In the context of the state space model, one of the traditional uses of the model is to predict future values of the unobservable state vector θ_t (or to predict future values of i_t). This model provides a convenient computational method for recursively obtaining predicted values.

Let $\hat{\theta}_{T+h|T} = \mathrm{E}[\theta_{T+h}|i_1, ..., i_T] \qquad h = 1, 2, 3, ...$

denote the minimum MSE linear predictor of θ_{T+h}, with covariance matrix

$$P_{T+h|T} = \mathrm{E}[(\theta_{T+h} - \hat{\theta}_{T+h|T})(\theta_{T+h} - \hat{\theta}_{T+h|T})']$$

So that under normality the conditional distribution of θ_{T+h} given $i_1, ..., i_T$ is normal with conditional mean $\hat{\theta}_{T+h|T}$ and covariance matrix $P_{T+h|T}$.

To evaluate $\hat{\theta}_{T+h|T}$ and $P_{T+h|T}$ we simply repeatedly apply the Kalman prediction

equations (2.1.4) and (2.1.5). This gives the h-step prediction equations.

$$\hat{\theta}_{T+h|T} = a_{T+h} + \Omega_{T+h}\hat{\theta}_{T+h-1|T} \tag{2.2.10}$$

and

$$P_{T+h|T} = \Omega_{T+h}P_{T+h-1|T}\Omega'_{T+h} + R_{T+h} \tag{2.2.11}$$

2.3 Kalman Smoothing

Another problem of interest with the state space model framework, particularly in applications to economics and business, is to obtain "smoothed" estimates of the past values of the state vector θ_t given observations $i_1, ..., i_T$ through some fixed time T. One convenient method to obtain the desired estimates, known as the *fixed-interval smoothing* algorithm, makes use of the Kalman filter estimates $\hat{\theta}_{t|t}$. The smoothing algorithm produces the minimum MSE estimator of the state vector θ_t given observations through time T, $\hat{\theta}_{t|T} = \mathrm{E}[\theta_t | i_1, ..., i_T]$. The covariance matrix of θ_t conditional on all T observations is denoted by

$$P_{t|T} = \mathrm{E}[(\theta - \hat{\theta}_{t|T})(\theta_t - \hat{\theta}_{t|T})']$$

The fixed-interval smoothing algorithm consists of a set of recursions which start with the final quantities, $\hat{\theta}_{T|T}$ and $P_{T|T}$, given by the Kalman filter and work backwards. The equations are

$$\hat{\theta}_{t|T} = \hat{\theta}_{t|t} + P_t^*(\hat{\theta}_{t+1|T} - \Omega_{t+1}\hat{\theta}_{t|t}) \tag{2.3.12}$$

and

$$P_{t|T} = P_{t|t} + P_t^*(P_{t+1|T} - P_{t+1|t})P_t^{*'} \tag{2.3.13}$$

where

$$P_T^* = P_{t|t}\Omega_{t+1}' P_{t+1|t}^{-1}, \qquad t = T-1, ..., 1.$$

The algorithm therefore requires that $\hat{\theta}_{t|t}$ and $P_{t|t}$ be stored for all t so that they can be combined with $\hat{\theta}_{t+1|T}$ and $P_{t+1|t}$.

Chapter 3

Parameter Estimation

3.1 Prediction Error Decomposition of the Likelihood

The system matrices in a state space model usually depend on a set of unknown parameters Ψ. The parameters are estimated using a maximum likelihood estimation procedure. here the Kalman filter technique is used to construct the likelihood function. The conditional probability density function is used to write the joint density function [Harvey (1989)] as

$$L(i,\theta) = \prod_{t=1}^{T} f(i_t|I_{t-1}) \tag{3.1.1}$$

where $f(i_t|I_{t-1})$ denotes the distribution of i_t, conditional on the information available at time $t-1$. That is

$$I_{t-1} = \{i_{t-1}, i_{t-2}, ..., i_1\}.$$

Note that given I_{t-1},

$$i_t \sim N(A_t\hat{\theta}_{t|t-1} + b_t, F_t)$$

therefore, the conditional p.d.f of i_t given I_{t-1} is given by

$$L = \frac{1}{(2\pi)^{\frac{NT}{2}} \prod_{t=1}^{T} |F_t|^{1/2}} e^{-\frac{1}{2}\sum_{t=1}^{T}(i_t - A_t\hat{\theta}_{t|t-1} - b_t)' F_t^{-1}(i_t - A_t\hat{\theta}_{t|t-1} - b_t)}. \tag{3.1.2}$$

Let $\hat{\nu}_t = i_t - A_t\hat{\theta}_{t|t-1} - b_t$, then the likelihood becomes

$$L = \frac{1}{(2\pi)^{\frac{NT}{2}} \prod_{t=1}^{T} |F_t|^{1/2}} e^{-\frac{1}{2}\sum_{t=1}^{T}\hat{\nu}_t' F_t^{-1}\hat{\nu}_t} \tag{3.1.3}$$

and therefore the log-likelihood is

$$\log L = -\frac{NT}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{T}\log|F_t| - \frac{1}{2}\sum_{t=1}^{T}\hat{\nu}_t' F_t^{-1}\hat{\nu}_t. \tag{3.1.4}$$

$\hat{\nu}_t$ is known as the vector of prediction errors.

If prior information is available on all elements of θ_0, then θ_0 has a proper prior distribution with known mean $\theta_{0|0}$ and bounded covariance matrix $P_{0|0}$. The Kalman filter then yields the exact likelihood function of the observations i, via the prediction error decomposition [Harvey (1989)]. Unfortunately, genuine prior information is rarely available. This has led some statisticians to conclude that Kalman filter techniques are only appropriate if (a) one is able to adopt a Bayesian approach in which a proper prior distribution for θ_0 is always specified, or (b) the sample size is so large that the specification of initial conditions is unimportant. As discussed in [Harvey (1989)], such a conclusion is unwarranted. We know that for a univariate series the Kalman filter can always be initialised with the mean and covariance matrix of the unconditional distribution of θ_t when θ_t is stationary. This is valid even if

the matrices A_t, R_t and b_t in the measurement equation are not time-invariant. For non-stationary state vectors with d non-stationary components, in a time-invariant model, observability of the d non-stationary components is sufficient for the construction of a proper distribution for θ_d. When this is the case, the joint distribution of $i_{d+1}, ..., i_T$ conditional on $i_1, ..., i_d$ is given by the prediction error decomposition, with summations running from $t = d + 1$ instead of $t = 1$. If $i_1, ..., i_d$ are regarded as being fixed, the joint density function is an unconditional one.

Once an algorithm for computing the likelihood has been found, it must be maximised with respect to the unknown parameters Ψ. This will normally be carried out by some kind of numerical optimisation procedure. However, in maximising the likelihood function it is usually advantageous to exploit any linearities in order to reduce the dimension of the search. In general, this is done by recognising that the model is *linear* in a subset of parameters. This means that if the remaining parameters are held constant, ML estimates of the first set of parameters may be obtained directly, without resorting to an iterative procedure. These estimates may be substituted into the likelihood function, and the resulting *concentrated* likelihood function maximised with respect to the second set of parameters only. For example, let $F_t = F$ a constant matrix, we can compute the ML estimate of F and then we concentrate F out of the likelihood.

