Latent factor commodity models: filtering and estimation.

by

Tran Huy

MSc. thesis.

Department of Mathematics and Statistics,

University of NSW.

June 2010

Abstract

In this thesis, we consider three different commodity models that take into account the mean reverting nature of commodity prices. The first model is a simple one-factor model based on the one proposed in [9]. In this model, the convenience yield and interest rate are assumed to be constant. The second model is the two-factor model developed in [21]. This model involves two factors: the short-term deviation and the equilibrium price level. The last model is a three-factor model proposed in [22]. We mainly focus on the second and third models since the first model and the second one are equivalent in general.

We study these three models based on two aspects: discretization and filtering. For the first study, we observe and compare theoretically and empirically two well-known discretization schemes, namely, the Euler scheme and the Milstein scheme. This study turns out to be useful for the filtering aspect. Indeed, once a model has been put in a state space form which can be obtained by using a discretization technique, then this enables filtering techniques to be applied to solve the filtering recursion problem for the model.

The second study aims to observe and compare the performance of the three well-known filtering techniques, namely, the Kalman filter, the Extended Kalman filter and the Particle filter. We implement these three filters for the second and third models via using Matlab. In addition, the data utilised to test the models involve futures contracts, since in most commodity markets the futures price is more flexible and easily observed than the spot price of a commodity.

Contents

1 Introduction

The study of commodity prices is an important issue for many application areas such in business and finance. There have been many commodity models constructed to imitate the stochastic behaviours of commodity prices. In this thesis, the study focuses on three types of commodity models which take into account the mean reverting nature of commodity prices. The first model is a one-factor model which was developed in [20]. The second model is a two-factor model proposed by [21], and the last model is a three-factor model constructed by [22].

We mainly focus on two aspects for these three models: filtering and estimation. Particularly, we concentrate on the discrete time state space approach to filtering and estimating the models. Under discrete time formulation, we present and compare two common discretization schemes, namely the Euler scheme and the Milstein scheme. The Milstein scheme is shown to be computationally and theoretically efficient for simulating the stochastic process for each of these three models. However, when the discretization interval is chosen to be very small, then the Euler and Milstein schemes appear to produce similar performance. Hence, we conclude that Milstein scheme provides a significant computational saving in these models. Especially in the three factor model, in which Milstein scheme takes a simple form. Moreover, we also examine the statistical properties of these schemes in terms of strong and weak convergence.

We then develop novel state space models and derive the observation equation for the futures price - followed by the state space model for each model under Milstein and Euler schemes.

For the filtering aspect, we present the filtering recursion problem, and then discuss and compare the performance of a few well-known filtering techniques: the Kalman filter, the Extended Kalman filter and the SIR Particle filter. The Kalman filter is optimal for the Gaussian and linear assumptions on the state space model. Specifically, the Kalman filter generates optimal estimators for the state space model at each discrete time point. However, when the state space model is no longer linear, such as the three-factor model, but still keeps the Gaussian assumption, then the EKF can be applied to achieve a good filtering result. For the nonlinear and non-Gaussian case, since the KF and EKF cannot be utilised, the Particle filter appears to give the solution for the filtering recursion problem.

Moreover, the data used to implement these filters involve five future contracts on a basis of daily observations. It should be noted that in most exchange markets, the spot price of a commodity is usually uncertain and not observed frequently, and so it cannot be used to

implement the filters. Due to this reason, the corresponding futures contract closest to maturity is utilised as a proxy for the spot price. Indeed, futures contracts are normally traded on many exchanges and hence their prices can be easily observed. Furthermore, for the models 1, 2 and 3, closed form solution for the futures price can be obtained, which may greatly simplify the comparative statics and empirical estimation.

The content of the thesis is organized as follows. The basic econometric concepts and models are introduced in section 3. The simulation study for the Euler and Milstein schemes are presented in sections 4 and 5. In sections 6 and 7, we introduce the state space model framework and state the filtering recursion problem as well as the assumptions on the state space model. In sections 8 and 9, we present filtering issues including techniques and empirical results, as well as study some impacts of futures contracts and the models on the performance of filtering techniques. Finally, section 10 concludes.

2 Notation and definitions

In this thesis, the following notations will be used:

- Capital letters represent stochastic processes. For identifying purpose, in sections 6, 7 and 8, we use bold type to account for random variables, and normal type to represent realisations or deterministic quantities.
- k (or t): discrete time.
- $\mathbb{N}$: the set of natural numbers.
- $\mathbb{R}$: the set of real numbers.
- $(.)^T$: transpose of a matrix. For example, M^T where M is a matrix.
- p(.) is the probability density function, while $p(.|.)$ is the conditional probability density function.
- $\mathbb{E}[\boldsymbol{x}]$ is the expectation of a random variable $\boldsymbol{x}$.
- $\mathbb{V}ar[\boldsymbol{x}]$ is the variance of a random variable $\boldsymbol{x}$.
- $\mathbb{C}ov[\boldsymbol{x}]$ is the covariance of a random variable $\boldsymbol{x}$.
- $\mathbb{S}td[\boldsymbol{x}]$ is the standard deviation of a random variable $\boldsymbol{x}$.
- $\mathcal{N}$(x; m, P) is a Gaussian density with argument x, mean m, and covariance P.
- $\tilde{\delta}(.)$ is a Dirac delta measure[1].
- $\boldsymbol{x}_{1:k} = \{\boldsymbol{x}_i, i = 1, ..., k\}$ is the set of all states up to time k.
- $\boldsymbol{z}_{1:k} = \{\boldsymbol{z}_i, i = 1, ..., k\}$ is the set of all measurements up to time k.
- w_k^i is the normalized weight of the ith particle at time k, while w_k^{*i} is referred to as the "true weight".

[1] A Dirac delta can be thought of as a function on the real line which is zero everywhere except the origin, where it is infinite, $\tilde{\delta}(x) = \left\{ \begin{matrix} +\infty, x = 0 \\ 0, x \neq 0 \end{matrix} \right\}$ and which is also constrained to satisfy the identity $\int_{-\infty}^{\infty} \tilde{\delta}(x)\, dx = 1$

- Supppose the state space at time k consists of discrete states $x_k^i, i = 1, ..., N_s$. For each state x_k^i, let the conditional probability of that state, given measurements up to time k be denoted by $w_{k|k}^i$, that is, $Pr(\boldsymbol{x}_k = x_k^i|\boldsymbol{z}_{1:k}) = w_{k|k}^i$. Similarly, we denote the conditional probability of the state $\boldsymbol{x}_k^i$, given measurements up to time k - 1, by $w_{k|k-1}^i$ (i.e. $Pr(\boldsymbol{x}_k = x_k^i|\boldsymbol{z}_{1:k-1}) = w_{k|k-1}^i$).

  Notation for Euler and Milstein schemes

- $\triangle$, $\triangle k$ or $\triangle t$: the time step length or the time interval.

- $\triangle W$ is the increment of a Wiener process W.

- X_0: the initial state of the true stochastic process X, $Y_0^{\triangle}$: the initial state of the simulation process Y based on the step length $\triangle$.

- p is the truncation which is provided in [14], who suggest $p \geq \frac{K}{\triangle t}$ for some positive constant K.

- dX_t^i is the i-th component of a general n-dimensional stochastic differential equation (SDE) with m-dimensional Wiener process. It is denoted as follows

$$dX_t^i = a^i\left(t, X_t\right) dt + \sum_{j=1}^{m} b^{i,j}\left(t, X_t\right) dW_t^j. \tag{2.1}$$

  where $a^i(t, X_t)$ and $b^{i,j}(t, X_t)$ are respectively the drift and the diffusion (or volatility) coefficients of the process X^i.

- L^j is the derivative operator which is defined by $L^j = \sum_{i=1}^{n} b^{i,j} \frac{\partial}{\partial \boldsymbol{x}_i}$.

- $I_{(j_1,j_1)\triangle t}$ is the multiple Ito integral given by

$$I_{(j_1,j_2)\triangle t} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_1} dW_{s_1}^{j_1} dW_{s_1}^{j_2}. \tag{2.2}$$

- $J_{(j_1,j_2)}$ is the multiple Stratonovich integral given by

$$J_{(j_1,j_2)\triangle t} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_1} dW_{s_1}^{j_1} dW_{s_1}^{j_2}. \tag{2.3}$$

- $J_{(j_1,j_2)}^p$ is the approximation of the multiple Stratonovich integral $J_{(j_1,j_2)}$. $J_{(j_1,j_2)}^p$ is given as follows:

$$\begin{aligned} J_{(j_1,j_2)}^p = & \triangle t \left(\frac{1}{2} \boldsymbol{\zeta}_{j_1} \boldsymbol{\zeta}_{j_2} + \sqrt{\boldsymbol{\rho}_p} \left(\boldsymbol{\mu}_{j_1,p} \boldsymbol{\zeta}_{j_2} - \boldsymbol{\mu}_{j_2,p} \boldsymbol{\zeta}_{j_1} \right) \right) \\ & + \frac{\triangle t}{2\pi} \sum_{r=1}^{p} \frac{1}{r} \left(\boldsymbol{\psi}_{j_1,r} \left(\sqrt{2} \boldsymbol{\zeta}_{j_2} + \boldsymbol{\nu}_{j_2,r} \right) - \boldsymbol{\psi}_{j_2,r} \left(\sqrt{2} \boldsymbol{\zeta}_{j_1} + \boldsymbol{\nu}_{j_1,r} \right) \right), \end{aligned} \tag{2.4}$$

where $\boldsymbol{\zeta}_j, \boldsymbol{\mu}_{j,p}, \boldsymbol{\nu}_{j,r}$ and $\boldsymbol{\psi}_{j,r}$ are all independent $\mathcal{N}(0;1)$ Gaussian random variables with,

$$\rho_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2}, \tag{2.5}$$

$$\boldsymbol{\zeta}_j = \frac{1}{\sqrt{\triangle t}}\triangle W^j. \tag{2.6}$$

- $n_{S,t-1}$, $n_{\delta,t-1}$ and $n_{\mu,t-1}$ are i.i.d. standard normal random variables (where S, δ and μ are the spot price, the convenience yield and the interest rate respectively).

3 Commodity models

3.1 Basic concepts in Econometric Theory

In this section, we first present some important background on commodity models as well as review some major commodity exchanges in the world. After that, we introduce basic concepts on forward and futures contracts, as well as market conditions on commodity.

3.1.1 Commodities and commodity exchanges

As defined in [5], a **commodity** is a physical substance, such as food, grains and metals, which is interchangeable with another product of the same type, and which investors buy or sell, usually through futures contracts. A commodity can be produced, consumed, transported or stored. The price of a commodity is subject to supply and demand of the market. More generally, a commodity is a product which trades on a commodity exchange; this would also include foreign currencies and financial instruments and indexes. There are many types of commodity in general, and the basic ones can be listed as follows

- **Energy :** crude oil, gasoline, natural gas, electricity, etc.
- **Metals :** copper, silver, gold, aluminum, zinc, etc.
- **Agricultural :** rice, wheat, salt, beans, coffee, pork bellies, grains, etc.
- **Others :** paper, chemicals, pulp, etc.

The commodity models presented in this thesis are focused on tangible commodities.

A **commodity exchange** (see [11]) is an exchange where various commodities are traded. Most commodity markets across the world trade in agricultural products and other raw materials (like sugar, milk, wheat, coffee, oil metals, etc.) and contracts based on them. These contracts can include legal details regarding spot prices, forwards, futures and options on futures. Commodity exchanges usually trade futures contracts on commodities, basically trading contracts to receive an amount of the commodity in a certain date in the future. The main commodity exchanges worldwide include [18]:

- **CME Group, Inc. (CME):** based in Chicago, is the world's largest derivatives exchange. It was formed by the 11.6 billion dollars merger in July 2007 of the 109-year-old Chicago

Mercantile Exchange[2] and the 159-year-old Chicago Board of Trade[3]. CME Group now handles around 90 percent of all futures in the United States. The CME's product complex spans all major asset classes, including: futures and options on interest rates, indexes, currencies, commodities, energy products, precious metals, and alternative investment instruments such as weather and real estate derivatives. The CME Group's total contract volume dropped 21 percent in 2009 to 2.590 billion, according to the annual Futures Industry Association (FIA)'s survey of the world's leading derivatives exchanges. The FIA report, published in early April 2010, notes that volume on the CME Group's Eurodollar futures contract fell by 26.7 percent to 437.6 million in 2009.

- **New York Mercantile Exchange (NYMEX):** opened in 1872, located in New York. It is now the world's largest energy and metals commodity exchange, and a unit of CME Group Inc. Its two principal divisions are the New York Mercantile Exchange and Commodity Exchange, Inc (COMEX) which were once separated but are now merged. NYMEX offers trading in crude oil, petroleum products, natural gas, coal, electricity, gold, silver, copper, aluminum, platinum group metals, emissions, and soft commodities contracts for trading and clearing virtually 24 hours a day. NYMEX's average daily volume for 2007 was 1.485 million contracts, a 25 percent increase over 2006. The floor of the NYMEX is regulated by the Commodity Futures Trading Commission, an independent agency of the United States government. The NYMEX is one of the few exchanges in the world to maintain the open outcry system where traders employ shouting and complex hand gestures on the physical trading floor.

- **Eurex:** is one of the world's largest and most diverse derivatives exchanges. It provides clearing services for derivatives, equities, bonds. Founded in 1998, Eurex is jointly operated by Deutsche Borse and the SIX Swiss Exchange, with the German group holding 50 percent of the voting rights and 85 percent of the share capital. The agreement was extended for 10 years in 2003. Eurex ranked as the world's second-largest derivatives exchange by contract

[2]The former Chicago Mercantile Exchange is a global futures and options exchange that in 2007 acquired the Chicago Board of Trade to become CME group. The Chicago Mercantile Exchange's products include futures and options on interest rates, foreign currencies, stock indexes, commodities and investment products (real estate, weather).

[3]The former Chicago Board of Trade (CBOT) is the oldest and one of the largest futures exchanges in the world. The CBOT's products include futures and options on futures: agricultural, interest rate, Dow stock indexes and precious metals.

volume in 2009, according to the annual Futures Industry Association (FIA)'s survey of the world's leading derivatives exchanges. In April 2010, the FIA notes that the number of futures and options traded on Eurex fell 16.6 percent in 2009 to 2.647 billion, dropping it behind the Korea Exchange for the top spot.

- **Korea Exchange (KRX):** was founded in 2005 from the merger of four domestic Korean exchanges: the Korea Stock Exchange (KSE), the Korea Futures Exchange (Kofex), the Kosdaq Market and the Kosdaq committee. It provides an electronic platform for the trading, clearing and settlement of cash equities, bonds and derivatives. Ownership of the KRX was split in 2007 between 44 financial and government entities, led by Woori Investment Securities (6.3 percent), Daewoo Securities (3.2 percent) and Daishin Securities (3.2 percent). Overseas owners include JP Morgan, Citigroup and Australia's Macquarie Bank. According to the 2009 volume rankings of world derivatives exchanges by the Futures Industry Association (FIA), the KRX ranked first, with over 3.1 billion futures and options traded and cleared. Moreover, KRX increased its 2009 trading volume by 8.3 percent over 2008, when it ranked third behind only CME Group and Eurex.

- **Shanghai Futures Exchange (SHFE):** formed in 1998 as part of the restructuring of China's futures industry. Particularly, it was merged from the Shanghai Metal Exchange, Shanghai Cereals and Oil Exchange and Shanghai Commodity Exchange. SHFE involves contracts in steel, copper, aluminum, natural rubber, fuel oil, zinc and gold. In 2009, it was ranked the world's tenth-largest derivative exchange by contract volume, according to the annual Futures Industry Association (FIA)'s survey of the world's leading derivatives exchanges. The FIA report, published in early April 2010, notes that the number of futures and options traded on SHFE more than trebled from 2008 to 2009 to reach a volume of 434.9 million. The SHFE's leading contract by volume in 2009 was its Fuel Oil Futures, which rose almost 50 percent on 2008 to rank as the FIA's fifth-largest energy derivatives contract by volume.

- **London Metal Exchange (LME):** established over 130 years ago and located in the centre of London. The MLE offers futures and options contracts for plastics, aluminium, copper, nickel, tin, zinc and lead, plus two regional aluminium alloy contracts. It was ranked as the world's 19th-largest derivatives exchange by volume in 2009, according to the Futures Industry Association (FIA). The FIA report, published in early April 2010, notes that the MLE's total volume for 2009 reached 111.93 million. The LME is a highly

liquid market and in 2007 achieved volumes of 93 million lots, equivalent to 9,500 billion dollars annually and between 35 billion dollars to 45 billion dollars on an average business day.

This thesis aims to study estimation and modelling on flexible multifactor stochastic models to capture the price behaviour of commodities in each of these exchanges.

3.1.2 Forward vs Futures contracts

A forward contract is an agreement to buy or sell an asset at a certain future time for a certain price. It is usually traded in the over-the-counter market[4] and there is no standard contract size or standard delivery arrangements. A single delivery date is usually specified and the contract is usually held to the end of its life and then settled.
Like a forward contract, a futures contract is an agreement between two parties to buy or sell an asset at a certain time in the future for a certain price. Unlike forward contracts, futures contract are normally traded on an exchange. To make trading possible, the exchange specifies certain standardized features of the contract. As the two parties to the contract do not necessarily know each other, the exchange also provides a mechanism that gives the two parties a guarantee that the contract will be honored. In addition, a range of delivery dates is usually specified. A futures contract is settled daily and usually closed out prior to maturity. The largest exchanges on which futures contract are traded are the Chicago Board of Trade (CBOT) and the Chicago Mercantile Exchange (CME). On these and other exchanges throughout the world, a very wide range of commodities and financial assets form the underlying assets in the various contracts. The traded commodities include pork bellies, live cattle, sugar, wool, lumber, copper, aluminum, gold and tin. The financial assets include stock indices, currencies and Treasury bonds.

3.1.3 Commodity Market Conditions: Contango vs Normal backwardation

- **Contango** is when the futures price is above the expected future spot price. Since the futures price must converge to the expected future spot price, contango implies that futures prices are falling over time as new information brings them into line with the expected

[4]The over-the-counter market is a telephone- and computer-linked network of dealers in which trades are done over the phone and are usually between two financial institutions or between a financial institution and one of its clients.

future spot price.

- **Normal backwardation** is when the futures price is below the expected future spot price. This is desirable for speculators who are "net long" in their positions: they want the futures price to increase. Therefore, normal backwardation is when the futures prices are increasing.

3.2 Commodity models

The stochastic models of commodity prices play a crucial role when evaluating commodity-related securities and projects. Recently, a number of authors have considered the use of mean-reverting price models and argued that these models are appropriate for many commodities (for example, see [9],[22],[19]). Intuitively, by the law of supply and demand, when the price of a commodity is higher than some long-run mean or equilibrium price level, the supply of the commodity will go up since higher cost producers of the commodity will enter the market. New production comes on line, older production expected to go off line stays on line, therefore putting a downward pressure on prices. Conversely, when prices are relatively low, supply will decrease since some of high-cost producers will exit the market, putting an upward pressure on prices. The impact of relative prices on the supply of the commodity will induce mean reversion in commodity prices. Moreover, when these entries and exits are not instantaneous, prices may be temporarily high or low but will eventually revert toward the equilibrium level.
In this thesis, we present three models that in different ways take into account the mean reverting nature of commodity prices, and also estimate them using futures data. The first model is a simple one-factor model in which the convenience yield and interest rate are assumed to be constant. The second model is the two-factor model developed in [21]. For most commodities, there seem to be some mean reversion in prices but there is also uncertainty about the equilibrium price to which prices revert. This two-factor model captures both of these effects by using two factors: the short-term deviation and the equilibrium price level. In this model, the equilibrium price level is assumed to follow a geometric Brownian motion with drift reflecting expectations of the exhaustion of existing supply, improving technology for the production and discovery of the commodity, inflation, political and regulatory effects. The short-term deviations which are defined as the difference between spot and equilibrium prices are typically expected to revert toward zero following an Ornstein-Uhlenbeck process [21]. These deviations may reflect, for instance, short-term changes in demand resulting from variations in the weather or intermitent

supply disruptions, and are tempered by the ability of market participants to adjust inventory levels in response to changing market conditions. The last model is a three-factor model proposed in [22]. This model takes into account the convenience yield factor which stands for the benefit from physically holding an asset rather than owning a futures contract. Specifically, when a shortage of the asset occurs in the market, it is better to own the asset already than to buy it since it is likely to be at a very high price due to high demand of the market. In addition, in this model, the convenience yield of the commodity and the interest rate are also assumed to follow mean reverting processes [22]. These assumptions under the multifactor models in this thesis result in closed form solutions for the resulting future prices derived in section 12.

One of the main difficulties in the empirical implementation of commodity price models is that the factors or state variables of these models are frequently not directly observable. In many cases, the spot price of a commodity is so uncertain that the corresponding futures contract closest to maturity is used as a proxy for the spot price. Futures contracts, nevertheless, are normally traded on several exchanges and their prices can be easily observed. Fortunately, for these three models, closed form solutions for the prices of futures and forward contracts can be obtained, which greatly simplifies the comparative statics and empirical estimation. In addition, for all three models the logarithm of the futures price is linear in the underlying factors, a property which turns out to be very useful in respect of the econometric technique used to estimate the parameters of the models.

