Doctoral Thesis:

Nonlinear, Non-Gaussian, and Non-stationary State Space Models and Applications to Economic and Financial Time Series

Koiti Yano [1]

June 20, 2007

[1]Koiti Yano, Department of Statistical Science, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies. koiti.yano@gmail.com.

The author would like to deeply appreciate Seisho Sato and Yoshiyasu Tamura, Akihiko Takahashi, Satoshi Yamashita, and Hajime Wago for their helpful comments and so kind supports. This paper is indebted to Tomoyuki Higuchi, Yukito Iba, HC. Jimbo, Genshiro Kitagawa, Sha Gyo Kyo, Yoshihiko Ogata, Toru Ozaki, Takashi Tsuchiya, Kevin Wong, Siew Hai Yen, two anonymous referees and an associate editor of Computatioanl Statistics for their helpful comments and supports. The author also appreciates all professors and all staffs in the Institute of Statistical Mathematics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Nonlinear, Non-Gaussian, and Non-stationary State Space Model and Economic and Financial Time Series

Financial markets and the economy are changing rapidly. On financial markets, many financial time series exhibit changes of volatility (variance) over time. Moreover, many financial time series are well known to have non-Gaussian heavy-tailed distributions. These facts indicate that a nonlinear non-Gaussian time series analysis is needed. Regarding the economy, as one example, the Japanese economy has the experience of the "bubble economy" in the late 1980s. After bursting of the "bubble economy", the economy entered a decade of economic stagnation, which is often called "the lost decade". These facts indicate that conventional linear regression based on ordinary least squares might be ineffective to analyze a non-stationary

economy because the coefficients of linear regression are fixed. This paper shows several statistical approaches based on nonlinear non-Gaussian state space modeling and time-varying coefficient autoregressive modeling. These approaches are novel studies of financial markets and the economy.

In this chapter 1, the Monte Carlo filter are introduced. It is a minimal introduction to nonlinear non-Gaussian state-space modeling.

In chapter 2, we propose a method to seek initial distributions of parameters for a self-organizing state space model proposed by [Kit98]. Our method is based on the simplex Nelder-Mead algorithm for solving nonlinear and discontinuous optimization problems. We show the effectiveness of our method by applying it to a linear Gaussian model, a linear non-Gaussian model, a nonlinear Gaussian model, and a stochastic volatility model.

In chapter 3, we propose a smoothing algorithm based on the Monte Carlo filter and the inverse function of a system equation (an inverse system function). Our method is applicable to any nonlinear non-Gaussian state space model if an inverse system equation is given analytically. Moreover, we propose a filter initialization algorithm based on a smoothing distribution obtained by our smoothing algorithm and an inverse system equation.

In chapter 4, we illustrate the effectiveness of our approach by applying it to stochastic volatility models and stochastic volatility models with heavy-tailed distributions for the daily return of the Yen/Dollar exchange rate.

In chapter 5, we propose a method that estimates a time-varying linear system equation based on time-varying coefficients' vector autoregressive modeling (time-varying VAR), and which controls the system. In our framework, an optimal feedback is determined using linear quadratic dynamic pro-

gramming in each period. The coefficients of time-varying VAR are assumed to change gradually (this assumption is widely known as smoothness priors of the Bayesian procedure). The coefficients are estimated using the Kalman filter. In our empirical analyses, we show the effectiveness of our approach by applying it to monetary policy, in particular, the inflation targeting of the United Kingdom and the nominal growth rate targeting of Japan. Furthermore, we emphasize that monetary policy must be forecast-based because transmission lags pertain from monetary policy to the economy. Our approach is convenient and effective for central bank practitioners when they are unaware of the true model of the economy. Additionally, we find that the coefficients of time-varying VAR change in response to changes of monetary policy.

In chapter 6, we estimate the β of a single factor model that is often used by financial practitioners. In this chapter, we assume that β changes "gradually" over time; this assumption is identical to that in chapter 5. Using our approach, we can estimate β, even if it is time varying. We apply our approach to the Japanese Stock Markets and show its effectiveness. Although we adopt a very restrictive method (we assume smoothness priors and use the Kalman filter, which is based on linear state space modeling and the Gaussian distribution), we can obtain good estimates of β.

1.2 Monte Carlo Filter

A nonlinear non-Gaussian state space model for the time series $\boldsymbol{y}_t,\ t = \{1, 2, \cdots, T\}$ is defined as

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_s, \boldsymbol{v}_t), \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{\xi}_m, \boldsymbol{\epsilon}_t), \end{aligned} \tag{1.1}$$

where $\boldsymbol{x}_t$ is an unknown $n_x \times 1$ state vector, $\boldsymbol{v}_t$ is the $n_v \times 1$ system noise vector with a density function $q(\boldsymbol{v})$, $\boldsymbol{\epsilon}_t$ is the $n_\epsilon \times 1$ observation noise vector with a density function $r(\boldsymbol{\epsilon})$, $\boldsymbol{\xi}_s$ is the $n_s \times 1$ system parameter vector of the function f, and $\boldsymbol{\xi}_m$ is the $n_m \times 1$ observation parameter vector of the function h. The function $f : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_v} \to \boldsymbol{R}^{n_x}$ is a possibly nonlinear function and the function $h : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_\epsilon} \to \boldsymbol{R}^{n_y}$ is a possibly nonlinear function. The first equation of (1.1) is called a system equation and the second equation is called a measurement equation. This nonlinear non-Gaussian state space model specifies the two following conditional density functions:

$$\begin{aligned} &p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \\ &p(\boldsymbol{y}_t|\boldsymbol{x}_t). \end{aligned} \tag{1.2}$$

We define a parameter vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\xi}_s \\ \boldsymbol{\xi}_m \end{bmatrix}. \tag{1.3}$$

In the state estimation problem of a nonlinear non-Gaussian state space model (e.g. [KG96]), the predictor, filter, and smoother are defined as

$$\begin{aligned} &\text{Predictor: } p(\boldsymbol{x}_t|\boldsymbol{y}_{1:(t-1)}), \\ &\text{Filter: } p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}), \\ &\text{Smoother: } p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}), \end{aligned} \tag{1.4}$$

where $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_t\}$. Most state estimation algorithms are Bayesian recursive tracking; they are based on Bayes' theorem (See [AMGC02]), which is

$$\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:(t-1)})}{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}_{1:(t-1)})},\ t \geq 1, \tag{1.5}$$

where $\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ is the prior probability, $\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is the likelihood, $\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ is the posterior probability, and $\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$ is the normalizing constant. We denote an initial probability $\mathbb{P}(\boldsymbol{x}_1) = \mathbb{P}(\boldsymbol{x}_1|\emptyset)$, where the empty set $\emptyset$ indicates that we have no observations. In the state estimation problem, determining an initial probability $\mathbb{P}(\boldsymbol{x}_1)$, which is called filter initialization, is important because a proper initial probability improves a posterior probability. In general, however, an initial probability is unknown.

In the Monte Carlo filter (the MC filter), the posterior density distribution at time t is approximated as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{1}{\sum_{i=1}^{M} w_t^i} \sum_{i=0}^{M} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i), \tag{1.6}$$

where w_t^i is the weight of a particle $\boldsymbol{x}_t^i$, M is the number of particles, and

δ is the Dirac's delta function [1]. The definition of w_t^i is described below. In the standard algorithm of the MC filter, particles are resampled with sampling probabilities proportional to the weights w_t^i at every time t. After resampling, the weights are reset to $w_t^i = 1/M$. Therefore, Eq. (1.6) is rewritten as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{1}{M}\sum_{i=1}^{M}\delta(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t^i) \tag{1.7}$$

where $\hat{\boldsymbol{x}}_t^i$ are particles after resampling. Using Eq. (1.7), the predictor $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:(t-1)})$ can be approximated by

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:(t-1)}) &= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:(t-1)})d\boldsymbol{x}_{t-1} \\ &= \frac{1}{M}\sum_{i=1}^{M}\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\delta(\boldsymbol{x}_{t-1} - \hat{\boldsymbol{x}}_{t-1}^i)d\boldsymbol{x}_{t-1} \\ &= \frac{1}{M}\sum_{i=1}^{M}p(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t-1}^i) \\ &\simeq \frac{1}{M}\sum_{i=1}^{M}\delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i). \end{aligned} \tag{1.8}$$

Note that $\boldsymbol{x}_t^i$ are obtained from

$$\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t-1}^i). \tag{1.9}$$

[1] The Dirac delta function is defined as

$$\delta(x) = 0, \text{ if } x \neq 0,$$
$$\int_{\infty}^{\infty}\delta(x)dx = 1.$$

Substituting Eq. (1.8) to Eq. (1.5), we obtain the following equation.

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \propto & p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \\ \propto & \frac{1}{M}p(\boldsymbol{y}_t|\boldsymbol{x}_t)\sum_{i=1}^{M}\delta(\boldsymbol{x}_t-\boldsymbol{x}_t^i) \\ = & \frac{1}{M}\sum_{i=1}^{M}p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)\delta(\boldsymbol{x}_t-\boldsymbol{x}_t^i). \end{aligned} \tag{1.10}$$

Comparing Eq. (1.6) and Eq. (1.10) indicates that weights w_t^i are obtain by

$$w_t^i \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^i). \tag{1.11}$$

Therefore, weight w_t^i is defined as

$$w_t^i \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^i) = r(\psi(\boldsymbol{y}_t,\boldsymbol{x}_t^i))\Big|\frac{\partial\psi}{\partial y}\Big|, i=\{1,\cdots,M\}, \tag{1.12}$$

where ψ is the inverse function of the function h. The right hand side of Eq. (1.12) is the likelihood function of a nonlinear non-Gaussian state space model. The algorithm of the MC filter is shown as Algorithm 1.

Algorithm 1: The Monte Carlo Filter

$[\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M, \text{llk}] = \text{MCPfilter}[\{\boldsymbol{x}_{t-1}^i\}_{i=1}^M,\ \boldsymbol{y}_t]$

{

FOR i=1,...M

Predict: $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, v_t^i)$

Weight: $w_t^i = p(\boldsymbol{y}_t|\boldsymbol{x}_t^i, \epsilon)$

ENDFOR

Sum of Weights: $sw = \sum_{i=1}^{M} w_t^i$

Log-Likelihood: $llk = log(sw/M)$

FOR i=1,...,M

Normalize: $\tilde{w}_t^i = \frac{w_t^i}{sw}$

ENDFOR

Resampling: $[\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M]$ =resample$[\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M]$

RETURN$[\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M, llk]$

}

MCPmain $[\{\boldsymbol{x}_0^i\}_{i=1}^M,\ \{\boldsymbol{y}_t\}_{t=1}^T]$ {

Initialize: $llk = 0$

FOR t=1,...,T

mcp = MCPfilter$[\{\boldsymbol{x}_{t-1}^i\}_{i=1}^M,\ \boldsymbol{y}_t]$

$llk = llk + (llk\ in\ $ mcp$)$

$\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M = (\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M\ in\ $ mcp$)$

ENDFOR

RETURN$[\{\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M\}_{t=1}^T, llk]$

}

A precedent work, [Kit96], shows that the likelihood of the parameter vector $\boldsymbol{\theta}$ is approximated by

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{t-1}, \boldsymbol{\theta}) \\ &= \prod_{t=1}^{T} \frac{1}{M} \sum_{i=1}^{M} w_t^i. \end{aligned} \tag{1.13}$$

Therefore, the log-likelihood of the parameter vector $\boldsymbol{\theta}$ is

$$l(\boldsymbol{\theta}) \simeq \sum_{t=1}^{T} \log(\sum_{i=1}^{M} w_t^i) - T \log M, \tag{1.14}$$

where T is the number of observations. This approximation shows that the log-likelihood is affected by sampling error from the Monte Carlo algorithm. In addition, [Kit98] points out that numerous particles are necessary to obtain a closely accurate log-likelihood result [2]. The sampling error of Eq. (1.14) prevents the calculation of accurate maximum likelihood estimates. Moreover, we cannot use optimization algorithms that require derivations of functions because Eq. (1.14) is approximated by the weights of particles. Therefore, we require an alternative approach that allows the estimation of parameters without maximizing the log-likelihood accurately.

If we have several statistical models, the goodness of fit of each model is evaluated using the Akaike Information Criterion (AIC), proposed by [Aka73]. The AIC is defined as

$$\text{AIC} = -2l(\boldsymbol{\theta}^*) + 2q, \tag{1.15}$$

[2] [Hig95] reports the sampling error of Eq. (1.14) in some cases.

where $\boldsymbol{\theta}^*$ is the maximum likelihood estimate and q is the number of unknown parameters.

Chapter 2

A Self-Organizing State Space Model and Simplex Initial Distribution Search

2.1 Introduction

The Monte Carlo (MC) filter is proposed by [GSS93] and [Kit96]. The filter is an algorithm to estimate states for a nonlinear non-Gaussian state space model[1]. In recent years, the filter has been applied to various problems[2]. In spite of its widespread practical application, a problem exists in parameter estimation in the MC filter. It is difficult to estimate parameters accurately using the MC filter. The problem results from the sampling error of the likelihood of the MC filter from the Monte Carlo algorithm. Consequently,

[1] [AMGC02] is a readable tutorial on the Monte Carlo filter.
[2] See [DdFG01].

it is very difficult to obtain accurate maximum likelihood estimates because the likelihood is affected by the sampling error from the Monte Carlo algorithm. [Kit98] points out that numerous particles are necessary to obtain a close accurate likelihood. Moreover, the likelihood of the MC filter is approximated by "particles". Therefore, the likelihood is not differentiable. Therefore, we cannot use function optimization algorithms, which require the derivatives of functions.

To solve the problem, [Kit98] proposes a self-organizing state space model[3]. In [Kit98], a state vector is augmented to include states and parameters. The augmented state vector is estimated using the MC filter. Therefore, states and parameters are estimated simultaneously without maximizing the likelihood of the MC filter. On a self-organizing state space model, however, [HK01] points out a problem: determination of initial distributions of parameters for a self-organizing state space model. The estimated parameters of a self-organizing state space model comprise a subset of the initial distributions of parameters. We must know the posterior distributions of parameters to estimate parameters adequately. However, the posterior distributions of the parameters are generally unknown. Parameter estimation fails if we do not know appropriate their initial distributions. We propose a method to seek initial distributions of parameters for a self-organizing state space model using the simplex Nelder-Mead algorithm to solve the problem. The algorithm solves nonlinear and discontinuous optimization problems. In our approach, the simplex Nelder-Mead algorithm is used to minimize the negative log-likelihood of a self-organizing state space model. Never-

[3]Another excellent approach is proposed by [DT03].

theless, the negative log-likelihood is affected by sampling error from the Monte Carlo algorithm, a self-organizing state space model is able to estimate parameters because minimizing the negative log-likelihood renders the initial distributions as close to "true" parameters. [LWH04] uses the simplex Nelder-Mead method to maximize the likelihood of the MC filter. However, we use the likelihood of a self-organizing state space model. Moreover, in our approach, parameters are estimated using a self-organizing state space model[4].

The motivation of our algorithm is to improve the accuracy of the estimates of parameters rather than [LWH04]. In the algorithm of [LWH04], they maximize the likelihood of the MC filter to obtain the estimates of parameters. However, the likelihood is affected by sampling error from the Monte Carlo algorithm. Thus, the simplex Nelder-Mead method in [LWH04] may not be terminated well or less accurate estimates are obtained. This problem is avoided by our alogrithm bacause parameters are estimated using a self-organizing state space model. In appendix B, we show that the estimation of our algorithm are better than the estimation of [LWH04] .

This paper is organized as follows. In section 2, we describe a self-organizing state space model and our initial distribution search algorithm. In section 3, we show examples for some models. In section 4, we describe this study's salient conclusions.

[4][Kit98] states that the difficulty of obtaining the close approximation of the likelihood motivates the development of a self-organizing state space model.

2.2 Model

2.2.1 Monte Carlo Filter

A nonlinear non-Gaussian state space model for the time series $\boldsymbol{y}_t,\ t = \{1, 2, \cdots, T\}$ is defined as follows.

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_s, \boldsymbol{v}_t), \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{\xi}_m, \boldsymbol{\epsilon}_t), \end{aligned} \tag{2.1}$$

where $\boldsymbol{x}_t$ is an unknown $n_x \times 1$ state vector, $\boldsymbol{v}_t$ is $n_v \times 1$ system noise vector with a density function $q(\boldsymbol{v})$ [5], $\boldsymbol{\epsilon}_t$ is $n_\epsilon \times 1$ observation noise vector with a density function $r(\boldsymbol{\epsilon})$, $\boldsymbol{\xi}_s$ is the $n_s \times 1$ system parameter vector of the function f, and $\boldsymbol{\xi}_m$ is the $n_m \times 1$ observation parameter vector of the function h. The function $f : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_v} \rightarrow \boldsymbol{R}^{n_x}$ is a possibly nonlinear function and the function $h : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_\epsilon} \rightarrow \boldsymbol{R}^{n_y}$ is a possibly nonlinear function. The first equation of (2.1) is called a system equation and the second equation is called a measurement equation. This nonlinear non-Gaussian state space model specifies the two following conditional density functions.

$$\begin{aligned} &p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\xi}_s), \\ &p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\xi}_m). \end{aligned} \tag{2.2}$$

We define a parameter vector $\boldsymbol{\theta}$ as follows.

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\xi}_s \\ \boldsymbol{\xi}_m \end{bmatrix}. \tag{2.3}$$

[5] The system noise vector is independent of past states and current states.

We denote that θ_j is the jth element of $\boldsymbol{\theta}$ and $J(= n_s + n_m)$ is the number of elements of $\boldsymbol{\theta}$.

This type of state space model (2.1) contains a broad class of linear, nonlinear, Gaussian, or non-Gaussian time series models. In spate space modeling, estimating the state space vector $\boldsymbol{x}_t$ is the most important problem. For the linear Gaussian state space model, the Kalman filter, which is proposed by [Kal60], is the most popular algorithm to estimate the state vector $\boldsymbol{x}_t$. For nonlinear or non-Gaussian state space model, there are many algorithms. For example, the extended Kalman filter ([Jaz70]) is the most popular algorithm and the other examples are the Gaussian-sum filter ([AS72]), the dynamic generalized model ([WHM85]), and the non-Gaussian filter and smoother ([Kit87]). In recent year, the MC filter for nonlinear non-Gaussian state space model is a popular algorithm because it is easily applicable to various time series models [6].

The MC filter is a variant of sequential Monte Carlo algorithms. In the MC filter, a posterior density function is approximated as "particles" that have weights, as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{1}{\sum_{i=1}^{M} w_t^i} \sum_{i=1}^{M} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i), \tag{2.4}$$

where w_t^i is the weight of a particle $\boldsymbol{x}_t^i$, M is the number of particles, and δ is the Dirac delta function[7]. We define $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_t\}$. Particle $\boldsymbol{x}_t^i$ is

[6] Many applications are shown in [DdFG01].

[7] The Dirac delta function is defined as follow.

$$\delta(x) = 0, \text{ if } x \neq 0,$$
$$\int_{\infty}^{\infty} \delta(x)dx = 1.$$

sampled from a system equation:

$$\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{\xi}_s). \tag{2.5}$$

Weight w_t^i is obtained as follows[8].

$$w_t^i = r(\psi(\boldsymbol{y}_t, \boldsymbol{x}_t^i))\Big|\frac{\partial \psi}{\partial \boldsymbol{y}_t}\Big|, \tag{2.6}$$

where ψ is the inverse function of the function h. The posterior density of Eq. (2.4) is obtained using resampling of particles of Eq. (2.5) with sampling probabilities proportional to $w_t^1, \cdots, w_t^M$. Resampling algorithms are discussed in [Kit96].

[Kit96] shows that the log-likelihood of the MC filter is approximated by

$$l(\boldsymbol{\theta}) \simeq \sum_{t=1}^{T} \log(\sum_{i=1}^{M} w_t^i) - T \log M, \tag{2.7}$$

where T is the number of observations. This approximation shows that the log-likelihood is affected by sampling error from the Monte Carlo algorithm. [Kit98] points out that numerous particles are necessary to obtain a closely accurate log-likelihood result. The sampling error of Eq. (2.7) prevents the calculation of accurate maximum likelihood estimates[9]. Moreover, we cannot use optimization algorithms that require the derivatives of functions because Eq. (2.7) is approximated by the weights of particles. Therefore, we need to develop an alternative approach that allows the estimation of

[8] See [Kit96].
[9] [Hig95] reports the sampling error of Eq. (2.7) in some cases.

parameters without maximizing a log-likelihood.

2.2.2 A Self-organizing State Space Model

The MC filter is problematic to estimate the parameter vector $\boldsymbol{\theta}$ because the likelihood of the filter contains error from the Monte Carlo method. [Kit98] proposes a self-organizing state space model to solve the problem. In [Kit98], an augmented state vector is defined as follows.

$$\boldsymbol{z}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{\theta} \end{bmatrix}, \tag{2.8}$$

An augmented system equation and an augmented measurement equation are defined as

$$\begin{aligned} \boldsymbol{z}_t &= F(\boldsymbol{z}_{t-1}, \boldsymbol{v}_t, \boldsymbol{\xi}_s), \\ \boldsymbol{y}_t &= H(\boldsymbol{z}_t, \boldsymbol{\epsilon}_t, \boldsymbol{\xi}_m), \end{aligned} \tag{2.9}$$

where

$$F(\boldsymbol{z}_{t-1}, \boldsymbol{v}_t, \boldsymbol{\xi}_s) = \begin{bmatrix} f(\boldsymbol{x}_{t-1}, \boldsymbol{v}_t, \boldsymbol{\xi}_s) \\ \boldsymbol{\theta} \end{bmatrix}$$

and

$$H(\boldsymbol{z}_t, \boldsymbol{\epsilon}_t, \boldsymbol{\xi}_m) = h(\boldsymbol{x}_t, \boldsymbol{\epsilon}_t, \boldsymbol{\xi}_m).$$

This nonlinear non-Gaussian state space model is called a self-organizing state space (SOSS) model. In our framework, the augmented state vector is estimated using the MC filter. The weights of particles are obtained by Eq. (2.6) and the log-likelihood of an SOSS model is approximated by Eq. (2.7).

States and parameters are estimated simultaneously without maximizing the log-likelihood of Eq. (2.9) because the parameter vector $\boldsymbol{\theta}$ in Eq. (2.9) is approximated by particles and it is estimated as the state vector in Eq. (2.8) [10]. However, the parameter vector $\boldsymbol{\theta}$ in Eq. (2.1) is not approximated by particles and it is estimated using maximizing the log-likelihood of Eq. (2.1). It is the difference between Eq. (2.1) and Eq. (2.9).

The initial particles $\boldsymbol{\theta}_0^i, \{i = 1, \cdots, M\}$ of a parameter vector $\boldsymbol{\theta}$ are sampled from the uniform distribution:

$$\theta_{j,0}^i \sim Uniform(P_j - r_j, P_j + r_j), \tag{2.10}$$

where $\theta_{j,0}^i$ is the jth element of $\boldsymbol{\theta}_0^i$, $Uniform(P_j - r_j, P_j + r_j)$ is uniform distribution from $P_j - r_j$ to $P_j + r_j$, P_j is the center of $(P_j - r_j, P_j + r_j)$, and r_j is a real number. The algorithm of an SOSS model is shown as Algorithm 2 [11].

Algorithm 2: A Self Organizing State Space

SOSS[$\{\boldsymbol{z}_{t-1}^i\}_{i=1}^M, \boldsymbol{y}_t$]

{

FOR i=1,...M

Predict: $\boldsymbol{z}_t^i \sim p(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}^i, v_t^i)$

Weight: w_t^i is obtained by Eq. (2.6)

ENDFOR

Sum of Weights: $sw = \sum_{i=1}^M w_t^i$

[10] The justification of an SOSS model is described in [Kit98].

[11] The negative log-likelihood is defined in the next subsection.

Log-Likelihood: $llk = log(sw/M)$

FOR i=1,...,M

Normalize: $\tilde{w}_t^i = \frac{w_t^i}{sw}$

ENDFOR

Resampling: $[\{\boldsymbol{z}_t^i,\ w_t^i\}_{i=1}^M]$ =resample$[\{\boldsymbol{z}_t^i,\ w_t^i\}_{i=1}^M]$

RETURN$[\{\boldsymbol{z}_t^i,\ w_t^i\}_{i=1}^M,\ llk]$

}

SOSSmain$[\{\boldsymbol{x}_0^i\}_{i=1}^M,\ \{\boldsymbol{y}\}_{t=1}^T,\ \boldsymbol{P}]$

{

Initialize the negative log-likelihood: $Y = 0$

$\boldsymbol{\theta}_0 \sim\ uniform(\boldsymbol{P} - r, \boldsymbol{P} + r)$

$\{z_0^i\}_{i=1}^M = (\{\boldsymbol{x}_0^i\}_{i=1}^M, \{\theta_0^i\}_{i=1}^M)$

FOR t=1,...,T

soss = SOSS$[\{\boldsymbol{z}_{t-1}^i\}_{i=1}^M,\ \boldsymbol{y}_t]$

$Y = Y - (llk\ in\ $ soss$)$

$\{\boldsymbol{z}_t^i,\ w_t^i\}_{i=1}^M = (\{\boldsymbol{z}_t^i,\ w_t^i\}_{i=1}^M\ in\ $ soss$)$

ENDFOR

RETURN$[\{\{\boldsymbol{z}_t^i,\ w_t^i\}_{i=1}^M\}_{t=1}^T,\ Y]$

}

On an SOSS model, [HK01] points out a problem: determination of initial distributions of a parameter vector $\boldsymbol{\theta}$ of an SOSS model. The difficulty of an SOSS model is that the filtered sample $\hat{\boldsymbol{\theta}}$ is a subsample of the prior sample $\boldsymbol{\theta}_0$ because $\hat{\boldsymbol{\theta}}$ has no noises (innovations). However, we do not generally know the location, spread, and shape of the posterior. For that reason,

it is difficult to determine the initial prior distributions of parameter $\boldsymbol{\theta}_0$. To solve that problem, [LW01] proposes adding artificial noise to parameters. The effectiveness of this approach is shown in [LW01] [12]. However, it is difficult to interpret the the artificial noise of invariant parameters. Therefore, we propose a method to estimate invariant parameters without using artificial innovations.

2.2.3 Simplex Initial Distribution Search

The Simplex Nelder-Mead (SNM) algorithm, which is proposed by [NM65], minimizes a function of J variables without constraints. The algorithm requires no derivatives of a function. Therefore, it works well for a non-differentiable function. The log-likelihood of an SOSS model is not a differentiable function because the log-likelihood is approximated by the weights of particles. The SNM method is applicable to the non-differentiable log-likelihood.

Vectors $\boldsymbol{P} = [\boldsymbol{P}_1, \cdots, \boldsymbol{P}_n, \cdots, \boldsymbol{P}_{J+1}]$ are the $(J+1)$ points in J-dimensional space that define the current "simplex"[13]. Note that the elements of $\boldsymbol{P}_n$ are the centers of the uniform distributions in Eq. (2.10) [14]. We denote that $\theta_{j,n},\ (1 \leq j \leq J,\ 1 \leq n \leq J+1)$, is the jth element of $\boldsymbol{\theta}$ at the nth point of the current "simplex". We write Y_n for the negative log-likelihood of an SOSS model at $\boldsymbol{P}_n$ and define that h has the suffix such that $Y_h = \max(Y_n)$ [h for "high"], s has the suffix the second highest Y_n, and l has the suffix such that $Y_l = \min(Y_n)$ [l for "low"]. We define the negative log-likelihood

[12] We compare our method with [LW01] in appendix A.
[13] Note that J is the number of elements of $\boldsymbol{\theta}$.
[14] The determination of the initial "simplex" $\boldsymbol{P}$ is described in appendix A.

as follows.

$$Y_n(\boldsymbol{P}_n) \simeq -\sum_{t=t_s}^{T} \log(\sum_{i=1}^{M} w_t^i) + T \log M, \tag{2.11}$$

where t_s is a positive integer. We set $t_s \geq 1$ because we would like to remove the initial particles of parameters until the particles are degenerated [15]. At each stage of the SNM method, $\boldsymbol{P}_h$ is replaced by a new point using three operations: reflection, expansion, and contraction. Reflection is defined as

$$\boldsymbol{P}_{ref} = (1+\alpha)\boldsymbol{P}_{cent} - \alpha \boldsymbol{P}_h, \ \alpha > 1, \tag{2.12}$$

where $\boldsymbol{P}_{cent}$ is the centroid of $\boldsymbol{P}_i$, $(i \neq h)$. Expansion is defined as

$$\boldsymbol{P}_{exp} = \gamma \boldsymbol{P}_{ref} + (1-\gamma)\boldsymbol{P}_{cent}, \ \gamma \geq 1. \tag{2.13}$$

Contraction is defined as

$$\boldsymbol{P}_{contr} = (1+\beta)\boldsymbol{P}_h - \beta \boldsymbol{P}_{cent}, \ \beta \in (0,1). \tag{2.14}$$

In our approach, we use the SNM algorithm to minimize the negative log-likelihood of an SOSS model along with the SOSS model estimate parameters. Although the log likelihood is affected by sampling error from the Monte Carlo algorithm, a self-organizing state space model can estimate parameters because minimizing the negative log-likelihood makes the initial distributions nearly equivalent to "true" parameters.

[15] We set $t_s = 20$ in our examples. Furthermore, we try to set $t_s = 1$. There exits a very little difference between two cases because particles of parameters are degenerated so rapidly.

Our procedure is terminated when the condition below is satisfied:

$$\sqrt{\sum_{j=1}^{J}\sum_{n=1}^{J+1}\frac{(\theta_{j,n}-\bar{\theta}_j)^2}{J(J+1)}} < \epsilon_h, \tag{2.15}$$

where $\bar{\theta}_j$ is the average of the jth element of $\boldsymbol{\theta}$ of $J+1$ points (the "simplex" at time T) and ϵ_h is a small positive real number. The algorithm of the simplex initial distribution search is shown in Algorithm 3 [16]. We denote that l is the number of iteration until the SNM procedure is stopped by the criterion given in Eq. (2.15).

[LWH04] adopts the simplex Nelder-Mead method to maximize the log-likelihood of the MC filter. The gradient of the log-likelihood around "true parameters" is nearly equal to zero in general. However, the log-likelihood is affected by sampling error from the Monte Carlo algorithm. Thus, the simplex Nelder-Mead method may not be terminated well because of the sampling error. To avoid the problem, numerous particles are necessary to obtain a closely accurate log-likelihood result. While, in our approach, parameters are estimated using an SOSS model. Thus, we don't need a closely accurate log-likelihood result as in [LWH04] [17].

[16] The algorithm of [NM65] is shown in Fig. 2.1.
[17] We compare our method with [LWH04] in appendix A.

Algorithm 3: Simplex Initial Distribution Search

SimplexSOSS[$\{\boldsymbol{y}\}_{t=1}^{T}$, $\boldsymbol{P} = [P_1, \cdots, P_{J+1}]$, J]

{

WHILE (TRUE)

FOR i=1,...,J+1

Y_i = SOSSMain($\{\boldsymbol{y}\}_{t=1}^{T}$, $\boldsymbol{P}_i$)

ENDFOR

Set highest negative log-likelihood: Yh

Set second highest negative log-likelihood: Ys

Set lowest negative log-likelihood: Yl

Calculate the centroid $\boldsymbol{P}_{cent}$ of $\{\boldsymbol{P}_i\}$, $i \neq h$

Reflection: $\boldsymbol{P}_{ref} = (1+\alpha)\boldsymbol{P}_{cent} - \alpha\boldsymbol{P}_h$

Y_{ref} = SOSSMain($\boldsymbol{P}_{ref}$)

FLAG=0

IF $Y_{ref} < Y_l$

Expansion: $\boldsymbol{P}_{exp} = \gamma\boldsymbol{P}_{ref} + (1-\gamma)\boldsymbol{P}_{cent}$

Y_{exp} = SOSSMain($\boldsymbol{P}_{exp}$)

IF $Y_{exp} < Y_l$

Replace P_h by P_{exp}

FLAG=1

ENDIF

ELSE IF $Y_{ref} <= Y_s$ AND $FLAG = 0$

Replace P_h by P_{ref}

FLAG=1

ELSE IF $Y_{ref} > Y_h$ AND $FLAG = 0$

(Do nothing)

ELSE

Replace P_h by P_{ref}

ENDIF

Contraction: $\boldsymbol{P}_{contr} = \beta \boldsymbol{P}_h + (1 - \beta)\boldsymbol{P}_{cent}$

Y_{contr} = SOSSMain($\boldsymbol{P}_{contr}$)

IF $Y_{contr} <= Y_h$

Replace P_h by P_{contr}

ELSE

Replace all P_i by $(P_i + P_l)/2$

ENDIF

IF Eq. (2.15) is satisfied THEN BREAK

ENDWHILE

RETURN[$\boldsymbol{P}$]

}

Section 3 describes the effectiveness of our method. In practice, the procedure of minimizing a negative log-likelihood stops before reaching a "true" parameter in some examples. That is, our initial distribution search is trapped in a local minimum because of the log-likelihood sampling error. This problem is solvable iteratively using our procedure. In other words, even if our initial distribution search is stopped, we restart our procedure at the stopped point. That restart is terminated when the parameter estimates

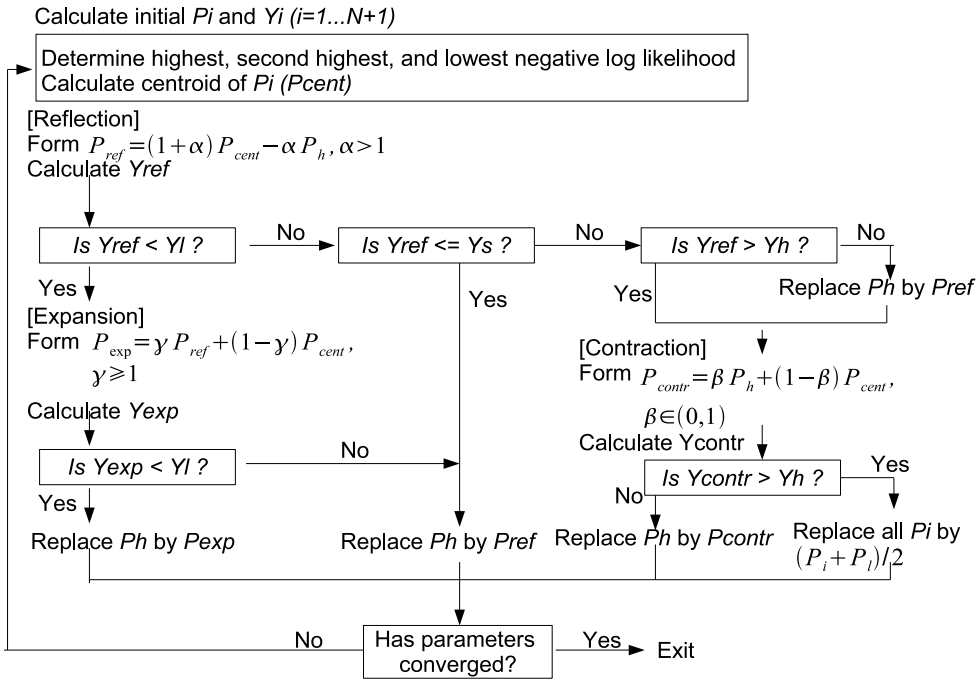

Figure 2.1: The Simplex Nelder-Mead Algorithm

are almost invariant. The next section shows that the problem of local minima is solvable if we iteratively use initial distribution searches two to four times.