Differentiating $\log L$ with respect to Ω is not straightforward. However, by making use of the results

$$\frac{\partial x'Ax}{\partial A} = xx' \tag{3.1.5}$$

and

$$\frac{\partial \log |A|}{\partial A} = (A')^{-1}, \tag{3.1.6}$$

the derivatives of $\log L$ with respect to the elements of the inverse of Ω are readily obtainable. Thus

$$\frac{\partial \log L}{\partial F^{-1}} = \frac{T}{2}F - \frac{1}{2}\sum_{t=1}^{T} \nu_t \nu_t', \tag{3.1.7}$$

having used the fact that

$$-\frac{T}{2}\log |F| = -\frac{T}{2}\log |F^{-1}|^{-1} = \frac{T}{2}\log |\Omega^{-1}| \,. \tag{3.1.8}$$

Because of the invariant property of ML estimators, the ML estimator of F is equal to the inverse of the ML estimator of F^{-1}. Thus setting (3.1.7) to a matrix of zeros and solving will yield the ML estimators of both F^{-1} and Ω. In fact it is F which is of primary interest, and the ML estimator is given directly as

$$\hat{F} = T^{-1}\sum_{t=1}^{T} \nu_t \nu_t'. \tag{3.1.9}$$

Maximising $\log L$ is equivalent to minimising S. With

$$S = T\log |F| + \sum_{t=1}^{T} \nu_t' F^{-1} \nu_t, \tag{3.1.10}$$

since

$$\sum \nu_t' \hat{F}^{-1} \nu_t = \mathrm{tr}(\hat{F}^{-1} \sum \nu_t \nu_t') = NT. \tag{3.1.11}$$

It follows from (3.1.10) that maximising the concentrated likelihood function is equivalent to minimising a generalised sum of squares function. In general, in order to obtain the maximum likelihood estimates we need to compute the information matrix.

3.2 Information Matrix

The information matrix provides an estimate of the covariance matrix of the maximum likelihood estimation and is also featured in algorithms used to compute these estimates. The prediction error decomposition likelihood function,(3.1.2), has the important property that it yields an information matrix which depends on the first derivatives only. The next section shows how these derivatives may be obtained either numerically or analytically when the model is in the state space form.

Expressions for the score vector and the information matrix for (3.1.2) are obtained as follows. First write the log-likelihood function in the form

$$\log L = \sum_{t=1}^{T} l_t$$

where l_t is the logarithm of the conditional density function of i_t given I_{t-1}, i.e.

$$l_t = \frac{1}{2}\log 2\pi - \frac{1}{2}\log|F_t| - \frac{1}{2}\nu_t' F^{-1}\nu$$

For any symmetric matrix A, the derivatives of the determinant and the inverse with respect to a variable, x, are

$$\frac{\partial|A|}{\partial x} = |A|\mathrm{tr}[A^{-1}\frac{\partial A}{\partial x}] \tag{3.2.12}$$

and

$$\frac{\partial A^{-1}}{\partial x} = -A^{-1}\frac{\partial A}{\partial x}A^{-1}. \tag{3.2.13}$$

Differentiating l_t with respect to the i-th element of Ψ, therefore gives

$$-\frac{1}{2}\mathrm{tr}\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}\right] - \frac{1}{2}\left[\frac{\partial \nu_t'}{\partial \psi_i}F_t^{-1}\nu_t - \nu_t' F_t^{-1}\frac{\partial F_t}{\partial \psi_i}F_t^{-1}\nu_t + \nu_t' F_t^{-1}\frac{\partial \nu_t}{\partial \psi_i}\right]. \tag{3.2.14}$$

Taking the trace of the last term allows this expression to be re-written as

$$\frac{\partial l_t}{\partial \psi_i} = -\frac{1}{2}\mathrm{tr}\left[\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}\right]\left[I - F_t^{-1}\nu_t\nu_t'\right]\right] - \left(\frac{\partial \nu_t}{\partial \psi_i}\right)' F_t^{-1}\nu_t. \tag{3.2.15}$$

Differentiating (3.2.15) with respect to the j-th element of Ψ gives

$$\begin{aligned}\frac{\partial^2 l_t}{\partial \psi_i \partial w_j} = &-\frac{1}{2}tr\left[\partial\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}\right]/\partial \psi_j\right]\left[I - F_t^{-1}\nu_t\nu_t'\right] \\ &-\frac{1}{2}\mathrm{tr}\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}F_t^{-1}\frac{\partial F_t}{\partial \psi_j}F_t^{-1}\nu_t\nu_t'\right] \\ &+\frac{1}{2}\mathrm{tr}\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}F_t^{-1}\left[\frac{\partial \nu_t}{\partial \psi_j}\nu_t' + \nu_t\frac{\partial \nu_t'}{\partial \psi_j}\right]\right] \\ &-\frac{\partial^2 \nu_t'}{\partial \psi_i \partial \psi_j}F_t^{-1}\nu_t - \frac{\partial \nu_t'}{\partial \psi_i}\frac{\partial F_t^{-1}}{\partial \psi_j}\nu_t - \frac{\partial \nu_t'}{\partial \psi_i}F_t^{-1}\frac{\partial \nu_t}{\partial \psi_j}.\end{aligned} \tag{3.2.16}$$

The ij-th element of the information matrix is by definition

$$-\mathrm{E}\left[\frac{\partial^2 \log L}{\partial \psi_i \partial \psi_j}\right] = -\mathrm{E}\left[\sum_{t=1}^{T}\frac{\partial^2 l_t}{\partial \psi_i \partial \psi_j}\right],$$

but its evaluation in the present context is simplified considerably by noticing that

$$\mathrm{E}(l_t) = \mathrm{E}\left[\mathrm{E}_{t-1}(l_t)\right], \tag{3.2.17}$$

where E_{t-1} is the expectation operator at time $t-i$. Now consider taking expectations of the terms in (3.2.16) conditional on the information at time $t-1$. The only random variables are the elements of the innovation vector, ν_t, and their first and second derivatives. However, the derivatives are fixed with respect to the expectation operator at time $t-1$. This follows because

$$\nu_t = i_t - \mathrm{E}_{t-1}(i_t),$$

and so

$$\frac{\partial \nu_t}{\partial \psi_i} = -\frac{\partial}{\partial \psi_i}\mathrm{E}_{t-1}(i_t).$$

The conditional expectation of ν_t is zero and therefore

$$\mathrm{E}_{t-1}\left[\frac{\partial \nu_t'}{\partial \psi_i}\nu_t\right] = \frac{\partial \nu_t'}{\partial \psi_i}\mathrm{E}_{t-1}(\nu_t) = 0\,. \tag{3.2.18}$$

A similar result holds for terms involving ν_t and its second derivatives. The net effect on the conditional expectation of (3.2.16) is that the third, fourth and fifth terms disappear. In addition the first term disappears because the conditional expectation of $\nu_t\nu_t'$ is F_t and for the same reason the second term simplifies. This leaves the following expression for the ij-th element of the information matrix

$$I_{ij}(\psi) = \frac{1}{2}\sum_t \left[\mathrm{tr}\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}F_t^{-1}\frac{\partial F_t}{\partial \psi_j}\right]\right]$$

$$+\mathrm{E}\left[\sum_t \left(\frac{\partial \nu_t}{\partial \psi_i}\right)' F_t^{-1}\frac{\partial \nu_t}{\partial \psi_j}\right], \quad i,j = 1...,n. \tag{3.2.19}$$

Dropping the expectation operator from the second term gives an expression which is asymptotically equivalent to (3.2.19) and which may, in some cases be easier to evaluate.