3.2.1 Model 1

The one-factor model is based on the one proposed in [9] where the convenience yield c and the interest rate r are assumed to be constant. The dynamics for the spot price under the real-world framework can be expressed as

$$dS = \mu S dt + \sigma S dZ. \tag{3.1}$$

Let X = $\ln S$. Then from Ito's Lemma [13], the log of the spot price follows the dynamics as

$$dX = d\ln S = \left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma dZ. \tag{3.2}$$

The dynamics for the spot price under the equivalent martingale (or risk neutral) measure is given by [20]:

$$dS = (r - c)\, S dt + \sigma S dZ^* \tag{3.3}$$

where dZ^* is an increment of a standard Brownian motion process under the risk neutral measure. In this model, since the interest rate is assumed to be constant, the futures price must equal to the forward price. In Appendix 12, via using the Kolmogorov backward equation, the futures price F with maturity T for the spot price S is derived to be

$$F(S,T) = Se^{(r-c)T}. \tag{3.4}$$

Moreover, the futures price can be expressed in the log form as

$$\ln F(X,T) = X + (r-c)T. \tag{3.5}$$

3.2.2 Model 2

This is a two-factor model which was developed in [21]. Let S_t denote the spot price of a commodity at time t. The spot price is then decomposed into two stochastic factors as $\ln(S_t) = \chi_t + \xi_t$, where χ_t is referred to as the short-term deviation in prices and ξ_t is the equilibrium price level. The dynamics for the short-run deviation and the equilibrium level are given in [21] as

$$d\chi_t = -\kappa\chi_t dt + \sigma_\chi dZ_\chi \tag{3.6}$$

$$d\xi_t = \mu_\xi dt + \sigma_\xi dZ_\xi \tag{3.7}$$

where the mean reversion coefficient κ describes the rate at which the short-term deviations are expected to disappear, dZ_χ and dZ_ξ are correlated increments of standard Brownian motion processes with $dZ_\chi dZ_\xi = \rho dt$, ρ being constant.

This model is called the short-term/long-term model [21]. Moreover, in order to value futures contracts as well as other commodity-related investments, this model can be transformed into risk-neutral stochastic processes as follows [21]

$$d\chi_t = (-\kappa\chi_t - \lambda_\chi)\,dt + \sigma_\chi dZ^*_\chi \tag{3.8}$$

$$d\xi_t = (\mu_\xi - \lambda_\xi)\,dt + \sigma_\xi dZ^*_\xi \tag{3.9}$$

where λ_χ and λ_ξ are the risk premiums for the short-term and long-term factors respectively, dZ^*_χ and dZ^*_ξ are increments of standard Brownian motion processes under the risk neutral measure with $dZ^*_\chi dZ^*_\xi = \rho dt$, ρ being constant. The risk-neutral process for the short-term deviation χ_t is now an Ornstein-Uhlenbeck process which reverts to $-\lambda_\chi/\kappa$, while the process for equilibrium prices is still a geometric Brownian motion, but now its drift becomes $\mu^*_\xi \equiv \mu_\xi - \lambda_\xi$.

Let $F_{T,0}$ denote the current market price for a futures contract with time T until maturity. In the risk-neutral framework, futures prices are equal to the expected future spot price at maturity T. Moreover, assume that interest rates are deterministic, then futures prices are equal to forward prices. In Appendix 12, via using the Kolmogorov backward equation, the formulation for the futures price $F_{T,0}$ is derived as

$$\ln\left(F_{T,0}\right) = \ln\left(\mathbb{E}^*\left[S_T\right]\right) = e^{-\kappa T}\chi_0 + \xi_0 + A\left(T\right) \tag{3.10}$$

where

$$A\left(T\right) = \mu_\xi^* T - \left(1 - e^{-\kappa T}\right)\frac{\lambda_\chi}{\kappa} + \frac{1}{2}\left(\left(1 - e^{-2\kappa T}\right)\frac{\sigma_\chi^2}{2\kappa} + \sigma_\xi^2 T + 2\left(1 - e^{-\kappa T}\right)\frac{\rho\sigma_\chi\sigma_\xi}{\kappa}\right).$$

3.2.3 Model 3

Model 3 is a three-factor model of commodity contingent claims. This model was constructed in [22]. The stochastic factors in this model are the spot price of the commodity, the instantaneous convenience yield and the instantaneous interest rate. Here the instantaneous interest rate is assumed to follow an Ornstein-Uhlenbeck process. The dynamics for these factors under the real-world framework can be expressed as [22]

$$dS = \left(\mu - \delta\right)Sdt + \sigma_1 S dZ_1 \tag{3.11}$$

$$d\delta = \kappa\left(\alpha - \delta\right)dt + \sigma_2 dZ_2 \tag{3.12}$$

$$dr = a\left(m - r\right)dt + \sigma_3 dZ_3 \tag{3.13}$$

$$dZ_1 dZ_2 = \rho_1 dt, \quad dZ_2 dZ_3 = \rho_2 dt, \quad dZ_1 dZ_3 = \rho_3 dt. \tag{3.14}$$

where μ is the total expected return on the spot price, κ and a are the speed coefficients of mean reversion for the convenience yield and interest rate respectively, α is the convenience yield long-run mean, that is, the level to which δ reverts as t goes to infinity, dZ_1, dZ_2 and dZ_3 are increments of standard Wiener processes and are correlated with each other.

Let X = $\ln S$. Then from Ito's Lemma [13], the dynamic for the log of the spot price can be obtained as

$$dX = d\ln S = \left(\mu - \delta - \frac{1}{2}\sigma_1^2\right)dt + \sigma_1 dZ_1. \tag{3.15}$$

In order to value futures contracts and other commodity-related investments, model 3 can be transformed into risk-neutral stochastic processes as follows [22]

$$dS = \left(r - \delta\right)Sdt + \sigma_1 S dZ_1^* \tag{3.16}$$

$$d\delta = \kappa\left(\hat{\alpha} - \delta\right)dt + \sigma_2 dZ_2^* \tag{3.17}$$

$$dr = a\left(m^* - r\right)dt + \sigma_3 dZ_3^* \tag{3.18}$$

$$dZ_1^* dZ_2^* = \rho_1 dt, \quad dZ_2^* dZ_3^* = \rho_2 dt, \quad dZ_1^* dZ_3^* = \rho_3 dt. \tag{3.19}$$

where $\hat{\alpha} = \alpha - \frac{\lambda_\delta}{\kappa}$, $m^* = m - \frac{\lambda_r}{a}$, λ_δ is the market price of convenience yield risk, λ_r is the risk of the interest rate; a and m^* are respectively the speed of adjustment coefficient and the risk adjusted mean short rate of the interest rate process.

Here we note that in model 3, the interest rate is a stochastic process. Therefore, futures prices are not equal to forward prices. In Appendix 12, via using the Kolmogorov backward equation, the futures prices are derived as

$$F\left(S,\delta,r,T\right) = S\exp\left[\frac{-\delta\left(1-e^{-\kappa T}\right)}{\kappa} + \frac{r\left(1-e^{-\alpha T}\right)}{a} + C\left(T\right)\right]. \tag{3.20}$$

Or in the log form:

$$\ln F\left(S,\delta,r,T\right) = \ln S - \frac{-\delta\left(1-e^{-\kappa T}\right)}{\kappa} + \frac{r\left(1-e^{-\alpha T}\right)}{a} + C\left(T\right) \tag{3.21}$$

where

$$C\left(T\right) = \frac{\left(\kappa\hat{\alpha}+\sigma_1\sigma_2\rho_1\right)\left(\left(1-e^{-\kappa T}\right)-\kappa T\right)}{\kappa^2} - \frac{\sigma_2^2\left(4\left(1-e^{-\kappa T}\right)-\left(1-e^{-2\kappa T}\right)-2\kappa T\right)}{4\kappa^3}$$
$$- \frac{\left(am^*+\sigma_1\sigma_3\rho_3\right)\left(\left(1-e^{-aT}\right)-aT\right)}{a^2} - \frac{\sigma_3^2\left(4\left(1-e^{-aT}\right)-\left(1-e^{-2aT}\right)-2aT\right)}{4a^3}$$
$$+ \sigma_2\sigma_3\rho_2\left(\frac{\left(1-e^{-\kappa T}\right)+\left(1-e^{-aT}\right)-\left(1-e^{-(\kappa+a)T}\right)}{\kappa a(\kappa+a)} + \frac{\kappa^2\left(1-e^{-aT}\right)+a^2\left(1-e^{-\kappa T}\right)-\kappa a^2 T-a\kappa^2 T}{\kappa^2 a^2(\kappa+a)}\right).$$

Next we present a summary of the properties of these three proposed models.

3.3 Properties of models

3.3.1 Models 1 and 2

In this sub-section, we will examine and derive the equivalence between model 1 and model 2. To some extent, model 1 is in fact nested in model 2. Unlike most other recent models of commodity prices, model 2 does not consider convenience yields - defined in [2] as "the flow of services which accrues to the owner of a physical inventory but not to the owner of a contract for future delivery" even when valuing futures contracts or options on these futures. However, this model is exactly equivalent to the stochastic convenience yield model proposed in [9] in that the factors in each model can be expressed as linear combinations of the factors in the other. To see this equivalence, we first briefly describe the Gibson and Schwartz's model. We let X_t

denote the log of the time-t current spot price, and let δ_t stand for the time-t convenience yield. The stochastic convenience yield model assumes the joint stochastic process for these factors as follows

$$dX_t = \left(\mu - \delta_t - \frac{1}{2}\sigma_1^2\right) dt + \sigma_1 dZ_1 \tag{3.22}$$

$$d\delta_t = \kappa\left(\alpha - \delta_t\right) dt + \sigma_2 dZ_2 \tag{3.23}$$

where dZ_1 and dZ_2 are correlated increments of standard Brownian motion process with $dZ_1 dZ_2 = \rho dt$. As shown in [9], the linear relationship between the variables in these two models can be expressed as follows:

$$\chi_t = \frac{1}{\kappa}\left(\delta_t - \alpha\right) \tag{3.24}$$

$$\xi_t = X_t - \chi_t = X_t - \frac{1}{\kappa}\left(\delta_t - \alpha\right). \tag{3.25}$$

To establish the equivalence of the two models, we can write the dynamics for the state variables of the short-term/long-term model using the dynamics of the variables in the stochastic convenience yield model, and then relate the parameters in the two models:

$$d\chi_t = \frac{1}{\kappa} d\delta_t = \left(\alpha - \delta_t\right) dt + \frac{\sigma_2}{\kappa} dZ_2 \tag{3.26}$$

$$d\xi_t = dX_t - \frac{1}{\kappa} d\delta_t = \left(\mu - \alpha - \frac{1}{2}\sigma_1^2\right) dt + \sigma_1 dZ_1 - \frac{\sigma_2}{\kappa} dZ_2. \tag{3.27}$$

The relationships between parameters in the two models can be summarized in the following table [21]:

Short-term/long-term model	Definition in terms of stochastic convenience yield model
κ	κ
σ_χ	$\frac{\sigma_2}{\kappa}$
dZ_χ	dZ_2
μ_ξ	$\mu - \alpha - \frac{1}{2}\sigma_1^2$
σ_ξ	$\left(\sigma_1^2 + \frac{\sigma_2^2}{\kappa^2} - \frac{2\rho\sigma_1\sigma_2}{\kappa}\right)^{\frac{1}{2}}$
dZ_ξ	$\left(\sigma_1 dZ_1 - \frac{\sigma_2}{\kappa} dZ_2\right)\left(\sigma_1^2 + \frac{\sigma_2^2}{\kappa^2} - \frac{2\rho\sigma_1\sigma_2}{\kappa}\right)^{-\frac{1}{2}}$
ρ	$\left(\rho\sigma_1 - \frac{\sigma_2}{\kappa}\right)\left(\sigma_1^2 + \frac{\sigma_2^2}{\kappa^2} - \frac{2\rho\sigma_1\sigma_2}{\kappa}\right)^{-\frac{1}{2}}$
λ_χ	$\frac{\lambda}{\kappa}$
λ_ξ	$\mu - r - \frac{\lambda}{\kappa}$

Table 1: The relationship between parameters in the short-term/long-term model and the stochastic convenience model proposed in [9]

Hence, we have established the equivalence between model 2 and the stochastic convenience yield model developed in [9]. Now we note that the stochastic convenience yield model is an extension of model 1 since it allows the convenience yield factor to be stochastic. To some extent, this implies that model 1 is nested in model 2 and therefore it suffices to consider the properties of model 2 rather than those of model 1.

Now given χ_0 and ξ_0 as initial values of the short-term deviation and the equilibrium price respectively. It is shown in Appendix 13 that χ_t and ξ_t are jointly normally distributed with mean vector and covariance matrix:

$$\mathbb{E}\left[(\chi_t, \xi_t)\right] = \left[e^{-\kappa t}\chi_0, \xi_0 + \mu_\xi t\right] \quad \text{and} \tag{3.28}$$

$$\mathbb{C}ov\left[(\chi_t, \xi_t)\right] = \begin{bmatrix} \left(1 - e^{-2\kappa t}\right)\frac{\sigma_\chi^2}{2\kappa} & \left(1 - e^{-\kappa t}\right)\frac{\rho\sigma_\chi\sigma_\xi}{\kappa} \\ \left(1 - e^{-\kappa t}\right)\frac{\rho\sigma_\chi\sigma_\xi}{\kappa} & \sigma_\xi^2 t \end{bmatrix}. \tag{3.29}$$

The log of the future spot price is then normally distributed with

$$\mathbb{E}\left[\ln\left(S_t\right)\right] = e^{-\kappa t}\chi_0 + \xi_0 + \mu_\xi t \quad \text{and} \tag{3.30}$$

$$\mathbb{V}ar\left[\ln\left(S_t\right)\right] = \left(1 - e^{-2\kappa t}\right)\frac{\sigma_\chi^2}{2\kappa} + \sigma_\xi^2 t + 2\left(1 - e^{-\kappa t}\right)\frac{\rho\sigma_\chi\sigma_\xi}{\kappa}. \tag{3.31}$$

The spot price is then log-normally distributed with the expected price given by

$$\mathbb{E}\left[S_t\right] = \exp\left(\mathbb{E}\left[\ln\left(S_t\right)\right] + \frac{1}{2}\mathbb{V}ar\left[\ln\left(S_t\right)\right]\right) \tag{3.32}$$

or

$$\begin{aligned}\ln\left(\mathbb{E}\left[S_t\right]\right) &= \mathbb{E}\left[\ln\left(S_t\right)\right] + \frac{1}{2}\mathbb{V}ar\left[\ln\left(S_t\right)\right] \\ &= e^{-\kappa t}\chi_0 + \xi_0 + \mu_\xi t + \frac{1}{2}\left(\left(1-e^{-2\kappa t}\right)\frac{\sigma_\chi^2}{2\kappa} + \sigma_\xi^2 t + 2\left(1-e^{-\kappa t}\right)\frac{\rho\sigma_\chi\sigma_\xi}{\kappa}\right).\end{aligned} \tag{3.33}$$

As the time t tends to infinity, the terms $e^{-\kappa t}$ and $e^{-2\kappa t}$ approach zero. Thus, the log of the expected spot price approaches

$$\xi_0 + \frac{\sigma_\chi^2}{4\kappa} + \frac{\rho\sigma_\chi\sigma_\xi}{\kappa} + \left(\mu_\xi + \frac{1}{2}\sigma_\xi^2\right)t. \tag{3.34}$$

Therefore, in the long run, the expected spot prices act as if they started at an "effective long-run price" of $\exp(\xi_0 + \frac{\sigma_\chi^2}{4\kappa} + \frac{\rho\sigma_\chi\sigma_\xi}{\kappa})$ and increase at a rate of $(\mu_\xi + \frac{1}{2}\sigma_\xi^2)$. The difference between this effective long-run price and the equilibrium price $(\exp(\xi_0))$ reflects the contribution of the short-term volatility to the expected spot prices.

Furthermore, differentiate both sides of equation (3.10) with respect to T, we get

$$\frac{d\ln\left(F_{T,0}\right)}{dT} = -\kappa e^{-\kappa T}\chi_0 + \mu_\xi^* - e^{-\kappa T}\lambda_\chi + \frac{1}{2}e^{-2\kappa T}\sigma_\chi^2 + \frac{1}{2}\sigma_\xi^2 + 2e^{-\kappa T}\rho\sigma_\chi\sigma_\xi \tag{3.35}$$

. Here we see that the volatility of the price $F_{T,0}$ or the instantaneous variance of $\ln(F_{T,0})$ is given by $e^{-2\kappa T}\sigma_\chi^2 + \sigma_\xi^2 + 2e^{-\kappa T}\rho\sigma_\chi\sigma_\xi$. The volatility is therefore independent of the state variables.

For near maturity futures contracts (T = 0), the volatility is equal to the sum of the volatilities of the short- and long-term factors (i.e., $\sigma_\chi^2 + \sigma_\xi^2$). As the maturity of the contract increases, the $e^{-2\kappa T}$ and $e^{-\kappa T}$ terms tend to 0, and the instantaneous volatility approaches the volatility of the equilibrium price level (σ_ξ^2). This also implies that the short-term deviations make less of a contribution to the volatility as $T \to \infty$.

This short-term/long-term model appears to be natural and intuitive since it still keeps the mean reverting nature of several common commodity price series. Indeed, the short-run deviation χ_t is assumed to revert toward zero following an Ornstein-Uhlenbeck process. The mean-reversion coefficient κ describes the rate at which the short-term deviation are expected to vanish.

Another important point is that model 2 has normal distribution. Due to this property, the Kalman filter which we shall consider in section 8.3 yields optimal solution for the state space model of model 2.

3.3.2 Model 3

We first observe that in this model, the convenience yield and the interest rate are both Gaussian, but the spot price is not. However, the transformation process $\mathrm{X} = \ln S$ is Gaussian. In section 8.3, we shall see that the Kalman filter achieves optimal filtering solution for a state space model under Gaussian and linear assumptions on the state space model. Therefore, when the Kalman filter is applied to model 3, the log of the spot price rather than the original spot price is made use of to obtain optimal filtering result. However, when the Extended Kalman filter, the Unscented Kalman filter or the Particle filter (see sections 8.4 and 8.5) is applied, we no longer need to utilise the transformation process X since these filtering techniques do not require the Gaussian assumption on a state space model.

To investigate the term structure of futures prices in the three-factor model, we might consider the rate of change in the futures price. This rate can be obtained by taking derivative of the futures price $F(S, \delta, r, T)$ with respect to time to maturity, dividing by the price, and taking the limit as time to maturity goes to infinity. In [22], this rate is shown to be

$$\frac{1}{F}\frac{\partial F}{\partial T}(T \to \infty) = m^* - \hat{\alpha} + \frac{\sigma_2^2}{2\kappa^2} - \frac{\rho_1\sigma_1\sigma_2}{\kappa} + \frac{\sigma_3^2}{2a^2} + \frac{\rho_3\sigma_1\sigma_3}{a} - \frac{\rho_2\sigma_2\sigma_3}{\kappa a}. \tag{3.36}$$

We interpret this to show that the term structure of futures prices will eventually turn upward and converge to a fixed rate of growth even if initially it is in strong backwardation. Otherwise, it will be decreasing over time and converge to a fixed rate if initially it is in contango.

In [22], Schwartz also shows that model 3 fits well the term structures of futures prices (for copper and oil commodities) with maturity less than 2 years. However, the model does a poor job when predicting longer term futures prices. A surprise arises in the term structure of the volatilities of futures returns as we shall see in the following.

Indeed, for this model, the volatility of futures return is shown in [22] as

$$\begin{aligned}\sigma_F^2(T) = \sigma_1^2 + \sigma_2^2\frac{\left(1-e^{-\kappa T}\right)^2}{\kappa^2} + \sigma_3^2\frac{\left(1-e^{-aT}\right)^2}{a^2} - 2\rho_1\sigma_1\sigma_2\frac{\left(1-e^{-\kappa T}\right)}{\kappa} + 2\rho_3\sigma_1\sigma_3\frac{\left(1-e^{-aT}\right)}{a} \\ - 2\rho_2\sigma_2\sigma_3\frac{\left(1-e^{-aT}\right)\left(1-e^{-\kappa T}\right)}{a\kappa}.\end{aligned} \tag{3.37}$$

A very important feature of this model is that we can see that the volatility is independent of the state variables and only depends on the time to maturity of the futures contracts. As the time to maturity tends to infinity, the volatility converges to

$$\sigma_F^2(\infty) = \sigma_1^2 + \frac{\sigma_2^2}{\kappa^2} + \frac{\sigma_3^2}{a^2} - \frac{2\rho_1\sigma_1\sigma_2}{\kappa} + \frac{2\rho_3\sigma_1\sigma_3}{a} - \frac{2\rho_2\sigma_2\sigma_3}{a\kappa}. \tag{3.38}$$

Thus the volatility of model 3 converges to a fixed value as maturity increases. However, as shown in [22], it can fit very well the volatility of the futures market data (including oil, copper and Enron oil data) for both short-term and long-term contracts. This contrasts to the poor fit ability of the futures prices implied by the model when dealing with long-term contracts.