2.3 Examples

In this section, we apply an initial distribution search algorithm to a linear Gaussian model, a linear non-Gaussian model, a nonlinear Gaussian model, and a stochastic volatility model.

In the following subsections: (1) we generate artificial time series ($T = 100$) based on each model, and (2) we estimate states and parameters using

a self-organizing state space model and initial distribution search iteratively. In our examples, we presume that an initial state x_0 is given. However, we assume that initial parameters are unknown. Therefore, we set all initial parameters zero as non-informative priors. We set the number of particles, M, to 10000 to estimate parameters[18].

2.3.1 Linear Gaussian State Space Model

In the case of two parameters, a linear Gaussian state space model is defined as

$$\begin{aligned} x_t &= x_{t-1} + v_t, \quad x_0 \sim N(0,1), \\ y_t &= x_t + \epsilon_t, \end{aligned} \tag{2.16}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim N(0, \sigma_m^2)$. Parameters are $\xi_s = \sigma_s$ and $\xi_m = \sigma_m$. The augmented state vector of this model is defined as

$$\boldsymbol{z}_t = \begin{bmatrix} x_t \\ \sigma_s \\ \sigma_m \end{bmatrix}. \tag{2.17}$$

The parameter estimates and the average of each parameter estimate are shown in Table 2.1 and the estimated state x_t based on the first iteration is shown in Fig. 2.2 (The dotted line is the estimated state x_t. The solid line is the real state). Table 2.1 shows that the parameter estimates are

[18] About the determination of M, we follow previous studies, for example, [Kit98]. We set $r_j = 0.1$, $\alpha = 2.0$, $\gamma = 2.0$, $\beta = 0.5$, and $\epsilon_h = 1.0 \times 10^{-4}$. We use $r_j = 0.1$ to estimate parameters to four places of decimals. Note that $0.1 \times 2/10000 = 2.0 \times 10^{-5}$. The centroid of $\{\boldsymbol{P}_i\}$, $i \neq h$, is obtained from the averages of the positions of J points.

close to the true parameter values in the first iteration. In Table 2.1, we show the estimates based on the Kalman filter. The results are better than our method, and it indicates the Kalman filter is better than our method in linear Gaussian state space models. Moreover, Fig. 2.2 shows that the estimated state is very similar to the true state.

Table 2.1: Linear Gaussian Model (2 Parameters)

Iteration	log-likelihood	σ_s	σ_m	M	l
Initial Parameter		0	0		
First Iteration	-548.878	4.0280	2.4153	10000	33
	-549.400	4.0279	2.4152		
	-549.718	4.0283	2.4154		
Average		4.0281	2.4154		
True Parameter Value		4	2		
Kalman Filter		4.0121	2.267		

2.3.2 Linear non-Gaussian State Space Model

In the case of two parameters, a linear non-Gaussian state space model is defined as

$$\begin{aligned} x_t &= x_{t-1} + \alpha v_t, \;\; x_0 \sim N(0,1), \\ y_t &= x_t + \epsilon_t, \end{aligned} \tag{2.18}$$

where $v_t \sim t(df = 1)$[Cauchy distribution] and $\epsilon_t \sim N(0, \sigma_m^2)$. The parameters are $\xi_s = \alpha$ and $\xi_m = \sigma_m$. The augmented state vector of this model is

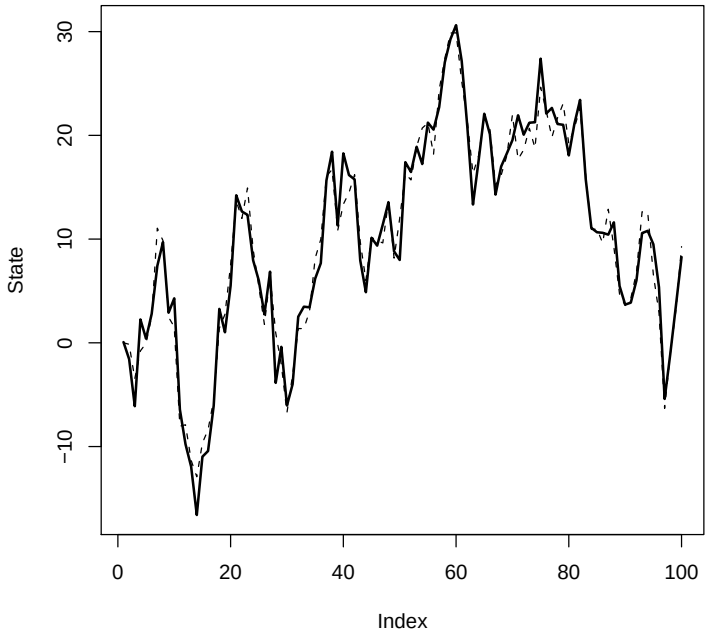

Figure 2.2: Linear Gaussian Model

defined as

$$\boldsymbol{z}_t = \begin{bmatrix} x_t \\ \alpha \\ \sigma_m \end{bmatrix}. \tag{2.19}$$

The parameter estimates and the average of each parameter estimate are shown in Table 2.2 and the estimated state x_t based on the first iteration is shown in Fig. 2.3. Table 2.2 shows that the parameter estimates are close to the true parameter values in the first iteration. Moreover, Fig. 2.3 shows that the estimated state is very close to the true state.

Table 2.2: Linear Non-Gaussian Model

Iteration	log-likelihood	α	σ_m	M	l
Initial Parameter		0	0		
First Iteration	-581.6198	1.2942	2.8583	10000	38
	-582.1342	1.2941	2.8582		
	-582.5045	1.2939	2.8580		
Average		1.2941	2.8582		
True Parameter Value		1	3		

2.3.3 Nonlinear Gaussian State Space Model

In the case of two parameters, a nonlinear Gaussian state space model is defined as

$$\begin{aligned} x_t &= \frac{1}{2}x_{t-1} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2(t-1)) + v_t, \;\; x_0 \sim N(0,1), \\ y_t &= \frac{x_t^2}{20} + \epsilon_t, \end{aligned} \tag{2.20}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim N(0, \sigma_m^2)$. Parameters are $\xi_s = \sigma_s$ and $\xi_m = \sigma_m$. The augmented state vector of this model is defined as

$$\boldsymbol{z}_t = \begin{bmatrix} x_t \\ \sigma_s \\ \sigma_m \end{bmatrix}. \tag{2.21}$$

The parameter estimates and the average of each parameter estimate are shown in Table 2.3 and the estimated state x_t based on the fourth iteration is shown in Fig. 2.4. Table 2.3 shows that the parameter estimates are close to the true parameter values in the fourth iteration. Moreover, Fig. 2.4

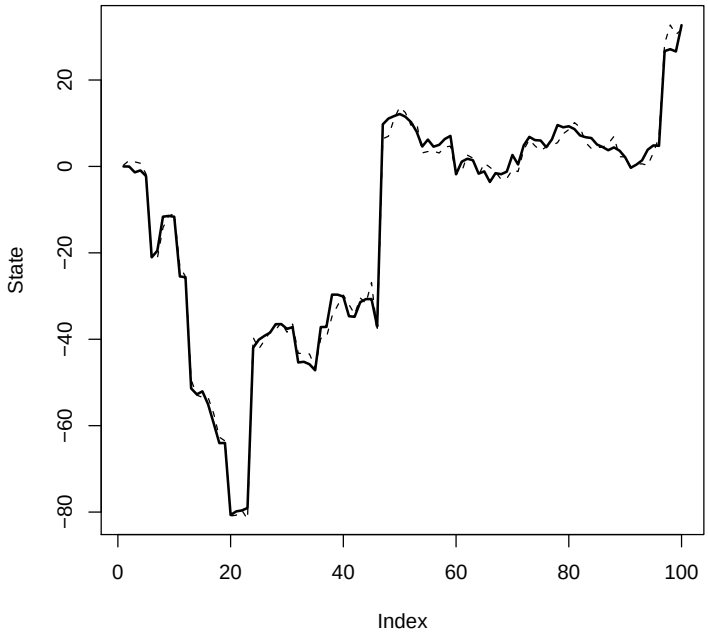

Figure 2.3: Linear Non-Gaussian Model

shows that the estimated state is a good fit for the true state.

In the case of three parameters, a nonlinear Gaussian state space model is defined as

$$\begin{aligned} x_t &= \frac{1}{2}x_{t-1} + \frac{\alpha x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2(t-1)) + v_t, \;\; x_0 \sim N(0,1), \\ y_t &= \frac{x_t^2}{20} + \epsilon_t. \end{aligned} \tag{2.22}$$

Parameters are $\boldsymbol{\xi}_s = (\sigma_s, \alpha)$ and $\xi_m = \sigma_m$. The augmented state vector of

Table 2.3: Nonlinear Gaussian Model (2 Parameters)

Iteration	log-likelihood	σ_s	σ_m	M	l
Initial Parameter		0	0		
First Iteration	-476.2196	2.6345	4.0271	10000	41
	-477.5585	2.6343	4.0268		
	-477.814	2.6344	4.0269		
Average		2.6344	4.0271		
Initial Parameter		2.6344	4.0271		
Second Iteration	-387.3325	3.6087	7.8855	10000	27
	-387.4781	3.6088	7.8857		
	-387.6288	3.6088	7.8857		
Average		3.6087	7.8855		
Initial Parameter		3.6087	7.8855		
Third Iteration	-383.2827	3.4785	10.2815	10000	15
	-383.2889	3.4783	10.2816		
	-383.4200	3.4783	10.2815		
Average		3.4783	10.2815		
Initial Parameter		3.4785	10.2815		
Fourth Iteration	-382.1269	1.4654	10.5987	10000	12
	-382.1665	1.4673	10.5985		
	-382.1712	1.4660	10.5984		
Average		1.4660	10.5986		
True Parameter Value		1	10		

this model is defined as

$$\boldsymbol{z}_t = \begin{bmatrix} x_t \\ \sigma_s \\ \alpha \\ \sigma_m \end{bmatrix}. \tag{2.23}$$

The parameter estimates and the average of each parameter estimate are shown in Table 2.4 and the estimated state x_t based on the fourth iteration is shown in Fig. 2.5. Table 2.4 shows that the parameter estimates are close

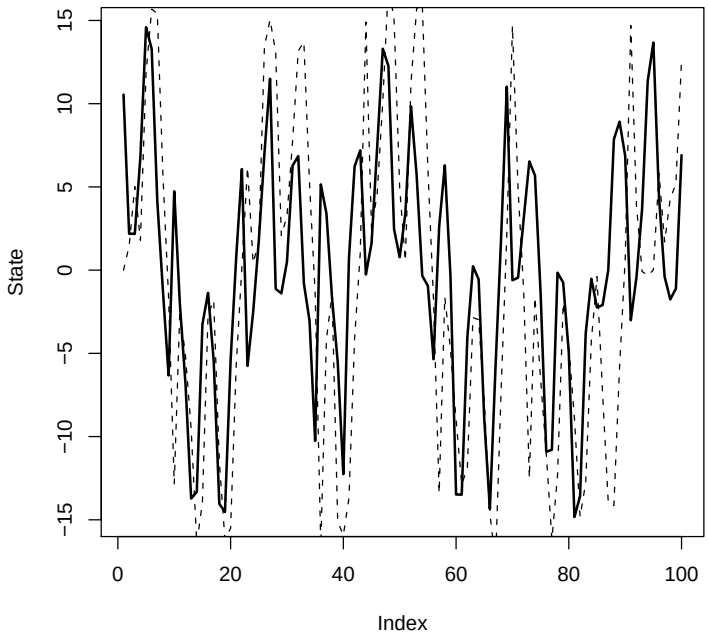

Figure 2.4: Nonlinear Gaussian Model (2 Parameters)

to the true parameter values after the fourth iteration. Moreover, Fig. 2.5 shows that the estimated state is a good fit for the true state.

2.3.4 Stochastic Volatility Model

In the case of two parameters, a stochastic volatility model is defined as

$$\begin{aligned} x_t &= 0.8x_{t-1} + v_t, \quad x_0 \sim N(0,1), \\ y_t &= \beta w_t \exp(\frac{x_t}{2}), \end{aligned} \tag{2.24}$$

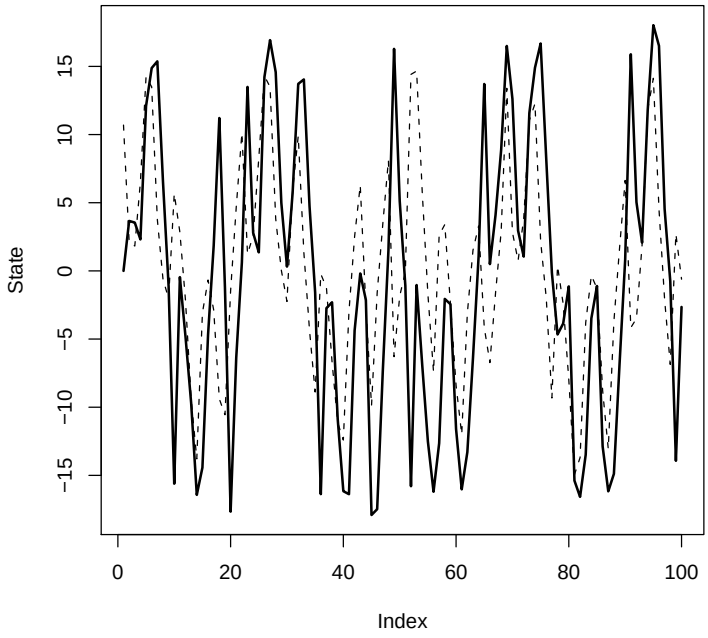

Figure 2.5: Nonlinear Gaussian Model (3 Parameters)

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim N(0, 1)$. Parameters are $\xi_s = \sigma_s$ and $\xi_m = \beta$. The augmented state vector of this model is defined as

$$\boldsymbol{z}_t = \begin{bmatrix} x_t \\ \sigma_s \\ \beta \end{bmatrix}. \tag{2.25}$$

The parameter estimates and the average of each parameter estimate are shown in Table 2.5 and the estimated state x_t based on the second iteration is shown in Fig. 2.6. Table 2.5 shows that the parameter estimates are close

to the true parameter values in the second iteration. Moreover, Fig. 2.6 shows that the estimated state is very close to the true state.

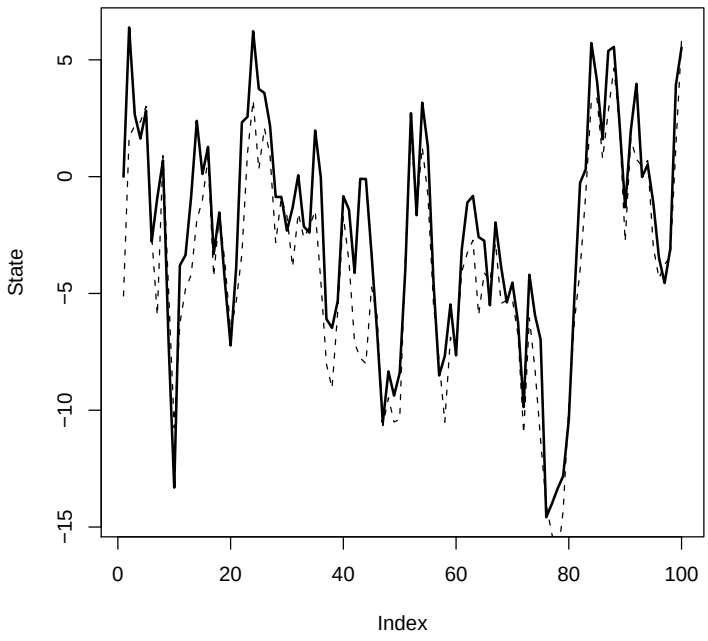

Figure 2.6: Stochastic Volatility Model

2.3.5 Threshold AR Model

Threshold Autoregressive models (threshold AR models) are proposed by [Ton90]. In the case of three parameters, a threshold AR(1) model [19] is

[19] See page 100, [Ton90].

defined as

$$x_t = \begin{cases} 1.0 + \beta_1 x_{t-1} + v_t, \text{ if } x_t \leq k, \\ -1.0 + \beta_2 x_{t-1} + v_t, \text{ if } x_t > k, \end{cases} \tag{2.26}$$

$$y_t = x_t + \epsilon_t,$$

where $v_t \sim N(0,1)$ and $\epsilon_t \sim N(0,1)$. Parameters are $\boldsymbol{\xi}_s = (k, \beta_1, \beta_2)$. In this example, we set $T$to 200 because the threshold AR(1) model consists of two AR(1) models.

$$\boldsymbol{z}_t = \begin{bmatrix} x_t \\ k \\ \beta_1 \\ \beta_2 \end{bmatrix}. \tag{2.27}$$

The parameter estimates and the average of each parameter estimate are shown in Table 2.6 and the estimated state x_t based on the first iteration is shown in Fig. 2.7. Table 2.6 shows that the parameter estimates are close to the true parameter values in the first iteration. Moreover, Fig. 2.7 shows that the estimated state is very close to the true state.

2.4 Conclusions

This paper proposes a method to seek initial distributions of parameters based on the simplex Nelder-Mead algorithm for a self-organizing state space model. Our method is applicable to any nonlinear non-Gaussian state space model because the simplex Nelder-Mead algorithm solves nonlinear discontinuous optimization problems. We demonstrate the effectiveness of our

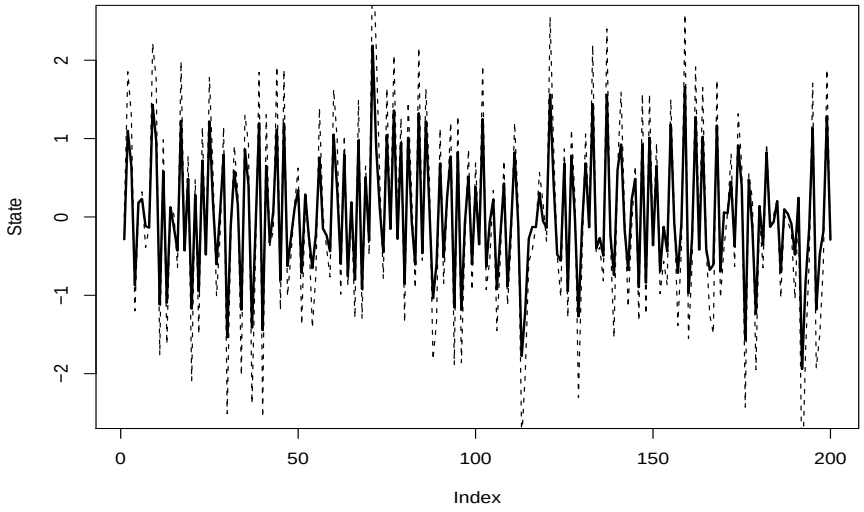

Figure 2.7: Threshold AR(1) Model

method by applying it to a linear Gaussian model, a linear non-Gaussian model, a nonlinear Gaussian model, and a stochastic volatility model. In practice, our initial distribution search is trapped in a local minimum in some examples. Our analyses, however, show that our method, used iteratively, can solve the local minimum problem. For further study, we would like to improve the iterative procedure by adopting a globalized Nelder-Mead algorithm proposed by [LR04].

Appendix A: Initialization of "Simplex"

Following the appendix of [NM65], the initial "simplex" $\boldsymbol{P}$ is determined as

follows.

$$
\begin{aligned}
\boldsymbol{P}_1' &= \Big[P_1, \quad P_2, \quad P_3, \quad \cdots P_j, \quad \cdots \quad P_{J-1}, \quad P_J\Big], \\
\boldsymbol{P}_2' &= \Big[P_1 + I, \quad P_2, \quad P_3, \quad \cdots P_j, \quad \cdots \quad P_{J-1}, \quad P_J\Big], \\
\boldsymbol{P}_3' &= \Big[P_1, \quad P_2 + I, \quad P_3, \quad \cdots P_j, \quad \cdots \quad P_{J-1}, \quad P_J\Big], \\
&\vdots \qquad\qquad\qquad \ddots \\
\boldsymbol{P}_J' &= \Big[P_1, \quad P_2, \quad P_3, \quad \cdots P_j, \quad \cdots \quad P_{J-1} + I, \quad P_J\Big], \\
\boldsymbol{P}_{J+1}' &= \Big[P_1, \quad P_2, \quad P_3, \quad \cdots P_j, \quad \cdots \quad P_{J-1}, \quad P_J + I\Big],
\end{aligned}
\tag{2.28}
$$

where $'$ means transposition and I is a real number. In our simulation studies, we set I to 1 [20]. Note that J is the number of elements of $\boldsymbol{\theta}$ and P_j is defined in Eq. (2.10). For example, in section 3.4, we use $\boldsymbol{P}_1' = [0,\ 0]$ in the first iteration and $\boldsymbol{P}_1' = [3.1717,\ 2.0070]$ in the second iteration [21].

Appendix B: Comparing Parameter Estimation Methods

We compare our method with other methods ([LWH04] and [LW01]) in the case of two parameters of a linear Gaussian state space model in section 3.1.

[LWH04] adopts the simplex Nelder-Mead (SNM) method to minimize the negative log-likelihood of the MC filter. The SNM procedure is termi-

[20] In the three parameters case of the nonlinear Gaussian model (section 3.3), we set I to 10.

[21] See Table 5.

nated when the condition below is satisfied:

$$\sqrt{\frac{\sum_{n=1}^{J+1}\left(Y_n-\bar{Y}\right)^2}{J+1}} < \epsilon_{nm}, \tag{2.29}$$

where Y_n is a negative log-likelihood given by Eq. (2.11), $\bar{Y}$ is the average of Y_n, and ϵ_{nm} is a small positive real number. We set ϵ_{nm} to 1.0×10^{-3} because the SNM procedure is not terminated when we set ϵ_{nm} to 1.0×10^{-4}. We estimate σ_s and σ_m in Eq. (2.16) a hundred times using our method and [LWH04]. In Table 2.7, we show the average and the variance of each parameter estimate. Table 2.7 shows that the variances of σ_s and σ_m, which are estimated by [LWH04], are larger than the variances of σ_s and σ_m, which are estimated by our method. We conclude that our method is better than [LWH04].

[LW01] proposes adding artificial noise to parameters. In [LW01], the conditional evolution density of a time-varying parameter vector $\boldsymbol{\theta}_t$ is defined by

$$p(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t) \sim N(a\boldsymbol{\theta}_t + (1-a)\bar{\boldsymbol{\theta}}_t, h^2\boldsymbol{V}_t), \tag{2.30}$$

where $\bar{\boldsymbol{\theta}}_t$ is the mean vector of particles of $\boldsymbol{\theta}_t$, $\boldsymbol{V}_t$ is the variance matrix of particles of $\boldsymbol{\theta}_t$, $h^2 = 1 - a^2$, $a = (3\delta_l - 1)/2\delta_l$, and δ_l is a discount factor in $(0, 1]$, typically around $0.95 - 0.99$. The main feature of [LW01] is that the variance matrix $\boldsymbol{V}_t$ shrinks step by step and it finally converges towards $\mathbf{0}$. In the case of two parameters of a linear Gaussian state space model in

section 3.1, $\boldsymbol{\theta}_t$ is defined by

$$\boldsymbol{\theta}_t = \begin{bmatrix} \sigma_{s,t} \\ \sigma_{m,t} \end{bmatrix} \tag{2.31}$$

where $\sigma_{s,t}$ and $\sigma_{m,t}$ are time-varying parameters. The evolutions of $\sigma_{s,t}$ and $\sigma_{m,t}$ are shown in Fig. 2.8 [22]. The right side of Fig. 2.8 shows $\sigma_{m,t}$ converges towards the true parameter value in Table 2.1. However, the left side of Fig. 2.8 shows $\sigma_{s,t}$ converges towards a wrong value because the $\boldsymbol{V}_t$ converges towards $\mathbf{0}$ before reaching the true parameter value in Table 2.1. This problem cannot be avoidable when we don't have the prior knowledge about parameters. However, our method does not require the prior knowledge about parameters.

[22] In our simulation, we set $\boldsymbol{V}_0$ to 2×2 identity matrix, $\bar{\boldsymbol{\theta}}_0$ to $\mathbf{0}$, and δ to 0.99.

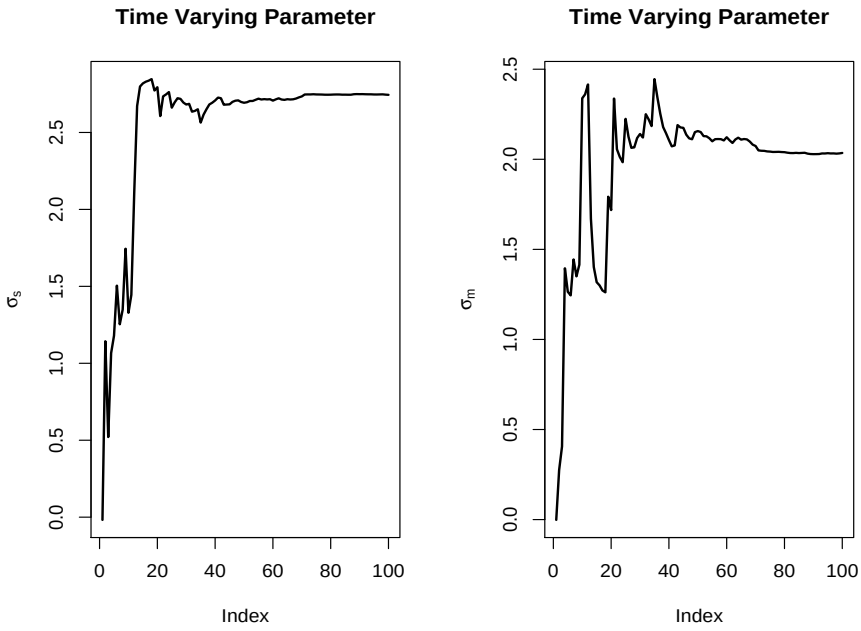

Figure 2.8: Time-Varying Parameter

Table 2.4: Nonlinear Gaussian Model (3 Parameters)

Iteration	log-likelihood	σ_s	α	σ_m	M	l
Initial Parameter		0	0	0		
First Iteration	-389.4667	4.2020	1.6419	11.3944	10000	38
	-389.4865	4.2020	1.6419	11.3946		
	-389.5318	4.2020	1.6419	11.3947		
	-389.5765	4.2020	1.6419	11.3946		
Average		4.2020	1.6419	11.3946		
Initial Parameter		4.2020	1.6419	11.3946		
Second Iteration	-388.1539	5.0995	7.6052	11.0038	10000	46
	-388.1856	5.0995	7.6051	11.0036		
	-388.2527	5.0996	7.6052	11.0038		
	-388.3268	5.0995	7.6051	11.0036		
Average		5.0995	7.6052	11.0037		
Initial Parameter		5.0995	7.6052	11.0037		
Third Iteration	-387.6866	5.1796	13.6353	10.5188	10000	52
	-387.7323	5.1798	13.6356	10.5191		
	-387.7649	5.1796	13.6353	10.5188		
	-387.8014	5.1797	13.6355	10.5190		
Average		5.1797	13.6354	10.5189		
Initial Parameter		5.1797	13.6354	10.5189		
Fourth Iteration	-385.8363	0.9657	23.9264	11.1893	10000	30
	-385.8460	0.9656	23.9265	11.1894		
	-385.9190	0.9646	23.9268	11.1897		
	-385.9277	0.9659	23.9263	11.1893		
Average		0.9655	23.9265	11.1894		
True Parameter Value		1	25	10		

Table 2.5: Stochastic Volatility Model

Iteration	log-likelihood	σ_s	σ_m	M	l
Initial Parameter		0	0		
First Iteration	-304.4514	3.1715	2.0069	10000	44
	-304.5505	3.1718	2.0070		
	-304.6005	3.1718	2.0070		
Average		3.1717	2.0070		
Initial Parameter		3.1717	2.0070		
Second Iteration	-304.3197	3.2489	1.8379	10000	19
	-304.4946	3.2492	1.8378		
	-304.5389	3.2486	1.8381		
		3.2489	1.8379		
Average		3.2489	1.8379		
True Parameter Value		3	1		

Table 2.6: Threshold AR(1) Model

Iteration	log-likelihood	k	β_1	β_2	M	l
Initial Parameter		0	0	0		
First Iteration	-347.638	0.0574	0.5963	0.4407	10000	14
	-347.904	0.0572	0.5963	0.4398		
	-348.010	0.0573	0.5961	0.4409		
	-348.177	0.0572	0.5963	0.4399		
Average		0.0573	0.5963	0.4403		
True Parameter Value		0	0.6	0.4		

Table 2.7: Our Method and Lin et al. (2004)

Our Method		
	σ_s	σ_m
Average	3.9488	1.7306
Variance	0.0132	0.0166

Lin et al. (2004)		
	σ_s	σ_m
Average	3.9575	1.6428
Variance	0.0362	0.0559

Chapter 3

Smoothing and Filter Initialization Based on an Inverse System Equation

3.1 Introduction

The Monte Carlo (MC) filter was proposed by [GSS93] and [Kit96]. The filter is an algorithm to estimate states for a nonlinear non-Gaussian state space model [1]. In recent years, the filter has been applied to various problems [2]. In spite of the widespread practical application of the MC filter, smoothing algorithms are less well established. The first smoothing algorithm, proposed by [Kit96], is based on the storing state vector. In that algorithm, the repetition of the resampling in the MC filter decreases the

[1] [AMGC02] is a readable tutorial on the Monte Carlo particle filter.
[2] see [DdFG01].

number of different realizations of the state vector. To resolve that problem, [Kit96] proposes to employ fixed $L-$lag smoothing. The paper recommends not to make L too large (say, 10 or 20, or 50 at the largest). A persistent problem is that the fixed $L-$lag smoothing cannot use all observations when the number of observations is larger than 10–50. To realize fixed-interval smoothing, researchers have developed alternative methods based on the two-filter formula ([Kit96]), a forward filtering – backward smoothing formula ([DGA00] and [GDW04]), and maximum *a posteriori* sequence estimation ([GDW01].)

This paper proposes a simple MC smoothing algorithm based on the inverse function of a system equation (an inverse system equation). Our method is applicable to any nonlinear non-Gaussian state space model if an inverse system equation is given analytically. This paper shows that our algorithm is fundamentally equivalent to the MC filter. Therefore, the computational complexity of our smoothing algorithm is equal to the complexity of the MC filter. Moreover, our algorithm is easily implemented because it can be realized by a minor modification of the MC filter. The main advantage of our algorithm is its simplicity. Furthermore, we propose a filter-initialization algorithm based on the smoothing distribution, which is obtained by our algorithm and an inverse system equation. Filter initialization is important to estimate a state vector of a nonlinear non-Gaussian state space model in Bayesian tracking like the MC filter. In this paper, we show the effectiveness of our method by applying it to a linear Gaussian state space model, a linear non-Gaussian state space model, a stochastic volatility model, and a stochastic volatility model with t-distribution.

This paper is organized as follows. In section 2, we describe our smoothing algorithm and filter initialization algorithm. In section 3, we show examples for some models. In section 4, we describe the salient conclusions of the paper.

3.2 Model

3.2.1 Monte Carlo Filter

A nonlinear non-Gaussian state space model for the time series $\boldsymbol{y}_t,\ t = \{1, 2, \cdots, T\}$ is defined as follows.

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_s, \boldsymbol{v}_t), \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{\xi}_m, \boldsymbol{\epsilon}_t), \end{aligned} \tag{3.1}$$

where $\boldsymbol{x}_t$ is an unknown $n_x \times 1$ state vector, $\boldsymbol{v}_t$ is the $n_v \times 1$ system noise vector with a density function $q(\boldsymbol{v})$, $\boldsymbol{\epsilon}_t$ is the $n_\epsilon \times 1$ observation noise vector with a density function $r(\boldsymbol{\epsilon})$, $\boldsymbol{\xi}_s$ is the $n_s \times 1$ system parameter vector of the function f, and $\boldsymbol{\xi}_m$ is the $n_m \times 1$ observation parameter vector of the function h. The function $f : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_v} \to \boldsymbol{R}^{n_x}$ and the function $h : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_\epsilon} \to \boldsymbol{R}^{n_y}$ are possibly nonlinear functions. The first equation of (3.1) is called a system equation and the second equation is called a measurement equation. This nonlinear non-Gaussian state space model specifies the following two conditional density functions.

$$\begin{aligned} &p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\xi}_s), \\ &p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\xi}_m). \end{aligned} \tag{3.2}$$

We define a parameter vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\xi}_s \\ \boldsymbol{\xi}_m \end{bmatrix}. \tag{3.3}$$

We denote that θ_j is the jth element of $\boldsymbol{\theta}$ and that $J(= n_s + n_m)$ is the number of elements of $\boldsymbol{\theta}$. In this paper, we suppose that the parameter $\boldsymbol{\theta}$ is known.

The MC filter is a variant of sequential Monte Carlo algorithms. In the MC filter, a posterior density function is approximated as "particles" that have weights, as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:(t-1)}) = \frac{1}{\sum_{i=1}^{M} w_t^i} \sum_{i=1}^{M} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i), \tag{3.4}$$

where w_t^i is the weight of a particle $\boldsymbol{x}_t^i$, M is the number of particles, and δ is a delta function. We define $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_t\}$. Weight w_t^i is defined as

$$w_t^i = r(\psi(\boldsymbol{y}_t, \boldsymbol{x}_t^i)) \left| \frac{\partial \psi}{\partial \boldsymbol{y}_t} \right|, \tag{3.5}$$

where ψ is the inverse function of the function h. The right hand side (RHS) of Eq. (3.5) is the likelihood function of a nonlinear non-Gaussian state space model. In the standard algorithm of the MC filter, the particles x_t^i are resampled with sampling probabilities proportional to $w_t^1, \cdots, w_t^M$. Resampling algorithms are discussed in [Kit96]. After resampling, we have

$w_t^i = 1/M$. Consequently, Eq. (3.4) is rewritten as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{1}{M}\sum_{i=1}^{M}\delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i),\ i = \{1, \cdots, M\}. \tag{3.6}$$

Particles $\boldsymbol{x}_t^i$ are sampled from a system equation:

$$\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{\xi}_s). \tag{3.7}$$

The algorithm of the MC filter is shown as Algorithm 1.

Actually, [Kit96] shows that the log likelihood of the MC filter is approximated as

$$l(\boldsymbol{\theta}) \simeq \sum_{t=1}^{T}\log(\sum_{i=1}^{M} w_t^i) - T\log M, \tag{3.8}$$

where T is the number of observations. This approximation shows that the log likelihood is affected by sampling error from the Monte Carlo algorithm. In addition, [Kit98] points out that numerous particles are necessary to obtain an accurate log-likelihood.

3.2.2 Smoothing and Filter Initialization Based on an Inverse System Equation

We propose a smoothing algorithm based on an inverse system function. Our algorithm is applicable to any nonlinear non-Gaussian state space model with an inverse system equation. We define an inverse system equation as:

$$\boldsymbol{x}_{t-1} = g(\boldsymbol{x}_t, \boldsymbol{\xi}_s, \boldsymbol{v}_t), \tag{3.9}$$

where g is the inverse function of f. The inverse system equation (3.9) specifies the following conditional density function.