3.3 Derivatives of the Likelihood Function

As was shown in the previous section the i-th element in the score vector is

$$\frac{\partial \log L}{\partial \psi_i} = -\frac{1}{2}\sum_t \left\{ \mathrm{tr}\left[\left[F_t^{-1}\frac{\partial F_t}{\partial \psi_i}\right](I - F_t^{-1}\nu_t\nu_t')\right] - \frac{\partial \nu_t'}{\partial \psi_i}F_t^{-1}\nu_t \right\}, \quad i = 1, ..., n \tag{3.3.20}$$

Evaluating the score vector therefore requires the evaluation of the $N_i \times N_\theta$ matrices of the derivatives, $\partial F_t/\partial \psi_i$, and the $N_i \times 1$ vectors of derivatives $\partial \nu_t/\partial \psi_i$ for $t = 1, ..., T$ and $i = 1, ..., n$. These same derivatives may then be used to compute the information matrix via (3.2.19).

The derivatives of F_t and ν_t for the state space form (2.1.1) and (2.1.2) may be evaluated numerically or analytically. Computing them numerically requires n additional passes of the Kalman filter. For $i = 1, ..., n$ a small amount, δ_i, is added to ψ_i, and the Kalman filter is run with this new value but with all the other elements in Ψ remaining at their original values. This yields a new set of innovations and their covariance matrices, $\nu_t^{(i)}$ and $F_t^{(i)}$. The expressions $\delta_i^{-1}[\nu_t^{(i)} - \nu_t]$ and $\delta_i^{-1}[F_t^{(i)} - F_t]$ are then numerical approximations to the required derivatives. Of course, numerically evaluating the derivatives is only useful if they are to be used in computing the information matrix. If all that is required is the score vector, the numerical derivatives of the log-likelihood function can be evaluated directly.

The derivatives of F_t and ν_t may be evaluated analytically by programming n sets of recursions to run in parallel with the Kalman filter. These recursions all have a

common form, which is set out below. The i-th set of recursions yields the quantities needed to calculate the derivatives of F_t and ν_t with respect to the i-th element of Ψ. Since

$$\nu_t = i_t - A_t\theta_t - b_t, \quad t = 1, ..., T\ , \tag{3.3.21}$$

the vector of derivatives with respect to ψ_i is

$$\frac{\partial \nu_t}{\partial \psi_i} = -A_t\frac{\partial \theta_{t|t-1}}{\partial \psi_i} - \frac{\partial A_t}{\partial \psi_i}\theta_{t|t-1} - \frac{\partial b_t}{\partial \psi_i}\ , \tag{3.3.22}$$

and so a recursion is needed to provide $\partial\theta_{t|t-1}/\partial\psi_i$. Similarly

$$\frac{\partial F_t}{\partial \psi_i} = \frac{\partial A_t}{\partial \psi_i}P_{t|t-1}A_t^{'} + A_t\frac{\partial P_{t|t-1}}{\partial \psi_i}A_t^{'} + A_tP_{t|t-1}\frac{\partial A_t^{'}}{\partial \psi_i} + \frac{\partial R_t}{\partial \psi_i} \tag{3.3.23}$$

and so a recursion is also needed for the derivatives of $P_{t|t-1}$.

The recursions for the derivatives of $\theta_{t|t-1}$ and $P_{t|t-1}$ are obtained by differentiating the Kalman filter prediction equations (2.1.4),(2.1.5) and updating equations (2.1.6), (2.1.7). Differentiating the prediction equations yields

$$\frac{\partial \theta_{t|t-1}}{\partial \psi_i} = \frac{\partial \Omega_t}{\partial \psi_i}\theta_{t-1|t-1} + \Omega_t\frac{\partial \theta_{t-1|t-1}}{\partial \psi_i} + \frac{\partial a_t}{\partial \psi_i} \tag{3.3.24}$$

and

$$\frac{\partial P_{t|t-1}}{\partial \psi_i} = \frac{\partial \Omega_t}{\partial \psi_i}P_{t-1|t-1}\Omega_t^{'} + \Omega_t\frac{\partial P_{t-1|t-1}}{\partial \psi_i}\Omega_t^{'} + \Omega_tP_{t-1|t-1}\frac{\partial \Omega_t^{'}}{\partial \psi_i} + \frac{\partial C_t}{\partial \psi_i}\ , \tag{3.3.25}$$

while for the updating equations

$$\frac{\partial \theta_{t|t}}{\partial \psi_i} = \frac{\partial \theta_{t|t-1}}{\partial \psi_i} + \frac{\partial P_{t|t-1}}{\partial \psi_i}A_t^{'}F_t^{-1}\nu_t + P_{t|t-1}\frac{\partial A_t^{'}}{\partial \psi_i}F_t^{-1}\nu_t$$

$$-P_{t|t-1}A_t^{'}F_t^{-1}\frac{\partial F_t}{\partial \psi_i}F_t^{-1}\nu_t + P_{t|t-1}A_t^{'}F_t^{-1}\frac{\partial \nu_t}{\partial \psi_i} \tag{3.3.26}$$

and

$$\frac{\partial P_{t|t}}{\partial \psi_i} = \frac{\partial P_{t|t-1}}{\partial \psi_i} - \frac{\partial P_{t|t-1}}{\partial \psi_i} A_t' F_t^{-1} A_t P_{t|t-1} - P_{t|t-1} \frac{\partial A_t'}{\partial \psi_i} F_t^{-1} A_t P_{t|t-1}$$

$$+ P_{t|t-1} A_t' F_t^{-1} \frac{\partial F_t}{\partial \psi_i} F_t^{-1} A_t P_{t|t-1} - P_{t|t-1} A_t' F_t' \frac{\partial A_t}{\partial \psi_i} P_{t|t-1}$$

$$- P_{t|t-1} A_t' F_t^{-1} A_t \frac{\partial P_{t|t-1}}{\partial \psi_i} , \qquad (3.3.27)$$

for $t = 1, ..., T$. Equations (3.3.24),(3.3.25),(3.3.26) and (3.3.27) together with (3.3.22) and (3.3.23) provide the required derivatives. The starting values depend on the starting values for the Kalman filter proper.

3.4 Bayesian Analysis

In this thesis we compare results obtained using the Bayesian approach and that obtained using classical method. At this juncture we give a brief discussion of Bayesian analysis. For a detailed description of the Bayesian statistics see [Berger (1989) or Lee (1999)].

Bayesian analysis is performed by combining prior beliefs about various possible hypothesis and then modify these prior beliefs in the light of relevant data which we have collected in order to arrive at posterior beliefs, from which all decisions and inferences are made.