4 Discretization of models and state space framework

Discretization is an indispensable approximation tool for simulating or estimating continuous processes. Since a continuous process contains infinitely many points, it is impossible to capture all these points in a real computer system. It is also therefore required when one wishes to perform estimation and filtering to first perform discretization. The discretization method chosen sorts out this problem. In theory, a discretized process converges to the real process as the time step tends to zero. Thus, the number of points of the real process can be reduced to a finite set so that they can be stored in a real computer system, and therefore the true process can be simulated in this manner under a discrete approximation. Furthermore, discretization can also be applied to simulate a complex system of continuous processes in which it may be very difficult to obtain analytic forms for each process in the system (model 3, for example). In this section, we present two common discretization schemes for multivariate stochastic processes together with a model representation under the state space framework. The first discretization scheme is known as the Euler scheme, and the second one is the Milstein scheme.
We first present the i-th component of a general n-dimensional stochastic differential equation (SDE), with m-dimensional Wiener process as,

$$dX_t^i = a^i\left(t, X_t\right) dt + \sum_{j=1}^{m} b^{i,j}\left(t, X_t\right) dW_t^j. \tag{4.1}$$

4.1 The Euler scheme

Throughout the remainder of this section we will adopt here the operator notation of [14]. In this book, page 341, the authors describe the i-th component of the Euler scheme under the Ito integrals as

$$X_t^i = X_{t-1}^i + a^i\left(t-1, X_{t-1}\right) \triangle t + \sum_{j=1}^{m} b^{i,j}\left(t-1, X_{t-1}\right) \triangle W_{t-1}^j \tag{4.2}$$

where

$$\triangle t = \tau_t - \tau_{t-1}$$

is the length of the time discretization subinterval $[\tau_{t-1}, \tau_t]$ and

$$\triangle W_{t-1}^j = W_{\tau_t}^j - W_{\tau_{t-1}}^j$$

is the $N(0; \triangle t)$ distributed increment of the jth component of the Wiener process W on $[\tau_{t-1}, \tau_t]$, and $\triangle W_{t-1}^{j_1}$ and $\triangle W_{t-1}^{j_2}$ are independent for $j_1 \neq j_2$.

We now use the Euler scheme to discretize the three models above.

4.1.1 Model 1: Euler Discretization

Model 1 is just a simple one-factor model and it can be discretized by the Euler scheme as

$$S_t = S_{t-1} + (r - c) S_{t-1} \triangle t + \sigma S_{t-1} \sqrt{\triangle t} n_{S,t-1} \tag{4.3}$$

where $n_{S,t-1}$ is a standard normal random variable.

4.1.2 Model 2: Euler Discretization

This model is first recast with respect to independent Wiener processes dW_1 and dW_2 as follows

$$d\chi_t = -\kappa \chi_t dt + \sigma_\chi dW_1 \tag{4.4}$$

$$d\xi_t = \mu_\xi dt + \sigma_\xi \rho dW_1 + \sigma_\xi \sqrt{1 - \rho^2} dW_2. \tag{4.5}$$

This leads to the following bivariate Euler discretization scheme:

$$\chi_t = \chi_{t-1} - \kappa \chi_{t-1} \triangle t + \sigma_\chi \sqrt{\triangle t} n_{\chi,t-1} = (1 - \kappa \triangle t) \chi_{t-1} + \sigma_\chi \sqrt{\triangle t} n_{\chi,t-1} \tag{4.6}$$

$$\xi_t = \xi_{t-1} + \mu_\xi \triangle t + \sigma_\xi \rho \sqrt{\triangle t} n_{\chi,t-1} + \sigma_\xi \sqrt{1 - \rho^2} \sqrt{\triangle t} n_{\xi,t-1} \tag{4.7}$$

where $n_{\chi,t-1}$ and $n_{\xi,t-1}$ are i.i.d. standard normal random variables.

4.1.3 Model 3: Euler Discretization

Model 3 is first recast with respect to independent Wiener processes dW_1, dW_2 and dW_3 as follows

$$dS_t = (\mu - \delta_t) S_t dt + \sigma_1 S_t dW_1 \tag{4.8}$$

$$d\delta_t = \kappa (\alpha - \delta_t) dt + \sigma_2 \left(\rho_1 dW_1 + \sqrt{1 - \rho_1^2} dW_2 \right) \tag{4.9}$$

$$dr_t = a (m - r_t) dt + \sigma_3 \left(\rho_3 dW_1 + \sqrt{1 - \rho_3^2} dW_3 \right). \tag{4.10}$$

This will result in the following specifications, (n=m=3):

$$a^1 (t, S_t) = (\mu - \delta_t) S_t; a^2 (t, \delta_t) = \kappa (\alpha - \delta_t); a^3 (t, r_t) = a (m - r_t)$$

$$b^{1,1} (t, S_t) = \sigma_1 S_t; b^{1,2} (t, S_t) = 0; b^{1,3} (t, S_t) = 0$$

$$b^{2,1} (t, \delta_t) = \sigma_2 \rho_1; b^{2,2} (t, \delta_t) = \sigma_2 \sqrt{1 - \rho_1^2}; b^{2,3} (t, \delta_t) = 0$$

$$b^{3,1} (t, r_t) = \sigma_3 \rho_3; b^{3,2} (t, r_t) = 0; b^{3,3} (t, r_t) = \sigma_3 \sqrt{1 - \rho_3^2}.$$

The trivariate Euler discretization scheme for model 3 can then easily be obtained as follows

$$S_t = S_{t-1} + (\mu - \delta_{t-1})\, S_{t-1} \triangle t + \sigma_1 S_{t-1} \sqrt{\triangle t} n_{S,t-1} \tag{4.11}$$

$$\delta_t = \delta_{t-1} + \kappa\,(\alpha - \delta_{t-1})\, \triangle t + \sigma_2 \rho_1 \sqrt{\triangle t} n_{S,t-1} + \sigma_2 \sqrt{1-\rho_1^2}\sqrt{\triangle t} n_{\delta,t-1} \tag{4.12}$$

$$r_t = r_{t-1} + a\,(m - r_{t-1})\, \triangle t + \sigma_3 \rho_3 \sqrt{\triangle t} n_{S,t-1} + \sigma_3 \sqrt{1-\rho_3^2}\sqrt{\triangle t} n_{r,t-1} \tag{4.13}$$

where $n_{S,t-1}$, $n_{\delta,t-1}$ and $n_{r,t-1}$ are i.i.d. standard normal random variables.

4.2 The Milstein scheme

The i-th component of the Milstein scheme under the Ito integrals is given in [14] (page 346) as follows

$$\begin{aligned} X_t^i = X_{t-1}^i &+ a^i\,(t-1, X_{t-1})\, \triangle t + \sum_{j=1}^{m} b^{i,j}\,(t-1, X_{t-1})\, \triangle W_{t-1}^j \\ &+ \sum_{j_1,j_2}^{m} L^{j_1} b^{i,j_2}\,(t-1, X_{t-1})\, I_{(j_1,j_2)\triangle t} \end{aligned} \tag{4.14}$$

where, $L^j = \sum_{i=1}^{n} b^{i,j} \frac{\partial}{\partial x_i}$ and the Ito multiple integral $I_{(j_1,j_1)\triangle t}$ is given by,

$$I_{(j_1,j_2)\triangle t} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_1} dW_{s_1}^{j_1} dW_{s_1}^{j_2}. \tag{4.15}$$

These integrals have the useful properties that,

$$I_{(j_1,j_1)} = \frac{1}{2}\{\left(\triangle W^{j_1}\right)^2 - \triangle t\}, \quad \text{and} \quad J_{(j_1,j_1)} = \frac{1}{2}\left(\triangle W^{j_1}\right)^2. \tag{4.16}$$

As pointed out in [14], when $j_1 \neq j_2$ the Ito and Stratonovich integrals are equal,

$$I_{(j_1,j_2)} = J_{(j_1,j_2)} = \int_{\tau_n}^{\tau_{n+1}} \int_{\tau_n}^{s_1} dW_{s_2}^{j_1} dW_{s_1}^{j_2}. \tag{4.17}$$

It is further argued in [14] that it will be easier to obtain the approximation under the Stratonovich integrals. This then results in a p-th order truncation given by

$$\begin{aligned} J_{(j_1,j_2)}^p = &\triangle t \left(\frac{1}{2}\zeta_{j_1}\zeta_{j_2} + \sqrt{\rho_p}\left(\mu_{j_1,p}\zeta_{j_2} - \mu_{j_2,p}\zeta_{j_1}\right)\right) \\ &+ \frac{\triangle t}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\left(\psi_{j_1,r}\left(\sqrt{2}\zeta_{j_2} + \nu_{j_2,r}\right) - \psi_{j_2,r}\left(\sqrt{2}\zeta_{j_1} + \nu_{j_1,r}\right)\right), \end{aligned} \tag{4.18}$$

where $\zeta_j, \mu_{j,p}, \nu_{j,r}$ and $\psi_{j,r}$ are all independent $\mathcal{N}(0;1)$ Gaussian random variables with,

$$\rho_p = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{p}\frac{1}{r^2}, \tag{4.19}$$

$$\zeta_j = \frac{1}{\sqrt{\triangle t}} \triangle W^j. \tag{4.20}$$

Here a recommendation for p is provided in [14] where the authors suggest $p \geq \frac{K}{\triangle t}$ for some positive constant K.

We now use the Milstein scheme to discretize the three models as follows

4.2.1 Model 1: Milstein Discretization

In this model, we have:

$$a = (r - c)\, S \quad \text{and} \quad b = \sigma S.$$

Model 1 can be discretized by the Milstein scheme as

$$\begin{aligned} S_t &= S_{t-1} + a\triangle t + b\triangle W + \frac{1}{2} bb' \left\{ (\triangle W)^2 - \triangle t \right\} \\ &= S_{t-1} + (r - c)\, S_{t-1} \triangle t + \sigma S_{t-1} \sqrt{\triangle t} n_{S,t-1} + \frac{1}{2} \sigma^2 S \left(\triangle t . n_{S,t-1}^2 - \triangle t \right) \end{aligned} \tag{4.21}$$

where $n_{S,t-1}$ is a standard normal random variable.

4.2.2 Model 2: Milstein Discretization

This model is first recast with respect to independent Wiener processes dW_1 and dW_2 as follows

$$d\chi_t = -\kappa \chi_t dt + \sigma_\chi dW_1 \tag{4.22}$$

$$d\xi_t = \mu_\xi dt + \sigma_\xi \rho dW_1 + \sigma_\xi \sqrt{1 - \rho^2} dW_2 \tag{4.23}$$

where $b^{1,1} = \sigma_\chi$, $b^{1,2} = 0$, $b^{2,1} = \sigma_\xi \rho$ and $b^{2,2} = \sigma_\xi \sqrt{1 - \rho^2}$.
Since $b^{1,1}$, $b^{1,2}$, $b^{2,1}$ and $b^{2,2}$ are constants, this follows directly that $L^1 b^{1,1}$, $L^2 b^{1,1}$, $L^1 b^{1,2}$, $L^2 b^{1,2}$, $L^1 b^{2,1}$, $L^2 b^{2,1}$, $L^1 b^{2,2}$ and $L^2 b^{2,2}$ are all equal to 0.
This leads to the following bivariate Milstein discretization scheme:

$$\chi_t = \chi_{t-1} - \kappa \chi_{t-1} \triangle t + \sigma_\chi \sqrt{\triangle t} n_{\chi,t-1} = (1 - \kappa \triangle t)\, \chi_{t-1} + \sigma_\chi \sqrt{\triangle t} n_{\chi,t-1} \tag{4.24}$$

$$\xi_t = \xi_{t-1} + \mu_\xi \triangle t + \sigma_\xi \rho \sqrt{\triangle t} n_{\chi,t-1} + \sigma_\xi \sqrt{1 - \rho^2} \sqrt{\triangle t} n_{\xi,t-1} \tag{4.25}$$

where $n_{\chi,t-1}$ and $n_{\xi,t-1}$ are i.i.d. standard normal random variables.

4.2.3 Model 3: Milstein Discretization

The discretization for model 3 can be found in Appendix 14 as

$$S_t = S_{t-1} + (\mu - \delta_{t-1}) S_{t-1} \triangle t + \sigma_1 S_{t-1} \sqrt{\triangle t} n_{S,t-1} + \sigma_1^2 S_{t-1} \frac{1}{2} \left(\triangle t n_{S,t-1}^2 - \triangle t \right) \tag{4.26}$$

$$\delta_t = \delta_{t-1} + \kappa (\alpha - \delta_{t-1}) \triangle t + \sigma_2 \rho_1 \sqrt{\triangle t} n_{S,t-1} + \sigma_2 \sqrt{1 - \rho_1^2} \sqrt{\triangle t} n_{\delta,t-1} \tag{4.27}$$

$$r_t = r_{t-1} + a (m - r_{t-1}) \triangle t + \sigma_3 \rho_3 \sqrt{\triangle t} n_{S,t-1} + \sigma_3 \sqrt{1 - \rho_3^2} \sqrt{\triangle t} n_{r,t-1} \tag{4.28}$$

where $n_{S,t-1}$, $n_{\delta,t-1}$ and $n_{r,t-1}$ are i.i.d. standard normal random variables.

5 Theoretical and numerical studies of Euler vs Milstein schemes

5.1 Theoretical results and discussion

Consider the i-th component of a general n-dimensional SDE, with m-dimensional Wiener process as follows

$$dX_t^i = a^i\left(t, X_t\right) dt + \sum_{j=1}^{m} b^{i,j}\left(t, X_t\right) dW_t^j. \tag{5.1}$$

where $a^i(t, X_t)$ and $b^{i,j}(t, X_t)$ are respectively the drift and the volatility coefficients of the process X.

In this sub-section, we investigate the rate of convergence of the Euler and Milstein schemes under certain conditions. The two following theorems provide necessary and sufficient conditions to ensure that the Euler and Milstein schemes each converge. These theorems together with their proofs can be found in [14].

5.1.1 Theorem 1 - Conditions for the convergence of the Euler scheme

Let X_0 be the initial state of the true process, $\triangle$ be the time interval and $Y_0^{\triangle}$ be the initial state of the simulation process generated by the Euler scheme. Suppose that

$$\mathbb{E}\left[|X_0|^2\right] < \infty \tag{5.2}$$

$$\mathbb{E}\left[\left|X_0 - Y_0^{\triangle}\right|^2\right]^{\frac{1}{2}} \leq K_1 \triangle^{\frac{1}{2}} \tag{5.3}$$

$$|a\left(t,x\right) - a\left(t,y\right)| + |b\left(t,x\right) - b\left(t,y\right)| \leq K_2\left|x-y\right| \tag{5.4}$$

$$|a\left(t,x\right)| + |b\left(t,x\right)| \leq K_3\left(1+|x|\right) \tag{5.5}$$

$$|a\left(s,x\right) - a\left(t,x\right)| + |b\left(s,x\right) - b\left(t,x\right)| \leq K_4\left(1+|x|\right)|s-t|^{\frac{1}{2}} \tag{5.6}$$

for all $s, t \in [0, T]$ and $x, y \in \mathcal{R}^d$, where the constants $K_1, \ldots, K_4$ do not depend on the time step $\triangle$. Then, for the Euler approximation $Y^{\triangle}$, the estimate

$$\mathbb{E}\left[\left|X_T - Y^{\triangle}\left(T\right)\right|\right] \leq K_5 \triangle^{\frac{1}{2}} \tag{5.7}$$

holds, where the constant K_5 does not depend on $\triangle$.

Remark 1

This theorem shows that one may bound the discretization error between the true process at time

T and the discretization as a function of a constant and the discretization interval. Therefore, one can obtain as accurate solution as required by a reduction of $\triangle$ only and no transform of the process required. The conditions for this to hold are:

The first condition (5.2) of theorem 1 implies that the initial state of the true process X must be finite in the mean square sense. The condition (5.3) indicates that the initial state of the simulation $Y^{\triangle}$ must be chosen such that the square root of the mean square error between X_0 and $Y_0^{\triangle}$ is bounded by $K_1 \triangle^{\frac{1}{2}}$. To some extent, (5.3) means that we must choose the initial state of the simulation such that the "difference" between it and the initial true state is small enough and bounded by a given $\triangle$. The Lipschitz condition (5.4) implies that the drift and the diffusion are differentiable everywhere in $\mathcal{R}^d$ for any $s, t \in [0, T]$. To some extent, (5.4) also guarantees the continuity of the drift and the diffusion coefficients in terms of their second component. The linear growth condition (5.5) implies that the growths of the drift $a(t, X_t)$ and the diffusion $b(t, X_t)$ must be bounded by the linear growth of $K_3(1 + |x|)$.

The three conditions (5.2), (5.4) and (5.5) follow the existence and uniqueness of a strong solution for the SDE (5.1) (see, for example, chapter 4 in [13]). The last condition (5.6) is a Hölder condition of order $\frac{1}{2}$ (for a fixed x in $\mathcal{R}^d$). Recall that a real or complex-valued function f on d-dimensional Euclidean space satisfies a Hölder condition, or is Hölder continuous, when there are nonnegative real constants C, α, such that $|f(x) - f(y)| \leq C|x - y|^{\alpha}$ for all x and y in the domain of f. Moreover, every Hölder continuous function is uniformly continuous. Therefore, the condition (5.6) guarantees the continuity and differentiability of the drift and diffusion in terms of their first component.

5.1.2 Theorem 2 - Conditions for the convergence of the Milstein scheme

Let X_0 be the initial state of the true process, $\triangle$ be the time interval and $Y_0^{\triangle}$ be the initial state of the simulation process generated by the Milstein scheme. Suppose that

$$\mathbb{E}\left[|X_0|^2\right] < \infty \tag{5.8}$$

$$\mathbb{E}\left[\left|X_0 - Y_0^{\triangle}\right|^2\right]^{\frac{1}{2}} \leq K_1 \triangle^{\frac{1}{2}} \tag{5.9}$$

$$|\underline{a}(t,x) - \underline{a}(t,y)| \leq K_2 |x-y| \tag{5.10}$$

$$\left|b^{j_1}(t,x) - b^{j_1}(t,y)\right| \leq K_2 |x-y| \tag{5.11}$$

$$\left|\underline{L}^{j_1} b^{j_2}(t,x) - \underline{L}^{j_1} b^{j_2}(t,y)\right| \leq K_2 |x-y| \tag{5.12}$$

$$|\underline{a}(t,x)| + \left|\underline{L}^{j} \underline{a}(t,x)\right| \leq K_3 (1+|x|) \tag{5.13}$$

$$\left|b^{j_1}(t,x)\right| + \left|\underline{L}^{j} b^{j_2}(t,x)\right| \leq K_3 (1+|x|) \tag{5.14}$$

$$\left|\underline{L}^{j} \underline{L}^{j_1} b^{j_2}(t,x)\right| \leq K_3 (1+|x|) \tag{5.15}$$

$$|\underline{a}(s,x) - \underline{a}(t,x)| \leq K_4 (1+|x|) |s-t|^{\frac{1}{2}} \tag{5.16}$$

$$\left|b^{j_1}(s,x) - b^{j_1}(t,x)\right| \leq K_4 (1+|x|) |s-t|^{\frac{1}{2}} \tag{5.17}$$

$$\left|\underline{L}^{j_1} b^{j_2}(s,x) - \underline{L}^{j_1} b^{j_2}(t,x)\right| \leq K_4 (1+|x|) |s-t|^{\frac{1}{2}} \tag{5.18}$$

$$\text{where } \underline{a} = a - \frac{1}{2} b b' \text{ and } \underline{L}^{j} = \sum_{k=1}^{d} b^{k,j} \frac{\partial}{\partial x^k}$$

for all $s,t \in [0,T]$ and $x,y \in \mathcal{R}^d$, $j = 0,...,m$ and $j_1,\ j_2 = 1,...,m$, where the constants $K_1,\ldots,K_4$ do not depend on $\triangle$. Then, for the Milstein approximation $Y^{\triangle}$, the estimate

$$\mathbb{E}\left[\left|X_T - Y^{\triangle}(T)\right|\right] \leq K_5 \triangle \tag{5.19}$$

holds, where the constant K_5 does not depend on $\triangle$.

Remark 2

Similarly to theorem 1, theorem 2 reveals that one may bound the discretization error and the discretization at maturity T as a function of a constant and the discretization interval. Hence, one can achieve as accurate solution as required by a reduction of the time interval $\triangle$ only. The conditions for this to hold are:

The conditions (5.8) and (5.10)-(5.15) are to guarantee the existence and uniqueness of the solution for the SDE (5.1) (see chapter 4 in [13]). The condition (5.9) indicates that the initial state of the simulation $Y^{\triangle}$ must be chosen such that the square root of the mean square error

between X_0 and $Y_0^{\triangle}$ is bounded by $K_1 \triangle^{\frac{1}{2}}$. To some extent, (5.9) means that we must choose the initial state of the simulation such that the "difference" between it and the initial true state is small enough and bounded by a given $\triangle$. The Lipschitz conditions (5.10)-(5.12) imply that the drift and the diffusion are differentiable everywhere in $\mathcal{R}^d$ for any $s, t \in [0, T]$. Hence, (5.10)-(5.12) also guarantee the continuity of the drift and the diffusion coefficients. The linear growth conditions (5.13)-(5.15) imply that the growths of the drift a(t, x) and the diffusion b(t, x) must be bounded by a linear growth of $K_3(1 + |x|)$. The last three conditions (5.16)-(5.18) are Hölder conditions of order $\frac{1}{2}$ (for a fixed x in $\mathcal{R}^d$). These three conditions guarantee the continuity and differentiability of the drift and diffusion coefficients in terms of their first component.

5.1.3 Discussion of theoretical comparison between Euler and Milstein schemes

To understand the difference between these two schemes, we might look at the results provided by each scheme. As $\triangle < 1$, if we want to improve the accuracy of the simulation, then we must reduce the time discretization step $\triangle$ to less than 100 times for the Euler scheme, whereas we only need to reduce it 10 times for the Milstein scheme. This implies that the solution by the Milstein scheme converges to the truth faster than the Euler scheme as $\triangle < 1$. This provides a significant computational saving. However, as $\triangle \geq 1$, then we note that $\triangle^{\frac{1}{2}} < \triangle$. This indicates that, the difference between the simulation and the truth is bounded by a smaller lower bound for the Euler scheme than for the Milstein scheme. Therefore, in some sense, this implies that when $\triangle \geq 1$, the Milstein scheme might produce worse results than the counterpart.