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \tag{3.10}$$

We derive the smoothing distribution $p(\boldsymbol{x}_{T-l}|\boldsymbol{y}_{1:T})$ from the definition of conditional probability. If the probability distribution $\mathbb{P}(\boldsymbol{y}_{1:T}) > 0$, then the conditional probability of $\boldsymbol{x}_t$ given $\boldsymbol{y}_{1:T}$ is

$$\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) = \frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_{1:T})}{\mathbb{P}(\boldsymbol{y}_{1:T})}, \tag{3.11}$$

where t $(1 \leq t < T)$ is an integer. We define $\boldsymbol{y}'_{1:T} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{t-1}, \boldsymbol{y}_{t+1}, \cdots, \boldsymbol{y}_T\}$. We rewrite the conditional probability as follows.

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) \\
=&\frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{y}'_{1:T})}{\mathbb{P}(\boldsymbol{y}_t, \boldsymbol{y}'_{1:T})} \\
=&\frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{y}'_{1:T})}{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}'_{1:T})} \frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}'_{1:T})/\mathbb{P}(\boldsymbol{y}'_{1:T})}{\mathbb{P}(\boldsymbol{y}_t, \boldsymbol{y}'_{1:T})/\mathbb{P}(\boldsymbol{y}'_{1:T})} \\
=&\frac{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}'_{1:T})\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}'_{1:T})}{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}'_{1:T})} \\
=&\frac{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}'_{1:T})}{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}'_{1:T})}.
\end{aligned} \tag{3.12}$$

In the fourth equality of Eq. (3.12), we assume that the likelihood, $\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)$, does not depend on $\boldsymbol{y}'_{1:T}$. We assume that $\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}'_{1:T})$ is factorized as follows

[3].

$$\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}'_{1:T}) = \mathbb{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})\mathbb{P}(\boldsymbol{x}_{t+1}|\boldsymbol{y}'_{1:T}) \tag{3.14}$$

Particles $\boldsymbol{x}_t^i$ are sampled from Eq. (3.9). If $\mathbb{P}(\boldsymbol{x}_{t+1}|\boldsymbol{y}'_{1:T})$ is nearly equal to $\mathbb{P}(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T})$, then the following equation is obtained by

$$\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) \propto \mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})\mathbb{P}(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}). \tag{3.15}$$

Note that $\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}'_{1:T})$ is the normalizing constant. In the MC filter, the smoothing distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$ is approximated by

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) = \frac{1}{\sum_{i=1}^M w_t^i} \sum_{i=1}^M w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i), \tag{3.16}$$

Therefore, Eq. (3.15) can be rewritten as

$$w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t+1}^i)w_{t+1}^i \delta(\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1}^i). \tag{3.17}$$

In the standard algorithm of the MC filter, resampling is applied at every time. After resampling at time $t+1$, we have $w_{t+1}^i = 1/M$. Therefore, we

[3]This factorization is justified as follows.

$$\begin{aligned} \mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}'_{1:T}) &= \frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}'_{1:T})}{\mathbb{P}(\boldsymbol{x}_{t+1})} \frac{\mathbb{P}(\boldsymbol{x}_{t+1})}{\mathbb{P}(\boldsymbol{y}'_{1:T})} \\ &= \mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}'_{1:T}|\boldsymbol{x}_{t+1})\mathbb{P}(\boldsymbol{x}_{t+1}|\boldsymbol{y}'_{1:T}) \\ &= \mathbb{P}(\boldsymbol{x}_t, |\boldsymbol{x}_{t+1})\mathbb{P}(\boldsymbol{x}_{t+1}|\boldsymbol{y}'_{1:T}) \end{aligned} \tag{3.13}$$

In the third equality of Eq. (3.13), we use the property that $\boldsymbol{x}_t$ depends only on $\boldsymbol{x}_{t+1}$ in Eq. (3.9).

can write

$$w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i) \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^i) p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t+1}^i) \frac{1}{M} \delta(\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1}^i). \tag{3.18}$$

The weight w_t^i is obtained as

$$w_t^i \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t^i). \tag{3.19}$$

In summary, the smoothing distribution, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$, can be obtained using the MC filter from time T to time 1 if the smoothing distribution, $p(\boldsymbol{x}_T|\boldsymbol{y}_{1:T})$, and an inverse system equation are given. Note that the smoothing distribution, $p(\boldsymbol{x}_T|\boldsymbol{y}_{1:T})$, can be obtained using the MC filter (Algorithm 1). The algorithm of smoothing based on an inverse system equation is shown as Algorithm 2. The computational complexity for our smoothing algorithm is $O(MT)$. It is equivalent to the computational complexity for [GDW04][4]. Furthermore, our smoothing algorithm requires $O(M)$ storage to save weights of particle because it requires only $W_T^i,\ \{1, \cdots, M\}$. In contrast, [GDW04] requires $O(MT)$ storage to save weights of particle because it requires $W_t^i,\ \{1, \cdots, M\},\ \{1, \cdots, T\}$. In other words, the advantages of our smoothing algorithm are its simplicity and small memory requirement. Our algorithm is a minor modification of the "standard" MC filter. However, an obstacle exists. In general, the inverse function of a system function cannot be obtained analytically.

We propose a filter initialization algorithm, which chooses an appropriate

[4]Computational complexities of the two-filter formula ([Kit96]) and the maximum *a posteriori* sequence estimation ([GDW01]) are $O(M^2T)$.

initial probability (a prior probability) $\mathbb{P}(\boldsymbol{x}_0)$ of the MC filter. In general, the initial probability $\mathbb{P}(\boldsymbol{x}_0)$ of the MC filter is unknown. State estimation based on the MC filter is improved if one can choose an appropriate initial probability. We propose a filter initialization algorithm based on the smoothing distribution $\mathbb{P}(\boldsymbol{x}_1|\boldsymbol{y}_{1:T})$ and an inverse system equation. We can obtain an appropriate initial probability $\hat{\mathbb{P}}(\boldsymbol{x}_0)$ as follows.

$$\hat{\mathbb{P}}(\boldsymbol{x}_0) \propto \mathbb{P}(\boldsymbol{x}_0|\boldsymbol{x}_1)\mathbb{P}(\boldsymbol{x}_1|\boldsymbol{y}_{1:T}). \tag{3.20}$$

Furthermore, we propose the following steps to estimate smoother $\mathbb{P}(\boldsymbol{x}_i|\boldsymbol{y}_{1:T}),\ \{i = 1, 2, \cdots, T\}$ using Algorithm 2 and Eq. (3.20) as follows.

1. Choose an arbitrary initial distribution $\mathbb{P}(\boldsymbol{x}_0)$.
2. Use Algorithm 2 with $\mathbb{P}(\boldsymbol{x}_0)$.
3. Calculate the initial distribution $\hat{\mathbb{P}}(\boldsymbol{x}_0)$ based on Eq. (3.20).
4. Use Algorithm 1 with $\hat{\mathbb{P}}(\boldsymbol{x}_0)$.

Our filter initialization is suitable for "on-line" state estimation when new observations are obtained in real-time. At the early stage of on-line state estimation, steps 1–3, as proposed above, are executed; then the MC filter estimates $\boldsymbol{x}_t$ in real-time based on the whole observation and $\hat{\mathbb{P}}(\boldsymbol{x}_0)$.

3.3 Examples

We apply our algorithms to a linear Gaussian state space model, a linear non-Gaussian state space model, the stochastic volatility model, and a stochastic

volatility model with t-distribution. In the following subsections: (1) we generate an artificial time series ($T = 100$) based on each model ($x_0 \sim N(0, 1^2)$), (2) we estimate a state x_t using our smoothing algorithm[5], and (3) we calculate $\hat{p}(x_0)$. We set the number of particles, M, to 10000 to estimate a state.

3.3.1 Linear Gaussian State Space Model

A linear Gaussian state space model is defined as

$$\begin{aligned} x_t &= x_{t-1} + v_t, \\ y_t &= x_t + \epsilon_t, \end{aligned} \tag{3.21}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim N(0, \sigma_m^2)$. We set $\{\sigma_s, \sigma_m\} = \{1, 3\}$. The estimated state x_t is shown in Fig. 3.1 In that figure, the thick black line is the estimated state x_t based on the inverse system equation smoothing. The dotted line is the estimated state x_t based on the MC filter; the thin black line is the real state. Figure 3.1 shows that state estimation based on our smoothing algorithm is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 3.2. That figure shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well. Linear Gaussian state space models, which are most fundamental in state spade models, are estimated using the Kalman filter. In this subsection, we apply our algorithm to the model and compare Kalman smoothing and our smoothing. We compare inverse system smoothing with Kalman smoothing.

[5]Initial particles are sampled from $Uniform(10, 11)$.

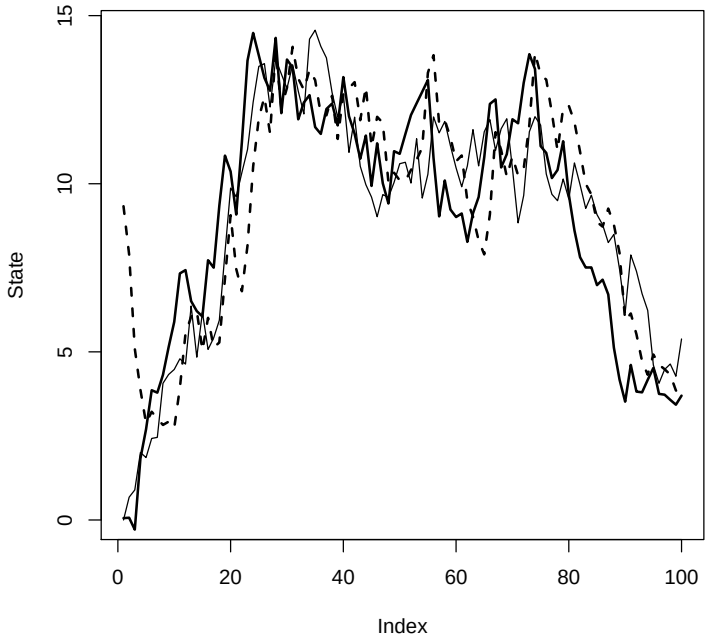

Figure 3.1: Linear Gaussian Model

Figure 3.3 shows that our algorithm realizes good state estimation as well as Kalman smoothing. This result shows that our smoothing algorithm can replace Kalman smoothing.

3.3.2 Linear Non-Gaussian State Space Model

In this subsection, we apply our algorithms to a simplest linear non-Gaussian state space model with $t-$distribution. A simple linear non-Gaussian state

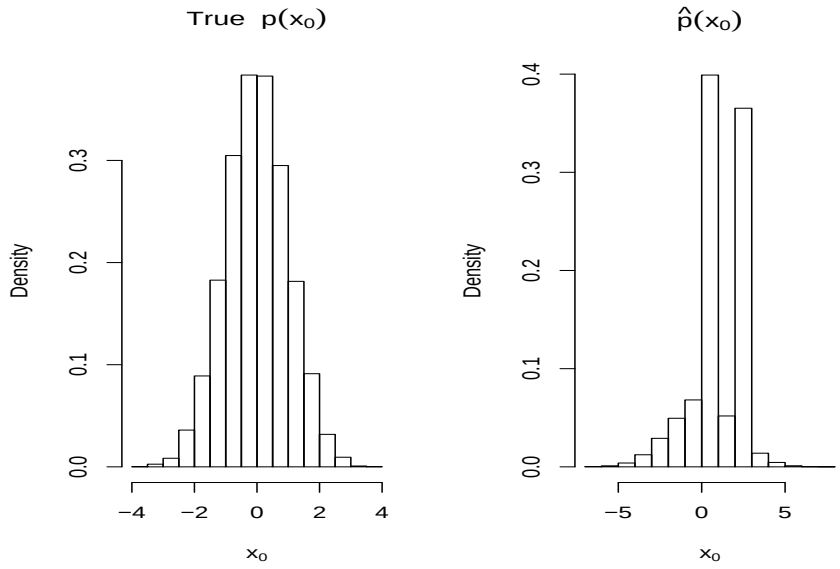

Figure 3.2: Initial Distribution (Linear Gaussian Model)

space model with $t-$distribution is defined as

$$\begin{aligned} x_t &= x_{t-1} + v_t, \\ y_t &= x_t + \epsilon_t, \end{aligned} \tag{3.22}$$

where $v_t \sim t(df)$[6] and $\epsilon_t \sim N(0, \sigma_m^2)$. We set $\{df, \sigma_m\} = \{8, 2\}$. The estimated state x_t is shown in Fig. 3.4. It shows that state estimation based on our smoothing algorithm is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 3.5. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

[6]The acronym *df* represents degrees of freedom of the $t-$distribution.

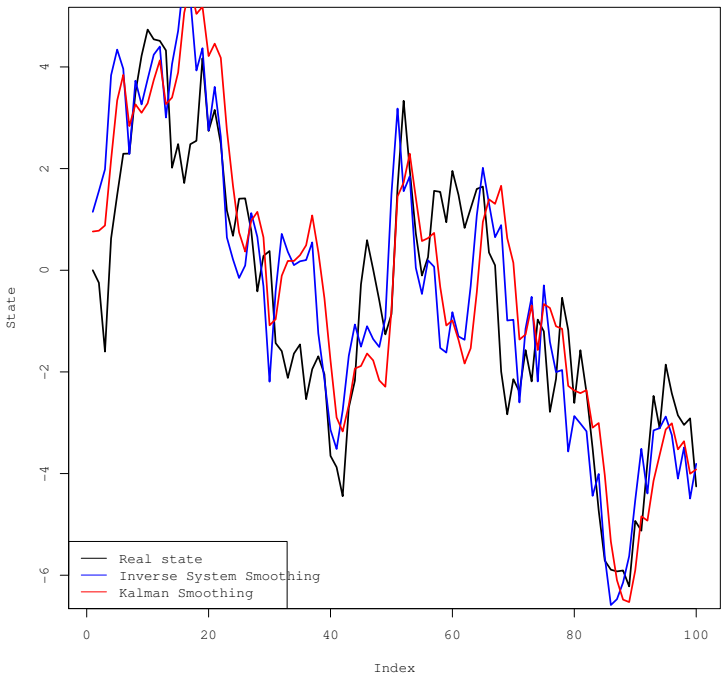

Figure 3.3: Compare inverse system smoothing with Kalman smoothing

3.3.3 Stochastic Volatility Model

The stochastic volatility model, which is introduced by [Tay86] is adopted to model the autoregressive behavior of the volatility and non-Normality in the returns that constitute a financial time series. The simplest stochastic volatility model is defined as

$$\begin{aligned} x_t &= \alpha_s x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{3.23}$$

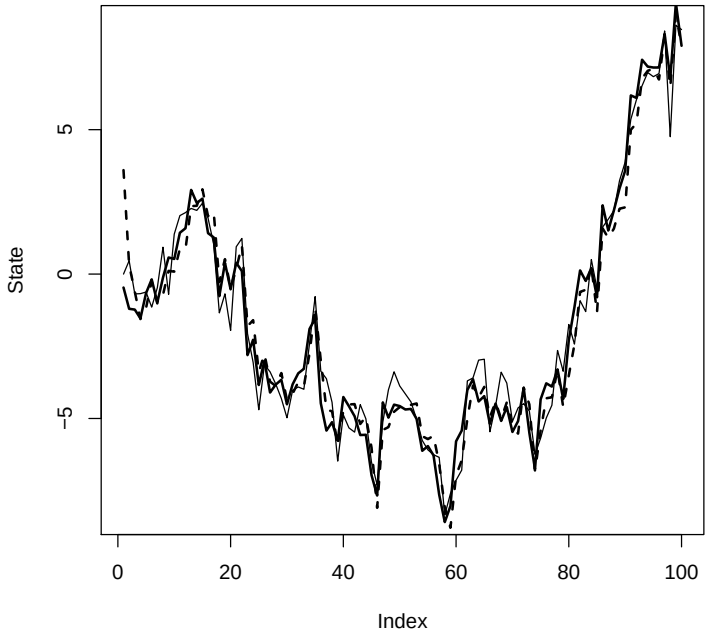

Figure 3.4: Linear Non-Gaussian Model

where $v_t \sim N(0, \sigma_s^2)$, and $\epsilon_t \sim N(0, \sigma_m^2)$. We set $\{\alpha_s, \sigma_s, \sigma_m\} = \{0.8, 1, 1\}$. The estimated state x_t is shown in Fig. 3.6. It shows that state estimation based on our smoothing algorithm is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 3.7. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

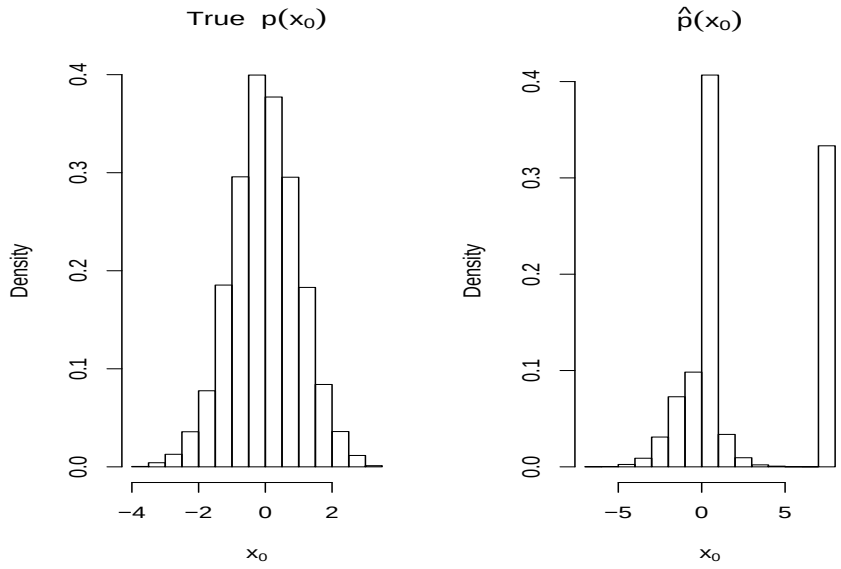

Figure 3.5: Initial Distribution (Linear Non-Gaussian Model)

3.3.4 Stochastic Volatility Model with t-distribution

A stochastic volatility model with a t-distribution, which is introduced by [LJ00], is defined as

$$\begin{aligned} x_t &= \alpha_s x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{3.24}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim t(df)$. We set $\{\alpha_s, \sigma_s, df\} = \{0.8, 1, 4\}$. The estimated state x_t is shown in Fig. 3.8. It shows that state estimation based on our smoothing algorithm is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 3.9. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

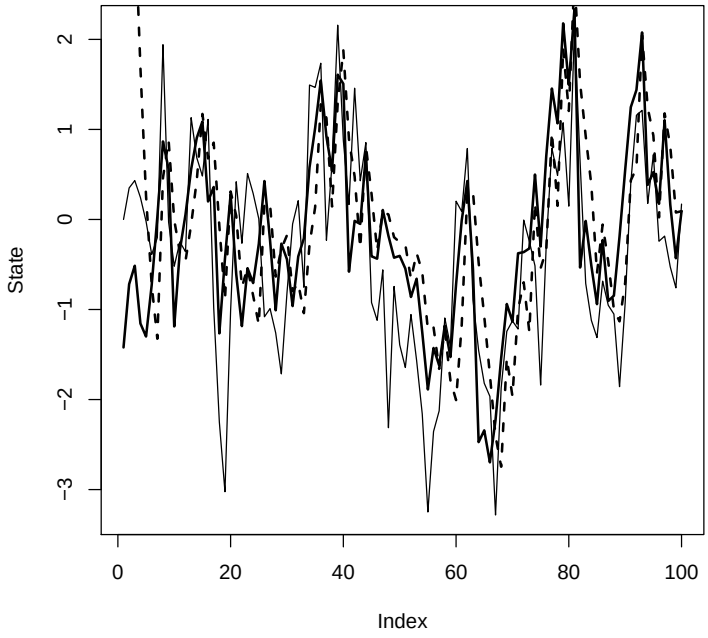

Figure 3.6: Stochastic Volatility Model

3.4 Conclusions

We proposed a smoothing algorithm based on the Monte Carlo filter and an inverse system function. Our method is applicable to any nonlinear non-Gaussian state space model if an inverse system equation is obtained analytically. The advantage of our smoothing algorithm is its simplicity. It is a minor modification of the "standard" MC filter. Moreover, our algorithm requires little memory to store the weights of particles. Nevertheless, an obstacle to its implementation remains: in general, the inverse function of a system function cannot be obtained analytically. Moreover, we pro-

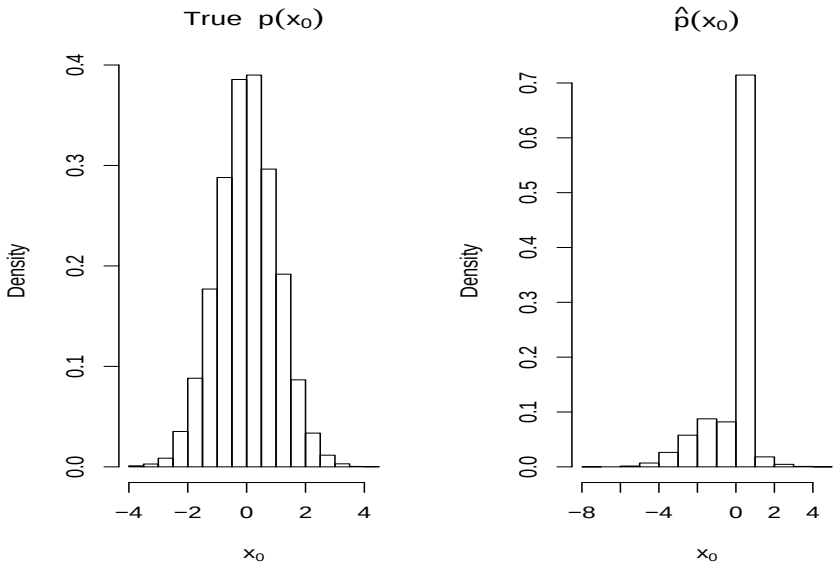

Figure 3.7: Initial Distribution (Stochastic Volatility Model)

pose a filter initialization algorithm based on the smoothing distribution $p(\boldsymbol{x}_1|\boldsymbol{y}_{1:T})$ and an inverse system equation. Our filter initialization algorithm is very simple to implement and realizes good approximation of a real initial distribution.

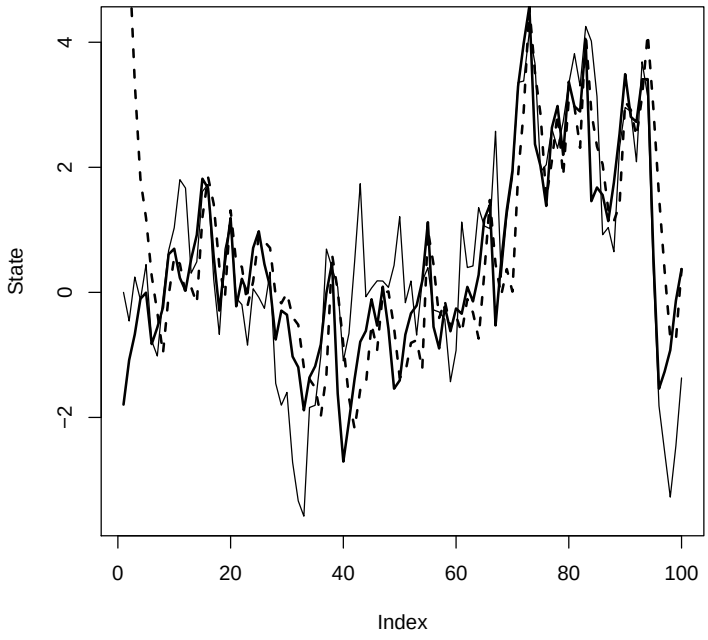

Figure 3.8: Stochastic Volatility Model with a t-distribution

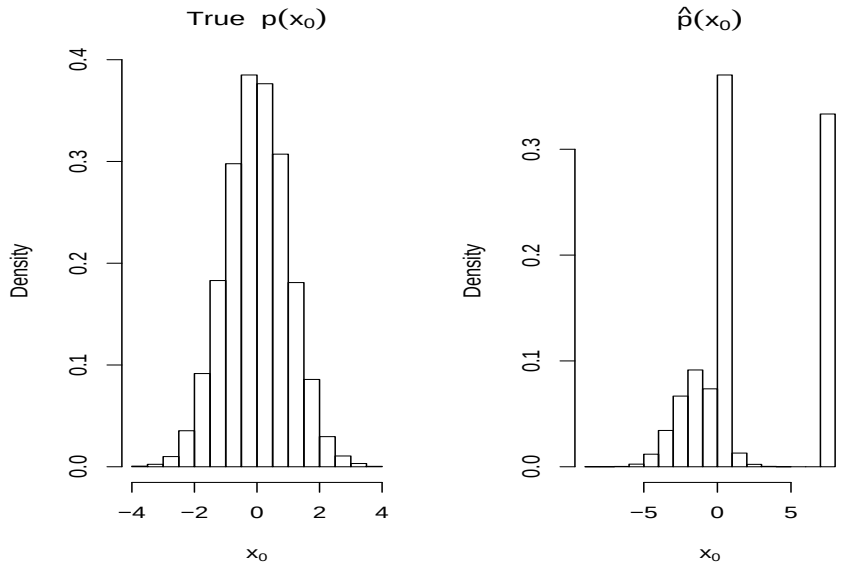

Figure 3.9: Initial Distribution (Stochastic Volatility Model with t-distribution)

Chapter 4

Yen/Dollar Exchange Rate and Stochastic Volatility Models with Heavy-tailed Distributions

4.1 Introduction

It is well known that the volatility of asset price return changes randomly over time. This phenomenon is referred to as volatility clustering. In [Man63], volatility clustering is noted: "large changes tend to be followed by large changes and small changes tend to be followed by small changes." It is also widely recognized that the distribution of asset price returns has fat-tails. There are many previous studies on these stylized facts of financial

time series (see [CLM97] and references therein). Modeling these stylized facts, continuous-time stochastic volatility models are often used in finance theory, for example [HW87]. In empirical analysis, two types of discrete-time stochastic volatility models are often used: the autoregressive heteroskedastic (ARCH) and stochastic volatility (SV) models. In this paper, we focus on SV models. A basic univariate SV model for the asset price return, y_t, is given by

$$\begin{aligned} x_t &= \mu + \phi x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{4.1}$$

where $v_t \sim N(0, \sigma_s^2)$, $\epsilon_t \sim N(0,1)$, μ and ϕ are constants. We assume $|\phi| < 1$. The symbol x_t is considered as the latent volatility of the asset price return.

In recent years, Markov Chain Monte Carlo (MCMC) method is introduced to estimate SV models by [JPR94] and MCMC techniques are further developed by many studies (See [She05]). Furthermore, [LJ00] proposes a method to estimate SV models with heavy-tailed distribution based on simulated maximum likelihood estimation. In the paper, ϵ_t is assumed to be t-distribution or generalized error distribution. [WA03] analyze the daily return of Yen/Dollar exchange rate and TOPIX using SV models with heavy-tailed distributions [1]. Unlike previous studies using MCMC, we propose an approach based on a self-organizing state space model, which is proposed by [Kit98], and a method to seek initial distribution for the model, which

[1] [WA03] uses MCMC techniques to estimate SV models with heavy-tailed distributions.

is proposed by [Yan07] to estimate parameters and volatility of SV models. Our approach is based on the Monte Carlo (MC) filter, which is proposed by [Kit96] and [GSS93][2]. The main feature of our approach is parameters and volatility are estimated simultaneously using a self-organizing state space model with simplex initial distribution search.

This paper is organized as follows. In section 2, we describe the MC filter, a self-organizing state space model, and simplex initial distribution search. In section 3, we apply our method to the daily return of Yen/Dollar exchange rate. In section 4, we describe conclusions and discussions.

4.2 Empirical Application

We apply our method to the daily return of Yen/Dollar exhange rate from 4th, January, 1990 to 28th, December, 1999. The daily returns are obtained from

$$y_t = 100 \times \big\{\log(P_{t+1}) - \log(P_t)\big\}, \tag{4.2}$$

where P_t is the closing price of day t. The descriptive statistics are shown in Table 4.1. In the following subsections, we estimate three types of SV models. The distributions of error terms of SV models are listed in Table 4.2. EPD means the Exponential Power Distribution.

[2][KSC98] proposes likelihood estimation based on the MC filter when parameters are obtained using MCMC.

Table 4.1: Descriptive Statistics of Daily Return (%) for the Yen/Dollar Exchange Rate

Statistics	Return of Yen/Dollar Rate
Sample Size	2510
Mean	-0.0134
St. Dev.	0.7388
Kurtosis	4.4365
Skewness	-0.5828

Table 4.2: SV, SVMT, and SVST models

Model	ϵ_t	v_t
SV	Normal distribution	Normal distribution
SVMT	t-distribution	Normal distribution
SVST	Normal distribution	t-distribution
SVME	EPD	Normal distribution
SVSE	Normal distribution	EPD

4.2.1 Stochastic Volatility Model

A SV model is given by

$$\begin{aligned} x_t &= \mu + \phi x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{4.3}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim N(0, 1)$. The results of our estimation of SV model are shown in Table 4.3 (the results of [WA03] are also shown). The result of $\exp(\frac{x_t}{2})$ is shown in Fig. 4.1[3].

[3]This result is estimated based on our smoothing algorithm in chapter 3.

Table 4.3: Results of SV model

	Log-likelihood	μ	ϕ	σ_s
Initial Value		0	0	0
Parameter Estimate	-2564.82	0.0049	0.8544	0.3092
[WA03]	–	-0.9218	0.9583	0.2113

4.2.2 Stochastic Volatility Models with t-distributions

A SVMT model, which is introduced by introduced by [LJ00], is given by

$$\begin{aligned} x_t &= \mu + \phi x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{4.4}$$

where $v_t \sim N(0, \sigma_s^2)$, $\epsilon_t \sim t(\nu)$, and ν is the degree of freedom of t-distribution. The results of our estimation of the SVMT model are shown in Table 4.4 (the results of [WA03] are also shown). The result of $\exp(\frac{x_t}{2})$ is shown in Fig. 4.2.

Table 4.4: Results of SVMT Model

	Log-likelihood	μ	ϕ	σ_s	ν
Initial Value		0	0	0	0
Parameter Estimate	-2556.66	-0.0246	0.8984	0.2922	3.7892
[WA03]	–	-0.8606	0.9827	0.1234	8.1161

SVST model is given by

$$\begin{aligned} x_t &= \mu + \phi x_{t-1} + \chi v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{4.5}$$

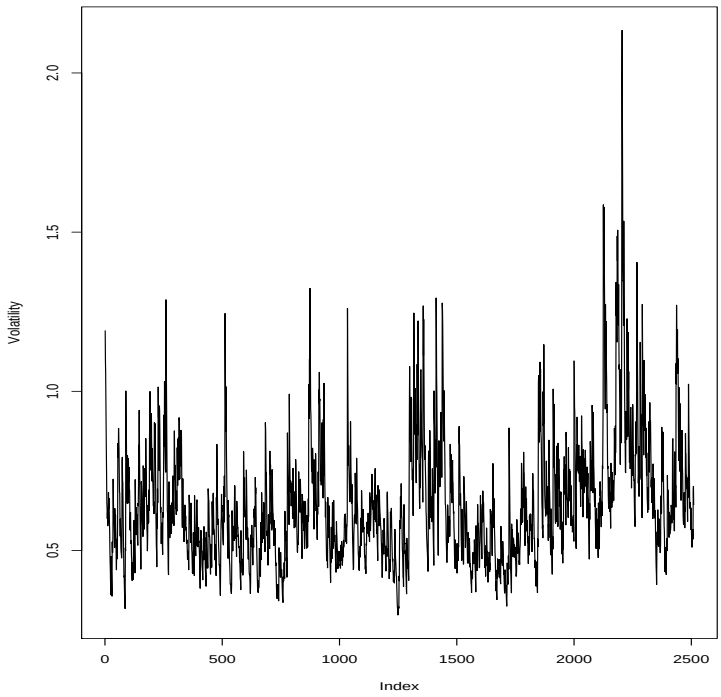

Figure 4.1: Volatility of SV Model

where $v_t \sim t(\nu)$, $\epsilon_t \sim N(0,1)$, ν is the degree of freedom of t-distribution, and χ is a constant. The results of our estimation of the SVST model are shown in Table 4.5. The result of $\exp(\frac{x_t}{2})$ is shown in Fig. 4.3.

4.2.3 Stochastic Volatility Models With Exponential Power Distribtuion

The Exponential Power Distribution has density function given by

$$f_p(x) = \frac{1}{2p^{(1/p)}\Gamma(1+1/p)\sigma_p} \exp(-\frac{|x-\mu_p|^p}{p\sigma_p^p}), \tag{4.6}$$

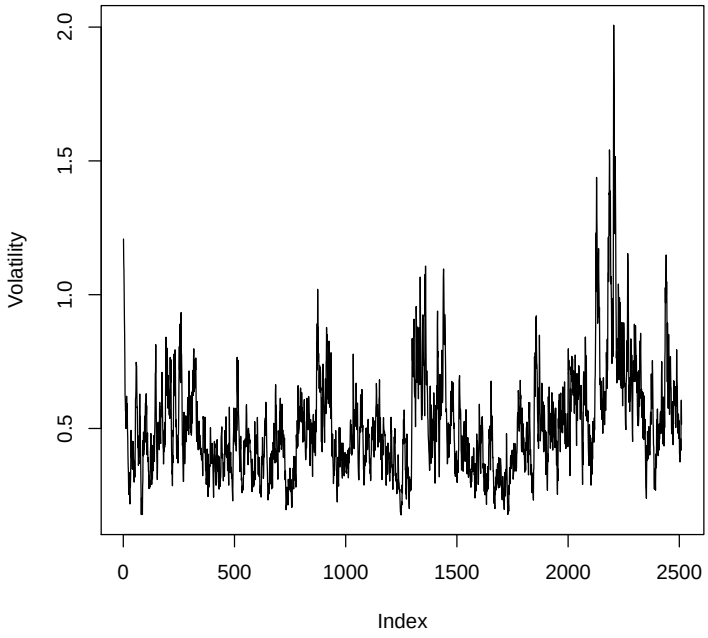

Figure 4.2: Volatility of SVMT model

where μ_p is the location parameter, σ_p is the scale parameter, and p is the shape parameter. When $p = 2$ the EPD becomes the Gaussian distribution, when $p \to \infty$ the EPD becomes the uniform distribution (See [BT73] and [MR05].). In this paper, we set μ_p to 0 and σ_p to 1.

A SVME model is given by

$$\begin{aligned} x_t &= \mu + \phi x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{4.7}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim EPD(p)$.

Table 4.5: Results of SVST Model

	Log-likelihood	μ	ϕ	χ	ν
Initial Value		0	0	0	0
Parameter Estimate	-2574.24	-0.0028	0.9239	0.0131	2.1383

Table 4.6: Results of SVME Model

	Log-likelihood	μ	ϕ	σ_s	p
Initial Value		0	0	0	2
Estimate	-2577.67	-0.0021	0.9417	0.4521	1.3451
[WA03]		0.1197	0.9627	0.2154	1.6975

A SVSE model is given by

$$\begin{aligned} x_t &= \mu + \phi x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{4.8}$$

where $v_t \sim EPD(p)$ and $\epsilon_t \sim N(0, \sigma_s^2)$.