The Posterior Distribution

Suppose that you are interested in the values of k unknown quantities

$$\theta = (\theta_1, \theta_2, ..., \theta_k) ,$$

(where k can be one or more than one) and that you have some *a priori* beliefs about their values which you can express in terms of the p.d.f $\pi(\theta)$.

Now suppose that you obtain some data relevant to their values. More precisely, suppose that we have n observations

$$\mathbf{X} = (X_1, X_2, ..., X_n) ,$$

which have a probability distribution that depends on these k unknown quantities as parameters, so that the p.d.f (continuous or discrete) of the vectors $\mathbf{X}$ depends on the vectors θ in a known way. Usually the components of θ and $\mathbf{X}$ will be integers or real numbers, so that the components of $\mathbf{X}$ are random variables, and so the dependence of $\mathbf{X}$ on θ can be expressed in terms of a p.d.f. $f(\mathbf{X}|\theta)$.

You then want to find a way of expressing your beliefs about θ taking into account both your prior beliefs and the data. Of course, it is possible that prior beliefs about θ may differ, but very often these agree on the way in which the data is related to θ [that is, on the form of $f(\mathbf{X}|\theta)$]. If this is so, posterior beliefs will differ (i.e. beliefs after the data has been obtained), but it will turn out that if enough data is collected, then posterior beliefs will usually become very close.

The basic tool is Bayes' theorem for random variables (generalized to deal with

random vectors). From this theorem we know that

$$f(\theta|\mathbf{X}) \propto \pi(\theta) f(\mathbf{X}|\theta).$$

Now we know that $f(\mathbf{X}|\theta)$ considered as a function of $\mathbf{X}$ for fixed θ is a density, but we will find that we often want to think of it as a function of θ for fixed $\mathbf{X}$. When we think of it in that way it does not have quite the same properties - for example, there is no reason why it should sum (or integrate) to unity. Thus in the extreme case where $f(\mathbf{X}|\theta)$ turns out not to depend on θ, then it is easily seen that it can quite well sum (or integrate) to ∞. When we are thinking of $f(\mathbf{X}|\theta)$ as a function of θ we call it the *likelihood* function. We sometimes write

$$l(\theta|\mathbf{X}) = f(\mathbf{X}|\theta).$$

Sometimes it is more natural to consider the log-likelihood function

$$L(\theta|\mathbf{X}) = \log l(\theta|\mathbf{X}).$$

We note that, because of the way we write Bayes' theorem with a porportionality sign, it does not alter the result if we multiply $l(\theta|\mathbf{X})$ by any constant or indeed more generally by any term which is a function of $\mathbf{X}$ alone. Accordingly, we can regard the definition of the likelihood as being any constant multiple of $f(\mathbf{X}|\theta)$ rather than necessarily equalling $f(\mathbf{X}|\theta)$ (and similarly the log-likelihood is undetermined up to an additive constant).

Sometimes the integral

$$\int l(\theta|\mathbf{X}) d\theta$$

(interpreted as a multiple integral $\int \int ... \int ... d\theta_1 d\theta_2 ... d\theta_k$ if $k > 1$ and interpreted as a summation or multiple summation in the discrete case), taken over the admissible

range of θ, is finite, although we have already noted that this is not always the case. When it is, it is occasionally convenient to refer to the quantity

$$\frac{l(\theta|\mathbf{X})}{\int l(\theta|\mathbf{X})d\theta}.$$

We shall call this the *standardized likelihood*, that is, the likelihood scaled so that the area, volume or hypervolume under the curve, surface or hypersurface is unity. To learn more about Bayesian statistical inference [see Berger (1989) and Lee (1999)].

Chapter 4

State Space Representation of the Inflation Model

4.1 Inflation Model

Empirical evidence has shown that the mean level of inflation may not be constant over certain periods [see Huber (1997)] which contradicts Wilkie's constant mean model. In this thesis we consider a model proposed by Arsad (1999), where the mean level of inflation is assumed to follow an AR(1)process.

Let μ_Q in equation (1.2.1) changes with time, so we can replace μ_Q with μ_{Qt}. That is

$$i_t = \mu_{Qt-1} + \alpha_1(i_{t-1} - \mu_{Qt-1}) + \sigma_{Q1}\epsilon_{Q1t} \,, \tag{4.1.1}$$

and suppose μ_{Qt} is also an $AR(1)$ process with mean μ, that is

$$\mu_{Qt} = \mu + \alpha_2(\mu_{Qt-1} - \mu) + \sigma_{Q2}\epsilon_{Q2t} \ . \tag{4.1.2}$$

Where ϵ_{Q2t} is an i.i.d unit standard normal variate. Let $\theta_{1t} = i_t - \mu$ and $\theta_{2t} = \mu_{Qt} - \mu$, then from equation (4.1.2) we get

$$\theta_{2t} = \alpha_2\theta_{2t-1} + \sigma_{Q2}\epsilon_{Q2t} \ , \tag{4.1.3}$$

and we can rewrite equation (4.1.1) as

$$i_t - \mu = \mu_{Qt} - \mu + \alpha_1(i_{t-1} - \mu + \mu - \mu_{Qt}) + \sigma_{Q1}\epsilon_{Q1t} \ ,$$

with

$$\theta_{1t} = \theta_{2t-1} + \alpha_1(\theta_{1t-1} - \theta_{2t-1}) + \sigma_{Q1}\epsilon_{Q1t} \ ,$$

which gives

$$\theta_{1t} = \alpha_1\theta_{1t-1} + (1 - \alpha_1)\theta_{2t-1} + \sigma_{Q1}\epsilon_{Q1t} \ . \tag{4.1.4}$$

Equations (4.1.3) and (4.1.4) can be written in the state space form as

$$\begin{pmatrix} \theta_{1t} \\ \theta_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_1 & 1-\alpha_1 \\ 0 & \alpha_2 \end{pmatrix} \begin{pmatrix} \theta_{1t-1} \\ \theta_{2t-1} \end{pmatrix} + \begin{pmatrix} \sigma_{Q1} & 0 \\ 0 & \sigma_{Q2} \end{pmatrix} \begin{pmatrix} \epsilon_{Q1t} \\ \epsilon_{Q2t} \end{pmatrix} \ , \tag{4.1.5}$$

where

$$\theta_t = \begin{pmatrix} \theta_{1t} \\ \theta_{2t} \end{pmatrix} , \ \Omega_t = \begin{pmatrix} \alpha_1 & 1-\alpha_1 \\ 0 & \alpha_2 \end{pmatrix} , \ \epsilon_t = \begin{pmatrix} \sigma_{Q1}\epsilon_{Q1t} \\ \sigma_{Q2}\epsilon_{Q2t} \end{pmatrix} \tag{4.1.6}$$

and

$$i_t = (1, 0)\theta_t + \mu. \tag{4.1.7}$$

4.2 Parameter Estimation

As discussed in Chapter 3. The likelihood function for a univariate model can be written as

$$\log L = -\frac{T}{2}\log 2\pi - \sum_{t=1}^{T}\log|\Omega_t| - \frac{1}{2}\sum_{t=1}^{T}\hat{\nu}_t'\Omega_t^{-1}\hat{\nu}_t. \tag{4.2.8}$$

Arsad (1999) proposes a Bayesian approach by incorporating prior distributions for elements of Ω (that is α_1 and α_2) and the variances σ_1^2, σ_2^2. His choice of priors are beta(2,2) distribution for α_1 and α_2. While σ_1^2 and σ_2^2 follows gamma(2,5000) and gamma(2,20000) respectively.