Under some Lipschitz and linear growth conditions on the coefficients of the drift and the diffusion given in the two above theorems, the Euler scheme attains the order of strong convergence $\gamma = 0.5$, whereas the Milstein scheme attains the order of strong convergence $\gamma = 1.0$. However, in special cases, the Euler scheme may actually achieve a higher order of strong convergence. For instance, when the noise is additive, that is when the diffusion coefficient has the form $b(t, x) \equiv b(t)$. This actually happens in the case of model 2 since $b_\chi = \sigma_\chi$ and $b_\xi = \sigma_\xi$. Hence we might expect that the Euler scheme may achieve higher order (rather than 0.5) when applied to model 2. Moreover, under appropriate smoothness assumptions on the drift and diffusion terms, it may turn out that the Euler scheme has order of strong convergence $\gamma = 1.0$.

The Milstein scheme is an extension of the Euler scheme by simply adding one more term. The strong order $\gamma = 1.0$ of the Milstein scheme corresponds to that of the Euler scheme in the deterministic case without any noise, that is with $b \equiv 0$. In this sense, we may regard the Milstein

scheme as a proper generalization of the deterministic Euler scheme for the strong convergence criterion since it gives the same order of strong convergence as for the deterministic case. In many practical problems, the diffusion coefficients may have special structures which allow the Milstein scheme to be simplified in such a way that avoids the use of double stochastic integrals involving different components of the Wiener process. For instance, with additive noise, the diffusion coefficients depend at most on time t and not on the x variable and the Milstein scheme reduces to the Euler scheme, which involves no double stochastic integrals. Moreover, model 3 is a typical example where its diffusion coefficients have special structure which do not require the computation of a Stratonovich integral given as in equation (4.18). Particularly, since the diffusions in the convenience yield and interest rate factors (model 3) are constant, it turns out that Milstein and Euler scheme produce the same result for these two factors. In addition, the diffusion of the spot price in model 3 is just a linear function of the spot price, it therefore simplifies the Milstein scheme to the case where the Stratonovich integrals disappear (see equation 4.26).

5.2 Simulation

In this sub-section, we examine and compare the Euler and Milstein schemes in terms of their convergence to the true process. There have been several criteria to judge the quality of a discretization scheme. For example, in [17], the authors use the bias and root mean square error (RMSE) criteria to appreciate and compare the accuracy of some schemes. In this analysis we will consider several comparative criteria, the first involves the ideas discussed in [17] regarding the root mean square error. In particular we will consider the averaged root mean square error, where we average over the sample paths generated for a given discretization interval and we also average over the discrete time steps to get an estimate of the average total error in the analysis. This measure provides a mix of the Monte Carlo error from simulating trajectories and the bias due to the discretization. We will also study the properties of strong convergence and weak convergence of the discretization schemes. For the strong convergence we compare the estimated empirical distribution to the known true distribution under a Kullback-Leibler divergence for the strong convergence analysis. For the weak convergence we analyse the first and second moment estimates and compare them to the true mean and variance of the process that we derived analytically for different maturities, in addition we also consider the skewness. To demonstrate the trajectories that we will simulate utilising these model parameters we first

provide the following plots in figures 1 and 2 for the three factor model with different discretization intervals. For both these plots, we use the same time period [0, T = 1]. Particularly, for the first plot, we choose $\triangle = 0.001$ and use 1000 simulation paths. For the second plot, we take $\triangle = 0.1$ and use 100 paths. Moreover, to simulate the means for the three factors in model 3, we choose $\triangle = 10^{-4}$ (since as the time interval is sufficiently small, the simulation almost does not distinguish with the truth). To generate the trajectories for model 3, we utilise the following parameters

κ	$\hat{\alpha}$	a	m^*	σ_1	σ_2	σ_3	ρ_1	ρ_2	ρ_3
0.3	1	0.18	0.76	0.25	0.15	0.1	0.24	0.3	0.08

Table 2: Parameters for model 3

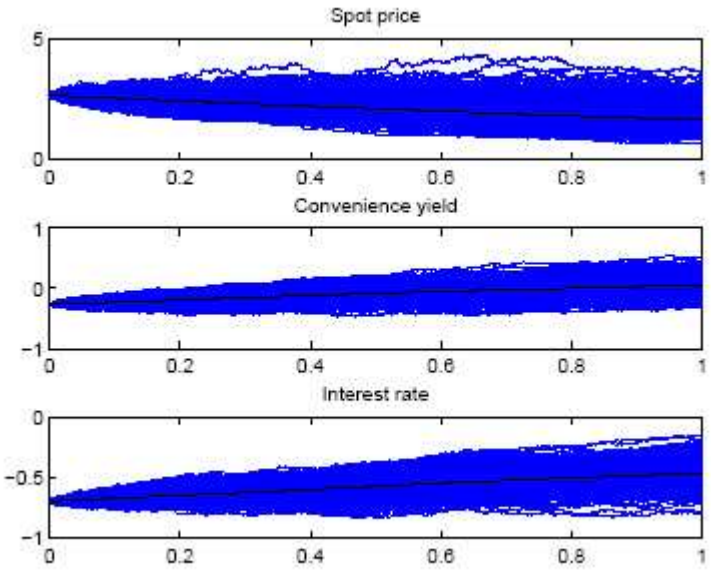

Figure 1: Simulation for model 3 using $\triangle = 0.001$ and 1000 simulation paths

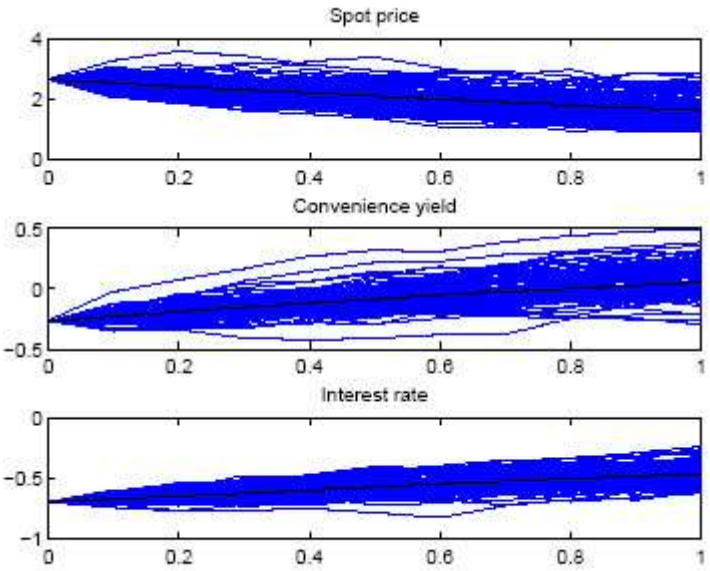

Figure 2: Simulation for model 3 using $\triangle = 0.1$ and 100 simulation paths

For figures 1 and 2, the blue lines stand for the simulation paths, whereas the black line represents the true mean of each factor. It should be noted here that the mean of 1000 simulation paths using a small $\triangle$ (0.001) appears to be closer to the true mean than that of 100 paths using a large $\triangle$ (0.1). In some sense, this reveals that the simulation achieves better results as we use more sample paths and a smaller time interval $\triangle$.

5.2.1 Strong Convergence Analysis

In this section we simulated 10,000 sample paths for maturities or time horizons of $T \in \{5, 10, 25, 50\}$, for a range of different time discretizations, $\triangle \in \{0.25, 0.5, 0.75, \ldots, 5\}$. We then consider the information theoretic concept of the Kullback-Leibler divergence [15] between the empirical histogram estimate of the sample paths at each maturity T versus the true distribution of the process at time T. We denote the empirical distributional estimate for each factor at time $t = T$ for a given discretization $\triangle$ by,

$$\widehat{p}_{\triangle}(t = T) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[X_i \leq x\right] \tag{5.20}$$

where N is the total number of sample paths generated, $\mathbb{I}\left[X_i(t = T) \leq x\right]$ is an indicator function which is one when the i-th path at time $t = T$ is less than x, producing the ECDF (estimated cumulative distribution function) estimate at time t. This can then be plugged into the Kullback-

Leibler divergence [15], which provides a non-symmetric measure of the difference between two probability distributions and is given by

$$D_{KL}(q(x(t)||p(x(t)) = \int_{-\infty}^{\infty} q(x(t)) \ln \frac{q(x(t))}{p(x(t))} dx(t). \tag{5.21}$$

Hence, if we substitute the ECDF for q we get the empirical KL divergence given by

$$D_{KL}(\widehat{p}_{\triangle}(X(t=T))||p(X(t=T)) = \sum_i \widehat{p}_{\triangle}(X(t=T)) \ln \frac{\widehat{p}_{\triangle}(X(t=T))}{p(X(t=T))} \tag{5.22}$$

and so we can measure the empirical "distance" between the true distribution and the estimated ECDF providing an analysis for different $\triangle$ of the strong convergence properties of a discretization scheme.

In the case of the two factor model we derived the distribution of the process at each maturity T analytically, since it is a Gaussian distribution and we can obtain explicitly the sufficient statistics as a function of t. Hence we can calculate exactly the empirical KL divergence, and in the case of the three factor model we must estimate the KL divergence empirically, where we assume that the true distribution is approximated exactly with a "true" distribution for the spot price factor estimated taking $\triangle = 10^{-5}$.
For this study, we utilise the following model parameters given in Tables 3 and 4.

κ	μ	σ_1	σ_2	ρ
0.02	0.11	0.44	0.22	0.15

Table 3: Parameters for model 2

κ	$\hat{\alpha}$	a	m^*	σ_1	σ_2	σ_3	ρ_1	ρ_2	ρ_3
0.4	0.81	0.28	0.7	0.17	0.15	0.08	0.24	0.3	0.08

Table 4: Parameters for model 3

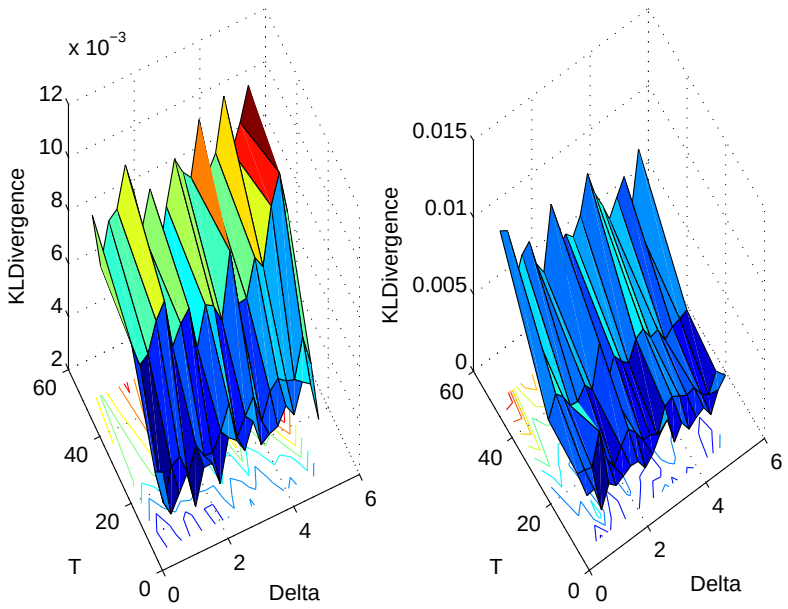

Figure 3: Model 2 - KL divergence for different maturities T in $\{5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.25, 0.5, 0.75, \ldots, 5\}$

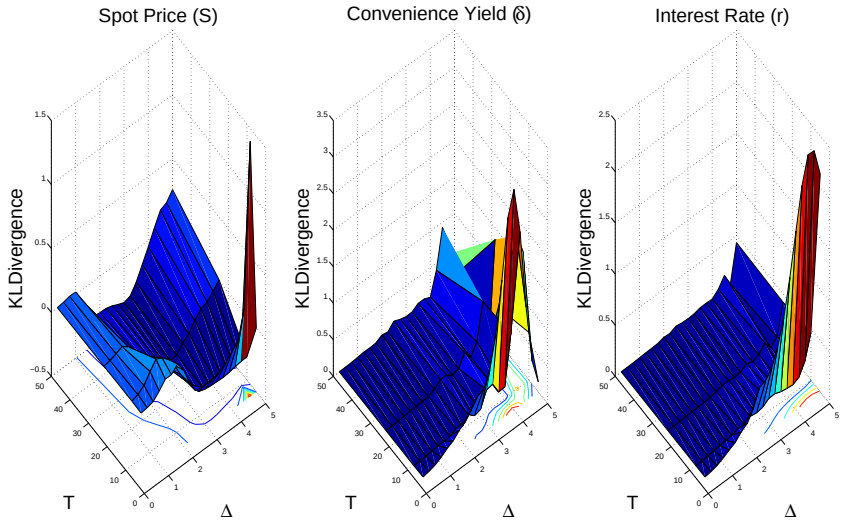

Figure 4: Model 3 - KL divergence for different maturities T in $\{5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.25, 0.5, 0.75, \ldots, 5\}$

From figures 3 and 4, we can observe that the KL divergence (for both model 2 and 3) appears to be increasing as the maturity T increases. This implies that when approximating the true process for a short maturity, the discretization schemes (Euler and Milstein) achieve very good result in the sense of distribution. Particularly, as the KL divergence is small, it indicates that the ECDF obtained by a simulation scheme almost does match the true distribution. Otherwise, for a large KL divergence, it implies a much less accuracy in the ECDF estimate. In addition, these figures also reveal a much uncertainty in the KL divergence as the time interval $\triangle$ is large. Specifically, for a small $\triangle$ from 0.25 to around 2.5, the KL divergence appears to be small and quite stable. But after that, it increases quickly up to $\triangle = 5$. This fact implies that the KL divergence is also affected by the time interval $\triangle$. To conclude, a simulation scheme achieves a good result in terms of distribution for a short maturity and a small time interval. Its result gets worse when either the maturity or the time interval is expanded, and the worse case happens when both the maturity and time interval become very large.

5.2.2 Weak Convergence Analysis

Recall that a time discrete approximation $Y^{\triangle}$ corresponding to a time interval $\triangle$ converges weakly to the truth X at time T as $\triangle \downarrow 0$ with respect to a class $\mathcal{C}$ of test functions $g : \mathcal{R}^d \to \mathcal{R}$ if we have

$$\lim_{\triangle \downarrow 0} \left| \mathbb{E}[g(X_T)] - \mathbb{E}[g(Y^{\triangle}(T))] \right| = 0 \tag{5.23}$$

for all $g \in \mathcal{C}$. If $\mathcal{C}$ contains all polynomials, then this definition reveals the convergence of all moments. Hence, for theoretical investigations in this aspect will require the existence of the moments.

In this section we simulated 10,000 sample paths for maturities or time horizons of $T \in \{1, 5, 10, 25, 50\}$. In addition, for model 2, we use a range of different discretizations $\triangle$ in $\{0.01, 0.02, 0.03, \ldots, 1\}$, and for model 3, we observe $\triangle$ in $\{0.01, 0.02, 0.03, \ldots, 1, 2.5, 5\}$. We then consider an analysis of the estimated mean, standard deviation and skewness which provide a comparison of the weak stationarity properties of the simulated sample paths. This allows us to estimate the effect of the discretization on the estimate of the moments. In probability theory and statistics, skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable. In practice, many models assume Gaussian distribution which has a skewness of zero (since the data points are symmetric about the mean). However, in reality, the data may not be symmetric. Therefore, an understanding about the skewness allows

us to see whether deviations from the mean are going to be positive or negative. We denote the empirical skewness estimate for each factor in the models for a given discretization $\triangle$ by

$$\gamma_{\triangle} = \frac{m_3}{m_2^{\frac{3}{2}}} = \frac{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^3}{(\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2)^{\frac{3}{2}}} \tag{5.24}$$

where N is the total number of sample paths generated, X_i's are the sample paths, $\bar{X}$ is the sample mean of N paths, m_3 is the sample third central moment, and m_2 is the sample variance. Intuitively, the skewness can be infinite when the sample variance m_2 equals 0 and the third central moment m_3 is positive or negative. Moreover, when both m_2 and m_3 are zeros, then the skewness is undefined. In such cases, the N sample paths do not distinguish to each other (since $m_2 = 0$ follows that $X_i = \bar{X}$ for all i). To deal with these cases, we need to sample another N paths until its sample variance is different from zero.

In the case of the two factor model we derived the distribution of the process at each maturity T analytically, since it is a Gaussian distribution and we can obtain explicitly the sufficient statistics as a function of t. Hence we can calculate exactly the moments, and in the case of the three factor model we must estimate these moments empirically, where we assume that true distribution is approximated exactly with a "true" distribution for the spot price factor estimated taking $\triangle = 10^{-3}$ under the Milstein scheme.

For this study, we utilise the following model parameters given in Tables 5 and 6.

κ	μ	σ_1	σ_2	ρ
1.5	-0.16	0.75	0.52	0.14

Table 5: Parameters for model 2

κ	$\hat{\alpha}$	a	m^*	σ_1	σ_2	σ_3	ρ_1	ρ_2	ρ_3
0.3	1	0.18	0.76	0.55	0.15	0.1	0.24	0.3	0.08

Table 6: Parameters for model 3

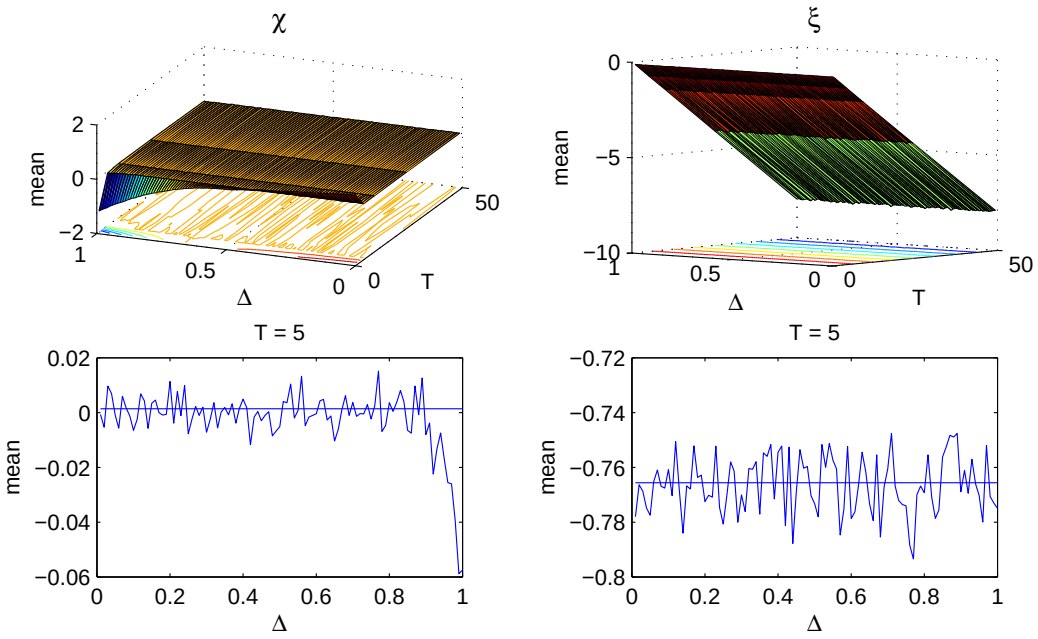

Figure 5: Model 2 - Estimated mean versus the true mean for different maturities T in $\{1, 5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1\}$

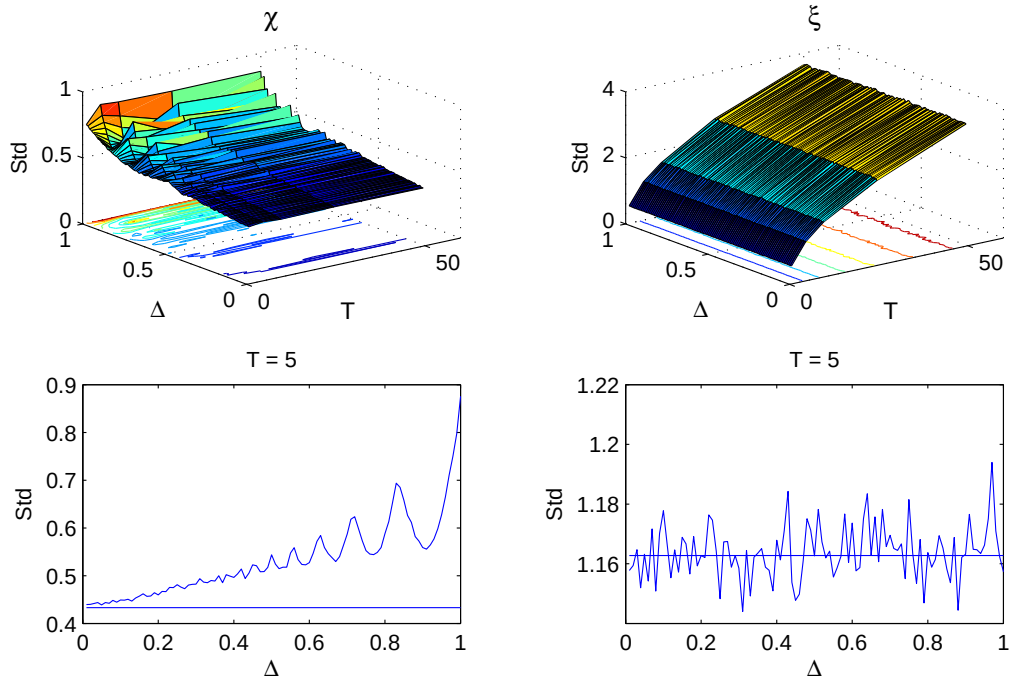

Figure 6: Model 2 - Estimated standard deviation for different maturities T in $\{1, 5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1\}$