Table 4.7: Results of SVSE Model

	Log-likelihood	μ	ϕ	χ	p
Initial Value		0	0	0	2
Estimate	-2595.66	-0.0010	0.8656	0.3623	1.2302

4.2.4 Comparison of SV Models

The log-likelihood of each model appears in Table 4.8. It shows that SVMT model is best, SV model is second best.

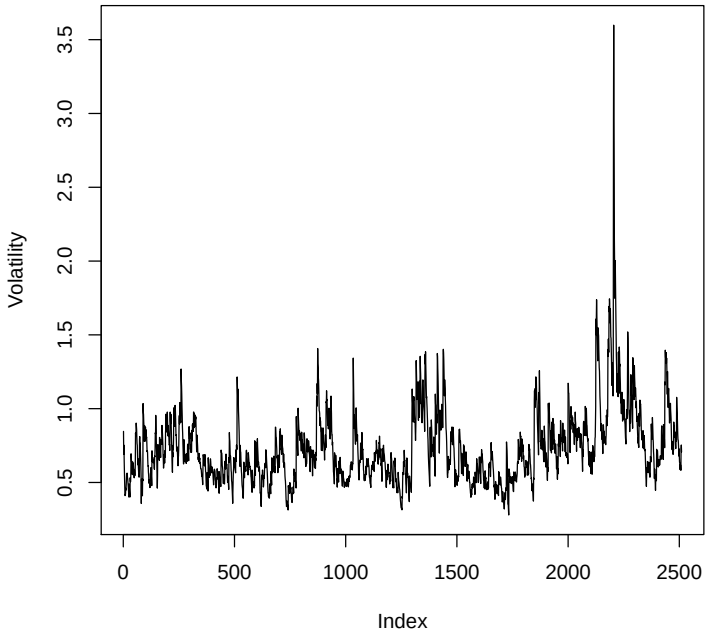

Figure 4.3: Volatility of SVST Model

The AIC of each model appears in Table 4.9. It shows that SVMT model is best, SV model is second best, and SVST is worst according to AIC.

4.3 Conclusions and Discussions

In this paper, we propose an approach based on a self-organizing state space model, which is proposed by [Kit98] and a method to seek initial distribution for the model, which is proposed by [Yan07] to estimate parameters and volatility of stochastic volatility models. We show the effectiveness of our approach by applying it to stochastic volatility models and stochastic

Table 4.8: Results of Log-likelihood

Model	Log-likelihood
SVMT	-2556.66
SV	-2564.82
SVST	-2574.24
SVME	-2577.67
SVST	-2595.66

Table 4.9: AIC

Model	AIC
SV	5153.64
SVMT	5121.32
SVST	5156.48

volatility models with t-distributions.

Chapter 5

Estimating Time Varying Linear Systems and Control: Applications to Monetary Policy

5.1 Introduction

We propose a method to estimate a time varying linear system equation based on time varying coefficients vector autoregressive modeling (time varying VAR) and control the system. In our framework, an optimal feedback is determined by linear quadratic dynamic programming in each period. In our empirical analyses, we apply our method to monetary policy because the economy can be regarded as a time varying system. Recently, several re-

searchers have focused on instrument rules and targeting regimes[1]. Various types of instrument rules have been proposed in several studies. For example, the Taylor rule, proposed by [Tay93], is the best-known simple instrument rule, and there exist many variants of it. Additionally, in recent years, targeting regimes, typically inflation targeting, have been adopted by many central banks. For the central bank which must achieve some specific targets, we propose a method to construct instrument rules whose coefficients are time varying. Our approach is a statistical tool for the practitioners in the central bank. The approach consists of two elements: (1) time varying coefficients Vector Autoregressive modeling (time varying VAR) with the vector of control variables (the control vector) and (2) linear quadratic dynamic programming (LQDP). Time varying VAR is proposed by [JK93]. The coefficients of time varying VAR are assumed to change "gradually." This assumption is widely known as smoothness priors of the Bayesian procedure, proposed by [Kit83]. The coefficients are estimated by the Kalman filer. The first improvement of this paper is that we include the vector of control variables (the control vector) in time varying VAR. Another improvement is that we estimate the transmission lags from monetary policy to the states of the economy (the monetary policy lags) based on Akaike Information Criterion (AIC), proposed by [Aka73], because previous studies, for example, [CEE99] reports that there exist the monetary policy lags. We combine the estimated time varying VAR with the control vector and LQDP to construct instrument rules. This approach is a natural extension

[1] [Wal03] and [Woo03] is an excellent, comprehensive, and readable survey on monetary theory and policy.

of [Aka70] and [Aka89]. In our empirical analyses, we show the effectiveness of our approach by applying it to the inflation target of the United Kingdom and the nominal growth rate target of Japan. According to the empirical analyses, we conclude that (1) our method can duplicate the actual monetary policies of the United Kingdom and Japan, (2) it minimizes the fluctuations of control variables, (3) the optimal policy lags are time varying, and (4) the changes of monetary policies affect the coefficients of control variables.

The basic concept of this paper is that the central bank under targeting regimes generates an instrument rule whose coefficients are time varying to achieve its targets. We refer to the generated rule as *the dynamic instrument rule*. The dynamic instrument rule is defined as follows:

$$\boldsymbol{U}(s) = -\boldsymbol{F}(s)(\boldsymbol{Y}(\hat{s}) - \tilde{\boldsymbol{Y}}), \tag{5.1}$$

where $\boldsymbol{U}$ is a vector of control variables, $\boldsymbol{Y}$ is a vector of state variables, $\tilde{\boldsymbol{Y}}$ is a vector of targets, and $\boldsymbol{F}$ is a time varying vector (state variables, control variables, and targets are discussed in the following paragraph) [2]. The s and $\hat{s}$ are the time indices. $\boldsymbol{U}$, $\boldsymbol{Y}$, and $\boldsymbol{F}$ are time dependent. In the first step to construct the dynamic instrument rule, the time varying VAR with the control vector is estimated (we call the time varying VAR with the control vector "time varying VAR" hereafter). The coefficients of time varying VAR are estimated based on the Kalman filter using recently updated data. Therefore, the coefficients always reflect the newest conditions of the

[2] Vectors and matrices are bold-faced types in this paper.

economy. In the second step, the dynamic instrument rule is derived based on the estimated time varying VAR and standard LQDP. These two steps are repeated in each period. The central bank can construct its own rule in a flexible manner because we propose the generalized approach to construct the dynamic instrument rule. Thus, our approach is convenient and effective for the practitioners in the central bank when they are unaware of the true model of the economy, due to the existence of model uncertainty which is pointed out by [Sma] [3]. Model uncertainty can be minimized based on AIC which is an unbiased estimator of the Kullback-Leiber information. The Kullback-Leiber information is a pseudo-distance between a true model and a statistical model. Thus, we minimize model uncertainty based on AIC

The necessity of the dynamic instrument rules is explained. The most influential instrument rule is the Taylor rule, proposed by [Tay93]. The rule is expressed as follows:

$$R_t = \bar{R} + \pi_t + f_\pi(\pi_t - \pi^*) + f_y \tilde{y}_t, \tag{5.2}$$

where R_t is the short nominal interest rate, $\bar{R}$ is the long average real rate of interest, π_t is an average of the recent inflation rates, π^* is the target inflation rate of a central bank, and $\tilde{y}_t$ is an output gap. f_π and f_y are coefficients. In the original formulation of [Tay93], f_π and f_y are 0.5 and 0.5, respectively, and the target inflation rate of the Federal reserve bank is set at 2. [Tay93] demonstrates that the rule explains well the path of the federal funds rate. However, there might exist many variants of targeting because

[3] [Sma] describes that there exist different views on the optimal monetary policy: Keynesian IS/LM, new classical equilibrium business cycle, monetarism, and public choice.

several central banks might have several specific conditions of their own countries. For example, the Japanese government has adopted a nominal growth rate target of 2% from fourth quarter of 2003, and the Bank of Japan must collaborate with the Japanese government to achieve the target. Thus, there exists a practical problem to tune the coefficients of a instrument rule for each country. Furthermore, the coefficients of instrument rules might change in response to the changes of current and future policies because [Luc76] points out that the changes of current and future policies might affect the behaviors of economic agents. There is a possibility that the changes of monetary policy affect the coefficients of the instrument rules. The coefficients of the Taylor rule are assumed to be fixed. In contrast, the coefficients of dynamic instrument rules are time varying to reflect the newly updated condition of the economy.

The way to construct the dynamic instrument rules is described in terms of the control theory. For the dynamic instrument rules, we need to determine the following four components.

- Targets: a specific rate of inflation, a specific growth rate of nominal output, etc.
- State variables: the growth rate of the consumer price index, the growth rate of nominal output, etc.
- Control variables: the raise in nominal short interest, the growth rate of monetary base, etc.
- Policy Lags: quarters, months, etc.

First, targets are set by the consolidated government (the central bank and the government). Inflation targeting is considered as the most popular kind of targeting in the world. For example, the original formulation of the Taylor rule chooses the 2% inflation rate target. Second, state variables are decided depending on the choice of targets. The growth rate of an price index must be included in state variables, if inflation targeting is chosen. Additionally, the other state variables should be chosen if required. In the case of the Taylor rule, the long average real rate of interest, an average of recent inflation rates and an output gap are chosen. Third, control variables are chosen adequately in order to achieve the targets. The Taylor rule adopts the nominal rate of interest. Fourth, policy lags should be chosen in an appropriate statistical manner. In [Tay93], the problem of policy lags is not explicitly discussed. We determine the policy lags based on AIC and this problem is discussed in section 2.

[Lit] has the concepts that are similar to ours. [Lit] uses Bayesian vector autoregressive modeling and LQDP for minimizing the fluctuations of the money supply and the volatility of interest rates. The essential difference between [Lit] and our study is that our method is designed for constructing the dynamic instrument rule to achieve targets; however, the purpose of [Lit] is to minimize the fluctuations of the monetary variables. Our method is inspired from recent discussions of instrument rules and targeting regimes, but [Lit] lacks of this point of view. In addition, we use estimate the goodness of fit of a model based on AIC, and policy lags are also infered based on AIC

This paper proceeds in the following manner. In section 2, the general

formulation of the dynamic instrument rule is described. In section 3, we apply empirical analyses to the inflation targeting of the United Kingdom and the nominal growth rate targeting of Japan. In section 4, we describes discussions and conclusions.

5.2 The Model

Our method has three steps:

1. Estimate time varying VAR with instantaneous response based on the Kalman filter.

2. Derive time varying VAR and construct a state space model for LQDP.

3. Determine coefficients of a dynamic instrument rule.

These three steps are repeated in each period.

5.2.1 Time Varying VAR with Instantaneous Response

In the first step of our method, we define time varying VAR with instantaneous response and estimate it. In our environment, we assume that the time series of discrete data. We particularly concentrate on the non-stationary time series. We define time varying VAR with instantaneous response,

$$\boldsymbol{y}(t) = \boldsymbol{B}_0(t)\boldsymbol{y}(t) + \sum_{l=1}^{p} \boldsymbol{B}_l(t)\boldsymbol{y}(t-l) + \boldsymbol{D}(t)\boldsymbol{u}(t-\kappa) + \boldsymbol{\epsilon}(t), \tag{5.3}$$

where $\boldsymbol{\epsilon}(t) = \big(\epsilon_1(t), \cdots, \epsilon_k(t)\big)^T \sim N(0, \boldsymbol{V})$ with $\boldsymbol{V} = diag(\sigma_1^1, \sigma_2^2, \cdots, \sigma_k^2)$, $\boldsymbol{y}(t)$ is a $(k \times 1)$ vector of state variables at time t, $\boldsymbol{u}(t)$ is an $(m \times 1)$ vector

of control variables at time t, and $\kappa \geq 1$ is a policy lag[4]. The matrices of time varying coefficients are

$$\boldsymbol{B}_0(t) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ b_{2,1,0}(t) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ b_{k,1,0}(t) & \cdots & b_{k,k-1,0}(t) & 0 \end{bmatrix}, \tag{5.4}$$

$$\boldsymbol{B}_l(t) = \begin{bmatrix} b_{1,1,l}(t) & \cdots & b_{1,k,l}(t) \\ \vdots & \ddots & \vdots \\ b_{k,1,l}(t) & \cdots & b_{k,k,l}(t) \end{bmatrix}, \tag{5.5}$$

$$\boldsymbol{D}(t) = \begin{bmatrix} d_{1,1}(t) & \cdots & d_{1,m}(t) \\ \vdots & \ddots & \vdots \\ d_{k,1}(t) & \cdots & d_{k,m}(t) \end{bmatrix}, \tag{5.6}$$

$$l = 1, 2, \cdots, p.$$

We assume that $b_{i,j,l}$ and $d_{i,n}$ change gradually over time, and therefore, the shift of $b_{i,j,l}$ and $d_{i,n}$ is locally smooth. In our approach, we adopt Bayesian smoothness priors, which are expressed in the form of a difference equation excited by a Gaussian white noise. The time varying coefficients are estimated by the Kalman filter. We define the difference equation on

[4]Time varying VAR with instantaneous response is very convenient for applying the minimum AIC procedure on the ith component of equation (5.3) because it has the diagonal innovation matrix $\boldsymbol{V}$. This type of VAR with instantaneous response is introduced by [Aka68]. In recent years, structural VAR is used by economists. The structural VAR has the same formulation of the VAR with instantaneous response. See subsection 2.3 and [KG96].

$b_{i,j,l}(t)$ as follows:

$$\begin{aligned} \Delta^q b_{i,j,l}(t) &\equiv \nu_{i,j,l}(t), \\ i,j &= 1,2,\cdots,k, \;\; l = 0,1,\cdots,p, \end{aligned} \tag{5.7}$$

where Δ^q is the q-th order difference operator defined by

$$\begin{aligned} \Delta b_{i,j,l}(t) &= b_{i,j,l}(t) - b_{i,j,l}(t-1), \\ \Delta^q b_{i,j,l}(t) &= \Delta(\Delta^{q-1} b_{i,j,l}(t)). \end{aligned} \tag{5.8}$$

In general, $\nu_{i,j,l}(t)$ is assumed to be a Gaussian white noise with mean zero and variance $\tau^2_{i,j,l}$. In this paper, we assume that $\tau^2_{i,j,l} = \tau^2_i$ and $\nu_{i,j,l}(t) \sim N(0, \tau^2_i)$. This assumption is a key feature of time varying VAR because it reduces the number of hyper parameters τ to be estimated[5]. We discuss about our assumption in the subsection 2.4. In the same manner, we define the difference equation on $d_{i,n}(t)$,

$$\begin{aligned} \Delta^q d_{i,n}(t) &\equiv \xi_{i,n}(t), \\ i &= 1,2,\cdots,k, \;\; n = 0,1,\cdots,m, \end{aligned} \tag{5.9}$$

where Δ^q is the q-th order difference operator defined by

$$\begin{aligned} \Delta d_{i,n}(t) &= d_{i,n}(t) - d_{i,n}(t-1), \\ \Delta^q d_{i,n}(t) &= \Delta(\Delta^{q-1}_{i,n}(t)). \end{aligned} \tag{5.10}$$

We assume $\xi_{i,n}(t) \sim N(0, \rho^2_i)$, where ρ^2_i is variance.

[5]First order smoothness priors without this assumption are widely known as *random walk priors*. Smoothness priors are generalized more than random walk priors. See [CP76].

5.2.2 The Estimation of Time Varying Coefficients

The state space model for the Kalman filter algorithm is shown in this subsection. The $b_{i,j,l}(t)$ and $d_{i,n}(t)$ are estimated by using the Kalman filter. The state space representation is given by

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{F}\boldsymbol{x}(t-1) + \boldsymbol{G}\boldsymbol{w}(t), \\ y_i(t) &= \boldsymbol{H}(t)\boldsymbol{x}(t) + \epsilon_i(t), \\ i &= 1, 2, \cdots, k, \end{aligned} \tag{5.11}$$

where $\boldsymbol{F}$, $\boldsymbol{G}$, $\boldsymbol{H}(t)$ are $(M \times M)$, $(M \times L)$, and $(1 \times M)$ matrices, respectively. $\boldsymbol{x}(t)$ is an $(M \times 1)$ vector of coefficients, $\boldsymbol{w}(t)$ is an $(L \times 1)$ vector, and $y_i(t)$ is an observation. The detail of these vectors and matrices are explained in the following paragraphs.

In particular, for the difference equation order $q = 1, 2$, the matrices $\boldsymbol{F}$, $\boldsymbol{G}$ and the length M are given by

$$\begin{aligned} &q = 1: \ M = L, \ \boldsymbol{F} = (\boldsymbol{I}_L), \ \boldsymbol{G} = (\boldsymbol{I}_L), \\ &q = 2: \ M = 2L, \ \boldsymbol{F} = \begin{bmatrix} 2\boldsymbol{I}_L & -\boldsymbol{I}_L \\ \boldsymbol{I}_L & \boldsymbol{0} \end{bmatrix}, \ \boldsymbol{G} = \begin{bmatrix} \boldsymbol{I}_L \\ \boldsymbol{0} \end{bmatrix}, \end{aligned} \tag{5.12}$$

where $\boldsymbol{I}_L$ is an L-dimensional identity matrix and $L = kp + m + i - 1$.

For the convenience of the expression, we use the following notations:

$$
\begin{aligned}
\boldsymbol{\nu}_{i,0}(t) &= \big(\nu_{i,1,0}(t), \cdots, \nu_{i,i-1,0}(t)\big), \\
\boldsymbol{\nu}_{i}(t) &= \big(\nu_{i,1,1}(t), \nu_{i,2,1}(t), \cdots, \nu_{i,k,1}(t), \\
&\qquad \nu_{i,1,2}(t), \cdots, \nu_{i,k,2}(t), \cdots, \nu_{i,1,p}(t), \cdots, \nu_{i,k,p}(t)\big), \\
\boldsymbol{\xi}_{i}(t) &= \big(\xi_{i,1}(t), \xi_{i,2}(t), \cdots, \xi_{i,m}(t)\big), \\
\boldsymbol{b}_{i,0}(t) &= \big(b_{i,1,0}(t), \cdots, b_{i,i-1,0}(t)\big), \\
\boldsymbol{b}_{i}(t) &= \big(b_{i,1,1}(t), b_{i,2,1}(t), \cdots, b_{i,k,1}(t), \\
&\qquad b_{i,1,2}(t), \cdots, b_{i,k,2}(t), \cdots, b_{i,1,p}(t), \cdots, b_{i,k,p}(t)\big), \\
\boldsymbol{d}_{i}(t) &= \big(d_{i,1}(t), d_{i,2}(t), \cdots, d_{i,m}(t)\big), \\
\boldsymbol{r}_{i}(t) &= \big(y_1(t), y_2(t), \cdots, y_{i-1}(t)\big), \\
\boldsymbol{h}(t) &= \big(y_1(t-1), y_2(t-1), \cdots, y_k(t-1), \cdots, \\
&\qquad y_1(t-p), y_2(t-p), \cdots, y_k(t-p)\big), \\
\boldsymbol{f}(t) &= \big(u_1(t-\kappa), u_2(t-\kappa), \cdots, u_m(t-\kappa)\big).
\end{aligned}
\tag{5.13}
$$

The vectors, $\boldsymbol{x}(t)$, $\boldsymbol{H}(t)$, and $\boldsymbol{w}(t)$, can be defined as follows. For the first component of $\boldsymbol{y}(t)$, $i = 1$,

$$
\boldsymbol{w}(t) = \big(\boldsymbol{\nu}_1(t), \boldsymbol{\xi}_1(t)\big)^T. \tag{5.14}
$$

For $q = 1$ (the difference of $\boldsymbol{b}_{i,j,l}(t)$ is first order),

$$
\begin{aligned}
\boldsymbol{x}(t) &= \big(\boldsymbol{b}_1(t), \boldsymbol{d}_1(t)\big)^T, \\
\boldsymbol{H}(t) &= \big(\boldsymbol{h}(t), \boldsymbol{f}(t)\big).
\end{aligned}
\tag{5.15}
$$

For $q = 2$ (the difference of $\boldsymbol{b}_{i,j,l}(t)$ is second order),

$$\begin{aligned} \boldsymbol{x}(t) &= \big(\boldsymbol{b}_1(t), \boldsymbol{d}_1(t), \boldsymbol{b}_1(t-1), \boldsymbol{d}_1(t-1)\big)^T, \\ \boldsymbol{H}(t) &= \big(\boldsymbol{h}(t), \boldsymbol{f}(t), \boldsymbol{0}_L\big), \end{aligned} \tag{5.16}$$

where $\boldsymbol{0}_L$ is L variate zero vector. For the ith component of $\boldsymbol{y}(t)$, $1 < i \leq k$,

$$\boldsymbol{w}(t) = \big(\boldsymbol{\nu}_{i,0}(t), \boldsymbol{\nu}_i(t), \boldsymbol{\xi}_i(t)\big)^T. \tag{5.17}$$

For $q = 1$ (the difference of $\boldsymbol{b}_{i,j,l}(t)$ is first order),

$$\begin{aligned} \boldsymbol{x}(t) &= \big(\boldsymbol{b}_{i,0}(t), \boldsymbol{b}_i(t), \boldsymbol{d}_i(t)\big)^T, \\ \boldsymbol{H}(t) &= \big(\boldsymbol{r}_i(t), \boldsymbol{h}(t), \boldsymbol{f}(t)\big). \end{aligned} \tag{5.18}$$

For $q = 2$ (the difference of $\boldsymbol{b}_{i,j,l}(t)$ is second order),

$$\begin{aligned} \boldsymbol{x}(t) &= \big(\boldsymbol{b}_{i,0}(t), \boldsymbol{b}_i(t), \boldsymbol{d}_i(t), \boldsymbol{b}_{i,0}(t-1), \boldsymbol{b}_i(t-1), \boldsymbol{d}_i(t-1)\big)^T, \\ \boldsymbol{H}(t) &= \big(\boldsymbol{r}_i(t), \boldsymbol{h}(t), \boldsymbol{f}(t), \boldsymbol{0}_L\big). \end{aligned} \tag{5.19}$$

We suppose the white noise sequence is given by

$$\begin{bmatrix} \boldsymbol{w}(t) \\ \boldsymbol{\epsilon}_i(t) \end{bmatrix} \sim N\Big(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{Q} \\ \sigma_i^2 \end{bmatrix} \Big), \tag{5.20}$$

where

$$\boldsymbol{Q} = \begin{bmatrix} \tau_i^2 \boldsymbol{I}_{kp+i-1} & \boldsymbol{0} \\ \boldsymbol{0} & \rho_i^2 \boldsymbol{I}_m \end{bmatrix}. \tag{5.21}$$

We use the Kalman filter to estimate a predictor $\boldsymbol{x}(t|t-1)$ and a filter $\boldsymbol{x}(t|t)$

of the state vector $\boldsymbol{x}(t)$. The algorithm is expressed as follows:

Prediction:

$$\begin{aligned} \boldsymbol{x}(t|t-1) &= \boldsymbol{F}\boldsymbol{x}(t-1|t-1), \\ \boldsymbol{V}(t|t-1) &= \boldsymbol{F}\boldsymbol{V}(t-1|t-1)\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T. \end{aligned} \tag{5.22}$$

Filtering:

$$\begin{aligned} \boldsymbol{K}(t) =& \boldsymbol{V}(t|t-1)\boldsymbol{H}(t)^T\{\boldsymbol{H}(t)\boldsymbol{V}(t|t-1)\boldsymbol{H}(t)^T + \sigma_i^2\}^{-1}, \\ \boldsymbol{x}(t|t) =& \boldsymbol{x}(t|t-1) + \boldsymbol{K}(t)\{y_i(t) - \boldsymbol{H}(t)\boldsymbol{x}(t|t-1)\}, \\ \boldsymbol{V}(t|t) =& \{\boldsymbol{I} - \boldsymbol{K}(t)\boldsymbol{H}(t)\}\boldsymbol{V}(t|t-1). \end{aligned} \tag{5.23}$$

We suppose $\boldsymbol{x}(0|0)$ and $\boldsymbol{v}(0|0)$ are given.

5.2.3 Model Identification

We estimate the goodness of fit of a model based on AIC. AIC consists of the log likelihood and the number of parameters to be estimated.

The log likelihood can be derived from the Kalman filter. For the ith component, the joint distribution of $\{y_i(1), \cdots, y_i(T)\}$ is

$$f\big(y_i(1), \cdots, y_i(T)|\boldsymbol{\theta_i}\big) = \prod_{t=1}^{T} f\big(y_i(t)|y_i(1), \cdots, y_i(t-1); \boldsymbol{\theta_i}\big), \tag{5.24}$$

where T is the size of the observations, $\boldsymbol{\theta}_i = \{\tau_i, \rho_i, \sigma_i\}$, and $f\big(y_i(t)|y_i(1), \cdots, y_i(t-$

$1); \boldsymbol{\theta_i}\big)$ is defined by

$$\begin{aligned} &f\big(y_i(t)|y_i(1),\cdots,y_i(t-1);\boldsymbol{\theta_i}\big) \\ =&\big(2\pi v^2(t)\big)^{-1/2}\exp\Big[-\frac{1}{2v^2(t)}\big(y_i(t)-\boldsymbol{H}(t)\boldsymbol{x}(t|t-1)\big)^2\Big], \end{aligned} \tag{5.25}$$

where $v^2(t) = \boldsymbol{H}(t)\boldsymbol{V}(t|t-1)\boldsymbol{H}(t)^T + \sigma_i^2$. Therefore, the log likelihood is obtained by

$$\begin{aligned} l(\boldsymbol{\theta}_i) = -\frac{1}{2}\Big\{&T\log(2\pi) + \sum_{t=1}^{T}\log v^2(t) \\ &+ \sum_{t=1}^{T}\frac{1}{v^2(t)}\big(y_i(t)-\boldsymbol{H}(t)\boldsymbol{x}(t|t-1)\big)^2\Big\}. \end{aligned} \tag{5.26}$$

AIC for the ith component of $\boldsymbol{y}(t)$ is defined as follows:

$$AIC(i) = -2l(\hat{\boldsymbol{\theta}}_i) + 2(\textit{the dimension of } \boldsymbol{\theta}_i), \tag{5.27}$$

where the dimension of $\boldsymbol{\theta}_i$ is three. Therefore, the total AIC of an entire model is given by

$$AIC = \sum_{i=1}^{k} AIC(i). \tag{5.28}$$

We estimate time varying coefficients and the optimal policy lags κ based on the total AIC.

We discuss the necessity of the assumption, which we consider as $\tau_{i,j,l} = \tau_i$. To estimate log-likelihood, nonlinear numerical minimization algorithm must be used. The algorithm usually needs rich computational requirement. Our assumption reduces the computational requirement to calculate numer-

ical minimization which is the main merit of the assumption. Furthermore, this assumption can be released if required.

5.2.4 Time Varying VAR and LQDP

In the second step of our method, we derive time varying VAR from time VAR with instantaneous response,

$$\boldsymbol{y}(t) = \sum_{l=1}^{p} \boldsymbol{A}_l(t)\boldsymbol{y}(t-l) + \boldsymbol{E}(t)\boldsymbol{u}(t-\kappa) + \boldsymbol{\eta}(t), \tag{5.29}$$

where $\boldsymbol{\eta}(t) \sim N(0, \boldsymbol{\Sigma})$,

$$\begin{aligned} \boldsymbol{A}_l(t) &= \big(\boldsymbol{I}_k - \boldsymbol{B}_0(t)\big)^{-1}\boldsymbol{B}_l(t), \\ \boldsymbol{E}(t) &= \big(\boldsymbol{I}_k - \boldsymbol{B}_0(t)\big)^{-1}\boldsymbol{D}(t), \\ \boldsymbol{\eta}(t) &= \big(\boldsymbol{I}_k - \boldsymbol{B}_0(t)\big)^{-1}\boldsymbol{\epsilon}(t), \\ \boldsymbol{\Sigma}(t) &= \big(\boldsymbol{I}_k - \boldsymbol{B}_0(t)\big)^{-1}\boldsymbol{V}\big(\boldsymbol{I}_k - \boldsymbol{B}_0(t)\big)^{-T}, \end{aligned} \tag{5.30}$$

where $\boldsymbol{A}_l(t)$, $\boldsymbol{E}(t)$, $\boldsymbol{\eta}(t)$, and $\boldsymbol{\Sigma}(t)$ are $(k \times k)$, $(k \times m)$, $(k \times 1)$, and $(k \times k)$ matrices, respectively. $\boldsymbol{I}_k$ is the k-dimensional identity matrix. The policy lags, κ, are optimal lags according to AIC.

We plan to use data in $\mathcal{T} = (1, 2, \cdots, t, \cdots, T)$ estimating equation (5.29) and use the estimated functional form of (5.29) at time T for LQDP. The system, which we consider below, is optimized in period $\mathcal{S} = (0, 1, \cdots, s, \cdots, \infty)$. You can consider $\mathcal{T}$ as the past and $\mathcal{S}$ as the future (see Figure 5.1).

The third step of our method, we combine time varying VAR and LQDP.

We define the system equation for LQDP,

$$\boldsymbol{Y}(s) = \boldsymbol{\Psi}\boldsymbol{Y}(s-1) + \boldsymbol{\Gamma}\boldsymbol{U}(s-1) + \Lambda(s), \tag{5.31}$$

where

$$\boldsymbol{Y}(s) = \begin{bmatrix} \boldsymbol{y}(s) \\ \boldsymbol{y}(s-1) \\ \vdots \\ \boldsymbol{y}(s-p+1) \end{bmatrix}, \tag{5.32}$$

$$\boldsymbol{U}(s) = \begin{bmatrix} \boldsymbol{u}(s) \\ \mathbf{0}_m^{(1)} \\ \vdots \\ \mathbf{0}_m^{(p-1)} \end{bmatrix}, \tag{5.33}$$

$$\boldsymbol{\Lambda}(s) = \begin{bmatrix} \boldsymbol{\eta}(s) \\ \mathbf{0}_k^{(1)} \\ \vdots \\ \mathbf{0}_k^{(p-1)} \end{bmatrix}, \tag{5.34}$$

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{A}_1(T) & \boldsymbol{A}_2(T) & \cdots & \boldsymbol{A}_{p-1}(T) & \boldsymbol{A}_p(T) \\ \boldsymbol{I}_k & \mathbf{0}_{k\times k} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{k\times k} & \boldsymbol{I}_k & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & & & \\ \mathbf{0} & \cdots & \cdots & \boldsymbol{I} & \mathbf{0} \end{bmatrix}, \tag{5.35}$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{E}(T) & \mathbf{0}_{k\times m}^{(1)} & \mathbf{0}_{k\times m}^{(2)} & \cdots & \mathbf{0}_{k\times m}^{(p-1)} \\ \mathbf{0}_{k\times m}^{(1)} & \mathbf{0}_{k\times m}^{(1)} & \cdots & \cdots & \mathbf{0}_{k\times m}^{(1)} \\ \vdots & & & & \\ \mathbf{0}_{k\times m}^{(p-1)} & & & & \end{bmatrix}, \tag{5.36}$$

where $\boldsymbol{Y}(s)$ is a $(kp\times 1)$ matrix, $\boldsymbol{U}(s)$ is a $(mp\times 1)$ matrix, $\boldsymbol{\Lambda}(s)$ is a $(kp\times 1)$ matrix, $\boldsymbol{\Psi}$ is a $(kp\times kp)$ matrix, and $\boldsymbol{\Gamma}$ is a $(kp\times mp)$ matrix. $\mathbf{0}_k$ is a $(k\times 1)$ zero vector, $\mathbf{0}_{k'\times k''}$ is a $(k'\times k'')$ zero matrix, and I_k is a k-variate identity matrix. The loss function of the central bank is defined by

$$\sum_{s=0}^{\infty}\beta^s E_0\Big\{\Big(\boldsymbol{Y}(s+\kappa)\Big)^T\boldsymbol{R}\Big(\boldsymbol{Y}(s+\kappa)\Big)+\boldsymbol{U}(s)^T\boldsymbol{Q}\boldsymbol{U}(s)\Big\}, \tag{5.37}$$

where E_0 is the conditional expectation, given the information known at period 0, $\tilde{\boldsymbol{Y}}$ is the vector of targets, $\boldsymbol{R}$ is a $(kp\times kp)$ real positive semidefinite matrix, $\boldsymbol{Q}$ is a $(mp\times mp)$ real symmetric positive definite matrix, and $\beta\in(0,1]$ is a discount factor or unity. The optimal feedback on $\boldsymbol{U}(0)$ can be given by

$$\begin{aligned}\boldsymbol{U}(0) &= -\boldsymbol{F}(E_0[\boldsymbol{Y}(\kappa-1)]) \\ &= -\boldsymbol{F}(E_0[\bar{\boldsymbol{Y}}(\kappa-1)]-\hat{\boldsymbol{Y}}),\end{aligned} \tag{5.38}$$

where

$$\boldsymbol{F}=\beta(\boldsymbol{Q}+\beta\boldsymbol{\Gamma}^T\boldsymbol{P}\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\boldsymbol{P}\boldsymbol{\Psi}, \tag{5.39}$$

where $\boldsymbol{F}$, which is a $(mp\times kp)$ matrix, is generated in each period, $\bar{\boldsymbol{Y}}$ is real time series, and $\hat{\boldsymbol{Y}}$ is a target. We refer to the optimal feedback as *the dynamic instrument rule*. $E_0[\boldsymbol{Y}(\kappa)]$ is an expectation of $\boldsymbol{Y}(\kappa)$ at time

$s = 0$, which is generated by iterations on equation (5.29), and therefore, the dynamic instrument rule is based on the forecast. This implies that optimal monetary policies must be forecast-based. $\boldsymbol{F}$ in equation (5.39) is derived in each period. This is the reason why we refer to equation (5.38) as the "dynamic" instrument rule. $\boldsymbol{P}$ can be obtained by iterations on the matrix Riccati equation [6] given by

$$\boldsymbol{P} = \boldsymbol{R} + \beta \boldsymbol{\Psi}^T \boldsymbol{P} \boldsymbol{\Psi} - \beta^2 \boldsymbol{\Psi}^T \boldsymbol{P} \boldsymbol{\Gamma} (\boldsymbol{Q} + \beta \boldsymbol{\Gamma}^T \boldsymbol{P} \boldsymbol{\Gamma})^{-1} \boldsymbol{\Gamma}^T \boldsymbol{P} \boldsymbol{\Psi}. \tag{5.40}$$

5.3 Empirical Analyses

5.3.1 Inflation Targeting: The United Kingdom

In this subsection, we simulate the optimal short nominal interest rate raises from 2001 to 2003. The Bank of England (BOE) has adopted inflation targeting from 1992, and the target was set to 2.5 % inflation based on the RPIX (the retail price index excluding mortgage interest payments) from from 1997 to 2003[7]. [Eng] states, "The Bank (of England) aims to meet the Government's inflation target by setting short-term interest rates." BOE implicitly adopts the inflation of the RPIX as a state variable (from 1997 to 2003) and the short interest rate as a control variable. It is an ideal case for our framework.

We use the year-on-year growth rate of the RPIX as a state variable and the raise in the nominal interest rate of three-month treasury bill as a

[6] See [LS04].