The probability density function for α_1 is given by

$$f(\alpha_1) = \begin{cases} \frac{\alpha_1(1-\alpha_1)}{B(2,2)}, & \text{if } 0 \le \alpha_1 \le 1 \\ 0, & \text{if } \alpha_1 < 0 \text{ or } \alpha_1 > 1 \end{cases} \tag{4.2.9}$$

The value of B(2,2) can be calculated from the formula

$$B(r,s) = \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)}, \tag{4.2.10}$$

where $\Gamma(u)$ is the gamma function. The probability density function for α_2 is

$$f(\alpha_2) = \begin{cases} \frac{\alpha_2(1-\alpha_2)}{B(2,2)}, & \text{if } 0 \le \alpha_2 \le 1 \\ 0, & \text{if } \alpha_2 < 0 \text{ or } \alpha_2 > 1 \end{cases} \tag{4.2.11}$$

While the p.d.f. for σ_1^2 is given by

$$f(\sigma_1^2) = \begin{cases} \frac{(5000)^2\sigma_1^2 e^{-5000\sigma_1^2}}{\Gamma(2)}, & \text{if } \sigma_1^2 \ge 0 \\ 0, & \text{if } \sigma_1^2 < 0 \end{cases} \tag{4.2.12}$$

and the p.d.f. for σ_2^2 is

$$f(\sigma_2^2) = \begin{cases} \frac{(20000)^2 \sigma_2^2 e^{-20000\sigma_2^2}}{\Gamma(2)}, & \text{if } \sigma_2^2 \geq 0 \\ 0, & \text{if } \sigma_2^2 < 0 \end{cases} \tag{4.2.13}$$

Thus the posterior density is

$$L(\alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2 | i) \propto \frac{1}{(2\pi)^{\frac{T}{2}} \prod_{t=1}^{T} |\Omega_t|^{\frac{1}{2}}} e^{-\frac{1}{2}\sum_{t=1}^{T} \hat{\nu}_t' \Omega_t^{-1} \hat{\nu}_t}$$

$$\times \alpha_1(1-\alpha_1)\alpha_2(1-\alpha_2)\sigma_1^2 e^{-5000\sigma_1^2} \sigma_2^2 e^{-20000\sigma_2^2} \tag{4.2.14}$$

The log-posterior density is hence

$$\log L(\alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2 | i) = -\frac{1}{2}\sum_{t=1}^{T} \log |\Omega_t| - \frac{1}{2}\sum_{t=1}^{T} \hat{\nu}_t' \Omega_t^{-1} \hat{\nu}_t + \log \alpha_1$$

$$+ \log(1-\alpha_1) + \log \alpha_2 + \log(1-\alpha_2)$$

$$+ \log \sigma_1^2 - 5000\sigma_1^2 + \log \sigma_2^2 - 20000\sigma_2^2 + constant. \tag{4.2.15}$$

Where the constant of proportionality is equal to $\log K$ and K is given by

$$K = \frac{1}{\int_0^1 \int_0^1 \int_0^\infty \int_0^\infty \frac{1}{(2\pi)^{\frac{T}{2}} \prod_{t=1}^{T} |\Omega_t|^{\frac{1}{2}}} e^{-\frac{1}{2}\sum_{t=1}^{T} \hat{\nu}_t' \Omega_t^{-1} \hat{\nu}_t} \alpha_1(1-\alpha_1)\alpha_2(1-\alpha_2)}$$

$$\times \sigma_1^2 e^{-5000\sigma_1^2} \sigma_2^2 e^{-20000\sigma_2^2} d\alpha_1 d\alpha_2 d\sigma_1^2 d\sigma_2^2 \tag{4.2.16}$$

4.3 Results

The parameters of both models are estimated using the SAS software. We call the model using the Bayesian approach Model II and the other using the classical method Model I. The data set used is a simulated inflation data with 200 repetitions. The graph of the data set is shown in figure 4.1. A summary of the results obtained is given in table 4.1.

The first column contains the parameters to be estimated. In the second column we have the exact values of the parameters. While in columns 3 and 4 we have the estimates of the parameters using classical and Bayesian approach respectively.

As can been seen from the results, there is no significant difference in using either method. However, the Bayesian approach tends to be closer to the true values (except in the estimate of μ) than the classical method. An explanation could be that an appropriate prior distribution for the initial state is been specified, as a result the Kalman filter yields an almost exact likelihood function of the observations.

The model thus obtained can be use for filtering, prediction and smoothing depending on the interest of the user.

Parameters	Exact	Model I	Model II
α_1	0.4	0.571178	0.5020872
α_2	0.5	0.418059	0.4993201
σ_1	0.1	0.146985	0.1187586
σ_2	0.2	0.11782	0.14915
μ	0.5	0.115363	0.1170604

Table 4.1: A summary of Estimation Results

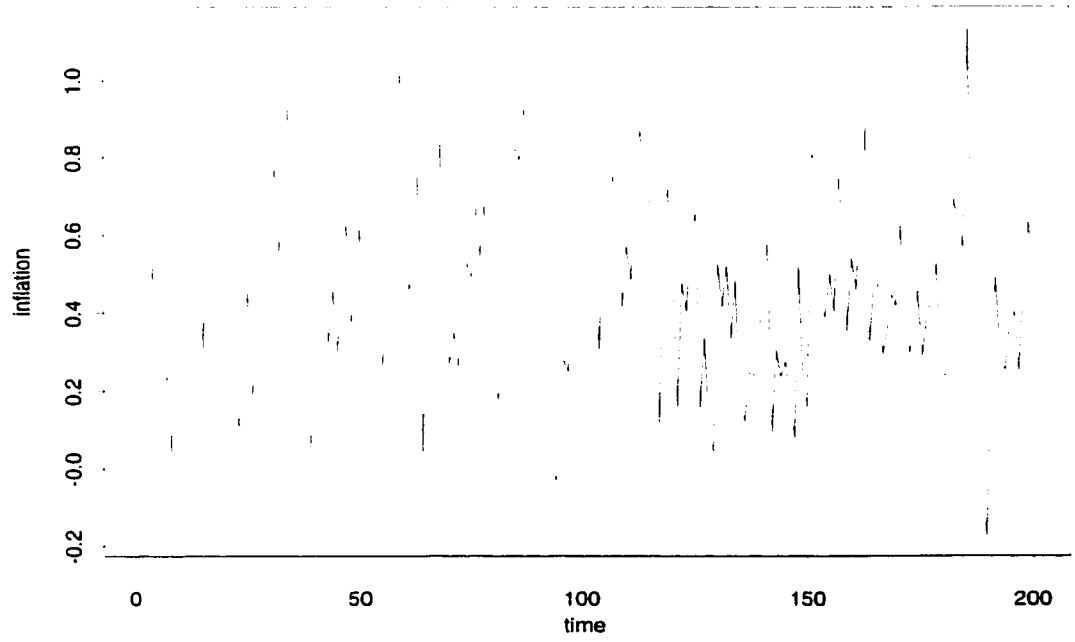

Figure 4.1: A graphical representation of the inflation data

Conclusions

In this thesis we investigated the model for the rate of inflation proposed by Arsad, which is an extension of the Wilkie AR(1) model. The proposed extension is to allow the mean of the inflation in Wilkie AR(1) process to change according to an AR(1) process. The use of the state space model formulation and the application of the Kalman filter has the advantage that it allows the mean reversion level of inflation to be handled as unobservable variables. We compared results obtained when the prior initial state is assumed known to that without any prior knowledge of the initial state. Our results suggest that there is no significant difference between both methods. especially if appropriate prior distributions are used.