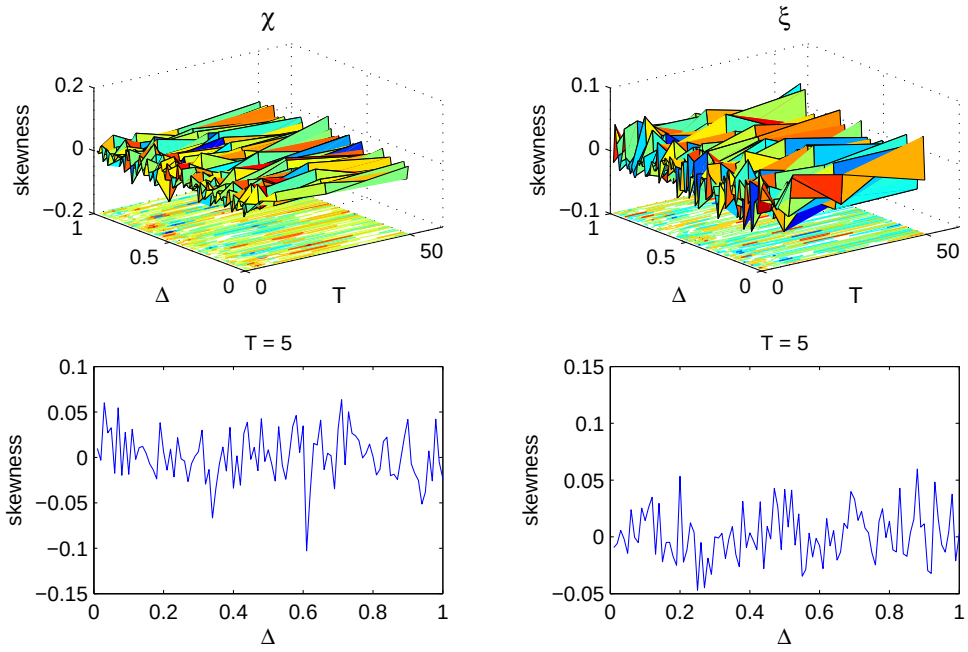

Figure 7: Model 2 - Estimated skewness for different maturities T in $\{1, 5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1\}$

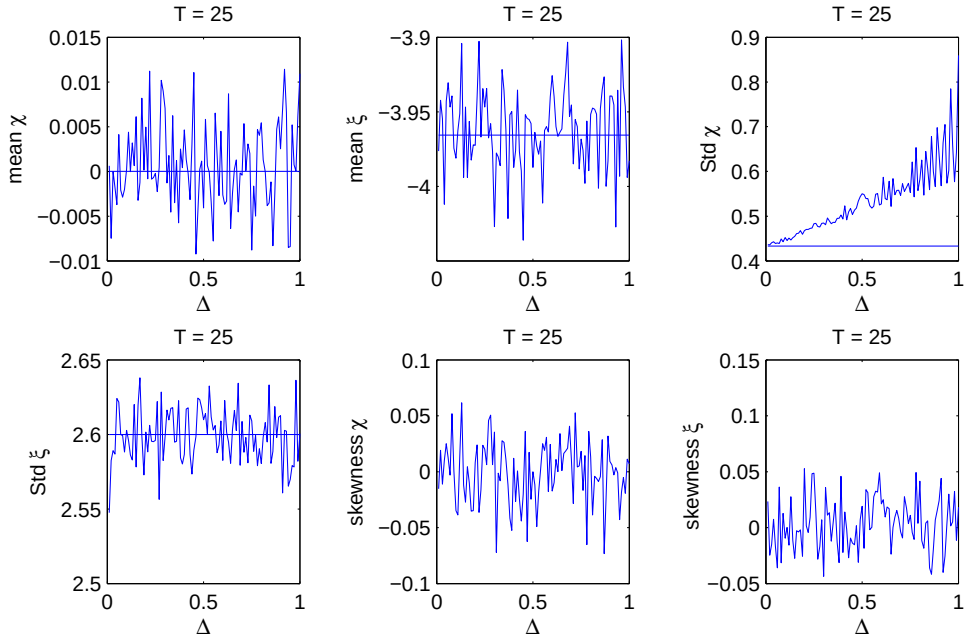

Figure 8: Model 2 - Estimated mean, standard deviation and skewness at maturity T = 25 for different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1, 2.5, 5\}$

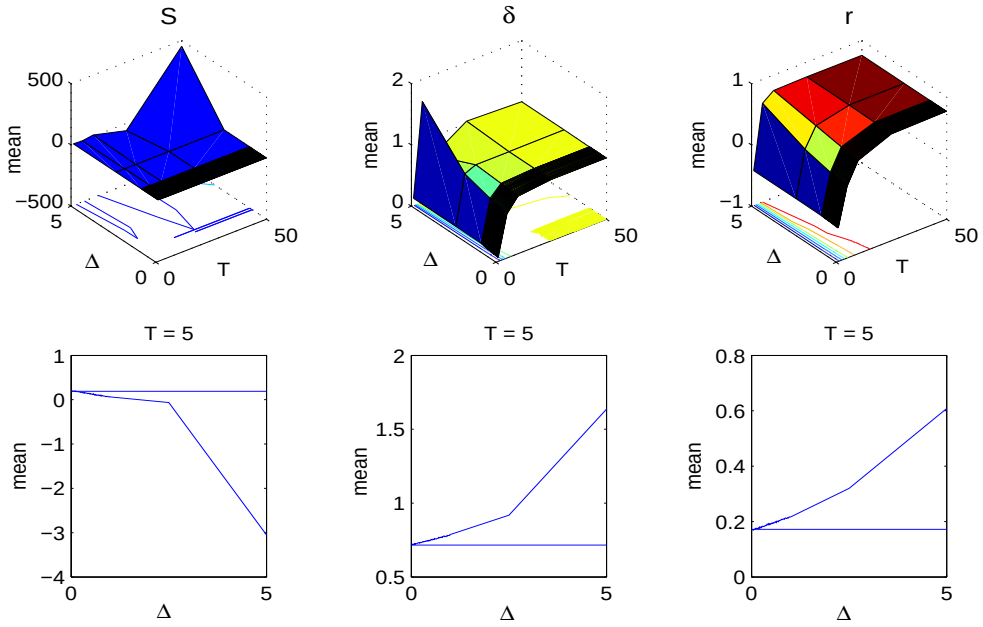

Figure 9: Model 3 - Estimated mean versus the true mean for different maturities T in $\{1, 5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1, 2.5, 5\}$

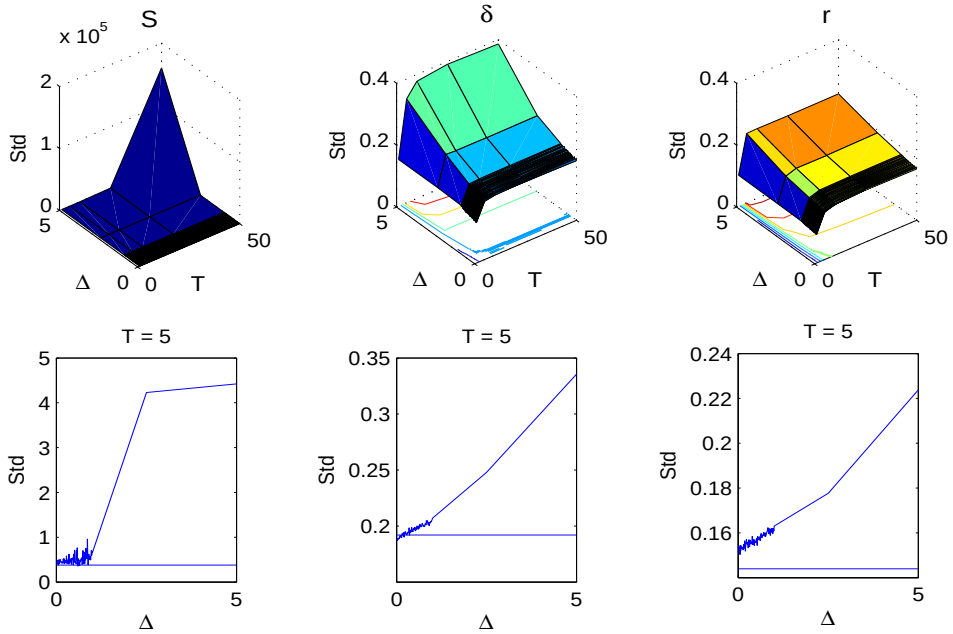

Figure 10: Model 3 - Estimated standard deviation for different maturities T in $\{1, 5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1, 2.5, 5\}$

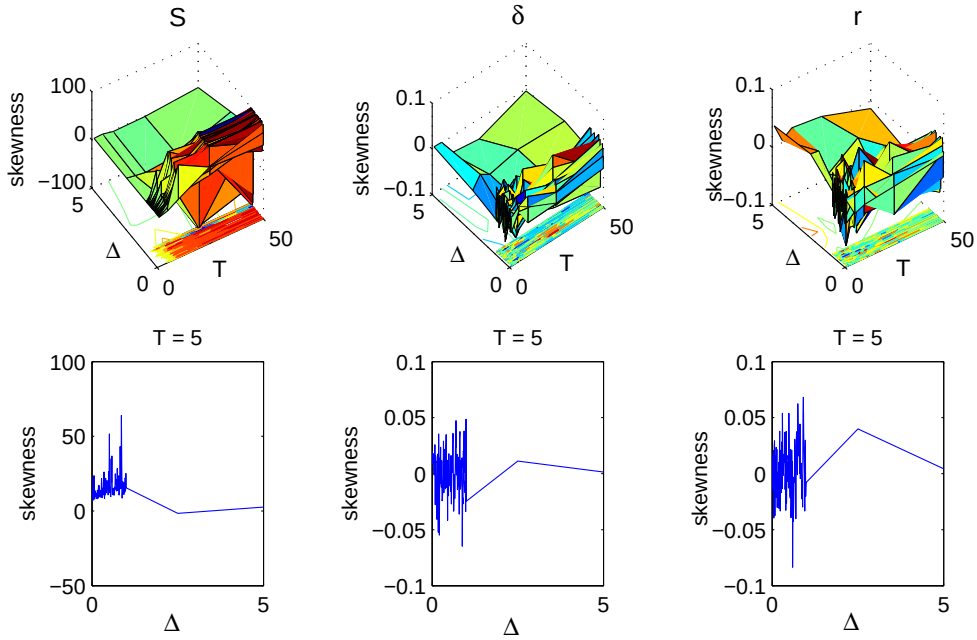

Figure 11: Model 3 - Estimated skewness for different maturities T in $\{1, 5, 10, 25, 50\}$ and different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1, 2.5, 5\}$

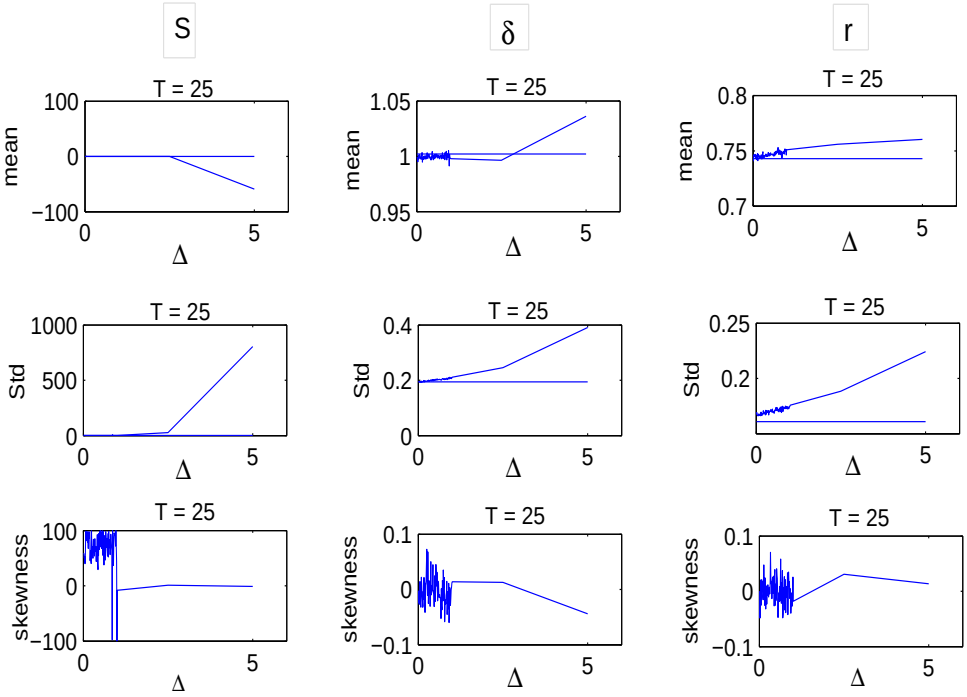

Figure 12: Model 3 - Estimated mean, standard deviation and skewness at maturity T = 25 for different time intervals $\triangle$ in $\{0.01, 0.02, \ldots, 1, 2.5, 5\}$

From figures 5, 6, 7, 8, 9, 10, 11 and 12, we observe that as the time interval is small, both the estimated mean and standard deviation get very close to the true ones. Otherwise, the estimated mean and standard deviation appear to diverge as the time interval becomes bigger. This implies that in order to obtain good approximation for the true process, we need to make the time interval $\triangle$ to be as small as possible. As a result, the discrete approximation $Y^{\triangle}$ we have generated converges weakly to the truth with respect to its first, second and third moments. This provides insight into properties of the models 1, 2 and 3 under different discretizations which will be valuable when we perform filtering.

5.2.3 Bias and Monte Carlo Uncertainty Analysis via Ave.RMSE

In the analysis of the average root mean square error (AveRMSE) and the standard deviation are taken into account to judge the accuracy and convergence of the Euler and Milstein schemes (using models 2 and 3). The standard deviation is taken with respect to the AveRMSE. To obtain accurate estimates of the AveRMSE, we simulate sets of 1000 sample paths for the factors in the models 2 and 3, until the relative uncertainty in our estimates is less than 20%, for each discretization interval. In addition, to evaluate the standard deviation of the AveRMSE, we use 20 sample sets (50 simulation paths for each set). Furthermore, we use 100 different time intervals to observe the accuracy of each scheme as the time discretization is made finer.
We divide these studies into two parts. In the first part, we consider taking the AveRMSE only at the points generated by the time step $\triangle$. In the second part, we extend taking the AveRMSE over the whole 100 points: 0.01, 0.02, ..., 1. In the latter study, the result contains more information than in the first one, since it also considers the points which may not be generated by $\triangle$. Hence, for the latter study, we might have a more clearly view about the "difference" between the simulation and the truth by using each scheme.
Recall that the average root mean square error is given as

$$\text{AveRMSE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\mathbb{E}\left[X_{1:T} - \mu_{1:T}\right]^2} \tag{5.25}$$

Or more explicitly

$$\text{AveRMSE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(X^j_{1:T} - \mu^j_{1:T}\right)^2} \tag{5.26}$$

where T is the total time observed, m is the number of simulation paths, $X^j_{1:T}$ is the j-th simulation path, $\mu^j_{1:T}$ is the j-th true mean path.

The standard deviation of the AveRMSE is given as

$$\mathbb{S}\text{td}[\text{AveRMSE}] = \sqrt{\frac{1}{k}\sum_{s=1}^{k}\left(\text{AveRMSE} - \overline{\text{AveRMSE}}\right)^2} \tag{5.27}$$

where $\overline{\text{AveRMSE}}$ is evaluated using m paths (N is the total number of simulation paths), k is the number of sets which is partitioned from N simulation paths, AveRMSE is evaluated using $\frac{N}{k}$ paths. Specifically, in our case, N = 1000 and k = 20. For studies 1 and 2 in this section, we utilise the following model parameters given in Tables 7 and 8.

κ	μ	σ_1	σ_2	ρ
1.5	-0.16	0.75	0.52	0.14

Table 7: Parameters for model 2

κ	$\hat{\alpha}$	a	m^*	σ_1	σ_2	σ_3	ρ_1	ρ_2	ρ_3
0.3	1	-0.18	0.76	0.35	0.44	0.4	0.24	0.3	-0.08

Table 8: Parameters for model 3

5.2.4 Study 1 - Errors at the points generated by $\triangle$

For this study, we calculate the AveRMSE between the simulation and the truth (for model 2 and model 3) at only the points produced by $\triangle$. A disadvantage of this study is that it does not take into account the points in the middle of two adjacent points when the time step is expanded to more than 0.01. In some sense, this implies that we might not be able to observe the "closeness" between the simulation and the truth paths.

Model 2

We note that since the volatilities in the short-term and long-term factors are constant, the Euler and Milstein schemes produce the same simulations for this model. Hence, we only restrict our comparisons here on the change of the time step. The figures 13, 14 and table 9 below summarize the results obtained by the Euler (or Milstein) scheme.

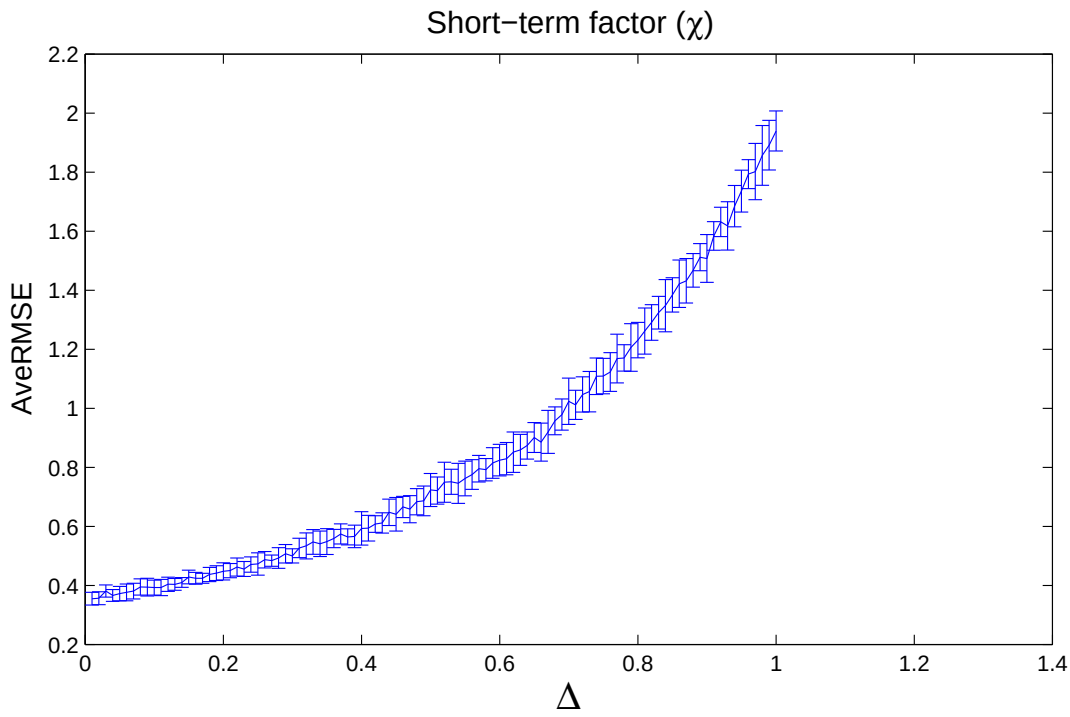

Figure 13: Short-term deviation versus Ave. RMSE for χ (study 1)

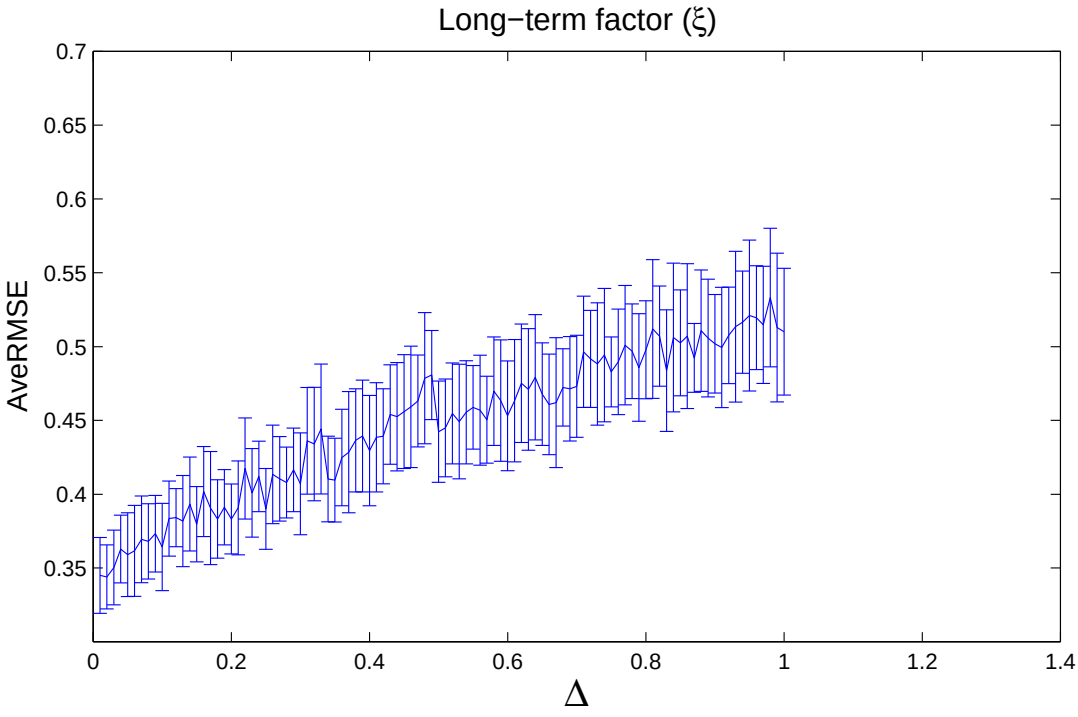

Figure 14: Long-term equilibrium price versus Ave. RMSE for ξ (study 1)

	χ		ξ	
$\triangle t$	AveRMSE 1	Standard deviation 1	AveRMSE 2	Standard deviation 2
0.01	0.3549	0.0216	0.3450	0.0257
0.02	0.3563	0.0218	0.3440	0.0217
0.05	0.3721	0.0243	0.3590	0.0284
0.1	0.3926	0.0254	0.3643	0.0295
0.2	0.4477	0.0290	0.3832	0.0237
0.25	0.4725	0.0377	0.3900	0.0274
0.5	0.7232	0.0559	0.4423	0.0343
0.6	0.8242	0.0535	0.4531	0.0372
0.8	1.2308	0.0601	0.4978	0.0332
1	1.9394	0.0679	0.5101	0.0430

Table 9: Simulation results for model 2 (study 1)

where AveRMSE 1 and Standard deviation 1 denote the AveRMSE and the standard deviation for the short-term factor respectively, AveRMSE 2 and Standard deviation 2 stand for the AveRMSE and the standard deviation for the long-term factor respectively.