[7] From 2004, BOE has adopted the inflation target of 2 %, measured by the 12-month increase in the consumer price index.

control variable. In this case, equation (5.31) can be written,

$$\pi(t) = a(t)\pi(t-1) + b(t)i(t-\kappa) + \eta(t), \tag{5.41}$$

where $\pi(t)$ is the inflation rate, $i(t)$ is the interest rate raise, $\eta(t)$ is a residual. We estimate the time varying coefficients $a(t)$ and $b(t)$, and calculate the F of the dynamic instrument rule. The estimation of coefficients are based on quarterly and seasonal adjusted data from 1987:I to 2003:IV. Optimal monetary policy is simulated from 2001:II to 2003:IV. We determine optimal estimates ($a(t)$ and $b(t)$) and optimal policy lag κ (quarters) of equation (5.41) based on AIC. Furthermore, we must check the stability of control. [AHMS96] surveys that $(\mathbf{\Psi} - \mathbf{F\Gamma})$ is said to be a stable matrix if the eigenvalues of $(\mathbf{\Psi} - \mathbf{F\Gamma})$ are all less than unity in absolute value. Thus, we can select R and Q in the loss function of a central bank [equation (5.37)] within the region the eigenvalues of $(\mathbf{\Psi} - \mathbf{F\Gamma})$ are all less than unity in absolute value. We try some sets of $\{R, Q\} = \{4000, 1\}, \{2000, 1\}, \{1000, 1\}$, and set $\beta = 1$. AICs and policy lags are shown in Table 5.1, and indices in Table 5.1 are show in Table 5.4.2. The optimal raise in the interest rate and the actual raise are shown in Figure 5.2. All results are shown in Table 5.3, 5.4, and 5.5. These results indicate that our method can duplicate the actual monetary policy. Moreover, we emphasize that the simulated raises of the nominal short interest rate is less than the actual one, and these results are consistent with [Lit]. Table 5.1 shows that the optimal policy lags are time varying. This result indicates that the central bank must estimate an optimal policy lag in each period. Additionally, Fs of equation (5.38) are shown

in Table 5.3, 5.4, and 5.5, and Fs are also time varying. Table 5.3, 5.4, and 5.5 show that policy lags are 2 and bs and Fs are positive from 2003:I to 2003:III. According to these results, equation (5.38) can be written,

$$i(t) = \bar{F}[E_t\bar{\pi}(t+1) - \hat{\pi}], \tag{5.42}$$

where i is an interest rate raise, $\bar{F}(< 0)$ is $-F$, $E_t\bar{\pi}(t+1)$ is the inflation rate expectation at time $t+1$, and $\hat{\pi}$ is the target of inflation. Equation (5.42) shows that the inflation rate expectation over the target rate decreases the nominal rate of short interest. This relation indicates that Fisher effect was dominant from from 2003:I to 2003:III.

The shifts of coefficients $a(t)$ and $b(t)$ are shown in Figure 5.3 and Figure 5.4, respectively. These results are based on quarterly data from 1991:II to 2003:IV. The $a(t)$, $b(t)$, and κ are estimated based on AIC, the κ is 9 quarters. The AICs are shown in Table 5.1. Figure 5.3 shows that the inflation rate of the United Kingdom becomes stable step by step, and we conclude that inflation targeting realizes price stability of the United Kingdom.

According to [McC], the Taylor rule explains well on monetary policy in the United Kingdom. Thus, we apply our method to simulate the optimal short nominal interest rate raises from 2001 to 2003 based on two state variables: year-on-year inflation rate and output gap. The output gap is the deviation from the trend of quarterly and seasonal adjusted real output[8].

[8]The trend is generated by Hodrirck-Prescott filter proposed by [HP97].

We use the $\mathbf{R}$ and $\mathbf{Q}$ as follows:

$$\mathbf{R} = \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix}, \quad \mathbf{Q} = 1, \tag{5.43}$$

where $R_{11} \geq 0$ and $R_{22} \geq 0$. In our analysis, we fix $R_{11} = 2000$ and try $R_{22} = 10$, $R_{22} = 5$, and $R_{22} = 2.5$. The $\mathbf{F}$ [equation (5.39)] can be written,

$$\mathbf{F} = \begin{bmatrix} F_1 & F_2 \end{bmatrix} \tag{5.44}$$

The optimal interest rate raise and the actual interest rate raise are shown in Figure 5.5. The optimal policy lags, F_1, and F_2 are shown in Table 5.6, 5.7, and 5.8. The full list of AICs and policy lags is available on request. Figure 5.5 shows that the fluctuation of the interest rate raise in this case is relatively unstable rather than the fluctuation of the interest rate raise in single state variable case.

5.3.2 Nominal Growth Rate Targeting: Japan

In this subsection, we simulate the optimal growth rate of the monetary base from 2002:I to 2004:II[9]. From 1999, the inflation of the CPI (the consumer price index excluding volatile food) is negative, and the Bank of Japan (BOJ) has adopted the zero interest rate policy from February, 1999. Nevertheless, the policy lead extremely low short interest rates, and the inflation rate did not become positive. Additionally, BOJ has adopted "quantitative easing" from March, 2001. Thus, from 2001, the control variable of BOJ is only

[9]The description in this paragraph is based on the [Jap] and [Mut].

the growth rate of the monetary base. Moreover, the Japanese government and the Liberal Democratic Party have committed to 2 % nominal growth rate targeting from the fourth quarter of 2003. In our simulation, we adopt the nominal growth rate as a state variable and the growth rate of the monetary base as a control variable. BOJ has not adopted formally targeting regimes. However we assume targeting regimes in the Japan's case because this assumption is convenient for our framework. Thus, our results of the Japanese economy are experimental.

We use the growth rate of nominal GDP as a state variable and the growth rate of monetary base as a control variable [10]. In this case, equation (5.31) can be written,

$$y_g(t) = a(t)y_g(t-1) + b(t)m_b(t-\kappa) + \eta(t), \tag{5.45}$$

where $y_g(t)$ is the growth rate of nominal output, $m_b(t)$ is the growth rate of monetary base, $\eta(t)$ is a residual. We estimate time varying coefficients $a(t)$ and $b(t)$, and calculate F of the dynamic instrument rule. The estimation of coefficients based on quarterly and seasonal adjusted data from 1980:I to 2004:III. Optimal monetary policy is simulated from 2002:IV to 2004:III. We decide the optimal estimates ($a(t)$ and $b(t)$) and the optimal policy lag κ (quarters) of equation (5.45) based on AIC. AICs and policy lags are shown in Table 5.9, and indices in Table 5.9 are shown in Table 5.10. Table 5.9 shows that AICs are monotonically decreasing in most cases. This indi-

[10]Estimating coefficients $a(t)$ and $b(t)$, the growth rate of monetary base is reduced by the average the growth rate of monetary base, and the average is added to the estimated optimal and actual growth rate of monetary base.

cates the occurrence of overfitting in nonlinear optimization. In numerical procedures, it is difficult to avoid overfitting; therefore, we must manually eliminate overfitting. We found that the coefficient $b(t)$ is negative in many periods when lags are more than 5, and an example of this case is shown in Figure 5.10. The negative $b(t)$ implies that the increasing monetary base causes decreasing nominal growth, and this can be problematic from the point of view of economics. Thus, we eliminate the problematic cases. We try some sets of $\{R, Q\} = \{100, 1\}, \{50, 1\}, \{25, 1\}$, and set $\beta = 1$. The optimal growth rate of monetary base and the actual growth rate are shown in Figure 5.6. All results are shown in Table 5.11, 5.12, and 5.13. These results indicate that the simulated optimal growth rates of monetary base are relatively stable as compared with the actual one and these are consistent with [Lit]. Additionally, Fs of equations (5.38) are shown in Table 5.11, 5.12, and 5.13, and they are also time varying. Moreover, in 2005:I, the growth rate of nominal GDP is 2.2 % (the real GDP growth rate is 4.6 %) based on initial reporting values. It is caused by about the about 20 % growth rate of monetary base from Jun. 2003, to Nov. 2003 because the monetary policy lag is about 5 quarters.

The shifts of coefficients $a(t)$ and $b(t)$ are shown in Figure 5.7 and Figure 5.8, respectively. This simulation is based on quarterly data from 1986:II to 2004:III. The $a(t)$, $b(t)$, and κ are estimated based on AIC, optimal κ is 5 quarters. The AICs are shown in Table 5.9. Figure 5.8 shows the coefficient $b(t)$ change in the first quarter of 2001, and it can be considered that the coefficient changes in response to the change of monetary policy of BOJ. This result indicates that the coefficient $b(t)$ depend on the expectation on

future monetary policy and the past time series data,

$$b(t) = E[b(t)|I_p, I_f], \tag{5.46}$$

where I_p is the information on the past and I_f is the information on the future. This is consistent with [Luc76]. We conclude that adopting time varying VAR in our method can be justified.

According to [Mut], BOJ has committed to continue the "quantitative easing" policy until the core CPI (Consumer Price Index) shows zero percent or above on a year-on-year basis in a stable manner. This commitment can be considered as zero inflation targeting.e Thus, we apply our method to simulate the optimal growth rate of monetary base from 2003:IV to 2004:III based on two state variables: the nominal growth rate and the year-on-year inflation rate[11]. In our analysis, we fix $R_{11} = 50$ and try $R_{22} = 100$, $R_{22} = 50$, and $R_{22} = 1$ in equation (5.43). The optimal growth rate of monetary base and the actual growth of monetary base are shown in Figure 5.9. The optimal policy lags, F_1, and F_2 are shown in Table 5.14, 5.15, and 5.16. The full list of AICs and policy lags is available on request.

5.4 Discussions and Conclusions

5.4.1 Discussions

In this subsection we discuss two points: (1) correlation does not imply causation, and (2) the problem of real-time data availability.

[11] In this analysis, we use the year-on-year GDP deflater as the year-on-year inflation rate because [Shi99] point out that Japanese CPI is biased.

Correlation does not imply causation. In our method, equation (5.29) indicates autocorrelation of state variables and correlation between state variables and control variables. Empirically, however, nominal short interest rates and the growth of the monetary base are instruments of monetary policy. Furthermore, our empirical analysis of the United Kingdom indicates that the short nominal interest rate raise can be considered as a control variable to control the inflation rate, and in the case of Japan indicates the growth rate of the monetary base can be considered as a control variable to control the growth rate of nominal output. In general, the phrase "correlation does not imply causation" is true. However, according to our empirical analyses, it can be considered that equation (5.29) estimates causality in the monetary policy.

Current economic data in period t could not be known until after the end of period t, and the initial reporting values are often revised later. Using the Taylor rule as an example, [Orp01] demonstrates that real-time policy recommendations differ from those obtained with ex post revised data. Our model is not affected by the problem indicated by [Orp01] because equation (5.38) determines the optimal control variables based on forecasted state values. In other words, our method does not require real-time data, but forecasted data. In our analyses, forecasted data are generated by equation (5.29) based on revised data. The practice of dynamic instrument rules involves further study. However, we suggest that forecasted data in practice can be calculated with equation (5.29) based on reliable and available data, which the central bank can acquire at that time to decide the monetary policy.

5.4.2 Conclusions

This paper proposes a statistical and practical approach that time varying coefficients monetary instrument rules are generated dynamically in each period by the central bank which must achieve some specific targets. In the empirical analyses of the United Kingdom and Japan, the simulation results show the effectiveness of our method. According to the results, we conclude (1) our method can duplicate the actual monetary policy, (2) it minimizes the fluctuation of a control variable, (3) the optimal policy lags are time varying and the optimal monetary policy must be forecast-based, and (4) policy changes affect the coefficients of control variables. The second point is consistent with [Lit] and the fourth point is consistent with [Luc76].

Index	Lag	AIC	Index	Lag	AIC	Index	Lag	AIC
1	1	-327.6496	2	1	-327.5197	3	1	-340.2939
1	2	-327.9179	2	2	-327.5108	3	2	-340.0787
1	3	-333.6233	2	3	-340.3128	3	3	-337.7012
1	4	-331.0292	2	4	-333.4488	3	4	-339.5507
1	5	-330.9126	2	5	-333.4344	3	5	-338.1729
1	6	-328.0539	2	6	-328.2055	3	6	-340.8968
1	7	-331.1925	2	7	-333.9774	3	7	-339.0937
1	8	-331.1218	2	8	-333.5171	3	8	-337.7656
1	9	-333.1306	2	9	-335.5315	3	9	-340.129
1	10	-327.2614	2	10	-328.0284	3	10	-338.7564
1	11	-338.0636	2	11	-339.4807	3	11	-341.2442
1	12	-331.5835	2	12	-335.6497	3	12	-337.6603
1	13	-330.2634	2	13	-333.3606	3	13	-337.8531

Index	Lag	AIC	Index	Lag	AIC	Index	Lag	AIC
4	1	-337.6356	5	1	-335.9952	6	1	-336.2268
4	2	-337.4686	5	2	-336.5502	6	2	-337.2904
4	3	-335.6503	5	3	-334.6152	6	3	-334.8725
4	4	-337.5717	5	4	-336.1568	6	4	-336.6558
4	5	-336.5497	5	5	-335.7691	6	5	-336.4238
4	6	-339.1487	5	6	-338.4865	6	6	-338.3796
4	7	-340.9611	5	7	-340.4571	6	7	-340.4608
4	8	-335.3499	5	8	-334.3672	6	8	-334.6222
4	9	-337.0654	5	9	-336.0952	6	9	-337.6319
4	10	-336.3081	5	10	-335.3722	6	10	-335.7087
4	11	-338.6519	5	11	-337.1396	6	11	-338.1534
4	12	-335.663	5	12	-334.4059	6	12	-334.8127
4	13	-336.4856	5	13	-334.4459	6	13	-335.2489

Index	Lag	AIC	Index	Lag	AIC	Index	Lag	AIC
7	1	-331.4329	8	1	-338.5258	9	1	-340.8725
7	2	-332.6374	8	2	-340.8283	9	2	-342.9302
7	3	-330.0669	8	3	-336.8451	9	3	-338.9037
7	4	-333.415	8	4	-337.9999	9	4	-340.1443
7	5	-331.0479	8	5	-336.8701	9	5	-338.8961

7	6	-333.5343	8	6	-338.9371	9	6	-339.9156
7	7	-335.9789	8	7	-339.0327	9	7	-340.111
7	8	-329.8088	8	8	-336.6994	9	8	-338.8253
7	9	-332.193	8	9	-338.506	9	9	-340.5519
7	10	-330.5314	8	10	-336.9317	9	10	-338.9329
7	11	-332.8134	8	11	-337.004	9	11	-338.9499
7	12	-329.6473	8	12	-336.8695	9	12	-340.0735
7	13	-330.1105	8	13	-336.5539	9	13	-338.6627

Index	Lag	AIC	Index	Lag	AIC			
10	1	-342.625	11	1	-343.4828			
10	2	-344.2918	11	2	-344.2531			
10	3	-339.9875	11	3	-342.4577			
10	4	-341.3353	11	4	-343.4962			
10	5	-339.979	11	5	-342.4616			
10	6	-340.7122	11	6	-343.1587			
10	7	-340.3404	11	7	-342.6803			
10	8	-340.0299	11	8	-343.013			
10	9	-342.493	11	9	-346.2899			
10	10	-340.3458	11	10	-342.918			
10	11	-339.9809	11	11	-342.7529			
10	12	-341.4635	11	12	-344.334			
10	13	-340.1753	11	13	-342.8376			

Table 5.1: AICs and policy lags of models of the United Kingdom

Table 5.2: Indices in table 1

Index (UK)	Data
1	1991:I-2001:II
2	1991:II-2001:III
3	1991:III-2001:IV
4	1991:IV-2002:I
5	1992:I-2002:II
6	1992:II-2002:III
7	1992:III-2002:IV
8	1992:IV-2003:I
9	1993:I-2003:II
10	1993:II-2003:III
11	1993:III-2003:IV

Data	Optimal Policy Lag	Actual	Optimal	a	b	F
2001:II	11	-0.3633	-0.0340	0.9453	-0.00008	-2.66284
2001:III	11	-0.3100	-0.2510	0.9562	-0.00073	-26.33202
2001:IV	11	-0.7967	-0.0676	0.9522	-0.00013	-5.06418
2002:I	7	-0.0333	-0.1476	0.9758	-0.00088	-40.24875
2002:II	7	0.0767	-0.2743	0.9803	-0.00050	-34.30706
2002:III	7	-0.1667	-0.2611	0.9805	-0.00065	-39.24066
2002:IV	7	-0.0033	0.2611	0.9904	-0.00315	-54.51951
2003:I	2	-0.2067	-0.1426	0.9893	0.00153	53.92889
2003:II	2	-0.1367	-0.1884	0.9948	0.00150	59.61648
2003:III	2	-0.0433	-0.1411	0.9990	0.00153	57.09896
2003:IV	9	0.3533	-0.0561	0.9833	-0.00134	-49.77213

Table 5.3: Optimal and actual short interest raises of the United Kingdom ($R = 4000$)

Data	Optimal Policy Lag	Actual	Optimal	a	b	F
2001:II	11	-0.3633	-0.0170	0.9453	-0.00008	-1.33274
2001:III	11	-0.3100	-0.1377	0.9562	-0.00073	-14.44196
2001:IV	11	-0.7967	-0.0339	0.9522	-0.00013	-2.54105
2002:I	7	-0.0333	-0.0897	0.9758	-0.00088	-24.45517
2002:II	7	0.0767	-0.1582	0.9803	-0.00050	-19.78973
2002:III	7	-0.1667	-0.1567	0.9805	-0.00065	-23.55361
2002:IV	7	-0.0033	0.1863	0.9904	-0.00315	-38.90766
2003:I	2	-0.2067	-0.0978	0.9893	0.00153	36.95990
2003:II	2	-0.1367	-0.1346	0.9948	0.00150	42.56660
2003:III	2	-0.0433	-0.0989	0.9990	0.00153	40.02637
2003:IV	9	0.3533	-0.0371	0.9833	-0.00134	0.35333

Table 5.4: Optimal and actual short interest raises of the United Kingdom ($R = 2000$)

Data	Optimal Policy Lag	Actual	Optimal	a	b	F
2001:II	11	-0.3633	-0.0085	0.9453	-0.00008	-0.66670
2001:III	11	-0.3100	-0.0728	0.9562	-0.00073	-7.63846
2001:IV	11	-0.7967	-0.0170	0.9522	-0.00013	-1.27279
2002:I	7	-0.0333	-0.0520	0.9758	-0.00088	-14.16625
2002:II	7	0.0767	-0.0872	0.9803	-0.00050	-10.90822
2002:III	7	-0.1667	-0.0897	0.9805	-0.00065	-13.48194
2002:IV	7	-0.0033	0.1307	0.9904	-0.00315	-27.30437
2003:I	2	-0.2067	-0.0655	0.9893	0.00153	24.76758
2003:II	2	-0.1367	-0.0955	0.9948	0.00150	30.21156
2003:III	2	-0.0433	-0.0684	0.9990	0.00153	27.68588
2003:IV	9	0.3533	-0.0237	0.9833	-0.00134	0.35333

Table 5.5: Optimal and actual short interest raises of the United Kingdom ($R = 1000$)

Data	Optimal Policy Lag	Actual	Optimal	F_1	F_2
2001:II	11	-0.3633	0.1160	-1.0373	1.6626
2001:III	11	-0.3100	-0.0160	2.2850	1.4208
2001:IV	3	-0.7967	0.3971	-14.3442	-1.6803
2002:I	3	-0.0333	0.3122	-20.2490	-1.5837
2002:II	3	0.0767	-0.2309	-18.3543	-1.7524
2002:III	3	-0.1667	0.0384	-18.7268	-1.6642
2002:IV	3	-0.0033	-0.2488	-19.9407	-1.6875
2003:I	3	-0.2067	-0.5326	-19.8521	-1.7919
2003:II	2	-0.1367	0.7050	27.0128	1.6671
2003:III	2	-0.0433	0.1038	27.5902	1.2775
2003:IV	9	0.3533	0.0243	-36.6125	-0.3188

Table 5.6: Optimal and actual short interest raises of the United Kingdom ($R_{11} = 2000,\ R_{22} = 10$)

Data	Optimal Policy Lag	Actual	Optimal	F_1	F_2
2001:II	11	-0.3633	0.0912	0.7277	1.1026
2001:III	11	-0.3100	-0.0074	1.9298	0.8501
2001:IV	3	-0.7967	0.2524	-19.0149	-1.2454
2002:I	3	-0.0333	0.2057	-24.6744	-1.1377
2002:II	3	0.0767	-0.2488	-22.7322	-1.2660
2002:III	3	-0.1667	-0.0256	-23.3301	-1.2084
2002:IV	3	-0.0033	-0.2099	-24.6491	-1.2469
2003:I	3	-0.2067	-0.4007	-24.8366	-1.3510
2003:II	2	-0.1367	0.4395	31.5519	1.1149
2003:III	2	-0.0433	0.0317	32.3514	0.8467
2003:IV	9	0.3533	0.0064	-35.5141	-0.1904

Table 5.7: Optimal and actual short interest raises of the United Kingdom ($R_{11} = 2000,\ R_{22} = 5$)

Data	Optimal Policy Lag	Actual	Optimal	F_1	F_2
2001:II	11	-0.3633	0.0784	2.5343	0.6946
2001:III	11	-0.3100	-0.0023	1.6299	0.4872
2001:IV	3	-0.7967	0.1345	-23.1821	-0.8971
2002:I	3	-0.0333	0.1258	-28.2073	-0.8048
2002:II	3	0.0767	-0.2650	-26.3072	-0.9061
2002:III	3	-0.1667	-0.0757	-27.0812	-0.8619
2002:IV	3	-0.0033	-0.1802	-28.5297	-0.9042
2003:I	3	-0.2067	-0.2934	-29.0278	-0.9924
2003:II	2	-0.1367	0.2453	35.1541	0.7122
2003:III	2	-0.0433	-0.0237	36.0716	0.5169
2003:IV	9	0.3533	-0.0027	-34.9374	-0.1252

Table 5.8: Optimal and actual short interest raises of the United Kingdom ($R_{11} = 2000,\ R_{22} = 2.5$)

Index	Lag	AIC	Overfit	Index	Lag	AIC	Overfit
1	1	314.9525		2	1	314.9922	
1	2	314.8504		2	2	314.9326	
1	3	314.5938		2	3	314.8149	
1	4	314.3304		2	4	314.5622	
1	5	314.2739		2	5	314.2865	
1	6	314.2739	overfitting	2	6	314.2468	overfitting
1	7	314.2625	overfitting	2	7	314.2542	overfitting
1	8	314.1945	overfitting	2	8	314.2502	overfitting
1	9	314.1767	overfitting	2	9	314.1751	overfitting
1	10	314.1634	overfitting	2	10	314.1488	overfitting
1	11	314.1665	overfitting	2	11	314.1388	overfitting
1	12	314.1302	overfitting	2	12	314.1443	overfitting

Index	Lag	AIC	Overfit	Index	Lag	AIC	Overfit
3	1	314.9636		4	1	314.9449	
3	2	314.9573		4	2	314.9197	
3	3	314.8815		4	3	314.8977	
3	4	314.7799		4	4	314.8425	
3	5	314.5172		4	5	314.7254	
3	6	314.2604	overfitting	4	6	314.4756	overfitting
3	7	314.2202	overfitting	4	7	314.2197	overfitting
3	8	314.2285	overfitting	4	8	314.1841	overfitting
3	9	314.218	overfitting	4	9	314.1856	overfitting
3	10	314.1391	overfitting	4	10	314.1708	overfitting
3	11	314.1152	overfitting	4	11	314.0928	overfitting
3	12	314.1054	overfitting	4	12	314.0706	overfitting

Index	Lag	AIC	Overfit	Index	Lag	AIC	Overfit
5	1	314.9708		6	1	314.969	
5	2	314.911		6	2	314.9377	

5	3	314.8672		6	3	314.8606	
5	4	314.8688		6	4	314.8383	
5	5	314.7986		6	5	314.8209	
5	6	314.694	overfitting	6	6	314.7666	overfitting
5	7	314.4446	overfitting	6	7	314.6624	overfitting
5	8	314.1944	overfitting	6	8	314.419	overfitting
5	9	314.1517	overfitting	6	9	314.1632	overfitting
5	10	314.149	overfitting	6	10	314.1198	overfitting
5	11	314.1348	overfitting	6	11	314.1185	overfitting
5	12	314.0585	overfitting	6	12	314.1027	overfitting

Index	Lag	AIC	Overfit	Index	Lag	AIC	Overfit
7	1	314.946		8	1	314.9297	
7	2	314.9321		8	2	314.9235	
7	3	314.878		8	3	314.8813	
7	4	314.8184		8	4	314.8579	
7	5	314.784		8	5	314.7913	
7	6	314.7765	overfitting	8	6	314.7642	overfitting
7	7	314.715	overfitting	8	7	314.7598	overfitting
7	8	314.617	overfitting	8	8	314.7081	overfitting
7	9	314.3693	overfitting	8	9	314.604	overfitting
7	10	314.1154	overfitting	8	10	314.3545	overfitting
7	11	314.0714	overfitting	8	11	314.0999	overfitting
7	12	314.0694	overfitting	8	12	314.0555	overfitting

Table 5.9: AICs and policy lags of models of Japan

Index (Japan)	Data
1	1984:III-2002:IV
2	1984:IV-2003:I
3	1985:I-2003:II
4	1985:II-2003:III
5	1985:III-2003:IV
6	1985:IV-2004:I
7	1986:I-2004:II
8	1986:II-2004:III

Table 5.10: Indices in table 5

Data	Optimal Policy Lag	Actual	Optimal	a	b	F
2002:IV	5	0.1877	0.1243	0.8796	0.0097	3.2451
2003:I	5	0.1164	0.1321	0.8724	0.0095	3.0369
2003:II	5	0.1473	0.1282	0.8629	0.0198	4.5780
2003:III	5	0.1865	0.1347	0.8628	0.0144	3.8441
2003:IV	5	0.1573	0.1235	0.8590	0.0200	4.5176
2004:I	5	0.1300	0.0620	0.8476	0.0359	5.3175
2004:II	5	0.0582	0.1073	0.8554	0.0308	5.2549
2004:III	5	0.0456	0.1260	0.8499	0.0309	5.1633

Table 5.11: Optimal and actual growth rates of the monetary base of Japan ($R = 100$)

Data	Optimal Policy Lag	Actual	Optimal	a	b	F
2002:IV	5	0.1877	0.1022	0.8796	0.0097	1.7387
2003:I	5	0.1164	0.1062	0.8724	0.0095	1.6150
2003:II	5	0.1473	0.1067	0.8629	0.0198	2.6620
2003:III	5	0.1865	0.1089	0.8628	0.0144	2.1309
2003:IV	5	0.1573	0.1039	0.8590	0.0200	2.6208
2004:I	5	0.1300	0.0674	0.8476	0.0359	3.3877
2004:II	5	0.0582	0.0959	0.8554	0.0308	3.2783
2004:III	5	0.0456	0.1074	0.8499	0.0309	3.2069

Table 5.12: Optimal and actual growth rates of the monetary base of Japan ($R = 50$)

Data	Optimal Policy Lag	Actual	Optimal	a	b	F
2002:IV	5	0.1877	0.0932	0.8796	0.0097	0.9043
2003:I	5	0.1164	0.0920	0.8724	0.0095	0.8359
2003:II	5	0.1473	0.0933	0.8629	0.0198	1.4691
2003:III	5	0.1865	0.0939	0.8628	0.0144	1.1339
2003:IV	5	0.1573	0.0917	0.8590	0.0200	1.4434
2004:I	5	0.1300	0.0712	0.8476	0.0359	2.0261
2004:II	5	0.0582	0.0880	0.8554	0.0308	1.9237
2004:III	5	0.0456	0.0947	0.8499	0.0309	1.8730

Table 5.13: Optimal and actual growth rates of the monetary base of Japan ($R = 25$)

Data	Optimal Policy Lag	Actual	Optimal	F_1	F_2
2002:IV	5	0.1877049	0.077643856	2.75836	3.905424
2003:I	5	0.11638155	0.13782743	2.689216	3.062527
2003:II	5	0.14727208	0.10361342	3.219914	2.989277
2003:III	5	0.186525	0.13052853	3.001859	1.862189
2003:IV	4	0.15732312	0.11070252	3.230283	-0.5292883
2004:I	4	0.13002511	0.06911243	3.335563	-0.7308527
2004:II	4	0.05820084	0.09330736	3.193513	-0.958046
2004:III	4	0.04558158	0.11100831	3.123806	-0.96308

Table 5.14: Optimal and actual growth rates of the monetary base of Japan ($R_{22} = 100$)

Data	Optimal Policy Lag	Actual	Optimal	F_1	F_2
2002:IV	5	0.1877049	0.08971726	2.80734	2.67537
2003:I	5	0.11638155	0.13501813	2.619805	2.081702
2003:II	5	0.14727208	0.10561177	3.193653	2.022997
2003:III	5	0.186525	0.12752814	2.901009	1.375564
2003:IV	4	0.15732312	0.12054111	3.352131	0.2043558
2004:I	4	0.13002511	0.074957944	3.443055	-0.02609152
2004:II	4	0.05820084	0.10395511	3.352056	-0.09655746
2004:III	4	0.04558158	0.11925632	3.282215	-0.1071787

Table 5.15: Optimal and actual growth rates of the monetary base of Japan ($R_{22} = 50$)

Data	Optimal Policy Lag	Actual	Optimal	F_1	F_2
2002:IV	5	0.1877049	0.10439074	2.874283	1.19116
2003:I	5	0.11638155	0.13186864	2.539471	1.020998
2003:II	5	0.14727208	0.10775704	3.163565	0.9772133
2003:III	5	0.186525	0.12446506	2.797158	0.8838948
2003:IV	4	0.15732312	0.13079583	3.478565	0.9696484
2004:I	4	0.13002511	0.081050666	3.554323	0.7084306
2004:II	4	0.05820084	0.11518348	3.519045	0.8120633
2004:III	4	0.04558158	0.12792758	3.448801	0.7925574

Table 5.16: Optimal and actual growth rates of the monetary base of Japan ($R_{22} = 25$)