Bibliography

[1] Arsad. Z. (1999). *Models for inflation using Kalman filter technique. Proceedings of the 3rd Insurance Mathematics and Economics Congress. London. July 1999.*

[2] Arsad. Z. (2000). *Models for inflation using Kalman filter technique. Unpublished Ph.D thesis. Heriot-Watt University. UK.*

[3] Berger. J.O. (1989).*Statistical Decision Theory: Foundations. Concepts. and Methods. Springer-Verlag. third edition. New York.*

[4] Brown. P.G and Hwang. P.Y.C. (1992). *Introduction to random signals and applied Kalman Filtering. 2nd Edition. J. Wiley. New York.*

[5] Engle. R.F. (1982). *Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. Econometrica.* **50**. *987.*

[6] Harvey. A.C. (1989). *Forecasting. structural time series models and the kalman filter. Cambridge University Press.*

[7] Harvey. A.C. (1990). *The econometric analysis of time series. 2nd edition. The MIT Press. Boston.*

[8] Huber. P.P. (1997). *A review of Wilkie's stochastic asset model. British Actuarial Journal. 3.***I**. *181-210.*

[9] Kalman. R.E. (1960). *A new approach to linear filtering and prediction problems. Journal of Basic Engineering. March. 35-45.*

[10] Lee. P.M. (1999). *Bayesian Statistics: An introduction. Oxford University Press.*

[11] Phadke. M.S. (1981). *Quality Audit Using Adaptive Kalman Filtering. ASQC Quality Congress Transactions. San Francisco. 1045-1052.*

[12] Wilkie. A.D. (1986). *A stochastic investment model for actuarial use. Transaction of the faculty of actuaries. 39. 341-403*

[13] Wilkie. A.D. (1995). *More on the stochastic asset model for actuarial use. British Actuarial Journal* **I**. *777-964.*

Appendix

A1: Model using Classical Analysis (Model I)

```
/*****************************************************************/
/*                  S A S   PROGRAM                              */
/*                                                               */
/*   TITLE: KALMAN FILTER MODEL FOR INFLATION                    */
/*   MODEL: MODEL I                                              */
/*    DATA: SIMULATED DATA SET                                    */
/*****************************************************************/

title 'Likelihood Evaluation of SSM';

%let DATASET = WORK.TEST;
%let ALPHA1   = 0.4;
%let ALPHA2    = 0.5;
%let MU     = 0.5;
```

```
%let SIGMA1  = 0.1;

%let SIGMA2  = 0.2;

%let NOBS    = 200;

%let BETA     =  0.9999;


/*-------------------------------------------------------------*/

/*- create test data set ----------------------------------------*/

/*-------------------------------------------------------------*/

data &DATASET;


/*- initialize random seeds -------------------------------------*/

    seed1  = 123456;

    seed2  = 654321;


/*- initialize state variables ----------------------------------*/

    lagmut = 0;

    lagy   = 0;




    do time = -100 to &NOBS;


    /*- normally distributed random components ------------------*/

        epsilon1 = rannor( seed1  )*&SIGMA1;
```

```
        epsilon2 = rannor( seed2  )*&SIGMA2;



    /*-  states equation   ----------------------------------*/

        mut   = &MU + &ALPHA2*(lagmut - &MU) + epsilon1;


    /*- observation equaition ----------------------------------------*/

        y = lagmut  + &ALPHA1*(lagy -lagmut) + epsilon2;


lagmut   = mut;

lagy = y;

if time > 0 then output;


    end;

    drop seed1 seed2 lagmut lagy;


run;

/*-------------------------------------------------------------------------------------

proc print data=test; run;


proc iml;


/*-----------------------------------------------------------------*/
```

```
/*- make  state transition matrix -- ----------------------   -*/
/*-------------------------------------------------------------*/
    start make_ome( alpha1, alpha2 );

        ome = j( 2, 2, 0 );
           ome[1, 1] =  alpha1;
           ome[1, 2] =  1 - alpha1;
           ome[2, 2] =  alpha2;
           omegat=ome;

        return( omegat );

    finish make_ome;

/*-------------------------------------------------------------*/
/*- make variance matrix -- ---------------------------------*/
/*-------------------------------------------------------------*/
    start make_var( sigma1,sigma2 );

        vec = j(3,3,0);
        vec[1,1] = (sigma1)**2;
        vec[2,2] = (sigma2)**2;
        vec[3,3] = (sigma2)**2;
        var = vec;
```

```
        return( var );

    finish make_var;

/*-----------------------------------------------------------------*/
/*------------ compute the log-likelihood                  ---- -*/
/*-----------------------------------------------------------------*/
    start lik(y,a,b,omega,h,var,z0,vz0);
        nz = ncol(h);  /*- number of state variables       -*/
        ny = nrow(y);  /*- number of observations          -*/
        k  = ncol(y);  /*- number of measurement variables -*/
        const = k*log(8*atan(1));
        if ( sum(z0=.) | sum(vz0 = . ) ) then
           call kalcvf(pred,vpred,filt,vfilt,y,0,a,omega,b,h,var);
        else
           call kalcvf(pred,vpred,filt,vfilt,y,0,a,omega,b,h,var,z0,vz0);
        et = y - pred*h';
        sum1 = 0;
        sum2 = 0;
        do i = 1 to ny;
           vpred_i = vpred[(i-1)*nz+1:i*nz,];
           et_i = et[i,];
           omegat = h*vpred_i*h' + var[nz+1:nz+k,nz+1:nz+k];
```

```
            sum1 = sum1 + log(det(omegat));
            sum2 = sum2 + et_i*inv(omegat)*et_i';
        end;
        return(-const-.5*(sum1+sum2)/ny);
    finish lik;


/*-----------------------------------------------------------------*/
/*- objective function                                            -*/
/*-----------------------------------------------------------------*/
    start fun( parm ) global(y);

        alpha1   = parm[1];
        alpha2   = parm[2];
       sigma1    = parm[3];
       sigma2    = parm[4];
           mu     = parm[5];
    /*- create the state-space model system matrices     ----------*/
    /*- you can make the time-invariant or time varying  ----------*/
        T   = nrow(y);
        omega = make_ome( alpha1, alpha2 );
        a   = j(nrow(omega),1,mu);

        h   = {1 0};
```

```
        var = make_var( sigma1, sigma2 );
        b   = j(nrow(h),1,0);



    /*- starting values -- these can be tweaked -------------------*/
        z0 = j(1,nrow(omega),0);
        vz0 = 1e-2# I(nrow(omega));



    /*- compute the log-likelihood --------------------------------*/
        logl = lik(y,a,b,omega,h,var,z0,vz0);
        return(logl);

    finish fun;

/*-----------------------------------------------------------------*/
/*- read the data set                                             -*/
/*-----------------------------------------------------------------*/
    use &DATASET;
    read all var {y};
    T = nrow(y);

/*-----------------------------------------------------------------*/
```

```
/*- starting value for optimization                                    -*/
/*---------------------------------------------------------------------*/
    x0    = j(1,5,0);
    x0[1] = &ALPHA1;
    x0[2] = &ALPHA2;
    x0[3] = &SIGMA1;
    x0[4] =   0.1 ;
    x0[5] = 0.4;
/*---------------------------------------------------------------------*/
/*- constraints                                                        -*/
/*  0<alpha1<1, 0<alpha2<1, sigma1>0, sigma2>0, -inf<mu<inf */
/*---------------------------------------------------------------------*/
    blc = {-&BETA  -&BETA  1e-6   1e-6  -1e10       . .,
            &BETA  &BETA  1e10  1e10  1e10   . .,
            .    .     .      .    .    . .,
            .    .     .      .    .    . . };

/*---------------------------------------------------------------------*/
/*- options -- maximize                                                -*/
/*---------------------------------------------------------------------*/
    opt = j(1,11,.); opt[1] = 1; opt[2] = 4;

/*---------------------------------------------------------------------*/
/*- optimize objective function using conjugate-gradient method   -*/
```

```
/*- other NLP* calls may be used                                    -*/
/*-----------------------------------------------------------------*/
    call nlpcg( rc, xr, "FUN", x0 ) blc=blc opt=opt;
    print rc xr;

quit;
```