From table 9, we see that as the time step increases, the AveRMSEs for both the short-term and long-term factors also increase. The AveRMSE for the short-term deviation begins at 0.3549 (for $\triangle t = 0.01$) and ends up at 1.9394 (for $\triangle t = 1$). Similarly, the AveRMSE for the long-term factor starts at 0.3450 (for $\triangle t = 0.01$), going up gradually and ends up at 0.5101 (for $\triangle t = 1$). These facts imply that the simulation generated by the Euler (or Milstein) scheme gives better approximation results as the time step becomes smaller. Regarding the standard deviations for the short- and long-term factors, even there are some declines at some time intervals, the general trend is still upward. The standard deviation for the short-term factor starts at 0.0216 for $\triangle t = 0.01$, and ends up at 0.0679 for $\triangle t = 1$ (an increase of approximately three-folds). Meanwhile, the standard deviation for the long-term factor makes an increase of about two-folds from 0.0257 (at $\triangle t = 0.01$) to 0.0430 (at $\triangle t = 1$). To some extent, these imply that the uncertainty of the simulation occurs most at $\triangle t = 1$, and when the time step is made smaller, the uncertainty of the simulation reduces too.

Model 3

We first note that since the diffusion coefficient in the spot price dynamics is not a constant, the Euler and Milstein schemes produce different simulation results for the spot price. In addition, since the diffusion coefficients in the convenience yield and the interest rate are constant, the results obtained by these schemes are not distinguishable. The figures 15, 16, 17, 18 and tables 10, 11 below summarize the results obtained by the Euler and Milstein schemes.

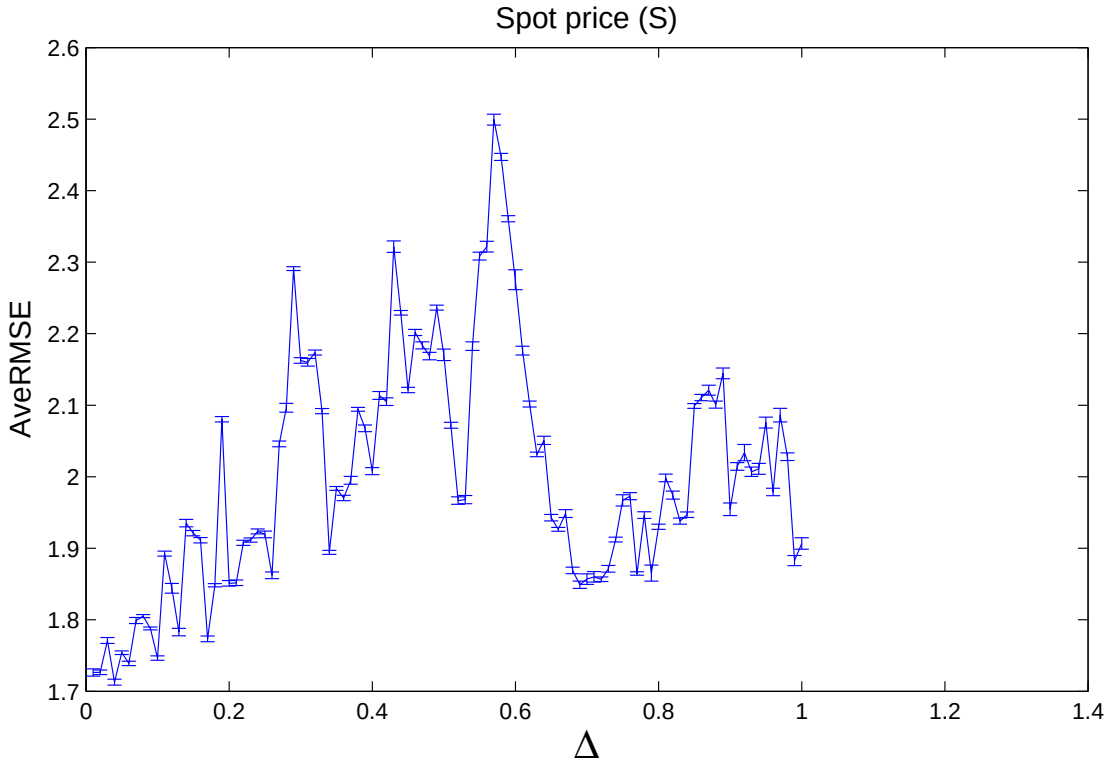

Figure 15: The spot price (Milstein scheme) versus Ave. RMSE (study 1)

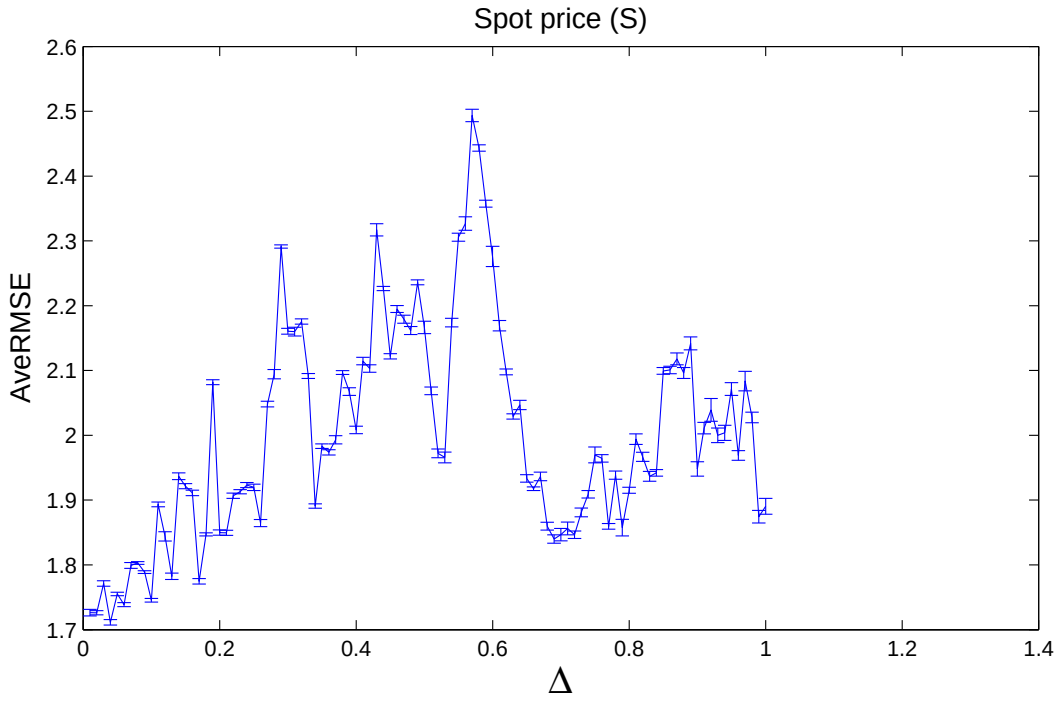

Figure 16: The spot price (Euler scheme) versus Ave. RMSE (study 1)

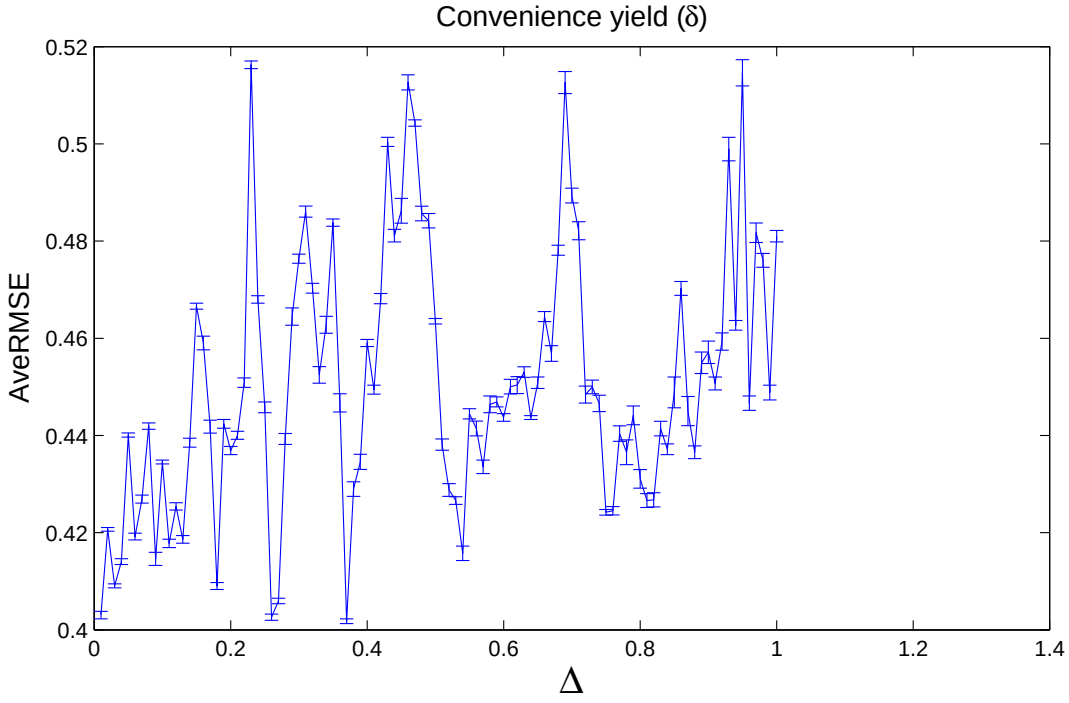

Figure 17: The convenience yield versus Ave. RMSE (study 1)

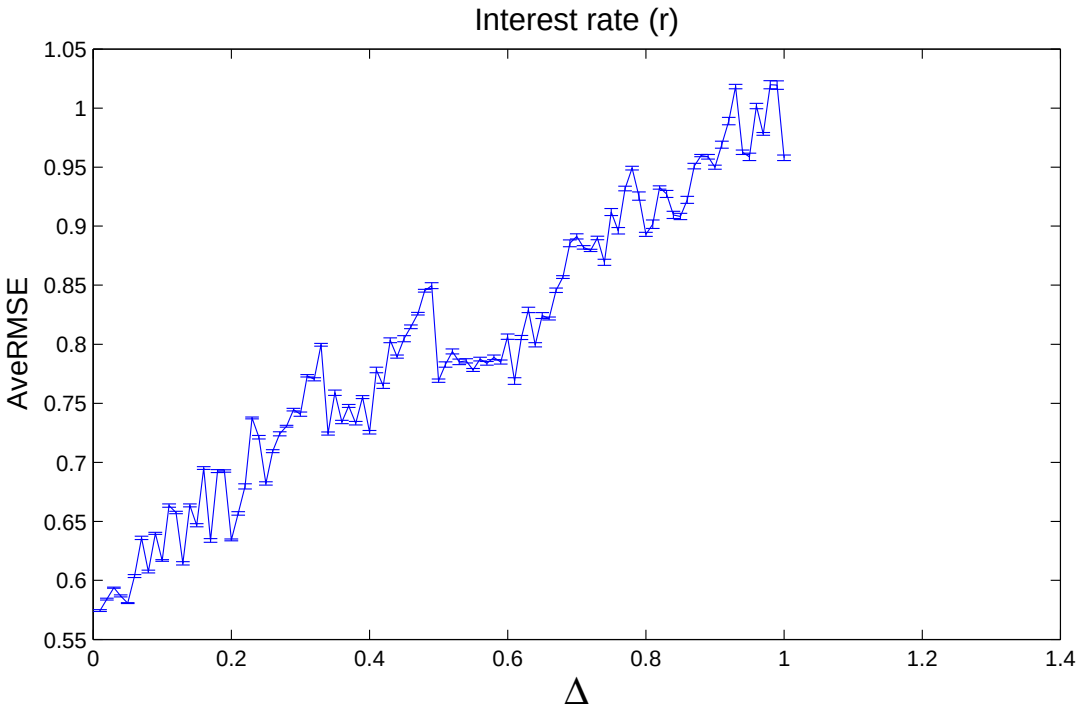

Figure 18: The interest rate versus Ave. RMSE (study 1)

	S		δ		r	
$\triangle t$	AveRMSE1	Std1	AveRMSE2	Std2	AveRMSE3	Std3
0.01	1.7263	0.0050	0.4030	0.0008	0.5744	0.0008
0.02	1.7266	0.0032	0.4207	0.0004	0.5841	0.0008
0.05	1.7540	0.0026	0.4401	0.0004	0.5808	0.0004
0.1	1.7465	0.0030	0.4345	0.0004	0.6169	0.0007
0.2	1.8509	0.0039	0.4369	0.0008	0.6343	0.0008
0.25	1.9194	0.0046	0.4458	0.0011	0.6821	0.0014
0.5	2.1705	0.0080	0.4635	0.0006	0.7692	0.0015
0.6	2.2754	0.0139	0.4437	0.0008	0.8064	0.0023
0.8	1.9301	0.0036	0.4311	0.0019	0.8931	0.0017
1	1.9067	0.0080	0.4810	0.0012	0.9579	0.0023

Table 10: Simulation results for model 3 using the Milstein scheme (study 1)

	S		δ		r	
$\triangle t$	AveRMSEe1	Stde1	AveRMSEe2	Stde2	AveRMSEe3	Stde3
0.01	1.7264	0.0050	0.4030	0.0008	0.5744	0.0008
0.02	1.7261	0.0032	0.4207	0.0004	0.5841	0.0008
0.05	1.7552	0.0027	0.4401	0.0004	0.5808	0.0004
0.1	1.7454	0.0029	0.4345	0.0004	0.6169	0.0007
0.2	1.8499	0.0041	0.4369	0.0008	0.6343	0.0008
0.25	1.9197	0.0050	0.4458	0.0011	0.6821	0.0014
0.5	2.1664	0.0096	0.4635	0.0006	0.7692	0.0015
0.6	2.2759	0.0156	0.4437	0.0008	0.8064	0.0023
0.8	1.9151	0.0047	0.4311	0.0019	0.8931	0.0017
1	1.8905	0.0122	0.4810	0.0012	0.9579	0.0023

Table 11: Simulation results for model 3 using the Euler scheme (study 1)

where:

- AveRMSE1 and Std1 denote the AveRMSE and the standard deviation for the spot price respectively (using Milstein scheme).
- AveRMSE2 and Std2 denote the AveRMSE and the standard deviation for the convenience yield respectively (using Milstein scheme).
- AveRMSE3 and Std3 denote the AveRMSE and the standard deviation for the interest rate respectively (using Milstein scheme).
- AveRMSEe1 and Stde1 denote the AveRMSE and the standard deviation for the spot price respectively (using Euler scheme).
- AveRMSEe2 and Stde2 denote the AveRMSE and the standard deviation for the convenience yield respectively (using Euler scheme).
- AveRMSEe3 and Stde3 denote the AveRMSE and the standard deviation for the interest rate respectively (using Euler scheme).

From tables 10 and 11, we observe that the AveRMSEs for the convenience yield and interest rate obtained by the Milstein and Euler schemes are not distinguishable since their diffusion

coefficients are constants. For the spot price factor, the AveRMSEs achieved by the both schemes appear to increase up to $\triangle t = 0.6$. For the Milstein scheme, its AveRMSE starts from 1.7263 at $\triangle t = 0.01$ and rises to 2.2754 at $\triangle t = 0.6$. For the Euler scheme, its AveRMSE begins from 1.7264 at $\triangle t = 0.01$ and increases to 2.2759 at $\triangle t = 0.6$. From $\triangle = 0.8$ to $\triangle = 1$, the AveRMSEs for both schemes decrease due to the fact that the spot price appears to be stationary and less changing in that interval. Regarding the standard deviation for the spot price, it can be seen that the standard deviation obtained by the Milstein scheme seems to be less than that of the Euler scheme. To some extent, this implies that the simulations generated by the Milstein gather closer to the truth than those generated by the Euler one. However, it should be noted that as the time interval becomes very small, the Euler appears to produce as a good result as its counterpart, as expected. This again verifies numerically the theoretical results presented previously. Indeed, at $\triangle t = 0.01$, the AveRMSEs for both these schemes almost do not distinguish with a very small difference of 0.0001, meanwhile the standard deviation for both schemes are the same (0.0050).

Regarding the results for the convenience yield factor, it can be seen that the AveRMSE for this factor is quite small and stable as compared to the spot price. It ranges from 0.4030 (at $\triangle t = 0.01$) to 0.4810 (at $\triangle t = 1$). Moreover, the standard deviation of the convenience yield seems to be very small as compared to its actual AveRMSE, ranging from 0.0004 to 0.0019. Another outstanding feature is that the AveRMSE of the interest rate increases as the time interval is expanded (except at $\triangle t = 0.02$ where its AveRMSE is 0.5841). To some extent, this implies that as the time interval is made smaller, the Milstein (or Euler) scheme gives better simulation for the interest rate. In addition, the standard deviation of the interest is quite small as compared to that of the spot price factor. The general trend of this standard deviation is upward which implies that as the time step increases, there may appear some simulation paths generated by the Milstein (or Euler) scheme being "far" away from the truth path.

5.2.5 Study 2 - Errors at fixed grid of values

For this study, we calculate the AveRMSE between the simulation and the truth (for model 2 and model 3) at only the points produced by $\triangle$. An advantage of this study is that it does take into account points in the middle of two adjacent discretization time points when the time step is expanded to more than 0.01. In some sense, this implies that we might be able to observe the "closeness" between the simulation and the truth paths.

Model 2

We note that since the volatilities in the short-term and long-term factors are constant, the Euler and Milstein schemes produce the same simulations for this model. Hence, we only restrict our comparisons here on the change of the time step. The figures 19, 20 and table 12 below summarize the results obtained by the Euler (or Milstein) scheme.