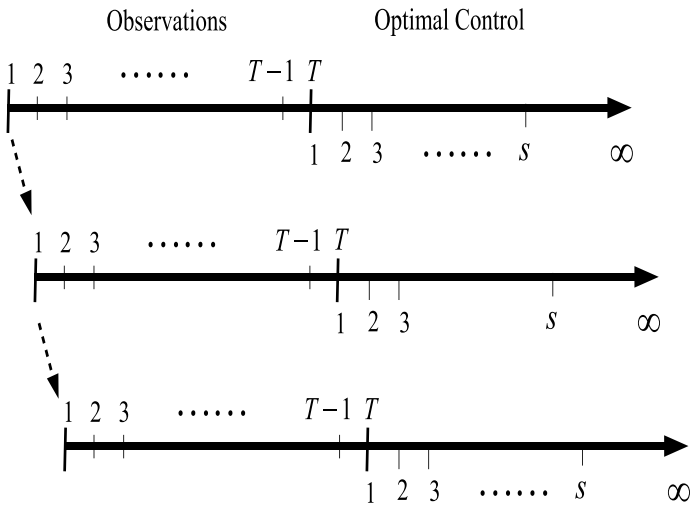

Figure 5.1: Recursive estimations of the dynamic instrument rule

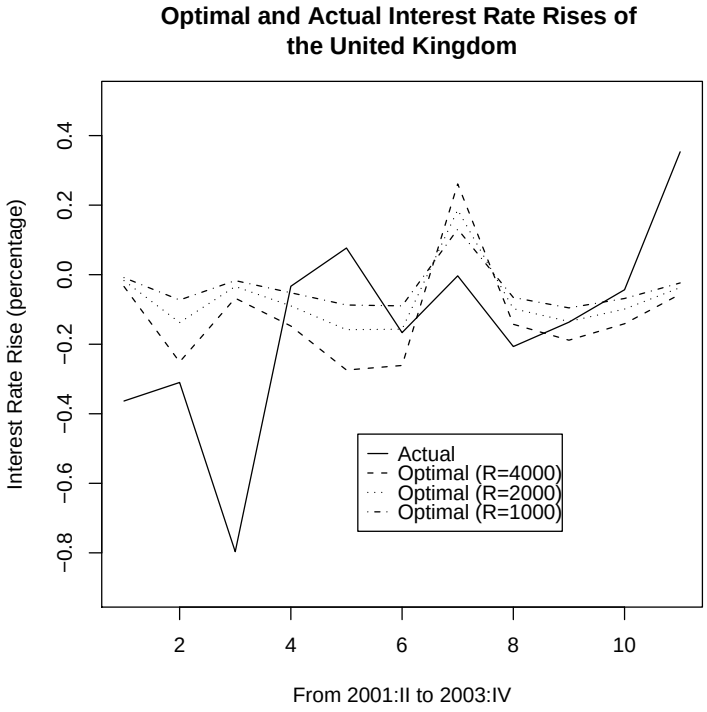

Figure 5.2: Optimal and actual monetary policy of the United Kingdom

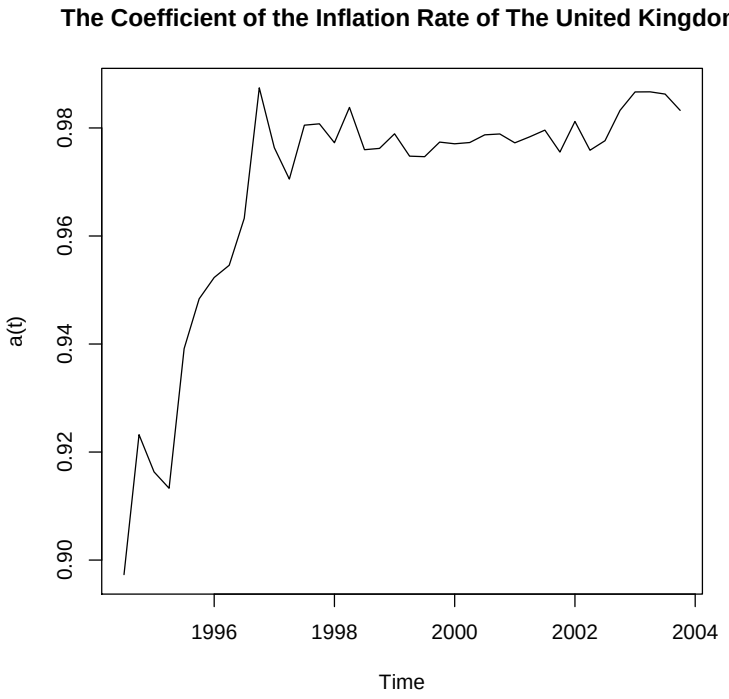

Figure 5.3: The Coefficient of the inflation rate of the United Kingdom

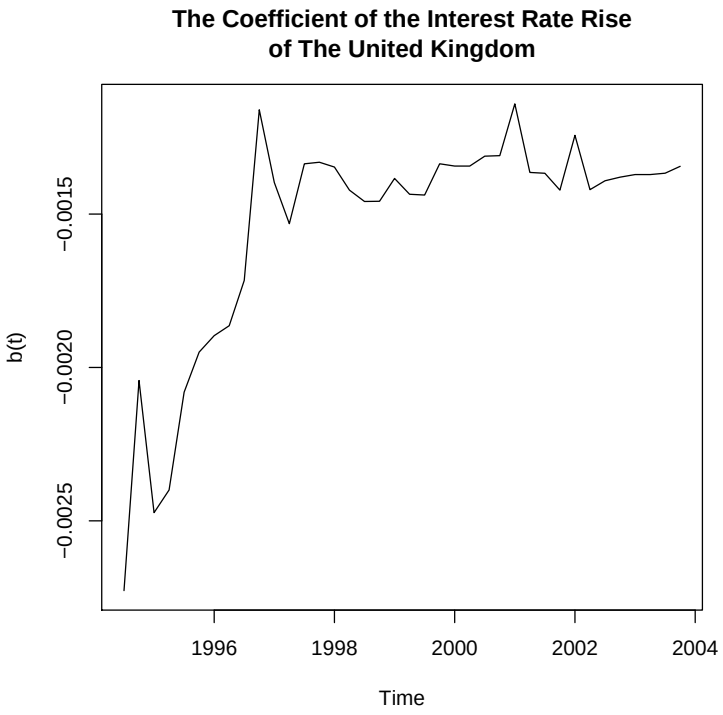

Figure 5.4: The coefficient of the interest rate raise of the United Kingdom

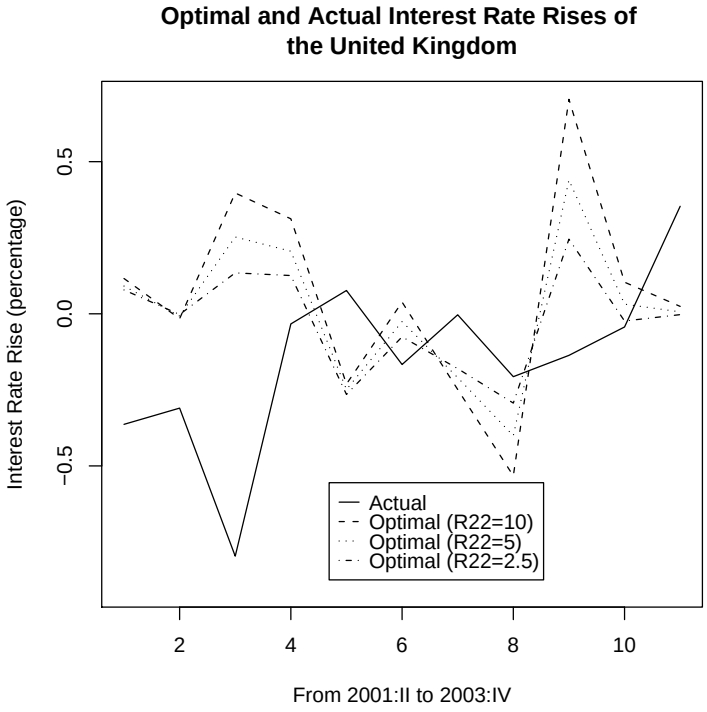

Figure 5.5: Optimal and actual monetary policy of the United Kingdom

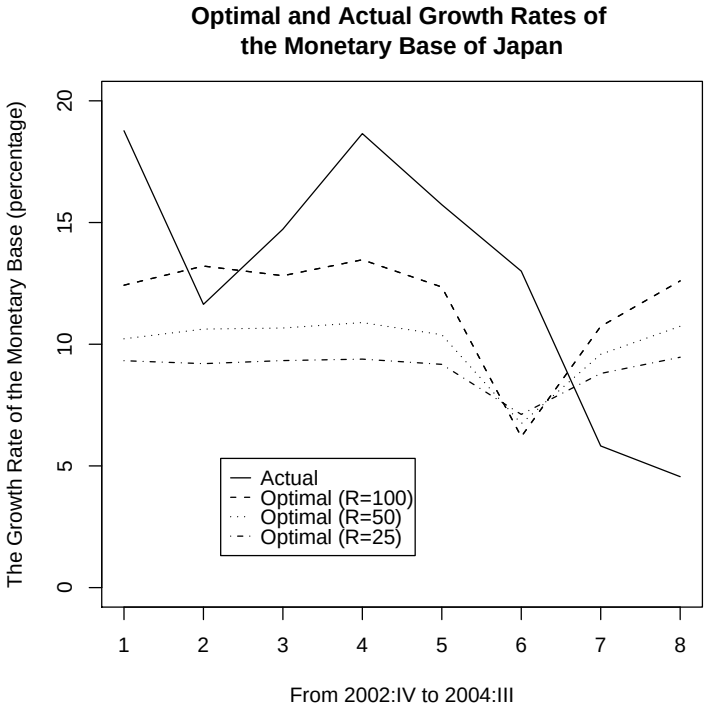

Figure 5.6: Optimal and actual monetary policy in Japan

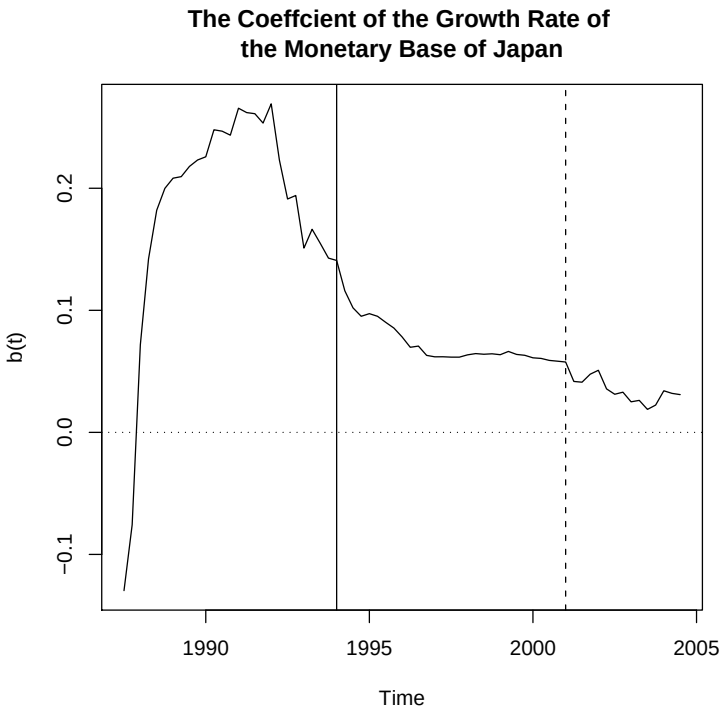

Figure 5.7: The coefficient of the growth rate of monetary base of Japan

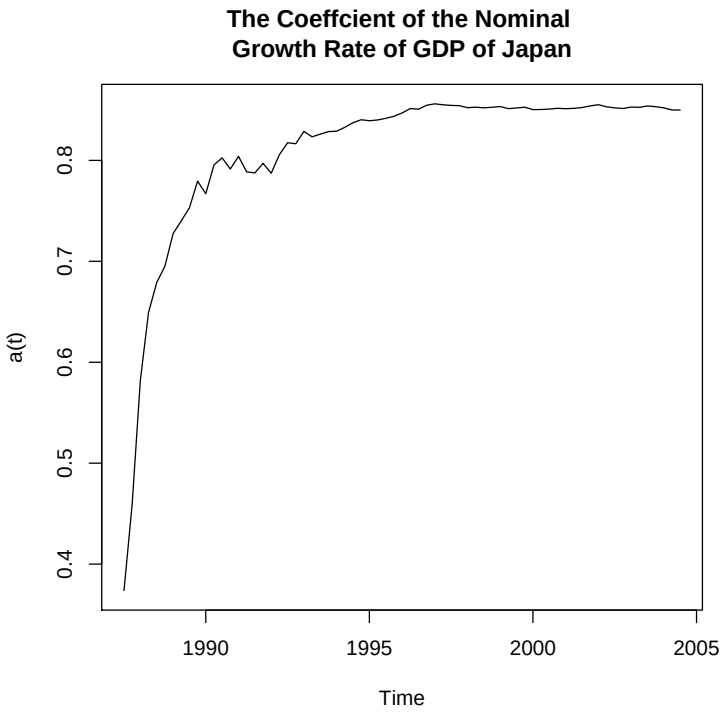

Figure 5.8: The coefficient of the nominal growth rate of Japan

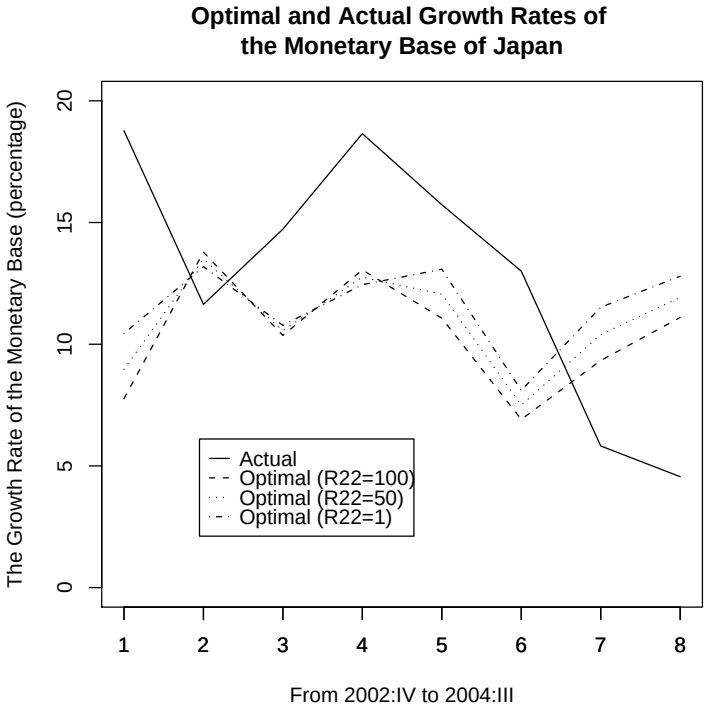

Figure 5.9: Optimal and actual monetary policy in Japan

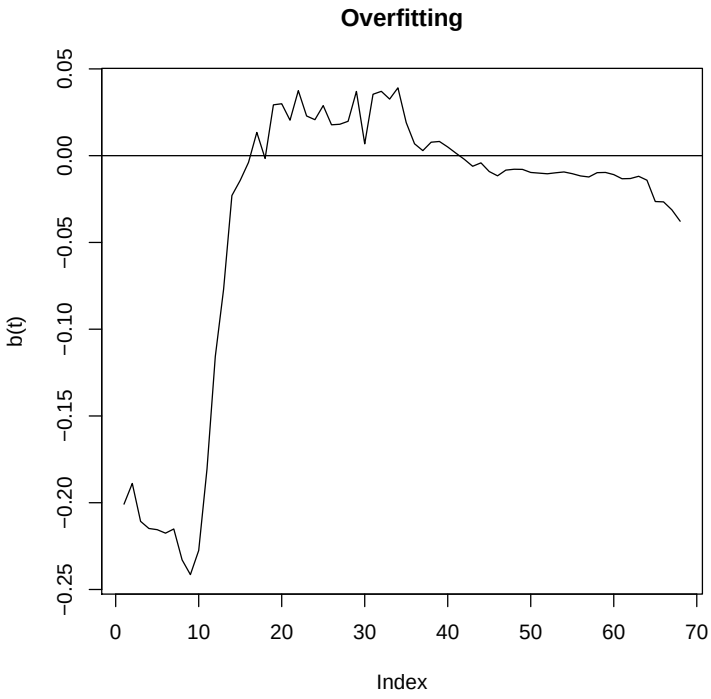

Figure 5.10: An example of overfitting

Chapter 6

Single Factor Model Estimation Based On The Kalman Filter

6.1 Introduction

In financial econometrics, a single factor model is often used to analyze the relation between the market portfolio and a stock price. In financial theory, the modern portfolio theory is proposed by [Mar52]. the Capital Asset Pricing Model (CAPM) is developed by [Sha64] and [Lin65]. The CAPM is the model of the single factor β. The single factor model in financial econometrics is similar with the CAPM, however, it is a purely statistical model. The single factor model is the linear relation between the return of

the market portfolio and the return of a security.

$$\begin{aligned} E[R_i] - R_f &= \beta_i (E[R_m] - R_f) \\ \beta_i &= \frac{Cov[R_i, R_m]}{Var[R_m]}, \end{aligned} \tag{6.1}$$

where R_i is the return of a security i, R_m is the return of the market portfolio, R_f is the return of a risk free asset. β_i is only a systematic risk factor of a security in the CAPM, therefore we concentrate on estimating β_i in this paper.

In financial econometric text books, for example, [CLM97], β_i is estimated by using ordinary least square and maximum likelihood method. In these studies it is often implicitly assumed that β_i is very stable. In a real stock markets, however, β_i may be time varying over time. Althoug there are many studies of discussing about the perfect market assumption of the CAPM, it is interestingly very few reseaches whether β_i is unstable and time varying. In this paper, we assume Bayesian smoothness proirs which is expressed in the form of a difference equation excited by a Gaussian white noise. Under our assumption, β_i changes gradually, and we estimate the time varying $\beta_i(t)$ by using the Kalman filter. The Kalman filter is an algorithm estimate a state vector of a linear system. Because equation (6.1) is a linear system on β_i, the Kalman filter can be used to estimate $\beta_i(t)$ of the CAPM. We adopt our method to the Japanse Stock Markets and discuss the the effectiveness of our method using residual analysis. According to our results, we conclude that our method is very effective in the cases the rapid changes of β_i are not happen. It is a significant finding of us, although we

use only "linear" frameworks (the linear relations of the CAMP, smoothness priors, and the Kalman filter). Because the nonlinearity of the financial markets is often discussed recently. We find the residuals of equation (6.1) are Gaussian in the most cases of us (even when the rapid changes of β).

6.2 Theoretical Background

The single factor model of security i can be shown by

$$Z_i(t) = \beta_i(t) Z_m(t) + \alpha_i(t) \tag{6.2}$$

where $Z_i(t)$ is $E[R_i - R_f]$, $Z_m(t)$ is $E[R_m - R_f]$ and $\alpha_i(t)$ is the intercept of security i. In the theory of the CAPM, $\alpha_i(t)$ is zero and $\beta_i(t)$ is a only risk factor of a security. In our empirical analysis, we suppose the $\alpha(t)$ is zero, and we concentrate on estimating $\beta_i(t)$. We assume β_i changes gradually over time, therefore, the shift of β_i is locally smooth. Our assumption is widely known as smoothness priors of Bayesian procedure [1]. We define the stochastic difference equation of $\beta_i(t)$ as follows,
The smoothiness prior of first order is

$$\nu_{i,1}(t) = \beta_i(t) - \beta_i(t-1), \tag{6.3}$$

where $\nu_{i,1}(t) \sim N(0, \tau_i)$. We suppose τ_i is a constant of a security i. The stochastic difference equation (6.3) is considered as the Gaussian smoothnss priors the time series varying coefficients in autoregressive model in [Kit83].

[1] See [Shi73] and [Kit83]

Equation (6.2), equation (6.3) can be rewritten to the state space representation.

The smoothness prior of first order is

$$\beta_i(t) = \beta_i(t-1) + \nu_{i,1}(t), \tag{6.4}$$

$$Z_i(t) = Z_m(t)\beta_i(t) + \epsilon_i(t), \tag{6.5}$$

where $Z_i(t)$ is $E[R_i - R_f]$, $Z_m(t)$ is $E[R_m - R_f]$, and $\epsilon_i(t)$ is a residual. We suppose $\epsilon_i(t) \sim N(0, \sigma_i)$, and σ_i is a constant of a security i. Equation (6.4) is a system equation and equation (6.5) is a measurement equation.

We use the Kalman filter[2] to estimate $\beta_i(t)$. Kalman filter which was proposed by Kalman (1960) is an algorithm for solving the linear filtering problem based on the state space reperesentaion. The linear filtering problem in the Kalman filer is formulated as state estimation. The $\beta_i(t)$ of the CAPM can be considered as a state in the financial market. Therefore we can use the state space representation described above to estimate $\beta_i(t)$ by using the Kalman filter. The Kalman filter consists of three components, (1) prediction, (2) filtering, and (3) smoothing. In this paper prediction and filtering are used. We denote $\boldsymbol{x}(t|t-1)$ as a predictor of $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t|t)$ as a filter of $\boldsymbol{x}(t)$. The algorithm of the Kalman filter is shown as follows.

Prediction algorithm:

$$\begin{aligned} \beta_i(t|t-1) &= \beta_i(t-1|t-1), \\ V(t|t-1) &= V(t-1|t-1) + \tau_i^2, \end{aligned} \tag{6.6}$$

[2]See [KG96].

Filtering algorithm:

$$\begin{aligned} K(t) =& V(t|t-1)Z_m(t)\{Z_m(t)V(t|t-1)Z_m(t)+\sigma_i^2\}^{-1}, \\ \beta_i(t|t) =& \beta_i(t|t-1)+K(t)\{Z_i(t)-Z_m(t)\beta_i(t|t-1)\}, \\ V(t|t) =& \{1-K(t)Z_m(t)\}V(t|t-1), \end{aligned} \tag{6.7}$$

where $V(t|t-1) = E[\{\beta_i(t|t) - \beta_i(t|t-1)\}^2]$. We suppose $\beta(0|0)$ is zero and $V(0|0)$ is a arbitrary huge number.

In our method, it is necessary to choose the best fit of β_i in the formulation described above. The state space representation and the Kalman filter yields an efficient algorithm for computing the likelihood of a time series model. For security i, the joint distribution of $\{Z_i(1), \cdots, Z_i(T)\}$ is

$$f(Z_i(1), \cdots, Z_i(T)|\boldsymbol{\theta}_i) = \prod_{t=1}^{T} f(Z_i(t)|Z_i(1), \cdots, Z_i(t); \boldsymbol{\theta}_i), \tag{6.8}$$

where $\boldsymbol{\theta}_i = (\sigma_i, \tau_i)$ and $f(Z_i(t)|Z_i(1), \cdots, Z_i(t); \boldsymbol{\theta}_i)$ is the conditional distribution defined by

$$\begin{aligned} & f(Z_i(t)|Z_i(1), \cdots, Z_i(t); \boldsymbol{\theta}_i) \\ =& \frac{1}{\sqrt{2\pi v^2(t)}} \exp\Big[-\frac{1}{2v^2(t)}(Z_i(t) - Z_m(t)\beta_i(t|t-1))^2\Big] \end{aligned} \tag{6.9}$$

Thus, we use the maximum log likelihood method to estimate the goodness

of the fit of a model. The log likelihood for $\boldsymbol{\theta}_i$ is obtain by

$$l(\boldsymbol{\theta}_i) = -\frac{1}{2}\Big\{T\log(2\pi) + \sum_{t=1}^{T}\log v^2(t) + \sum_{t=1}^{T}\frac{1}{v^2(t)}(Z_i(t) - Z_m(t)\beta_i(t|t-1))^2\Big\}, \tag{6.10}$$

where $v^2(t) = Z_m(t)V(t|t-1)Z_m(t) + \sigma_i^2$. The maximum likelihood estimate of parameter, $\hat{\boldsymbol{\theta}}_i$ is obtained by maximizing $l(\boldsymbol{\theta}_i)$ with respect to those parameters. In our program, we use the optimization function *nlm* in R, which use a Newton-type algorithm.

6.3 The Application to the Japanese Stock Markets

6.3.1 The Data of the Japanese Stock Markets

We use the monthly closed price data of securities in the first section of Tokyo Stock Exchange (TSE) from January, 1998 to December, 2003. The TSE domestic stock market is divided into the first and second sections. The first section is the market for stocks of relatively large companies. We choose 2 securities from 33 industry sectors of the first section of TSE (Toushou 33 gyoushu), and use the monthly closed prices of them. We adopt TOPIX which include all first section listed shares in TSE as a proxy of the market portfolio. TOPIX provides a comprehensive measure of stock price changes. We choose data period from 1998 to 2003 (6 years), and the data size is 72

Data (monthly closed price)	Periods
25 companies in the first sector of TSE	1998/01 - 2003/12
TOPIX	1998/01 - 2003/12
Libor Yen (3 month)	1998/01 - 2003/12

Table 6.1: The List of Data

(the size of monthly returns is 71) [3]. On calculating the returns of securities, we ignore the effects of dividends, because TOPIX does not include the effects of divideneds. Moreover, we don't adjust stock prices to inflation rates, because the inflation rates of recent Japan is relatively small. The inflation rates from 1998 to 2003 are from -0.8 to -1.6 based on the enterprise price index. At the choosing the securities, the liquidity and the credit risks of them are unconsidered. We adopt LIBOR Yen 3 month as a proxy of short term risk free rate, bacause LIBOR Yen 3 month is relatively stable rather than call rate 1 month in most cases. The list of data is shown in table 6.1.

The monthly returns of securities is defined by

$$R_i = \log \frac{P_t}{P_{t-1}}, \tag{6.11}$$

where P_t is a monthly closed price of a security. The monthly returns of TOPIX are caluculated in the same way. We make $Z_i(t) = R_i(t) - R_f(t)$ from equation (6.11) and LIBOR 3 month in the same period.

In our program, the log likelihood function [equation (6.10)] is maximized by optimization function nlm in R by using data sets described above, and

[3]Single foctor model is commonly estimated using 5 years of monthly data [pp. 184, Campbell, Lo, and MacKinlay (1996)].

the program returns the estimated $\beta_i(t)$, residuals of equation (6.5), τ_i, σ_i, AIC [is equal to the negative log likelihood in our case], and *nlm* return code [2 and 3 mean optimization is successful, and 4 means optimization is failed[4]]. The residual analysis is made based on the results of our program. We made the diagnostic checking for normality and serial correlation as residual analysis.

6.3.2 The Results of our Estimation

We show the the results of our estimation in Figure 1-25. We show the actual returns and estimated returns of securities and the estimated $\beta_i(t)$ in the upper panels of figure 1-25. The results of estimated values are shown in table 6.2. In table 6.2, "code" is a return code from *nlm*. It shows τ_i and σ_i is very small in successeful cases. Therefore, we conclude $\beta_i(T)$ is a very good proxy of β in successful cases. However, in some cases, $\beta_i(t)$ is changing rapidly and the optimizations of *nlm* are failed.

6.3.3 Residual Analysis

We show the results of residual analysis. The assumption of Kalman filter the disturbance of $\nu_i(t)$ and $\epsilon_i(t)$ are normally distributed and serially independent with constant variance. We made the diagnostic checking for normality and serial correlation as follows.

1. Quantile-Quantile plot

[4]We consider optimization is failed on the 500 loops of calculation of *nlm*. In most susseccful cases, the loops of calculation of *nlm* are less than 200.

NAME(No.)	Code	AIC	τ	σ
NTT	0	-149.3359	0.07534621	0.05882475
SONY	0	-64.40624	0.1435226	0.03899718
Tokyo Electric Power Co.	0	-170.9791	0.0643951	0.06440982
Nishimatsu	0	-119.4261	0.09381163	0.004122669
Ito-Yokado	0	-105.2728	0.1038671	0.03153384
Toyota	0	-152.0062	0.0730043	0.02377539
Takeda Pharmaceutical Co. 4502	0	-158.2433	0.0709385	0.02374597
Panasonic 6752	0	-142.9274	0.07851107	0.02417267
Sharp 6753	0	-119.2637	0.09456484	0.02551953
Denso 6902	0	-175.7362	0.06173184	0.02517676
Nissan 7201	0	-90.45424	0.1159477	0.03341797
Cannon 7751	0	-146.1517	0.07626406	0.02624219
Mitsubishi Co. 8058	0	-142.2841	0.07883462	0.02532104
JR East 9020	0	-170.8066	0.06367031	0.02346094
JR Tokai 9022	0	-201.2063	0.05186767	0.02268473
KDDI 9433	0	-72.4038	0.1305883	0.03283984
JT 2914	0	-140.844	0.08041787	0.02553174
Kao 4452	0	-180.9324	0.05882841	0.02141521
NEC 6701	0	-115.7291	0.09526309	0.02877051
Kyocera 6971	0	-89.49805	0.1159477	0.03341797
Nintendo 7974	0	-108.1091	0.1029107	0.03000534
Kanebo 3102	0	-93.36037	0.1159477	0.03341797

Table 6.2: Estimated Values

Name	Residuals		Coefficients	
	Bowman-Shenton (p value)	Ljung-Box (p value)	Bowman-Shenton (p value)	Ljung-Box (p value)
NTT	0.8300	0.0065 ***	0.0000 ***	0.1588
SONY	0.4734	0.0951	0.0016 **	0.6258
Tokyo Elec. Power Co.	0.4734	0.0951	0.0016 **	0.6258
Nishimatsu	0.4671	0.5509	0.0000 ***	0.2269
Ito-yokado	0.0000 ***	0.4553	0.0000 ***	0.9637
Toyota	0.0000 ***	0.0027 ***	0.0000 ***	0.1711
Takeda Pharm. Co.	0.9370	0.1464	0.0000 ***	0.3193
Panasonic	0.8450	0.5210	0.0000 ***	0.6653
Sharp	0.2481	0.4699	0.0000 ***	0.4783
Denso	0.7662	0.3247	0.0000 ***	0.0344 **
Nissan	0.0855	0.1385	0.0000 ***	0.3584
Cannon	0.0583	0.3719	0.0000 ***	0.5045
Mitsubishi Co.	0.0035 ***	0.3101	0.0000 ***	0.0772
JR East	0.7622	0.9843	0.0000 ***	0.5211
JR Tokai	0.1577	0.5774	0.0000 ***	0.8257
KDDI	0.1739	0.5256	0.0000 ***	0.0071 ***
JT	0.1515	0.6817	0.0000 ***	0.9975
Kao	0.9811	0.6888	0.0000 ***	0.0871
NEC	0.8509	0.5885	0.0000 ***	0.6851
Kyocera	0.0000 ***	0.6516	0.0000 ***	0.2925
Nintendo	0.8155	0.5098	0.0000 ***	0.2006
Kanebo	0.0000 ***	0.7953	0.0000 ***	0.7606

Table 6.3: Residual Analysis

2. Bowman-Shenton (Jarque-Bera) test: Tests the null hypothesis of normality

3. Ljung-Box test: Tests the null hypothesis of independence

The results of Bowman-Shenton test and Ljung-Box test are shown in table 6.3. The results of quantile-quantile plot are shown in lower panels of figure 1-25 [The left sides of lower panels].

6.4 Conclusion and Discussion

We conclude as follws.

1. The last estimate of $\beta_i(t)$ is a good proxy of β in successful cases.

2. the results of Bowman-Shenton test and Ljung-Box test on residuals are relatively good rather than the results of same tests on coefficients.

We think that we can improve our method of β_i estimation changing from the Kalman filter to non-Gaussinan filters, for example, the Monta Carlo filter proposed by Kitagawa (1996).

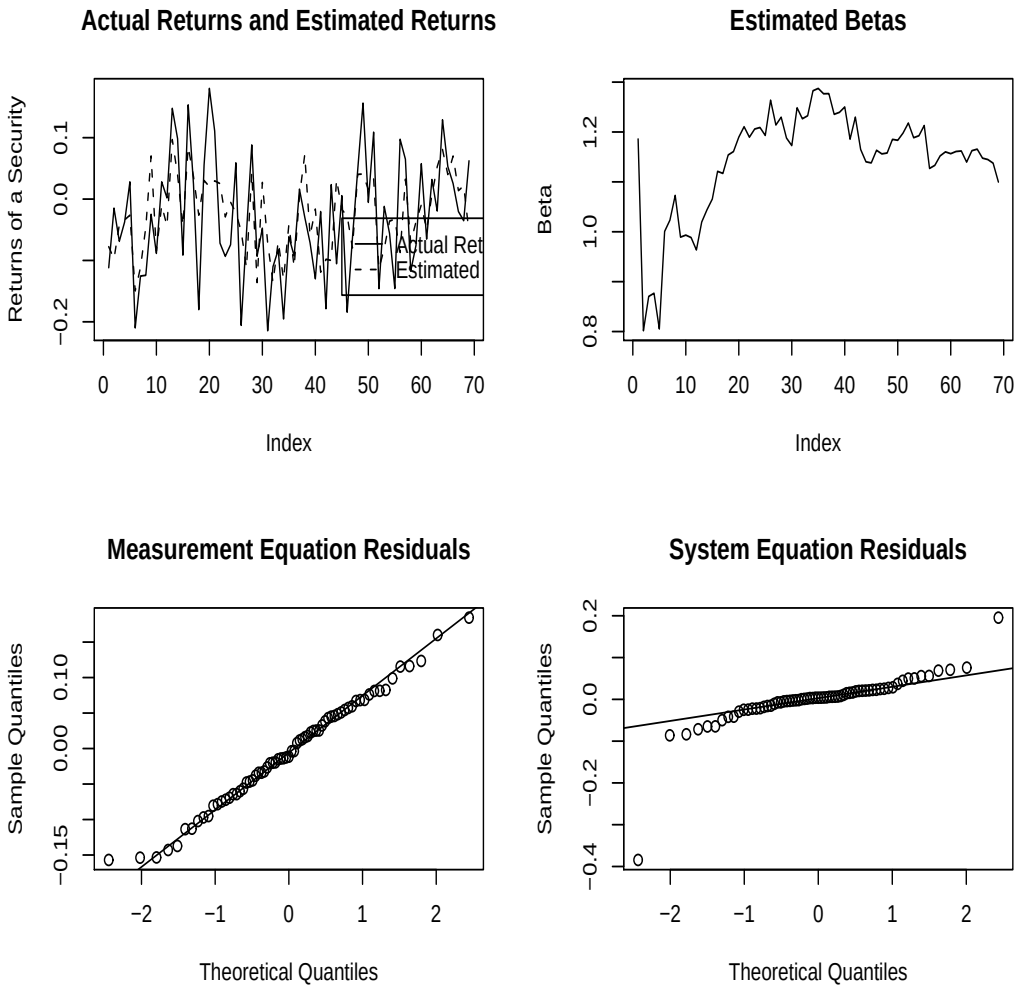

Figure 6.1: NTT

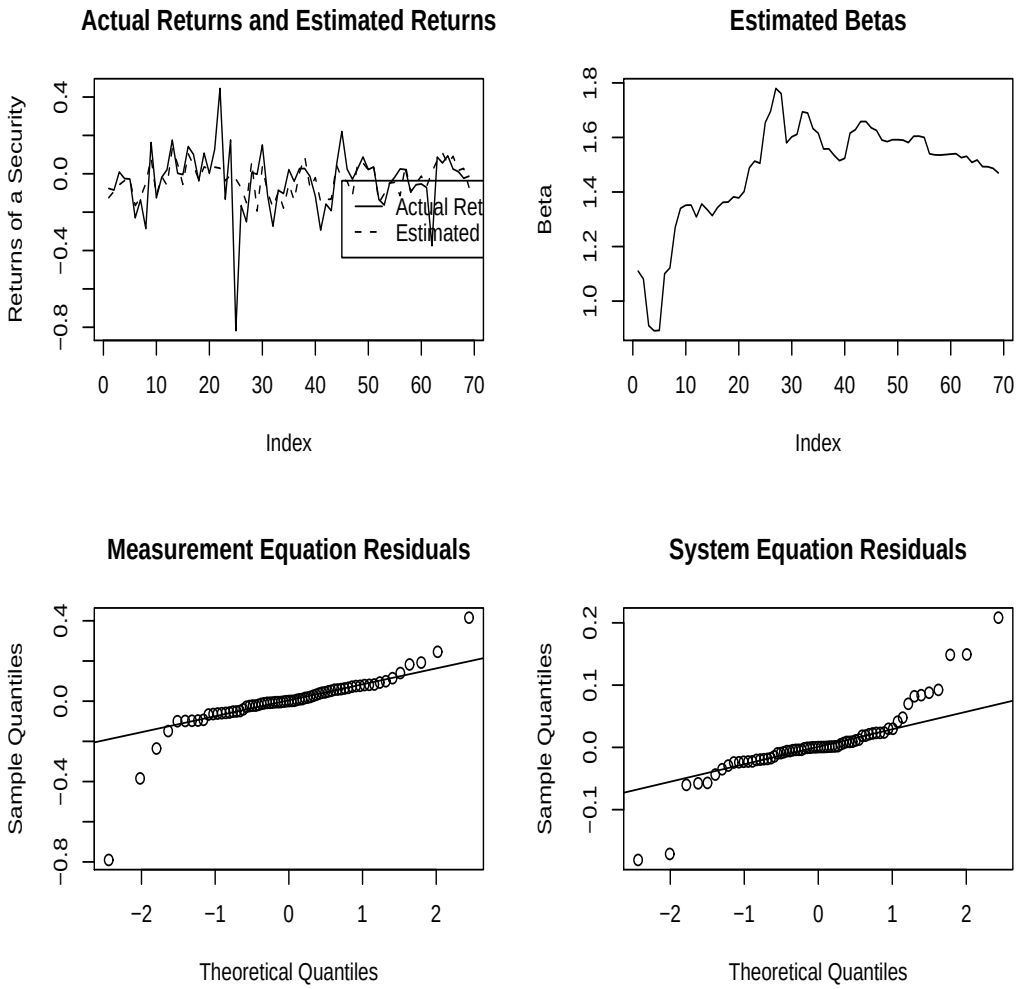

Figure 6.2: SONY

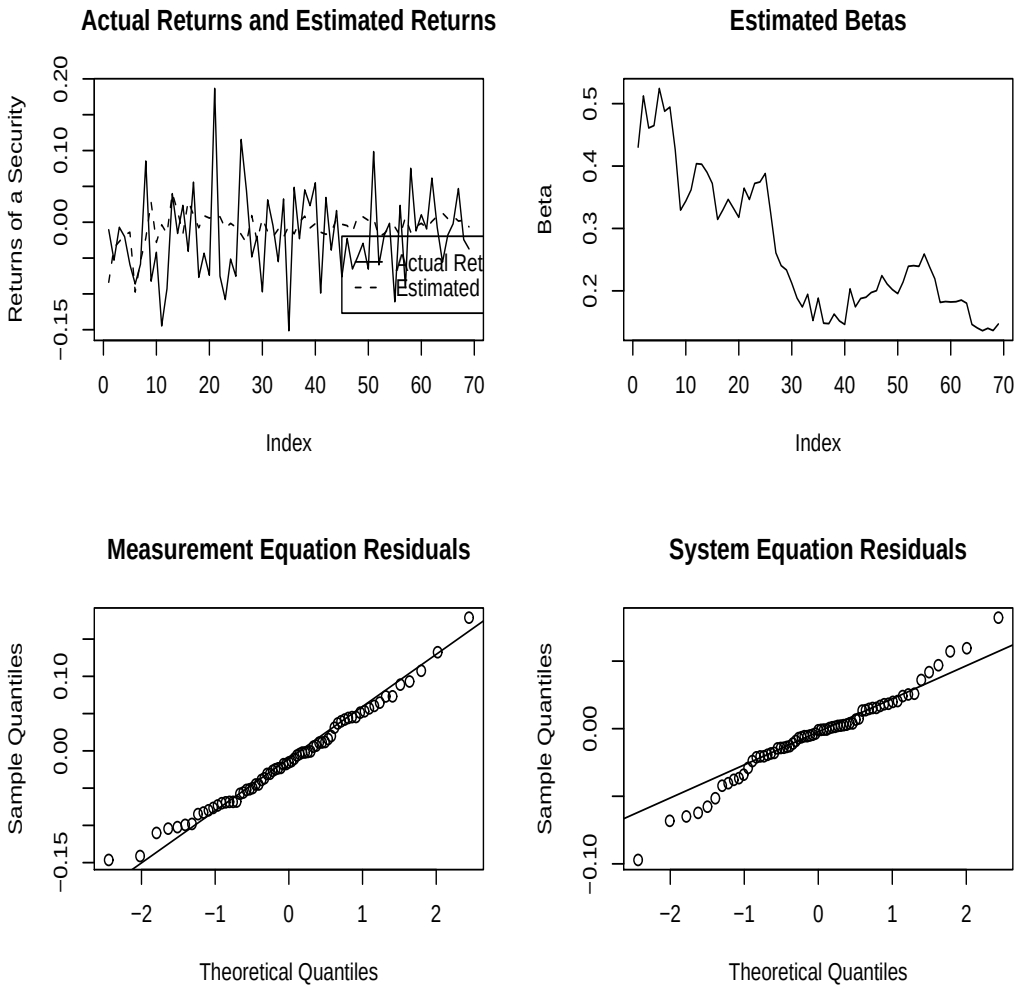

Figure 6.3: Tokyo Electric Power Co.