Output of the Program for Model I

Optimization Start

Parameter Estimates

Parameter		Estimate	Gradient	Lower BC	Upper BC
1	X1	0.400000	-6.54225	-0.99990	0.99990
2	X2	0.500000	-7.01796	-0.99990	0.99990
3	X3	0.100000	47.78141	1E-6	1E10
4	X4	0.100000	85.19007	1E-6	1E10
5	X5	0.400000	-46.57058	-1E10	1E10

Value of Objective Function = -7.190203525

Conjugate-Gradient Optimization

*** Termination Criteria ***

Minimum Iterations	0
Maximum Iterations	400
Maximum Function Calls	1000
ABSGCONV Gradient Criterion	0.0000100
GCONV Gradient Criterion	1E-8
ABSFCONV Function Criterion	0
FCONV Function Criterion	2.2204E-16
FCONV2 Function Criterion	0
FSIZE Parameter	0
ABSXCONV Parameter Change Criterion	0
XCONV Parameter Change Criterion	0
XSIZE Parameter	0
ABSCONV Function Criterion	1.3408E154

*** Other Control Parameters ***

Line Search Method 2: Starting Alpha	1.00000
Line Search Precision LSPRECISION	0.10000
DAMPSTEP Parameter for Line Search	.
MAXSTEP Parameter for Line Search	0
FD Derivatives: Accurate Digits in Obj.F.	15.65356
Singularity Tolerance (SINGULAR)	1E-8
Constraint Precision (LCEPS)	1E-8

```
Linearly Dependent Constraints (LCSING) . . . . .       1E-8

Releasing Active Constraints (LCDEACT)  . . . . .          .


                    Conjugate-Gradient Optimization

              Automatic Restart Update (Powell, 1977; Beale, 1972)

                    Gradient Computed by Finite Differences

                       Number of Parameter Estimates 5

                           Number of Lower Bounds 5

                           Number of Upper Bounds 5


Optimization Start: Active Constraints= 0  Criterion= -7.190 Maximum Gr

Optimization Results: Iterations= 71 Function Calls= 159 Gradient Calls= 94

Active Constraints= 0  Criterion= -0.76904823

Maximum Gradient Element= 0.0000821316 Slope= -1.58998E-7


NOTE:  GCONV convergence criterion satisfied.


                        Optimization Results

                        Parameter Estimates

------------------------------------------------------------

Parameter          Estimate     Gradient    Active BC

------------------------------------------------------------
```

1	X1	0.571178	-0.0000708
2	X2	0.418059	-0.0000821
3	X3	0.146985	-0.0000339
4	X4	0.117819	-0.0000790
5	X5	0.115363	0.0000230