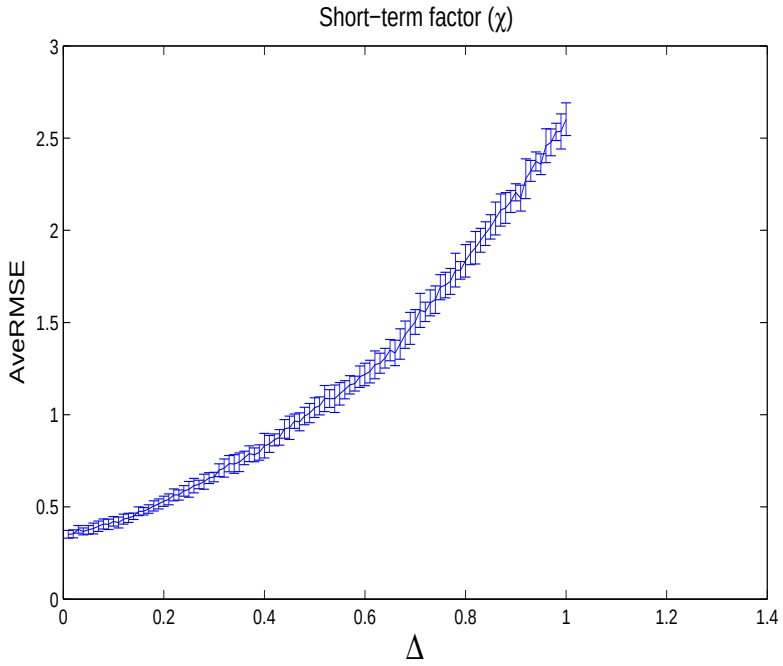

Figure 19: Short-term deviation versus Ave. RMSE for χ (study 2)

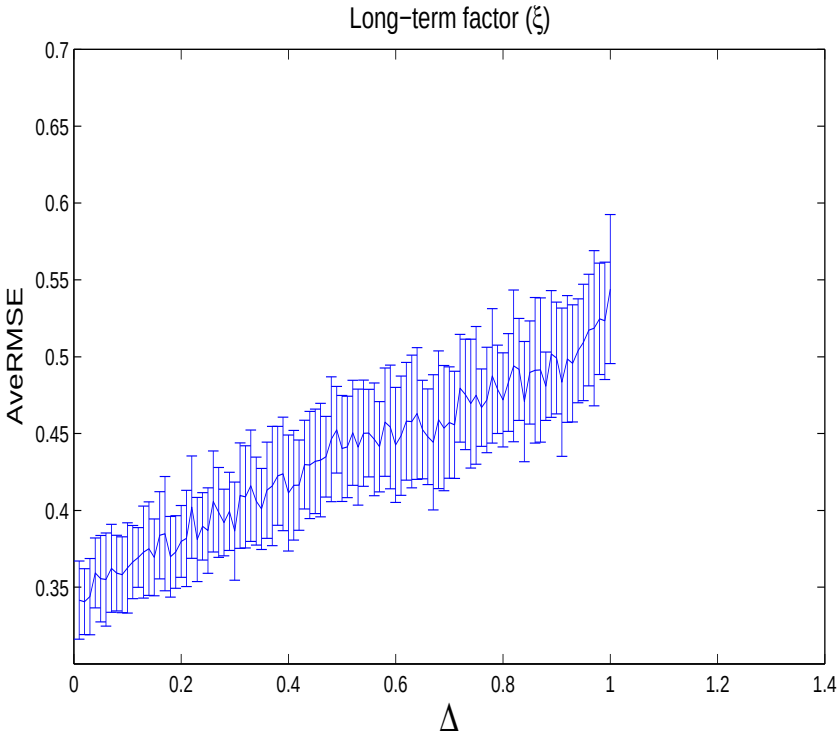

Figure 20: Long-term equilibrium price versus Ave. RMSE for ξ (study 2)

	χ		ξ	
$\triangle t$	AveRMSE 1	Standard deviation 1	AveRMSE 2	Standard deviation 2
0.01	0.3514	0.0214	0.3416	0.0254
0.02	0.3535	0.0215	0.3406	0.0214
0.05	0.3754	0.0235	0.3555	0.0281
0.1	0.4217	0.0259	0.3626	0.0294
0.2	0.5308	0.0278	0.3798	0.0233
0.25	0.5950	0.0427	0.3869	0.0278
0.5	1.0380	0.0533	0.4403	0.0345
0.6	1.2186	0.0609	0.4426	0.0375
0.8	1.8342	0.0884	0.4720	0.0306
1	2.6021	0.0886	0.5441	0.0485

Table 12: Simulation results for model 2 (study 2)

where AveRMSE 1 and Standard deviation 1 denote the AveRMSE and the standard deviation for the short-term factor respectively, AveRMSE 2 and Standard deviation 2 stand for the

AveRMSE and the standard deviation for the long-term factor respectively.

From table 12, we observe that as the time step increases, the AveRMSEs for both the short-term and long-term factors almost increase too. Particularly, the AveRMSE for the short-term deviation begins at 0.3514 (for $\triangle t = 0.01$) and ends up at 2.6021 (for $\triangle t = 1$). Similarly, the AveRMSE for the long-term factor starts at 0.3416 (for $\triangle t = 0.01$), going up gradually (even there is a small decline to 0.3406 at $\triangle t = 0.02$) and ends up at 0.5441 (for $\triangle t = 1$). These facts indicate that the simulation generated by the Euler (or Milstein) scheme gives better approximation results as the time step is made smaller. Regarding the standard deviations for the short- and long-term factors, even there are some declines at some time intervals (for the long-term factor), the general trend for both two factors is still upward. The standard deviation for the short-term factor starts at 0.0214 for $\triangle t = 0.01$, and ends up at 0.0886 for $\triangle t = 1$ (an increase of approximately four times). In addition, the standard deviation for the long-term factor makes an increase of about two-folds from 0.0254 (at $\triangle t = 0.01$) to 0.0485 (at $\triangle t = 1$). To some extent, these imply that the uncertainty of the simulation occurs most at $\triangle t = 1$, and when the time step is smaller, the uncertainty of the simulation decreases too.

Model 3

We first notice that since the diffusion coefficient in the spot price dynamics is not a constant, the Euler and Milstein schemes produce different simulation results for the spot price. On the other hand, since the diffusion coefficients in the convenience yield and the interest rate are constant, the results obtained by these schemes are not distinguishable. The figures 21, 22, 23, 24 and tables 13, 14 below summarize the results achieved by the Euler and Milstein schemes.

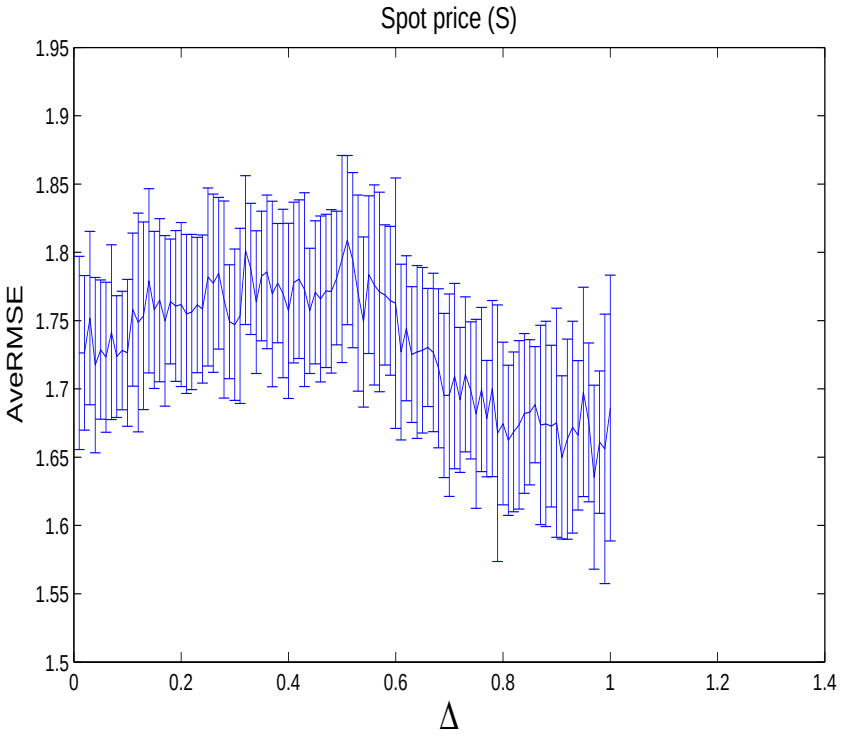

Figure 21: The spot price (Milstein scheme) versus Ave. RMSE (study 2)

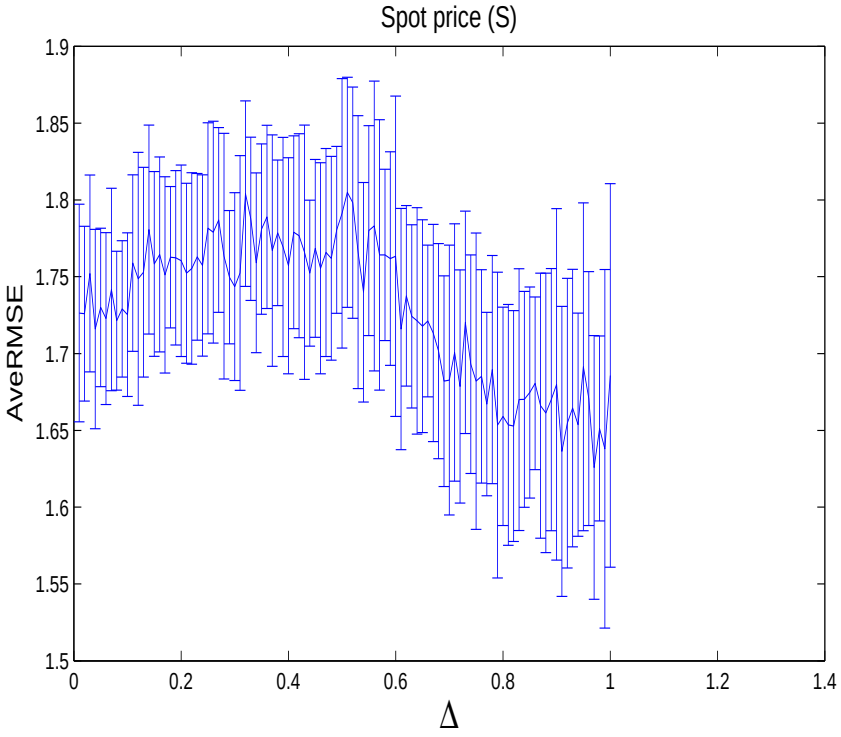

Figure 22: The spot price (Euler scheme) versus Ave. RMSE (study 2)

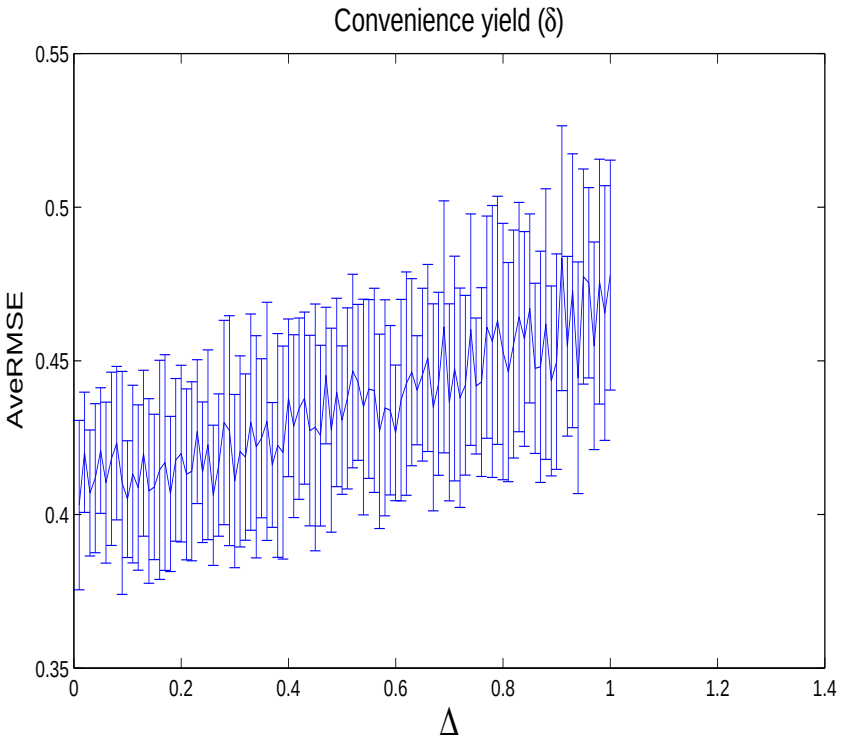

Figure 23: The convenience yield versus Ave. RMSE (study 2)

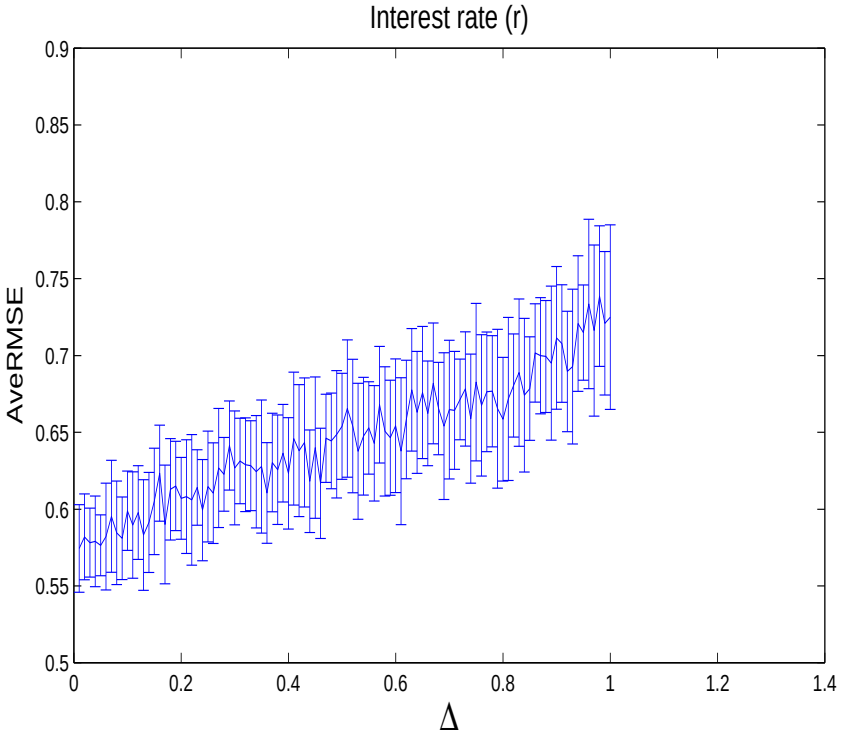

Figure 24: The interest rate versus Ave. RMSE (study 2)

	S		δ		r	
$\triangle t$	AveRMSE1	Std1	AveRMSE2	Std2	AveRMSE3	Std3
0.01	1.7263	0.0708	0.4030	0.0276	0.5744	0.0285
0.02	1.7264	0.0566	0.4202	0.0196	0.5820	0.0280
0.05	1.7288	0.0510	0.4208	0.0204	0.5765	0.0198
0.1	1.7264	0.0538	0.4050	0.0190	0.5990	0.0258
0.2	1.7618	0.0600	0.4198	0.0287	0.6071	0.0266
0.25	1.7819	0.0652	0.4227	0.0309	0.6147	0.0361
0.5	1.7951	0.0758	0.4307	0.0242	0.6539	0.0345
0.6	1.7628	0.0917	0.4341	0.0225	0.6543	0.0435
0.8	1.6747	0.0596	0.4556	0.0422	0.6586	0.0403
1	1.6860	0.0973	0.4779	0.0374	0.7250	0.0601

Table 13: Simulation results for model 3 using the Milstein scheme (study 2)

	S		δ		r	
$\triangle t$	AveRMSEe1	Stde1	AveRMSEe2	Stde2	AveRMSEe3	Stde3
0.01	1.7264	0.0708	0.4030	0.0276	0.5744	0.0285
0.02	1.7259	0.0568	0.4202	0.0196	0.5820	0.0280
0.05	1.7300	0.0515	0.4208	0.0204	0.5765	0.0198
0.1	1.7254	0.0533	0.4050	0.0190	0.5990	0.0258
0.2	1.7604	0.0623	0.4198	0.0287	0.6071	0.0266
0.25	1.7815	0.0686	0.4227	0.0309	0.6147	0.0361
0.5	1.7912	0.0877	0.4307	0.0242	0.6539	0.0345
0.6	1.7633	0.1042	0.4341	0.0225	0.6543	0.0435
0.8	1.6591	0.0711	0.4556	0.0422	0.6586	0.0403
1	1.6857	0.1248	0.4779	0.0374	0.7250	0.0601

Table 14: Simulation results for model 3 using the Euler scheme (study 2)

where:

- AveRMSE1 and Std1 denote the AveRMSE and the standard deviation for the spot price respectively (using Milstein scheme).

- AveRMSE2 and Std2 denote the AveRMSE and the standard deviation for the convenience yield respectively (using Milstein scheme).
- AveRMSE3 and Std3 denote the AveRMSE and the standard deviation for the interest rate respectively (using Milstein scheme).
- AveRMSEe1 and Stde1 denote the AveRMSE and the standard deviation for the spot price respectively (using Euler scheme).
- AveRMSEe2 and Stde2 denote the AveRMSE and the standard deviation for the convenience yield respectively (using Euler scheme).
- AveRMSEe3 and Stde3 denote the AveRMSE and the standard deviation for the interest rate respectively (using Euler scheme).

From tables 13 and 14, we first observe that the AveRMSE for the spot price (for both the Milstein and Euler schemes) appears to increase from $\triangle t = 0.01$ up to $\triangle t = 0.5$, and then decline until the end. The Milstein and Euler schemes produce almost the same results at $\triangle t = 0.01$ (1.7263 for the Milstein and 1.7264 for the Euler one). This is due to the fact that the Euler scheme achieves very good simulation result as the Milstein does as the time interval is made very small. The AveRMSE for the spot price rises gradually (even it declines at some certain $\triangle t$), and reaches 1.7951 and 1.7912 for the Milstein and Euler schemes respectively. It should be noted that from time 0.5 to time 1, the curve of the spot price appears to be less changing than from time 0 to time 0.5. To some extent, this leads to the declines in the AveRMSE of the spot price achieved by these two schemes. Regarding the standard deviation of the spot price, the numbers for both the Milstein and Euler schemes appear to be quite similar to each other for very small $\triangle t$. As $\triangle t$ increases, the Milstein seems to yield better result than its counterpart which implies that as the time interval goes up, the uncertainty appears more in the Euler scheme than in the Milstein one. In addition, the general trend for the standard deviation of the spot price is upward. Both the Milstein and Euler start at almost the same level of 0.0708 (at $\triangle t = 0.01$), and end up at 0.0973 for the Milstein scheme and 0.1248 for the Euler one respectively.

Regarding the results for the convenience yield and the interest rate factors, since the drift and diffusion coefficients are constants, therefore the Milstein and Euler schemes produce the same results. It first should be noted that the AveRMSEs for these two factors seem to rise up as the time interval is expanded. The AveRMSE for the convenience yield starts at 0.4030 (for $\triangle t =$

0.01), and then it fluctuates around 0.4200 until $\triangle t = 0.2$. For $\triangle t$ from 0.25 to 1, the AveRMSE for this factor goes up quite quickly, and reach 0.4779 at $\triangle t = 1$. For the interest rate factor, there is only a bit decrease at $\triangle t = 0.05$ where its AveRMSE is 0.5765. In general, the trend of the AveRMSE for this last factor is upwards. It begins at 0.5744 for $\triangle t = 0.01$, and ends up at 0.7250 for $\triangle t = 1$. Finally, regarding the standard deviation aspect, it can be seen that there are more uncertainty for large $\triangle t$ than for smaller $\triangle t$. For the convenience yield factor, the numbers at $\triangle t = 0.8$ and $\triangle t = 1$ are 0.0422 and 0.0374 respectively, as compared to 0.0276 and 0.0196 at $\triangle t = 0.01$ and $\triangle t = 0.02$ in that order. For the interest rate one, the numbers at $\triangle t = 0.8$ and $\triangle t = 1$ are 0.0403 and 0.0601 respectively, as compared to 0.0285 and 0.0280 at $\triangle t = 0.01$ and $\triangle t = 0.02$ in the same order.

6 State space models for commodities

In this section, the state space model is presented as in [10]. This model representation is a powerful tool to handle a wide range of time series models. Once a model has been put in state space form, the Kalman filter may be applied and this in turn leads to algorithms for prediction and smoothing. The general state space form applies to a multivariate time series, $\boldsymbol{z}_k$, containing N elements. These observable variables are related to an m $\times$ 1 vector, $\boldsymbol{x}_k$, known as the state vector, via a measurement equation [10]

$$\boldsymbol{z}_k = W_k \boldsymbol{x}_k + d_k + \boldsymbol{\epsilon}_k, \quad k = 1, ..., T \tag{6.1}$$

where W_k is an N x m matrix, d_k is an N x 1 vector and $\boldsymbol{\epsilon}_k$ is an N x 1 vector of serially uncorrelated disturbances with mean zero and covariance matrix H_k, that is

$$\mathbb{E}\left[\boldsymbol{\epsilon}_k\right] = 0 \quad \text{and} \quad \mathbb{V}ar\left[\boldsymbol{\epsilon}_k\right] = H_k \tag{6.2}$$

In general, the elements of $\boldsymbol{x}_k$ are not observable. However, they are known to be generated by a first-order Markov process [10]

$$\boldsymbol{x}_k = T_k \boldsymbol{x}_{k-1} + c_k + R_k \boldsymbol{\eta}_k, \quad k = 1, ..., T \tag{6.3}$$

where T_k is an m x m matrix, c_k is an m x 1 vector, R_k is an m x g matrix and $\boldsymbol{\eta}_k$ is a g x 1 vector of serially uncorrelated disturbances with mean zero and covariance matrix, Q_k, that is

$$\mathbb{E}\left[\boldsymbol{\eta}_k\right] = 0 \quad \text{and} \quad \mathbb{V}ar\left[\boldsymbol{\eta}_k\right] = Q_k \tag{6.4}$$

Equation (6.3) is the transition equation. To some extent, the inclusion of the matrix R_k is arbitrary. The disturbance term could always be redefined to have a covariance matrix $R_k Q_k R_k^T$. The specification of the state space system is completed by two further assumptions:
(a) the initial state vector, $\boldsymbol{x}_0$, has a mean of a_0 and a covariance matrix P_0, that is

$$\mathbb{E}\left[\boldsymbol{x}_0\right] = a_0 \quad \text{and} \quad \mathbb{V}ar\left[\boldsymbol{x}_0\right] = P_0 \tag{6.5}$$

(b) the disturbances $\boldsymbol{\epsilon}_k$ and $\boldsymbol{\eta}_k$ are uncorrelated with each other in all time periods, and uncorrelated with the initial state, that is

$$\mathbb{E}\left[\boldsymbol{\epsilon}_k \boldsymbol{\eta}_s^T\right] = 0 \quad \forall s, k = 1, ..., T \tag{6.6}$$

and

$$\mathbb{E}\left[\boldsymbol{\epsilon}_k \boldsymbol{x}_0^T\right] = 0, \quad \mathbb{E}\left[\boldsymbol{\eta}_k \boldsymbol{x}_0^T\right] = 0 \quad \text{for } k = 1, ..., T \tag{6.7}$$

The matrices W_k, d_k and H_k in the measurement equation and the matrices T_k, c_k, R_k and Q_k in the transition equation are referred to as the system matrices. If these matrices do not change over time, the model is said to be time-invariant or time-homogeneous. Stationary models are a special case. However, although the class of time-invariant models is broader than that of stationary models, many time-invariant models have a stationary from which can be obtained by applying a transformation such as differencing. The system matrices W_k, H_k, T_k, R_k and Q_k may also depend on a set of unknown parameters, and one of the main statistical tasks will often be the estimation of these parameters.