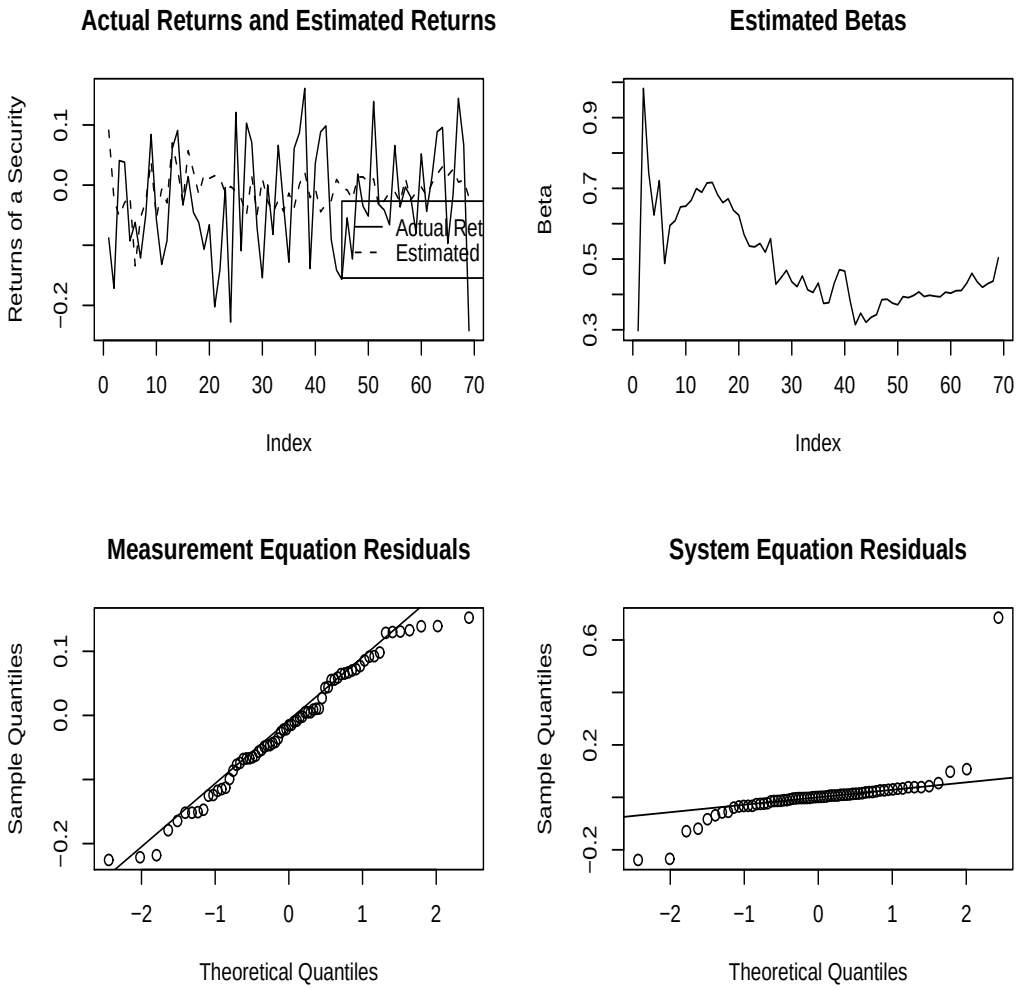

Figure 6.4: Nishimatsu

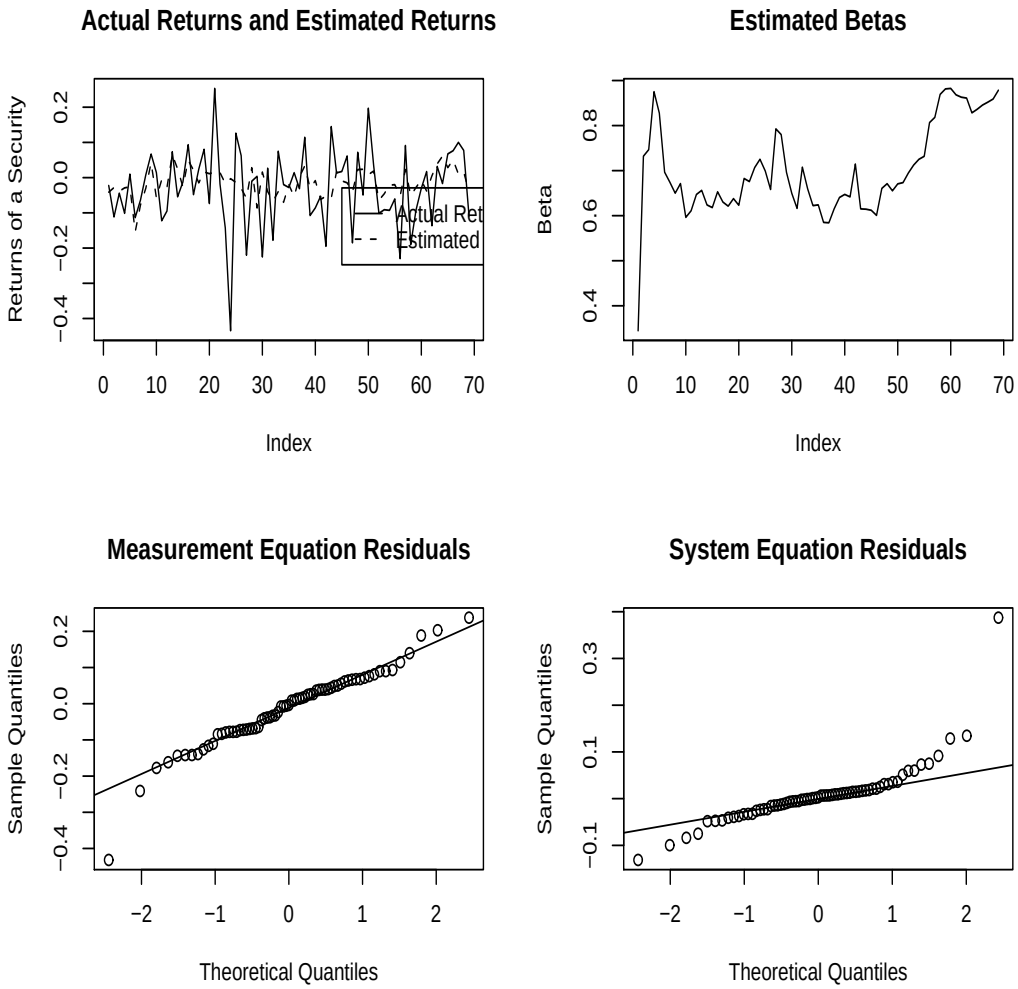

Figure 6.5: Ito-Yokado

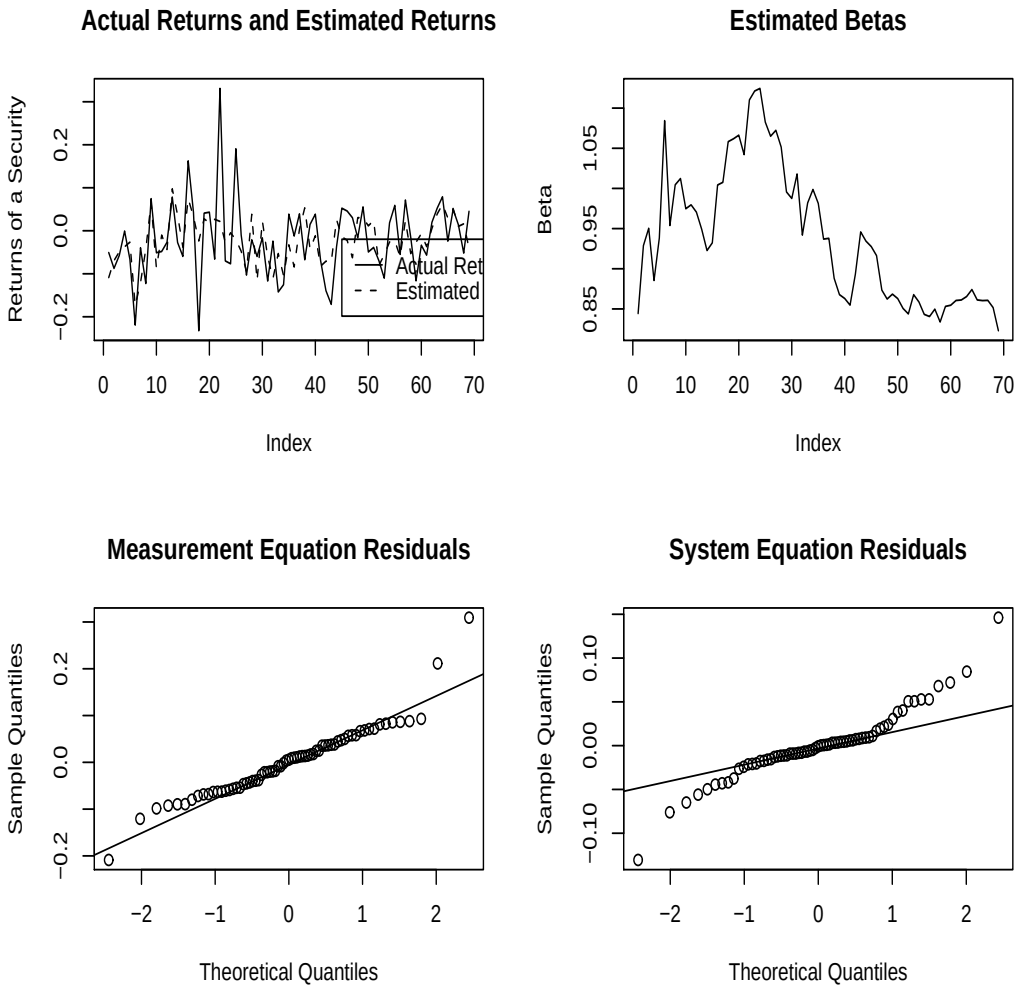

Figure 6.6: Toyota

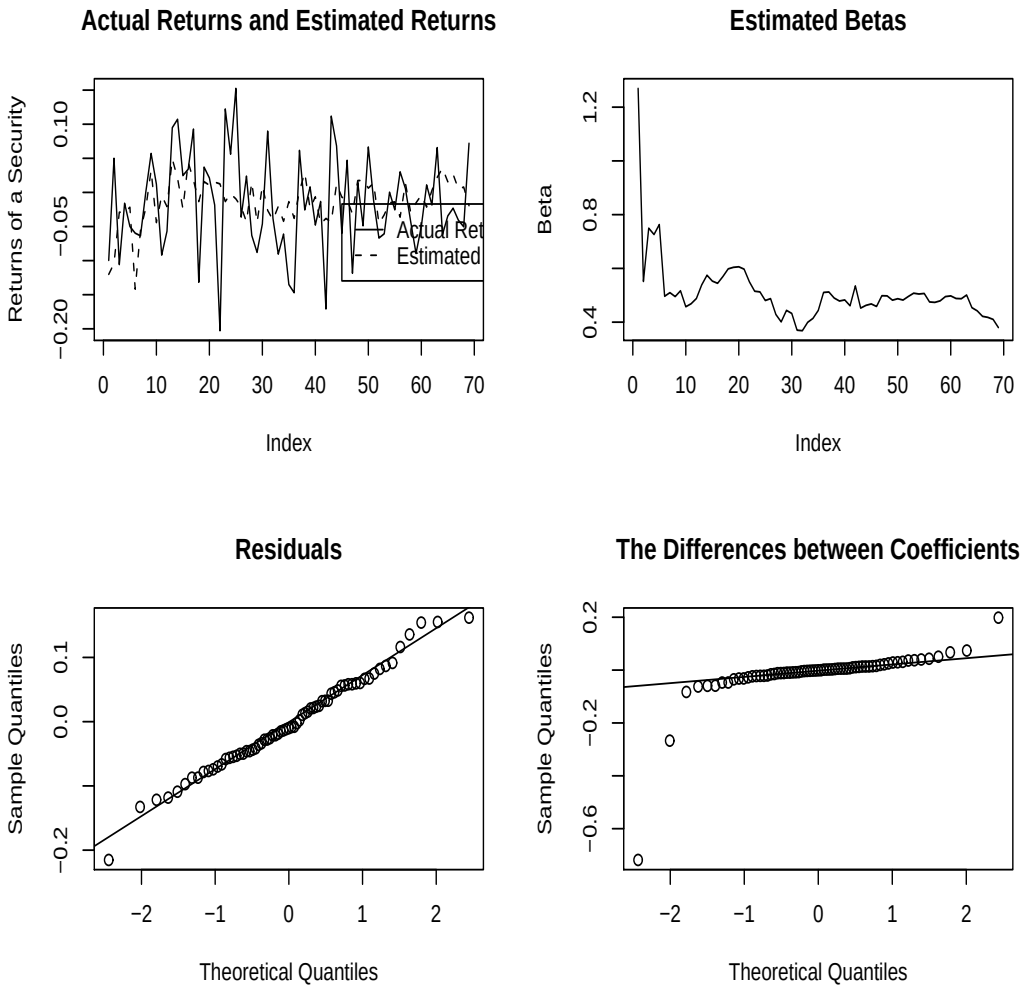

Figure 6.7: Takeda Pharmaceutical Co.

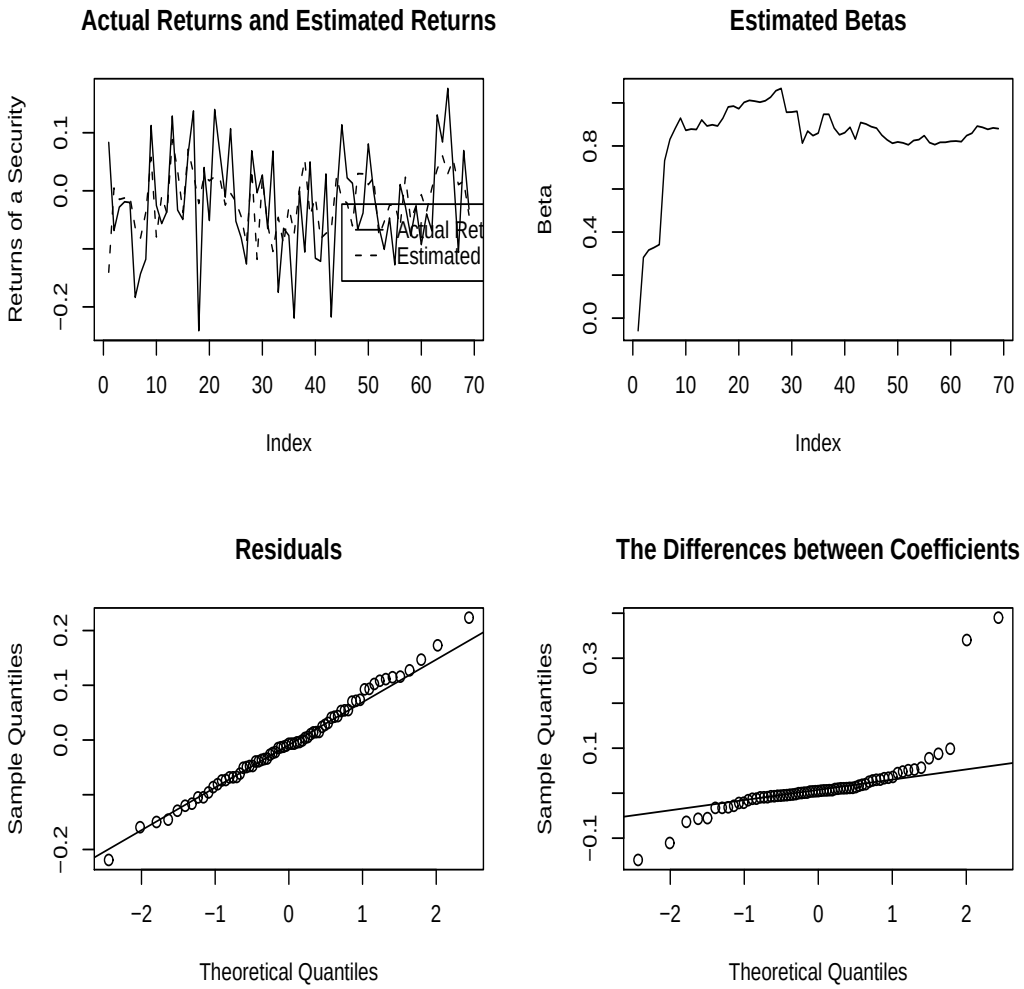

Figure 6.8: Panasonic

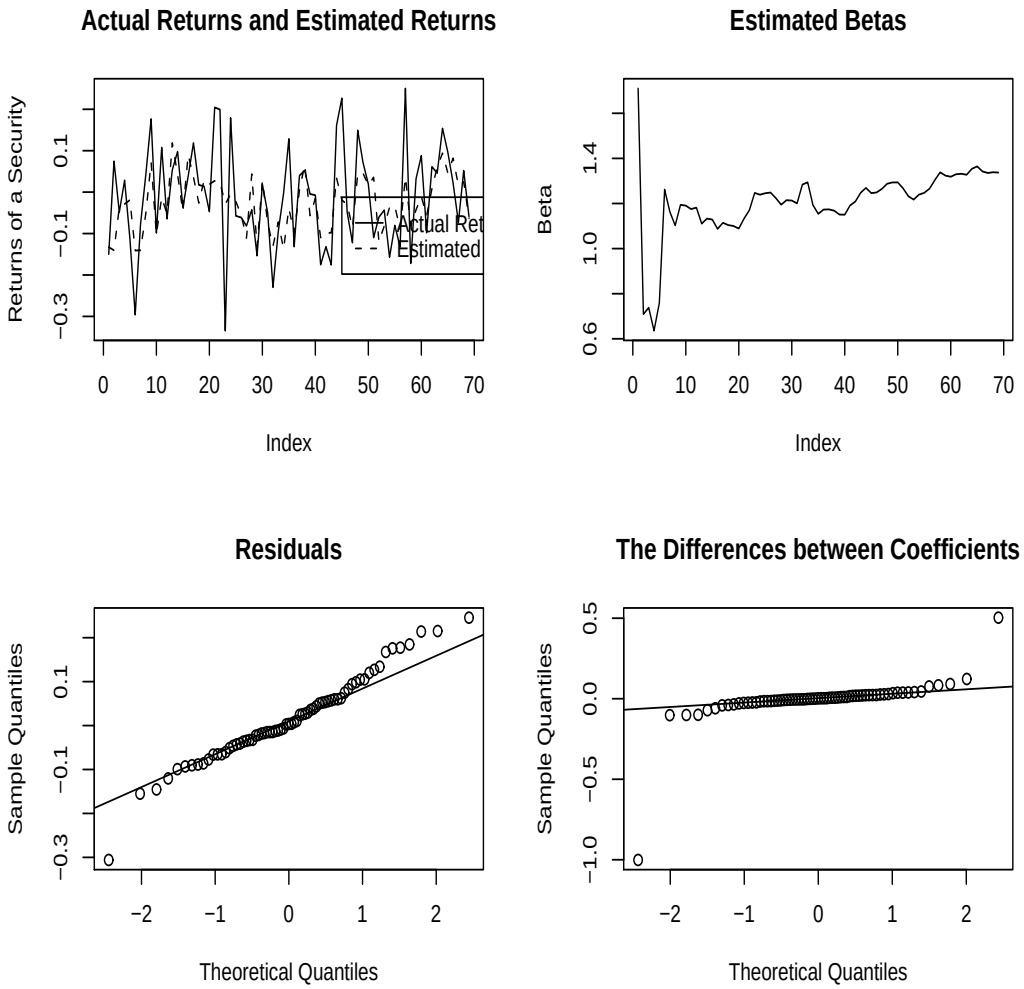

Figure 6.9: Sharp

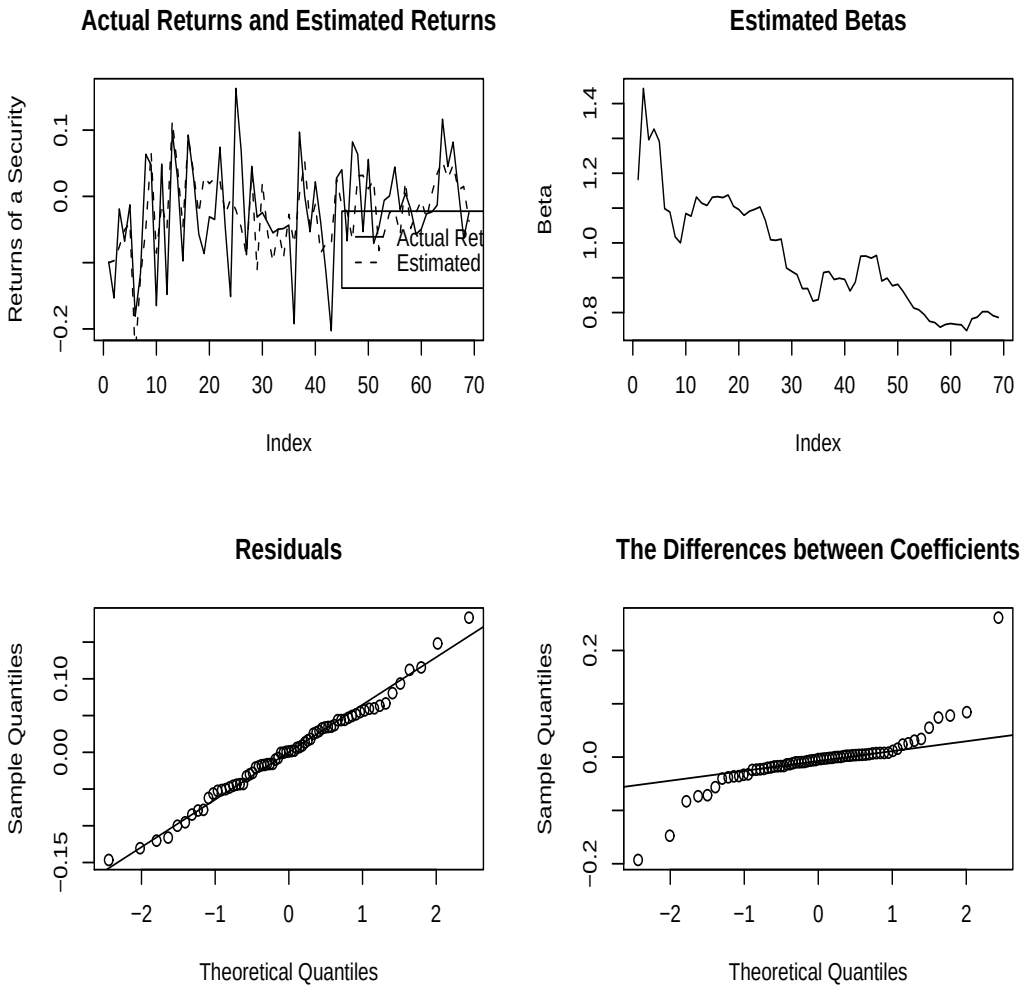

Figure 6.10: Denso

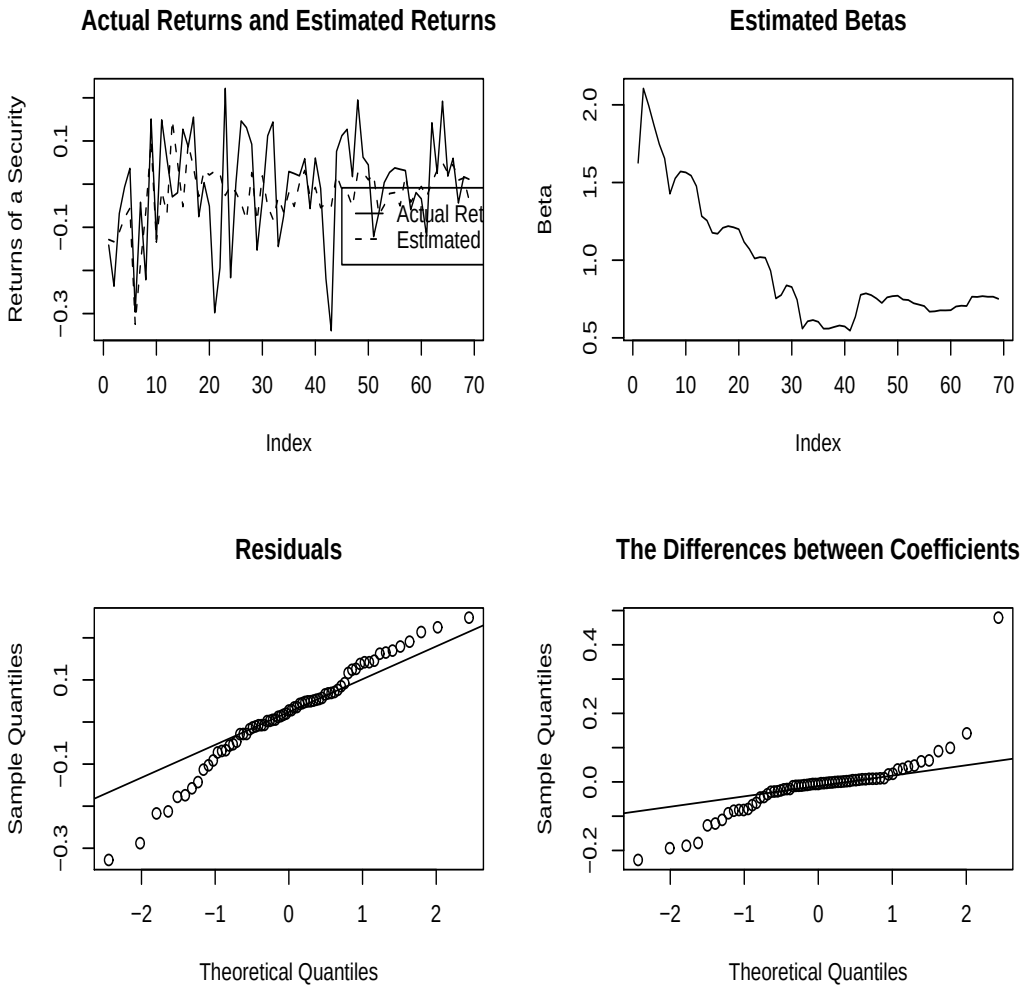

Figure 6.11: Nissan

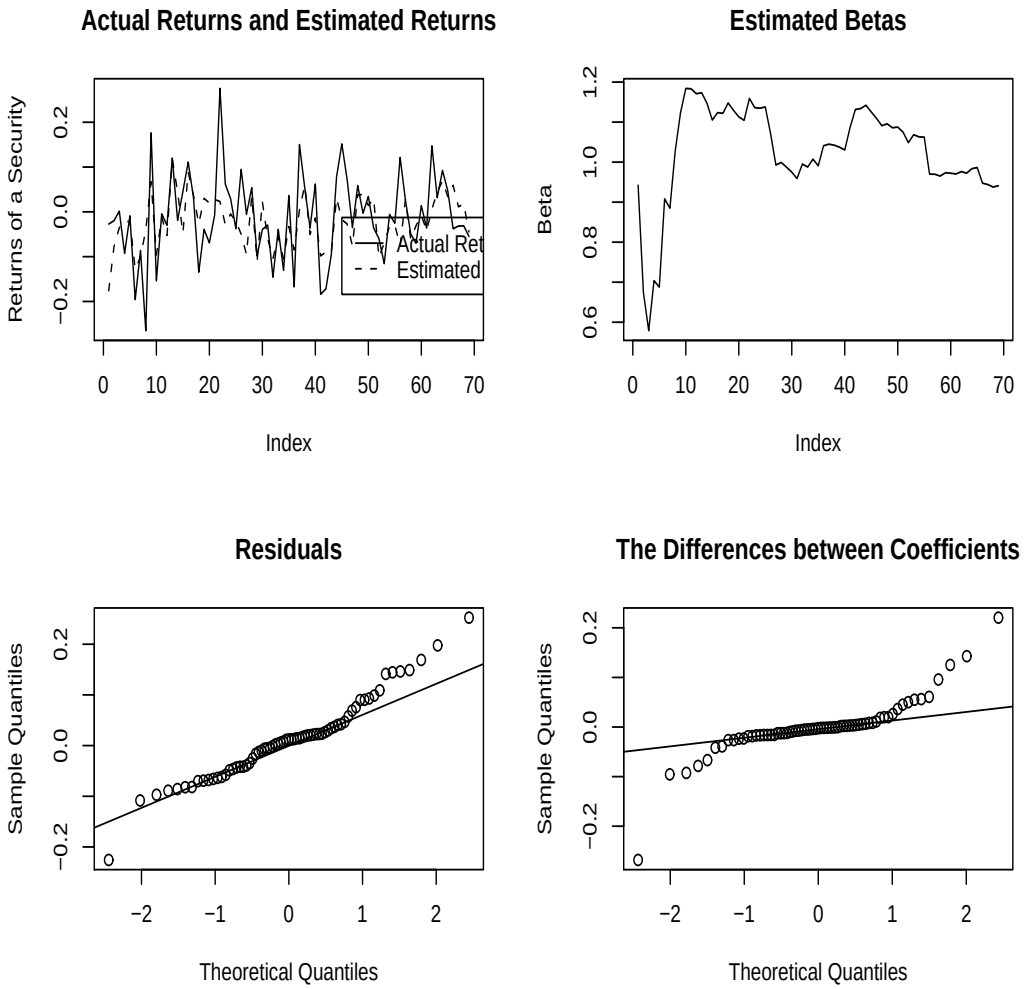

Figure 6.12: Cannon

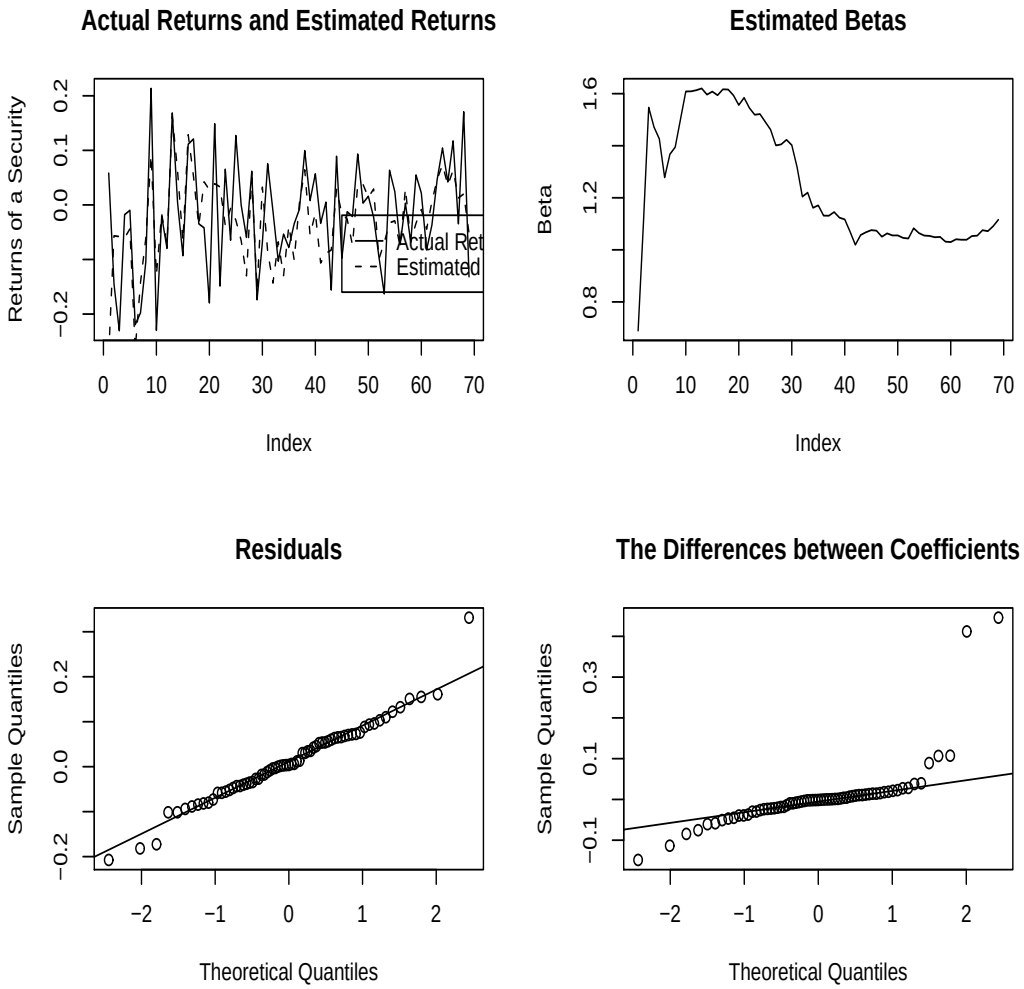

Figure 6.13: Mitsubishi Co.