Value of Objective Function = -0.769048227

RC	XR				
6	0.5711784	0.4180591	0.1469846	0.117819	0.1153634

A2: Model Using a Bayesian Approach (Model II)

```
/*****************************************************************/
/*          S A S   PROGRAM                                      */
/* TITLE: KALMAN FILTER MODEL FOR INFLATION                      */
/* MODEL: MODEL II                                               */
/* DATA:  SIMULATED DATA SET                                     */
 /*****************************************************************/


title 'Likelihood Evaluation of SSM';


%let DATASET = WORK.TEST;
%let ALPHA1   = 0.4;
%let ALPHA2    = 0.5;
%let MU     = 0.5;
%let SIGMA1  = 0.1;
%let SIGMA2  = 0.2;
%let NOBS    = 200;
%let BETA     =  0.9999;


/*---------------------------------------------------------------*/
/*- create test data set ------------------------------------------*/
/*---------------------------------------------------------------*/
data &DATASET;
```

```
/*- initialize random seeds ------------------------------------*/
    seed1  = 123456;
    seed2  = 654321;

/*- initialize state variables ---------------------------------*/
    lagmut = 0;
    lagy   = 0;



    do time = -100 to &NOBS;

    /*- normally distributed random components ------------------*/
        epsilon1 = rannor( seed1  )*&SIGMA1;
        epsilon2 = rannor( seed2  )*&SIGMA2;



    /*-  states equation   -----------------------------------*/
        mut   = &MU + &ALPHA2*(lagmut - &MU) + epsilon1;

    /*- observation equaition ---------------------------------------*/
        y = lagmut  + &ALPHA1*(lagy -lagmut) + epsilon2;

lagmut   = mut;
```

```
lagy = y;
if time > 0 then output;

    end;
    drop seed1 seed2 lagmut lagy;

run;
/*-----------------------------------------------------------------*/

proc print data=test; run;

proc iml;

/*-----------------------------------------------------------------*/
/*- make  state transition matrix -- ----------------------   -*/
/*-----------------------------------------------------------------*/
    start make_ome( alpha1, alpha2 );

        ome = j( 2, 2, 0 );
           ome[1, 1] =  alpha1;
           ome[1, 2] =  1 - alpha1;
           ome[2, 2] =  alpha2;
           omegat=ome;
```

```
        return( omegat );

    finish make_ome;

/*-----------------------------------------------------------------*/
/*- make variance matrix -- ----------------------------------*/
/*-----------------------------------------------------------------*/
    start make_var( sigma1,sigma2 );

        vec = j(3,3,0);
        vec[1,1] = (sigma1)**2;
        vec[2,2] = (sigma2)**2;
        vec[3,3] = (sigma2)**2;
        var = vec;

        return( var );

    finish make_var;

/*-----------------------------------------------------------------*/
/*------------ compute the log-likelihood                  ---- -*/
/*-----------------------------------------------------------------*/
    start lik(y,a,b,omega,h,var,z0,vz0);
        nz = ncol(h);  /*- number of state variables       -*/
```

```
ny = nrow(y);  /*- number of observations          -*/
k  = ncol(y);  /*- number of measurement variables -*/
const = k*log(8*atan(1));
if ( sum(z0=.) | sum(vz0 = . ) ) then
   call kalcvf(pred,vpred,filt,vfilt,y,0,a,omega,b,h,var);
else
   call kalcvf(pred,vpred,filt,vfilt,y,0,a,omega,b,h,var,z0,vz0);
et = y - pred*h';
sum1 = 0;
sum2 = 0;
x = {1,0};
yy = {0,1};
xx = {1,0,0};
yyy = {0,1,0};
alpha1 = x' * omega * x;
alpha2 = yy' * omega * yy;
sigma1 = xx' * var * xx;
sigma2 = yyy' * var * yyy;
do i = 1 to ny;
   vpred_i = vpred[(i-1)*nz+1:i*nz,];
   et_i = et[i,];
   omegat = h*vpred_i*h' + var[nz+1:nz+k,nz+1:nz+k];
   sum1 = sum1 + log(det(omegat));
   sum2 = sum2 + et_i*inv(omegat)*et_i';
```

```
        end;
 return(-const-.5*(sum1+sum2)/ny + log(alpha1) + log(1- alpha1) + log(alpha2)
 + log(1 - alpha2) + log(sigma1**2) + log(sigma2**2)- 5000*(sigma1**2)
 - 2000*(sigma2**2) + 167.27024);
    finish lik;



/*-----------------------------------------------------------------*/
/*- objective function                                            -*/
/*-----------------------------------------------------------------*/
    start fun( parm ) global(y);

        alpha1   = parm[1];
        alpha2   = parm[2];
       sigma1    = parm[3];
       sigma2    = parm[4];
           mu     = parm[5];
    /*- create the state-space model system matrices     ----------*/
    /*- you can make the time-invariant or time varying  ----------*/
        T   = nrow(y);
        omega = make_ome( alpha1, alpha2 );
        a   = j(nrow(omega),1,mu);

        h   = {1 0};
```

```
        var = make_var( sigma1, sigma2 );
        b   = j(nrow(h),1,0);


    /*- starting values -- these can be tweaked ------------------*/
        z0 = j(1,nrow(omega),0);
        vz0 = 1e-3 # I(nrow(omega));


    /*- compute the log-likelihood -------------------------------*/
        logl = lik(y,a,b,omega,h,var,z0,vz0);
        return(logl);

    finish fun;

/*-----------------------------------------------------------------*/
/*- read the data set                                            -*/
/*-----------------------------------------------------------------*/
    use &DATASET;
    read all var {y};
    T = nrow(y);

/*-----------------------------------------------------------------*/
```

```
/*- starting value for optimization                                    -*/
/*--------------------------------------------------------------------*/
    x0    = j(1,5,0);
    x0[1] = &ALPHA1;
    x0[2] = &ALPHA2;
    x0[3] = &SIGMA1;
    x0[4] =   0.1 ;
    x0[5] = 0.4;
/*--------------------------------------------------------------------*/
/*- constraints                                                       -*/
/*  0<alpha1<1, 0<alpha2<1, sigma1>0, sigma2>0, -inf<mu<inf */
/*--------------------------------------------------------------------*/
    blc = {-&BETA  -&BETA  1e-6   1e-6  -1e10       . .,
            &BETA  &BETA  1e10  1e10  1e10   . .,
            .    .     .      .    .    . .,
            .    .     .      .    .    . . };

/*--------------------------------------------------------------------*/
/*- options -- maximize                                               -*/
/*--------------------------------------------------------------------*/
    opt = j(1,11,.); opt[1] = 1; opt[2] = 4;

/*--------------------------------------------------------------------*/
/*- optimize objective function using conjugate-gradient method   -*/
```

```
/*- other NLP* calls may be used                                    -*/
/*-----------------------------------------------------------------*/
    call nlpcg( rc, xr, "FUN", x0 ) blc=blc opt=opt;
    print rc xr;

quit;
```

Output of the Program for Model II

Optimization Start

Parameter Estimates

	Parameter	Estimate	Gradient	Lower BC	Upper BC
1	X1	0.400000	-5.69964	-0.99990	0.99990
2	X2	0.500000	-7.04350	-0.99990	0.99990
3	X3	0.100000	67.82400	1E-6	1E10
4	X4	0.100000	118.31079	1E-6	1E10
5	X5	0.400000	-46.59158	-1E10	1E10

Value of Objective Function = 138.10735097

Conjugate-Gradient Optimization

*** Termination Criteria ***

Minimum Iterations	0
Maximum Iterations	400
Maximum Function Calls	1000
ABSGCONV Gradient Criterion	0.0000100
GCONV Gradient Criterion	1E-8
ABSFCONV Function Criterion	0
FCONV Function Criterion	2.2204E-16
FCONV2 Function Criterion	0
FSIZE Parameter	0
ABSXCONV Parameter Change Criterion	0
XCONV Parameter Change Criterion	0
XSIZE Parameter	0
ABSCONV Function Criterion	1.3408E154

*** Other Control Parameters ***

Line Search Method 2: Starting Alpha	1.00000

Line Search Precision LSPRECISION	0.10000

```
DAMPSTEP Parameter for Line Search . . . . . .
MAXSTEP Parameter for Line Search . . . . . 0
FD Derivatives: Accurate Digits in Obj.F. . . . . 15.65356
Singularity Tolerance (SINGULAR) . . . . . 1E-8
Constraint Precision (LCEPS) . . . . . 1E-8
Linearly Dependent Constraints (LCSING) . . . . . 1E-8
Releasing Active Constraints (LCDEACT) . . . . . .
```

Conjugate-Gradient Optimization

Automatic Restart Update (Powell, 1977; Beale, 1972)

Gradient Computed by Finite Differences

Number of Parameter Estimates 5

Number of Lower Bounds 5

Number of Upper Bounds 5

Optimization Start: Active Constraints= 0 Criterion= 138.107

Maximum Gradient Element= 118.311

Optimization Results: Iterations= 21 Function Calls= 59 Gradient Calls= 28

Active Constraints= 0 Criterion= 145.59069

Maximum Gradient Element= 0.00269949 Slope= -0.00160718

NOTE: GCONV convergence criterion satisfied.

NOTE: At least one element of the (projected) gradient is greater than 1e-3.

Optimization Results

Parameter Estimates

	Parameter	Estimate	Gradient	Active BC
1	X1	0.502087	0.0003492	
2	X2	0.499320	-0.00270	
3	X3	0.118759	-0.0000563	
4	X4	0.149150	0.0006025	
5	X5	0.117060	0.0005157	

Value of Objective Function = 145.59068903

RC	XR				
6	0.5020872	0.4993201	0.1187586	0.14915	0.1170604