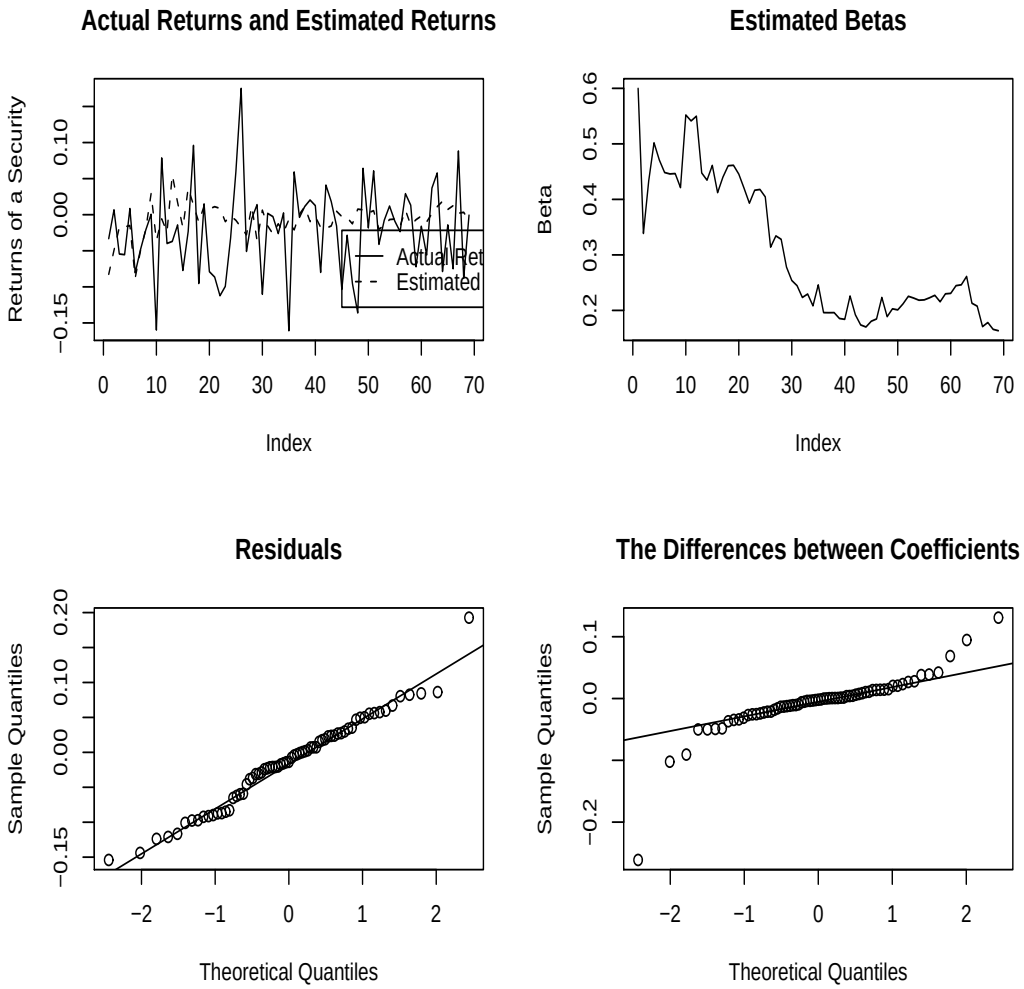

Figure 6.14: JR East

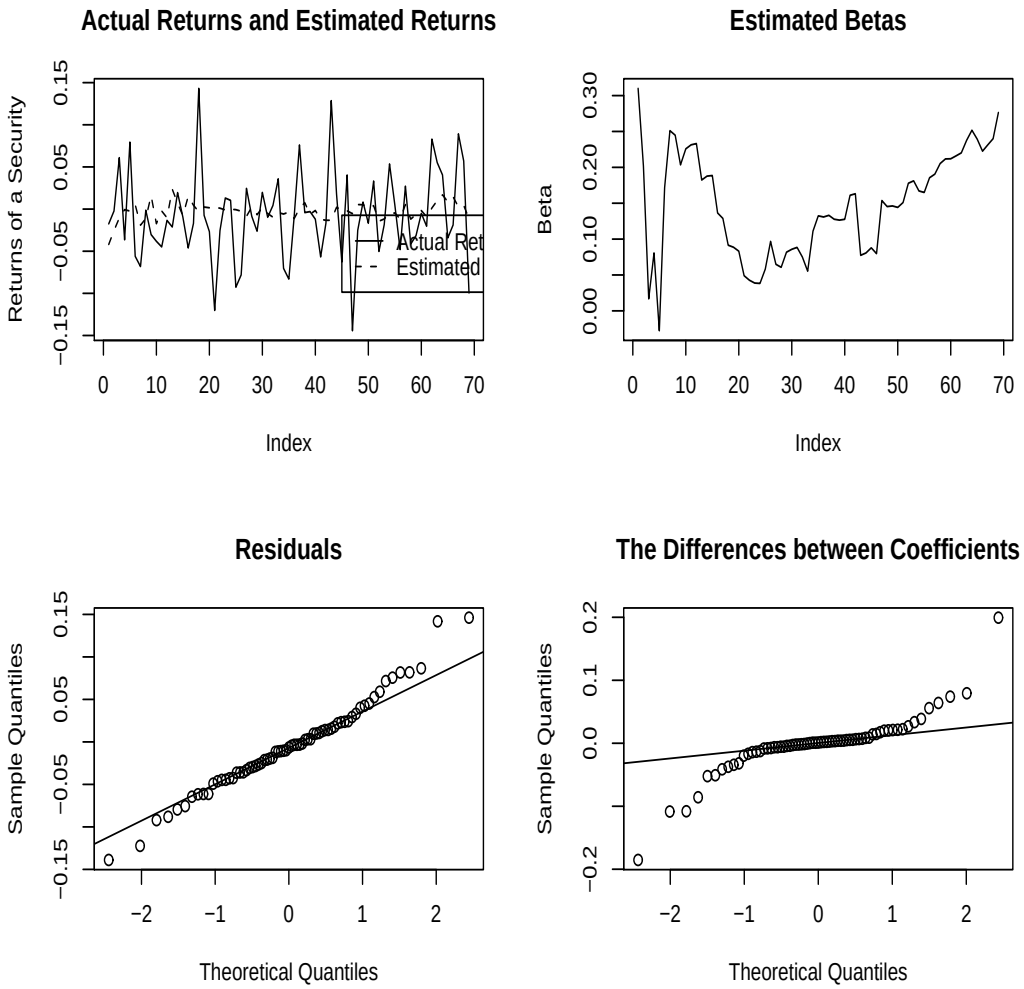

Figure 6.15: JR Toukai

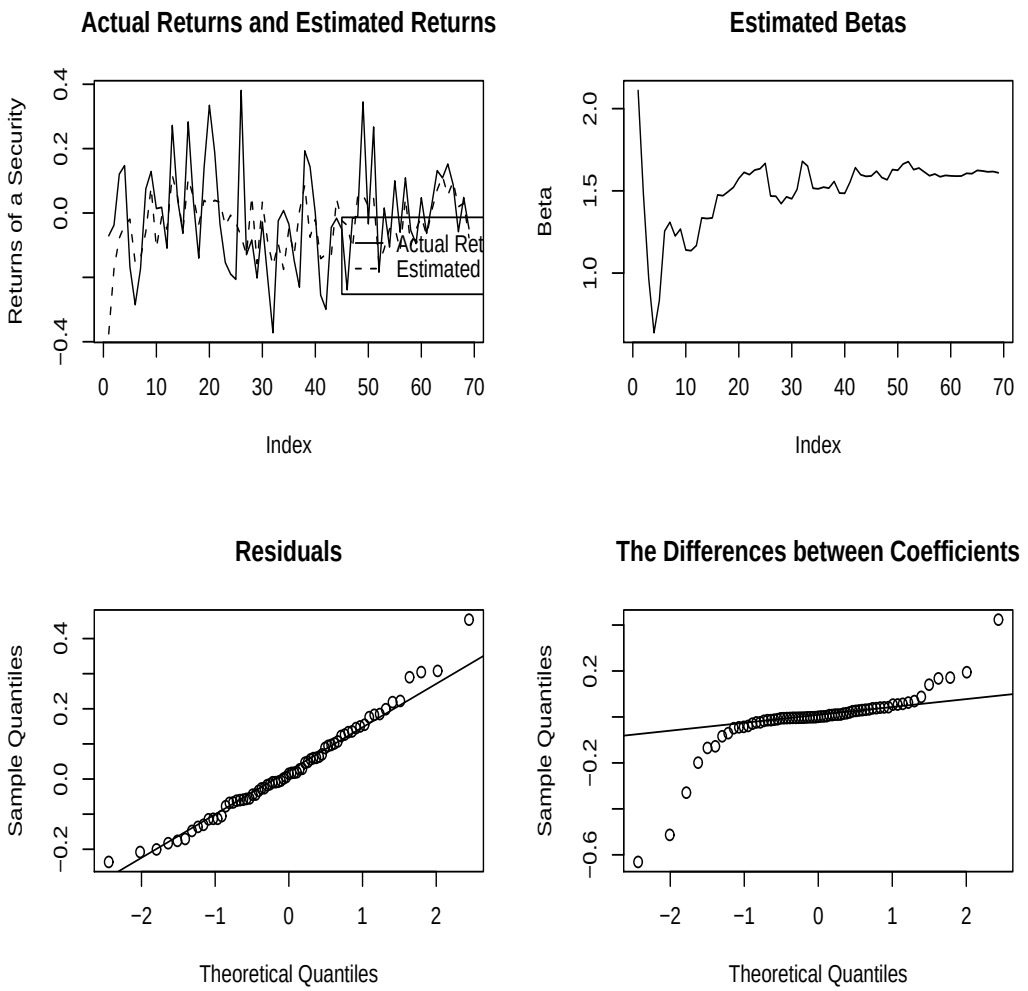

Figure 6.16: KDDI

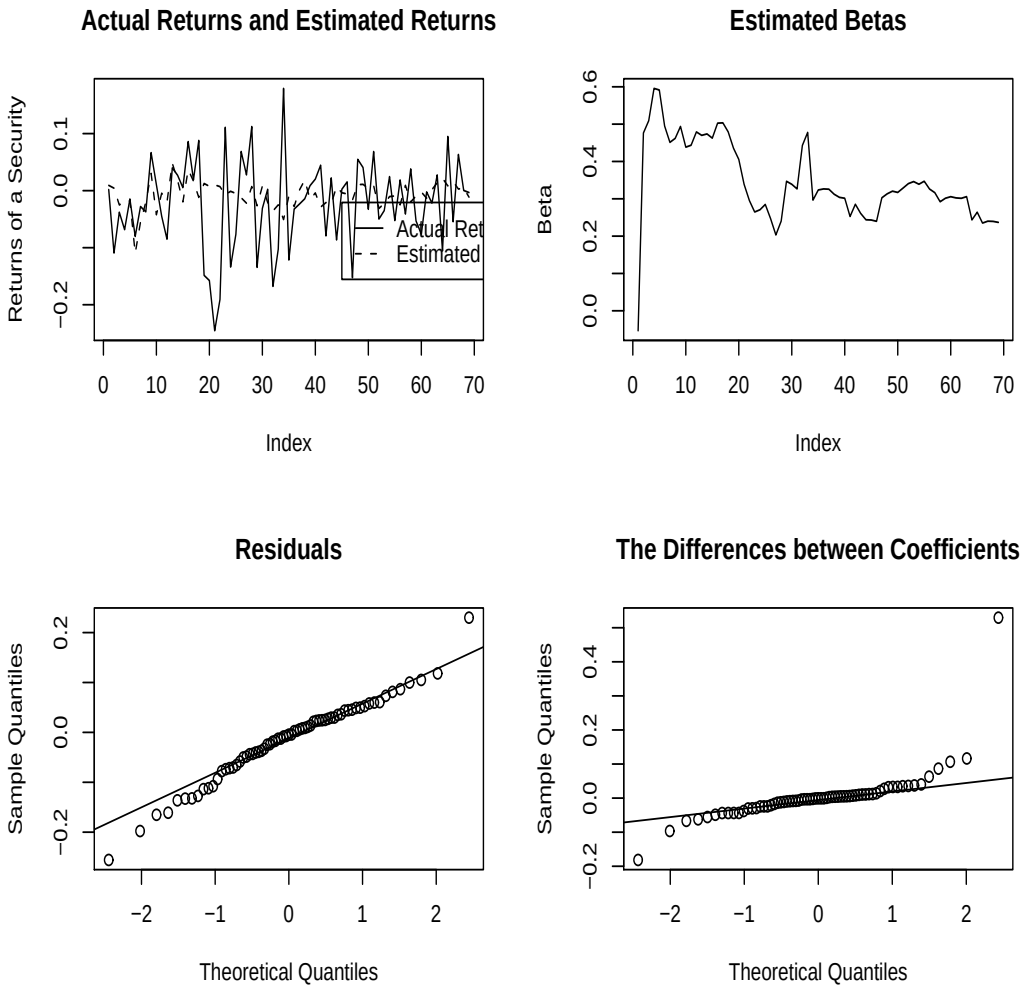

Figure 6.17: JT

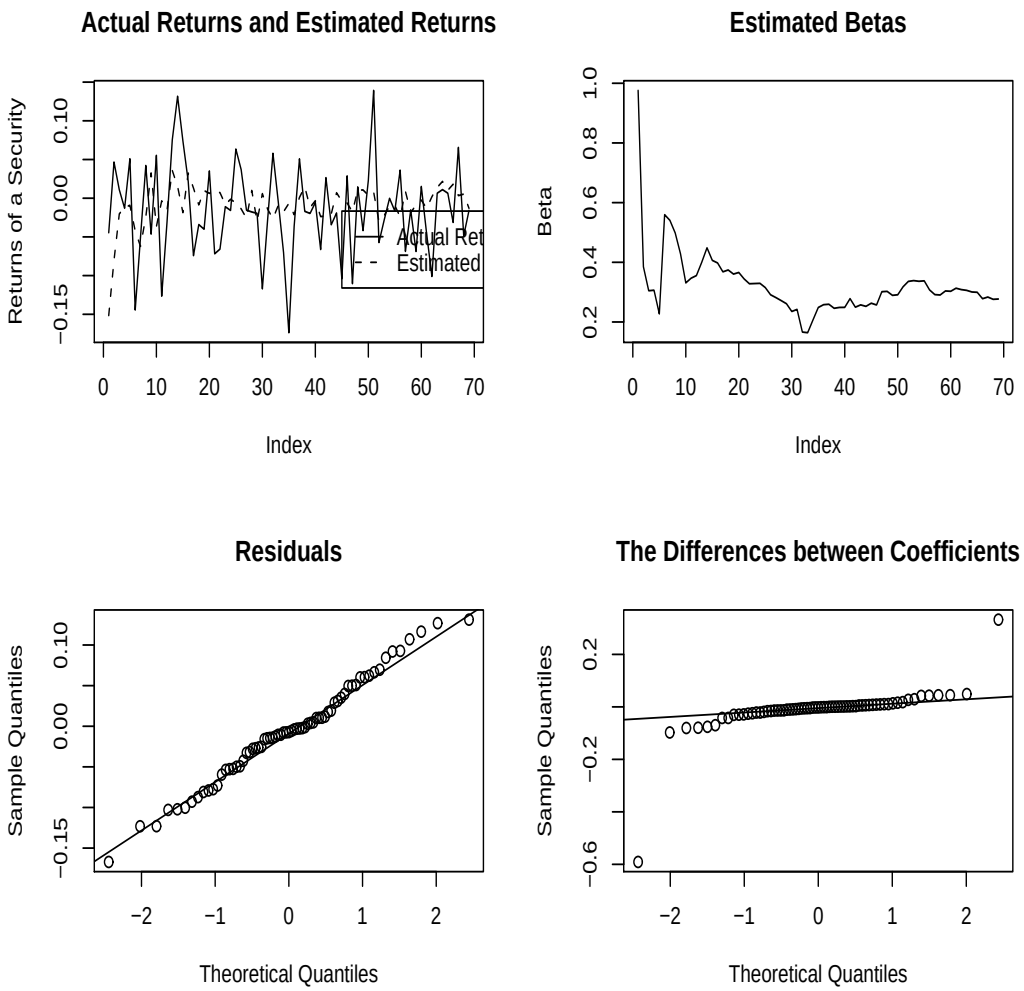

Figure 6.18: Kao

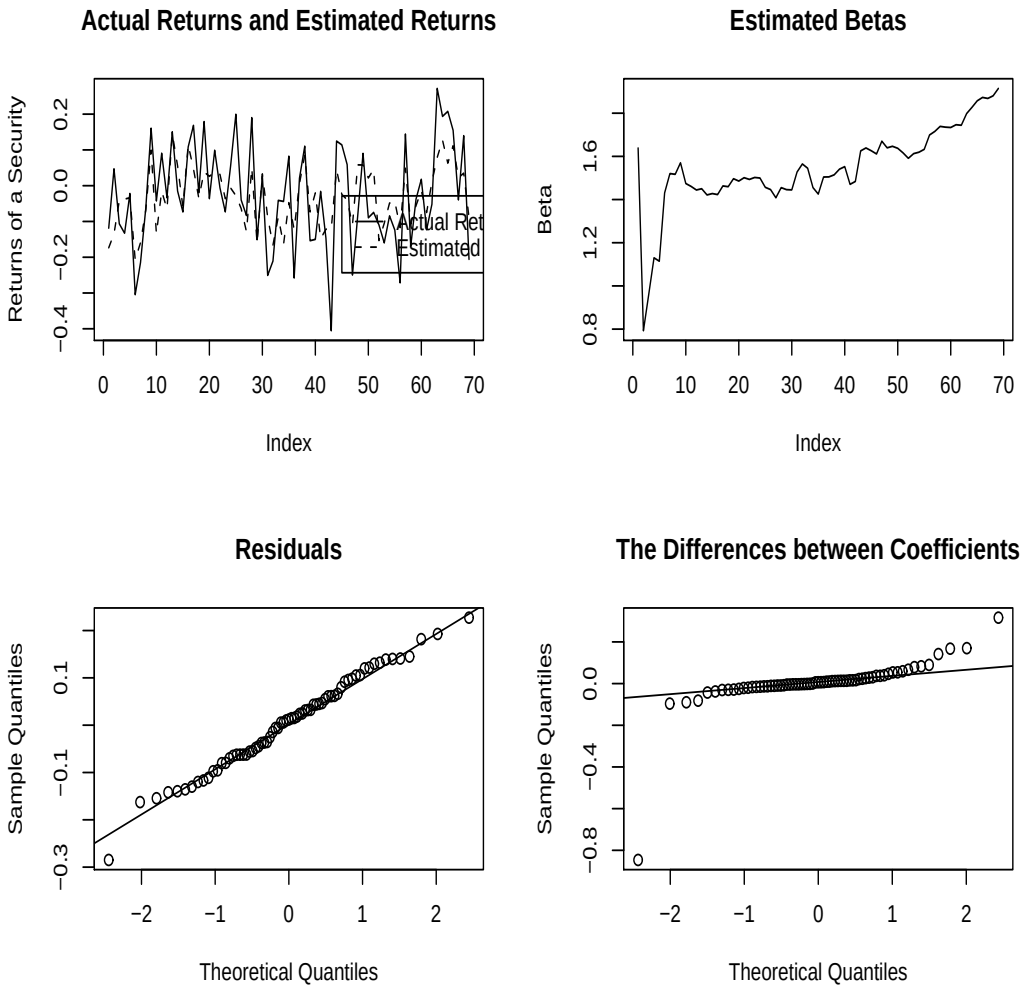

Figure 6.19: NEC

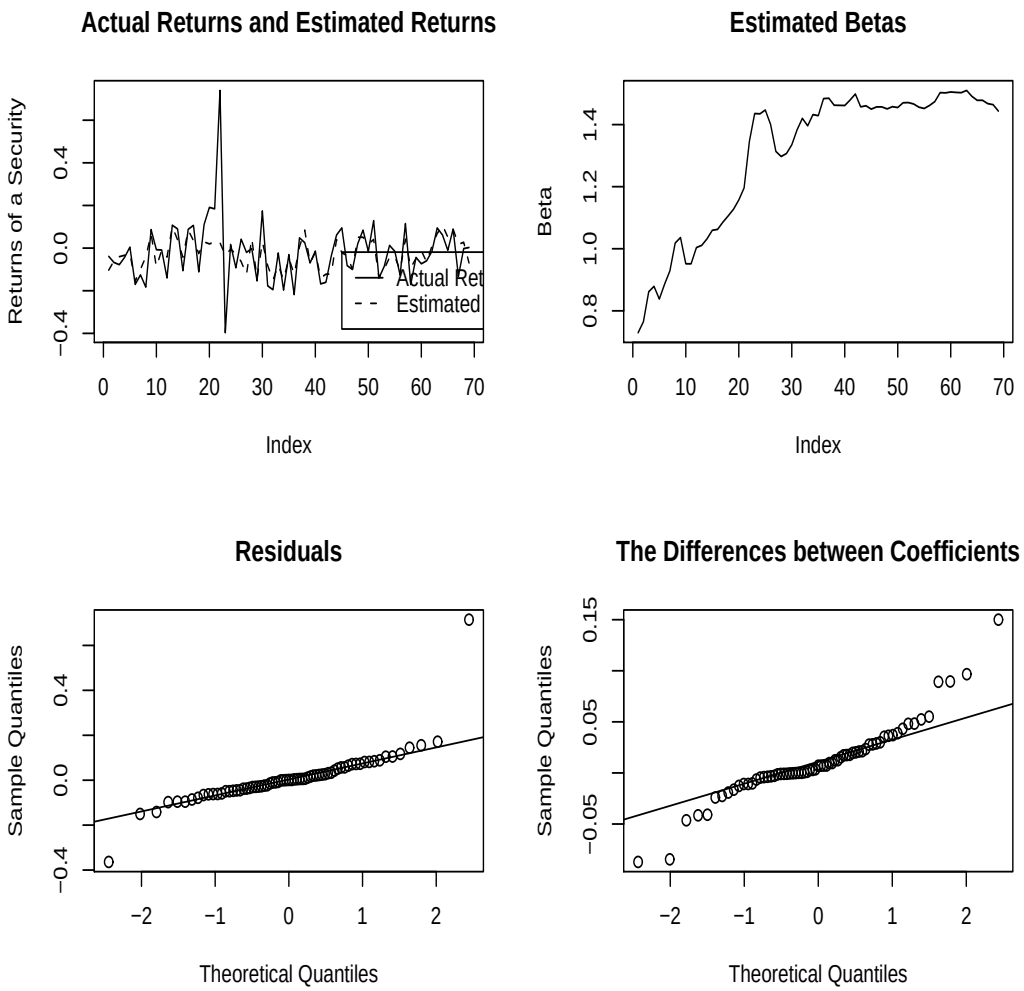

Figure 6.20: Kyocera

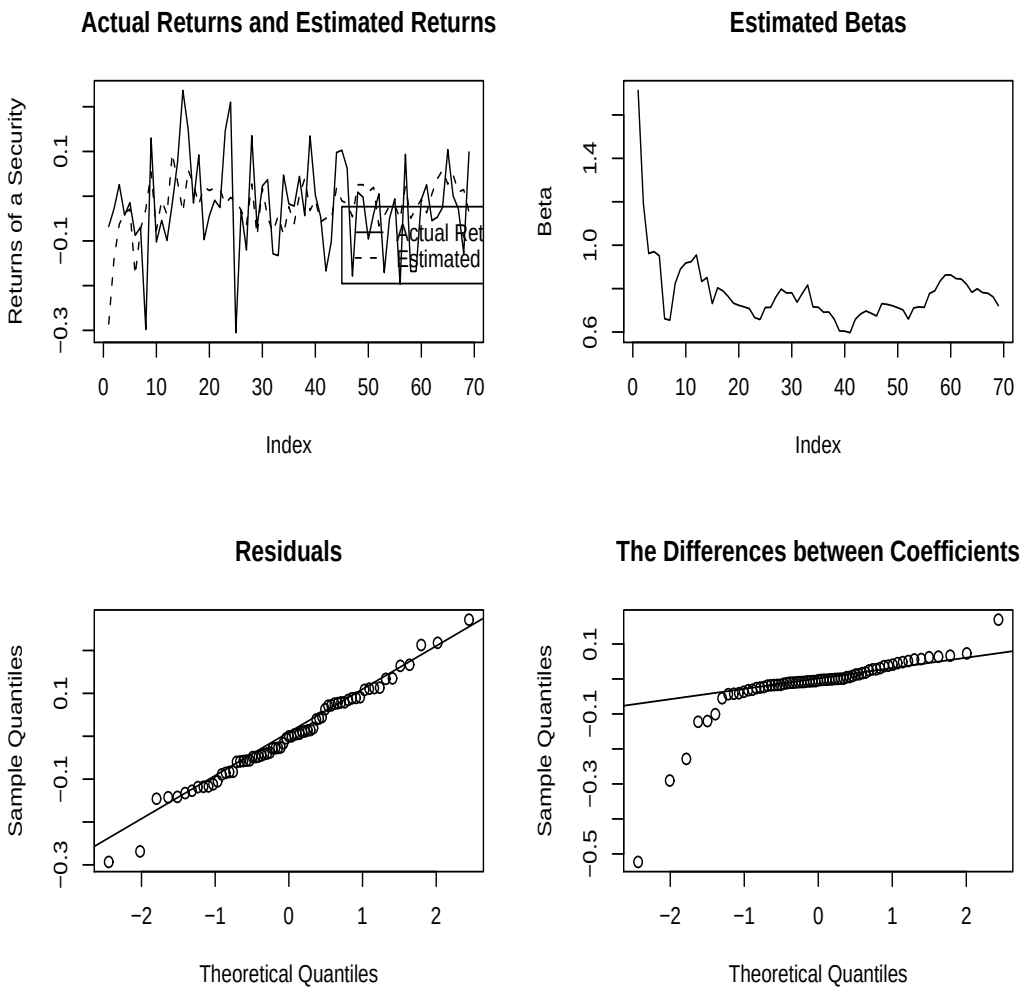

Figure 6.21: Nintendo

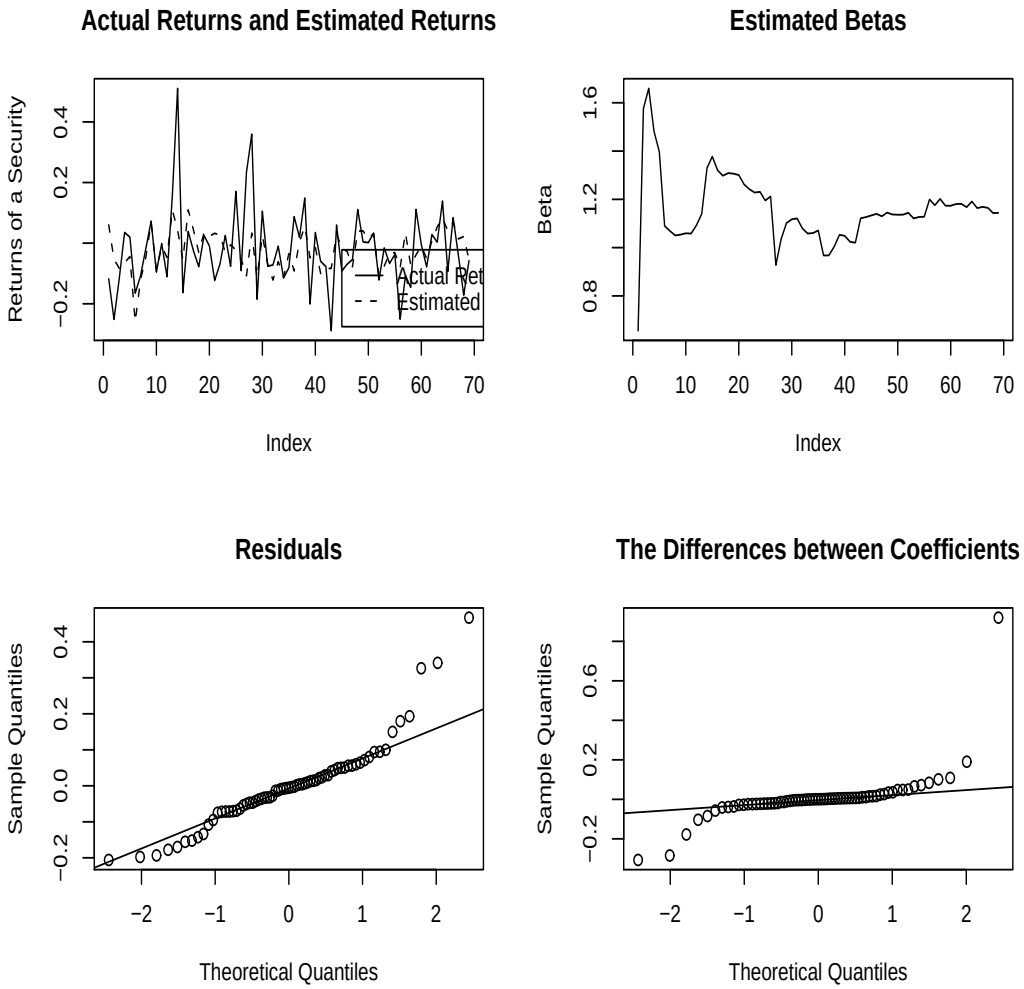

Figure 6.22: Kanebo

Bibliography

[AHMS96] E. W. Anderson, L. P. Hansen, E. R. McGrattan, and T. J. Sargent. On the mechanics of forming and estimating dynamic linear economies. In H. M. Amman, D. A. Kendrick, and John Rust, editors, *Handbook of Computational Economics*, volume 1, pages 171–252. Elsevier, Amsterdam, 1996.

[Aka68] H. Akaike. On the use of a linear model for the identification of feedback systems. *Annals of the Institute of Statistical Mathematics*, 20:425–440, 1968.

[Aka70] H. Akaike. Autoregressive model fitting for control. *Annals of the Institute of Statistical Mathematics*, 23:163–180, 1970.

[Aka73] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrov and F. Casaki, editors, *Proceedings of Second International Symposium on Information Theory*. Akademiai Kiado, 1973.

[Aka89] H. Akaike. Application of the multivariate autoregressive model. *Advances in Statistical Analysis and Statistical Com-*

puting, 2:43–58, 1989.

[AMGC02] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[AS72] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximation. *IEEE Transactions on Automatic Control*, 17:439–448, 1972.

[BT73] G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. Addison-Wesley, Massachusetts, 1973.

[CEE99] L. J. Christiano, M. Eichenbaum, and C. L. Evans. Monetary policy shocks: What have we learned and to what end? In J. B. Taylor and M. Woodford, editors, *Handbook of Macroeconomics*, volume 1, pages 65–148. Elsevier, Amsterdam, 1999.

[CLM97] J. Campbell, A. Lo, and C. MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, New Jersey, 1997.

[CP76] T. F. Cooley and E. C. Prescott. Estimation in the presence of stochastic parameter variation. *Econometrica*, 44:167–184, 1976.

[DdFG01] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[DGA00] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[DT03] A. Doucet and V. B. Tadic. Particle methods for parameter estimation in general state-space models. *Annals of the Institute of Statistical Mathematics*, 55:409–422, 2003.

[Eng] Bank Of England. Framework for monetary policy. The Web Site of the Bank of England.

[GDW01] S. J. Godsill, A. Doucet, and M. West. Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of The Institute of Statistical Mathematics*, 53:82–96, 2001.

[GDW04] S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for non-linear time series. *Journal of the American Statistical Association*, 99:156–168, 2004.

[GSS93] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings-F*, 140:107–113, 1993.

[Hig95] T. Higuchi. On the resampling scheme in the filtering procedure of the Kitagawa Monte Carlo filter. ISM Research Memo, No. 556, 1995.

[HK01] M. Hürseler and H. R. Künsch. Approximating and maximizing the likelihood for a general state-space model. In A. Doucet,

N. de Freitas, and Gordon N., editors, *Sequential Monte Carlo Methods in Practice*, pages 159–175. Springer-Verlag, New York, 2001.

[HP97] R. J. Hodrick and E. C. Prescott. Postwar u.s. business cycles: An empirical investigation. *Journal of Money, Credit, and Banking*, 29:1–16, 1997.

[HW87] J. Hull and A. White. The pricing of options on assets with stochastic volatilities. *Journal of Finance*, 42:281–300, 1987.

[Jap] Liberal Democratic Party Of Japan. The commitments of liberal democratic party 2003. The Web Site of Liberal Democratic Party of Japan (in Japanese).

[Jaz70] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[JK93] X. Jiang and G. Kitagawa. A time varying coefficient vector ar modeling of nonstationary covariance time series. *Signal Processing*, 33:315–331, 1993.

[JPR94] E. Jacquier, N. G. Polson, and P. E. Rossi. Bayesian analysis of stochastic volatility models. *Journal of business & economic statistics*, 12:69–87, 1994.

[Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering (Series D)*, 82:35–45, 1960.

[KG96] G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series*. Springer-Verlag, New York, 1996.

[Kit83] G. Kitagawa. Changing spectrum estimation. *Journal of Sound and Vibration*, 89:433–445, 1983.

[Kit87] G. Kitagawa. Non-Gaussian state space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1041, 1987.

[Kit96] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.

[Kit98] G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93:1203–1215, 1998.

[KSC98] S. Kim, N. Shephard, and S. Chib. Stochastic volatility: likelihood inference and comparison with ARCH models. *Review of Economic Studies*, 65:361–393, 1998.

[Lin65] J. Lintner. The valuation of risk assets and selection of risky investments in stock portfolios and capital budgets. *Review of Economics and Statistics*, 47:13–37, 1965.

[Lit] R. B. Litterman. Optimal control of the money supply. Federal Reserve Bank of Minneapolis, Staff Report, No. 82.

[LJ00] R. Liesenfeld and R. C. Jung. Stochastic volatility models: Conditional normality versus heavy-tailed distributions. *Journal of Applied Econometrics*, 15:137–160, 2000.

[LR04] M. A. Luersen and R. Le Riche. Globalized Nelder-Mead method for engineering optimization. *Computers and Structures*, 82:2251–2260, 2004.

[LS04] L. Ljungqvist and T. J. Sargent. *Recursive Macroeconomic Theory.* MIT Press, Cambridge, 2004.

[Luc76] R. Lucas. Econometric policy evaluation: A critique. *Carnegie-Rochester Conference Series on Public Policy*, 1:19–46, 1976.

[LW01] J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 197–223. Springer-Verlag, New York, 2001.

[LWH04] J. Y. Lin, Y. Wu, and T. S. Huang. 3D model-based hand tracking using stochastic direct search method. *Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 693–698, 2004.

[Man63] B. Mandelbrot. The variation of certain speculative prices. *The Journal of Business*, 36:394–419, 1963.

[Mar52] H. M. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

[McC] B. T. McCallum. Alternative monetary policy rules: A comparison with historical settings for the united states, the united kingdom, and japan. NBER Working Paper, No. w7725.

[MR05] A. M. Mineo and M. Ruggieri. A software tool for the exponential power distribution: The normalp package. *Journal of Statistical Software*, 12:1–24, 2005.

[Mut] T. Muto. Battle against deflation. The Web Site of The Bank of Japan.

[NM65] J. A. Nelder and R. Mead. A simplex algorithm for function minimization. *Computer Journal*, 7:308–313, 1965.

[Orp01] A. Orphanides. Monetary policy rules based on real-time data. *American Economic Review*, 91:964–985, 2001.

[Sha64] W. F. Sharpe. Capital asset prices: A theory of market equilibrium under condition of risk. *Journal of Finance*, 19:425–442, 1964.

[She05] Neil Shephard, editor. *Stochastic Volatility: Selected Readings*. Oxford University Press, 2005.

[Shi73] R. Shiller. A distributed lag estimator derived from smoothness priors. *Econometrica*, 41:775–788, 1973.

[Shi99] S. Shiratsuka. Measurement errors in japanese consumer price index. *Monetary and Economic Studies*, 17:69–102, 1999.

[Sma] D. J. C. Smant. Monetary policy strategy: Optimal control, discretion, fine tuning. Manuscript.

[Tay86] S. J. Taylor. *Modelling Financial Time Series*. John Wiley, Chichester, 1986.

[Tay93] J. B. Taylor. Discretion vs. policy rules in practice. *Carnegie-Rochester Conference Series on Public Policy*, 39:195–214, 1993.

[Ton90] H. Tong. *Non-linear time series: a dynamical system approach*. Oxford University Press, Oxford, 1990.

[WA03] T. Watanabe and M. Asai. Stochastic volatility models with heavy-tailed distributions: a Bayesian analysis. COE discussion paper series, No. 1, faculty of economics, Tokyo Metropolitan University, 2003.

[Wal03] C. Walsh. *Monetary Theory and Policy*. MIT Press, Cambridge, 2003.

[WHM85] M. West, P. J. Harrison, and H. S. Migon. Dynamic generalized linear models and Bayesian forecasting (with discussion). *Journal of the American Statistical Association*, 80:73–97, 1985.

[Woo03] M. Woodford. *Interest and Prices: Foundations of a Theory of Monetary Policy*. Princeton University Press, Princeton, 2003.

[Yan07] K. Yano. A self-organizing state space model and simplex initial distribution search. *Computational Statistics*, 22, 2007. Forthcoming.