The transition equation in (6.3) is sometimes shifted forward one period in order to become [10]

$$\boldsymbol{x}_{k+1} = T_k \boldsymbol{x}_k + c_k + R_k \boldsymbol{\eta}_k, \tag{6.8}$$

The definition of $\boldsymbol{x}_k$ for any particular statistical model is determined by construction. Its elements may or may not be identifiable with components which have a substantive interpretation, for instance as a trend or a seasonal. From a technical viewpoint, the aim of the state space formulation is to set up $\boldsymbol{x}_k$ in such a way that it contains all the relevant information on the system at time k and that it does so by having as small a number of elements as possible. A state space form which minimises the length of the state vector is said to be a minimal realisation. A minimal realisation is a basic criterion for a good state space representation.

6.1 Representation for model 1

From equation (3.5), the measurement equation of model 1 can be expressed as follows

$$\boldsymbol{z}_k = W_k \boldsymbol{x}_k + d_k + \boldsymbol{\epsilon}_k, \quad k = 1, ..., n_T \tag{6.9}$$

where:

- $\boldsymbol{z}_k = [\ln F_{T_1}, \ldots, \ln F_{T_n}]$ is a $n \times 1$ vector of observed (log) futures prices with maturities $T_1, \ldots, T_n$;
- $d_k = [(r-c)T_1, \ldots, (r-c)T_n]$ is a $n \times 1$ vector;
- $W_k = [1, \ldots, 1]$ is a $n \times 1$ matrix; and
- $\boldsymbol{\epsilon}_k$ is a $n \times 1$ vector of serially uncorrelated, normally distributed disturbances with:

$$\mathbb{E}[\boldsymbol{\epsilon}_k] = 0 \quad \text{and} \quad \mathbb{C}ov[\boldsymbol{\epsilon}_k] = H_k.$$

Now using the Euler scheme, the process $\boldsymbol{X}$ (the log of the spot price) can be discretized as

$$\boldsymbol{X}_k = \boldsymbol{X}_{k-1} + \left(\mu - \frac{1}{2}\sigma^2\right) \triangle k + \sigma\sqrt{\triangle k} n_{X,k-1} \tag{6.10}$$

where $n_{X,k-1}$ is a standard normal random variable.
Thus, from this equation, the transition equation can be expressed as follows

$$\boldsymbol{x}_k = T_k \boldsymbol{x}_{k-1} + c_k + R_k \boldsymbol{\eta}_k, \quad k = 1, ..., n_T \tag{6.11}$$

where:

- $\boldsymbol{x}_k = [\boldsymbol{X}_k]$ is a 1×1 vector of the state variable;
- $T_k = [1]$ is a 1×1 matrix;
- $c_k = [(\mu - \frac{1}{2}\sigma^2)\triangle k]$ is a 1×1 vector;
- R_k is a 1×1 identity matrix;
- $\boldsymbol{\eta}_k$ is a 1×1 vector of normally distributed disturbance with $\mathbb{E}[\boldsymbol{\eta}_k] = 0$ and $\mathbb{V}ar[\boldsymbol{\eta}_k] = Q_k = \mathbb{V}ar[\boldsymbol{X}_k] = [\sigma^2 \triangle k]$.

6.2 Representation for model 2

From equation (3.10), the measurement equation of the short-term/long-term model can be written as follows

$$\boldsymbol{z}_k = W_k \boldsymbol{x}_k + d_k + \boldsymbol{\epsilon}_k, \quad k = 1, ..., n_T \tag{6.12}$$

where:

- $\boldsymbol{z}_k = [\ln F_{T_1}, \ldots, \ln F_{T_n}]$ is a $n \times 1$ vector of observed (log) futures prices with maturities $T_1, \ldots, T_n$;
- $d_k = [A(T_1), \ldots, A(T_n)]$ is a $n \times 1$ vector;
- $W_k = [e^{-\kappa T_1} \quad 1, \ldots, e^{-\kappa T_n} \quad 1]$ is a $n \times 2$ matrix; and
- $\boldsymbol{\epsilon}_k$ is a $n \times 1$ vector of serially uncorrelated, normally distributed disturbances with:

$$\mathbb{E}[\boldsymbol{\epsilon}_k] = 0 \quad \text{and} \quad \mathbb{C}ov[\boldsymbol{\epsilon}_k] = H_k$$

The evolution of the state variables is described by the transition equation [10]:

$$\boldsymbol{x}_k = T_k \boldsymbol{x}_{k-1} + c_k + R_k \boldsymbol{\eta}_k, \quad k = 1, ..., n_T \tag{6.13}$$

where:

- $\boldsymbol{x}_k = [\boldsymbol{\chi}_k, \boldsymbol{\xi}_k]$ is a 2×1 vector of state variables;
- $T_k = \begin{bmatrix} e^{-\kappa \triangle k} & 0 \\ 0 & 1 \end{bmatrix}$ is a 2×2 matrix;
- $c_k = [0, \mu_\xi \triangle k]$ is a 2×1 vector;
- R_k is a 2×2 identity matrix;
- $\boldsymbol{\eta}_k$ is a 2×1 vector of serially uncorrelated, normally distributed disturbances with $\mathbb{E}[\boldsymbol{\eta}_k] = 0$ and $\mathbb{V}ar[\boldsymbol{\eta}_k] = Q_k = \mathbb{C}ov[(\boldsymbol{\chi}_{\triangle k}, \boldsymbol{\xi}_{\triangle k})] = \begin{bmatrix} \left(1 - e^{-2\kappa \triangle k}\right) \frac{\sigma_\chi^2}{2\kappa} & \left(1 - e^{-\kappa \triangle k}\right) \frac{\rho \sigma_\chi \sigma_\xi}{\kappa} \\ \left(1 - e^{-\kappa \triangle k}\right) \frac{\rho \sigma_\chi \sigma_\xi}{\kappa} & \sigma_\xi^2 \triangle k \end{bmatrix}$ (given by equation (3.29)).

6.3 Representation for model 3

For the purpose of estimation which we will consider later on, the spot price, the convenience yield and the interest rate should be estimated simultaneously from a time series and cross-section of futures prices. To simplify the estimation, we first estimate the parameters of the interest rate process, and then use them to estimate the parameters of the spot price and convenience yield processes. Hence, it is essentially assumed that the parameters of the interest rate process are not affected by commodity futures prices, which seems to be a reasonable assumption. Once the interest rate process has been estimated, we then only have to estimate the parameters and state variables from the spot price and convenience yield processes. For these reasons, we just need to express the measurement and transition equations in terms of the (log) spot price and the convenience yield factors.

From equation (3.21), the measurement equation of model 3 can be written as follows

$$\boldsymbol{z}_k = W_k \boldsymbol{x}_k + d_k + \boldsymbol{\epsilon}_k, \quad k = 1, ..., n_T \tag{6.14}$$

where:

- $\boldsymbol{z}_k = [\ln F_{T_k}]$, $k = 1, \ldots, n$ is a $n \times 1$ vector of observed (log) futures prices with maturities $T_1, \ldots, T_n$;

- $W_k = [1 \quad -\frac{1-e^{-\kappa T_k}}{\kappa}]$, $k = 1, \ldots, n$ is a $n \times 2$ matrix;
- $d_k = [\frac{r_k(1-e^{-aT_k})}{a} + C(T_k)]$, $k = 1, \ldots, n$ is a $n \times 1$ vector; and
- $\boldsymbol{\epsilon}_k$ is a $n \times 1$ vector of serially uncorrelated, normally distributed disturbances with:

$$\mathbb{E}[\boldsymbol{\epsilon}_k] = 0 \quad \text{and} \quad \mathbb{C}ov[\boldsymbol{\epsilon}_k] = H_k$$

The transition equation can be found in the Appendix as

$$\boldsymbol{x}_k = T_k \boldsymbol{x}_{k-1} + c_k + R_k \boldsymbol{\eta}_k, \quad k = 1, ..., n_T \tag{6.15}$$

where:

- $\boldsymbol{x}_k = [\boldsymbol{X}_k, \boldsymbol{\delta}_k]$ is a 2×1 vector of state variables;
- $T_k = \begin{bmatrix} 1 & -\triangle k \\ 0 & 1 - \kappa \triangle k \end{bmatrix}$ is a 2×2 matrix;
- $c_k = [(\mu - \frac{1}{2}\sigma_1^2)\triangle k, \kappa\alpha\triangle k]$ is a 2×1 vector;
- R_k is a 2×2 identity matrix;
- $\boldsymbol{\eta}_k$ is a 2×1 vector of serially uncorrelated, normally distributed disturbances with $\mathbb{E}[\boldsymbol{\eta}_k] = 0$ and $\mathbb{V}ar[\boldsymbol{\eta}_k] = Q_k = \mathbb{C}ov[(\boldsymbol{X}_k, \boldsymbol{\delta}_k)] = \begin{bmatrix} \sigma_1^2 \triangle k & \rho_1 \sigma_1 \sigma_2 \triangle k \\ \rho_1 \sigma_1 \sigma_2 \triangle k & \sigma_1^2 \triangle k \end{bmatrix}$.

7 Filtering recursions for commodities

Here a general framework relevant to each of the commodity state space models is developed for recursive estimation of the latent states. This approach is typically known as filtering.

7.1 Filtering problem

The filtering problem considers the evolution of the state sequence $\boldsymbol{x}_k, k \in \mathbb{N}$ given by

$$\boldsymbol{x}_k = f_k\left(\boldsymbol{x}_{k-1}, \boldsymbol{v}_{k-1}\right) \tag{7.1}$$

where $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ is a possibly nonlinear function of the state $\boldsymbol{x}_{k-1}$, $\boldsymbol{v}_{k-1}, k \in \mathbb{N}$ is an i.i.d process noise sequence, n_x, n_v are dimensions of the state and process noise vectors, respectively. The objective of the problem is to recursively estimate $\boldsymbol{x}_k$ from the measurements

$$\boldsymbol{z}_k = h_k\left(\boldsymbol{x}_k, n_k\right) \tag{7.2}$$

where $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \to \mathbb{R}^{n_z}$ is a possibly nonlinear function, $n_k, k \in \mathbb{N}$ is an i.i.d. measurement noise sequence, and n_z, n_n are dimensions of the measurement and measurement noise vectors respectively.

From a Bayesian perspective, the aim of the filtering problem is to recursively calculate some degree of belief in the state $\boldsymbol{x}_k$ at time k, taking different values, given the data $\boldsymbol{z}_{1:k}$ up to time k. Thus, it is required to construct the p.d.f. $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$. It is essentially assumed that the initial p.d.f. $p(\boldsymbol{x}_0|\boldsymbol{z}_0) \equiv p(\boldsymbol{x}_0)$ of the state vector, which is also known as the prior, is available ($\boldsymbol{z}_0$ being the set of no measurements).

7.2 Model definition and assumptions

To be able to solve the problem of filtering, a model is required for the dynamics of the state and for the measurement process. It is often assumed that the $\boldsymbol{x}_k$ process is Markov, so the state at a time step, $\boldsymbol{x}_{k-1}$, is a sufficient statistic of the history of the process, $\boldsymbol{x}_{1:k-1}$. Since the state captures all the information known about the system, the state at a time step is also assumed a sufficient statistic of the history of measurements, $\boldsymbol{z}_{1:k-1}$. The current state, $\boldsymbol{x}_k$, is therefore independent of the history of states and measurements if the previous state, $\boldsymbol{x}_{k-1}$, is known:

$$p\left(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1}, \boldsymbol{z}_{1:k-1}\right) = p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{x}_{1:k-2}, \boldsymbol{z}_{1:k-1}\right) = p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right) \tag{7.3}$$

While in general, the measurement could be a function of the entire history of states, $\boldsymbol{x}_{1:k}$, and the previous measurements, $\boldsymbol{z}_{1:k-1}$, the case often encountered is that the measurement is independent of the history of states and the previous measurements:

$$p\left(\boldsymbol{z}_k|\boldsymbol{x}_{1:k}, \boldsymbol{z}_{1:k-1}\right) = p\left(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{x}_{1:k-1}, \boldsymbol{z}_{1:k-1}\right) = p\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right) \tag{7.4}$$

In principle, the procedure for solving the filtering problem consists of two stages: prediction and update.

7.2.1 Prediction stage

Suppose that the required p.d.f. $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ at time k-1 is available. The prediction stage involves using the state equation of $\boldsymbol{x}_k$ to obtain the prior p.d.f. $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ at time k via the Chapman-Kolmogorov equation (see, for example, [14] page 69)

$$\begin{aligned} p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}\right) &= \int p\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}\right) d\boldsymbol{x}_{k-1} \\ &= \int p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{1:k-1}\right) p\left(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}\right) d\boldsymbol{x}_{k-1} \\ &= \int p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right) p\left(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}\right) d\boldsymbol{x}_{k-1} \end{aligned} \tag{7.5}$$

In this equation, the fact that $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ has been used since the state equation is described as a Markov process of order one. The probabilistic of the state evolution $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is defined by the system equation of $\boldsymbol{x}_k$ and the known statistics of $\boldsymbol{v}_{k-1}$.

7.2.2 Update stage

At time step k, when a measurement $\boldsymbol{z}_k$ becomes available, it may be used to update the prior via Bayes' rule

$$p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}\right) = \frac{p\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}\right)}{p\left(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}\right)} \tag{7.6}$$

where the normalizing constant

$$p\left(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}\right) = \int p\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}\right) d\boldsymbol{x}_k \tag{7.7}$$

depends on the likelihood function $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ defined by the measurement equation and the known statistics of n_k. In the update stage, the measurement $\boldsymbol{z}_k$ is used to modify the prior density to obtain the required posterior density of the current state.

8 Filtering solutions for commodity state space models

In this section, derivation of the properties of estimation under a filtering framework in a state space model will be presented. In particular, derivation of optimality of the Kalman filter in the linear Gaussian setting will be considered ([10], [8], [1]).

8.1 Minimum mean square error criterion

In statistics, the mean square error (MSE) of an estimator is a method to quantify the difference between an estimator and the true value. MSE is the expected value of the squared error loss or quadratic loss, and it measures the average of the square of the error. Therefore, minimizing MSE is a key criterion in selecting estimators.

Here we denote $\hat{\boldsymbol{x}}_k(\boldsymbol{z}_{1:k})$ by $\hat{\boldsymbol{x}}_{k|k}$ - the state estimator at time k based on the observations up to time k. The conditional mean estimator yields

$$\hat{x}_{k|k} = \mathbb{E}\left[\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right] = \int \boldsymbol{x}_k p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k \tag{8.1}$$

This is a unique estimate which minimizes the MSE:

$$\mathbb{E}\left[\left\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\right\|^2 |\boldsymbol{z}_{1:k} = z_{1:k}\right] \tag{8.2}$$

and is also called the minimum error variance estimate, since

$$\mathbb{E}\left[\left\|\boldsymbol{x}_k - \hat{x}_{k|k}\right\|^2 |\boldsymbol{z}_{1:k} = z_{1:k}\right] \leq \mathbb{E}\left[\left\|\boldsymbol{x}_k - y\right\|^2 |\boldsymbol{z}_{1:k} = z_{1:k}\right] \tag{8.3}$$

for all estimators y = $f(\boldsymbol{z}_{1:k})$.

Proof

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{x}_k - f\left(\boldsymbol{z}_{1:k}\right)\right\|^2 |\boldsymbol{z}_{1:k} = z_{1:k}\right] \\
=&\int\int \left(\boldsymbol{x}_k - f\left(\boldsymbol{z}_{1:k}\right)\right)^2 p\left(\boldsymbol{x}_k, \boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k d\boldsymbol{z}_{1:k} \\
=&\int \left(\int \left(\boldsymbol{x}_k - f\left(\boldsymbol{z}_{1:k}\right)\right)^2 p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k\right) p\left(\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{z}_{1:k}
\end{aligned}$$

To minimize $\mathbb{E}\left[\left\|\boldsymbol{x}_k - f\left(\boldsymbol{z}_{1:k}\right)\right\|^2 |\boldsymbol{z}_{1:k} = z_{1:k}\right]$, we minimize $\int \left(\boldsymbol{x}_k - f\left(\boldsymbol{z}_{1:k}\right)\right)^2 p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k$

$$\begin{aligned}
&\frac{\partial}{\partial f}\int \left(\boldsymbol{x}_k - f\left(\boldsymbol{z}_{1:k}\right)\right)^2 p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k \\
=&\frac{\partial}{\partial f}\left\{\int \boldsymbol{x}_k^2 p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k - 2f\left(\boldsymbol{z}_{1:k}\right)\int \boldsymbol{x}_k p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k + f\left(\boldsymbol{z}_{1:k}\right)^2 \int p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k\right\} \\
=&-2\int \boldsymbol{x}_k p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k + 2f\left(\boldsymbol{z}_{1:k}\right)\int p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right) d\boldsymbol{x}_k \\
=&-2\mathbb{E}\left[\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right] + 2f\left(\boldsymbol{z}_{1:k}\right)
\end{aligned}$$

Set the last expression to zero, then: $f(\boldsymbol{z}_{1:k}) = \mathbb{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}]$

Remarks

Let $\tilde{\boldsymbol{x}}_{k|k}$ be the MSE estimation error, then it is given by

$$\tilde{\boldsymbol{x}}_{k|k} = \boldsymbol{x}_k - \hat{x}_{k|k} \tag{8.4}$$

The estimator $\hat{\boldsymbol{x}}_{k|k}$ has the following properties:

1. *$\hat{\boldsymbol{x}}_{k|k}$ is conditionally unbiased. That is,*

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right] = 0 \tag{8.5}$$

2. *$\hat{\boldsymbol{x}}_{k|k}$ is both conditionally and unconditionally unbiased:*

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right] = 0 \tag{8.6}$$

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}\right] = 0 \tag{8.7}$$

Proof

1. We use the above result that $\hat{\boldsymbol{x}}_{k|k} = \mathbb{E}\left[\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right]$. Then this follows that:

$$\begin{aligned}
&\mathbb{E}\left[\hat{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right] = \mathbb{E}\left[\boldsymbol{x}_k|\boldsymbol{z}_{1:k} = z_{1:k}\right] \\
&\Rightarrow \mathbb{E}\left[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right] = 0 \\
&\Rightarrow \mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right] = 0
\end{aligned}$$

2. We use the result of item 1: $\mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right] = 0$. Take expectation of both hand sides yields:

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{E}\left(\tilde{\boldsymbol{x}}_{k|k}|\boldsymbol{z}_{1:k} = z_{1:k}\right)\right] = 0 \\
&\Rightarrow \mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}\right] = 0
\end{aligned}$$

   where we use a property of conditional expectation that: $\mathbb{E}[\mathbb{E}[A|B]] = \mathbb{E}[A]$.

Result (Orthogonality principle)

In statistics, the orthogonality principle is a necessary and sufficient condition for the optimality of a Bayesian estimator. It says that the error associated with optimal estimate is orthogonal to any function of the observations with respect to the inner product $\mathbb{E}[\boldsymbol{x}\boldsymbol{y}^T]$. Thus, an estimator

is optimal according to the minimum mean square error criterion if and only if it is unbiased and the orthogonality condition holds, that is

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k}\right] = 0, \text{ and} \tag{8.8}$$

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}_{k|k} f\left(\boldsymbol{z}\right)\right] = 0 \tag{8.9}$$

The second condition is equivalent to

$$\tilde{\boldsymbol{x}} \perp \mathcal{Z}^k \tag{8.10}$$

where $\mathcal{Z}^k$ denotes the linear vector space generated by $\boldsymbol{z} = \boldsymbol{z}_{1:k}$. Hence, the least linear mean square estimator of $\boldsymbol{x}$ given $\boldsymbol{z}$ is simply the projection of $\boldsymbol{x}$ onto the space $\mathcal{Z}^k$, that is

$$\begin{aligned} \hat{\boldsymbol{x}} &= proj_{\mathcal{Z}} \boldsymbol{x} \\ &= \frac{\mathbb{E}\left[\boldsymbol{x}\boldsymbol{z}^T\right]}{\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^T\right]} \boldsymbol{z} \end{aligned} \tag{8.11}$$

As shown earlier, the conditional mean is an optimal estimator in the least mean square sense. Hence, from the orthogonality principle, it follows automatically that the conditional mean estimator is orthogonal to the linear space generated by the observations available at the current time.

We have so far made no assumptions on the state or observations, as well as have not defined any particular state space models yet. From now on, we will concentrate on problems with specific assumptions made on the state and the observations. In the following part, we consider the case where the state and the observations are jointly multivariate normal. This is particularly relevant for the Kalman filter optimal solution.

8.2 Properties of multivariate normal random variables

Suppose the state $\boldsymbol{x}_k$ and the observations $\boldsymbol{z}_{1:k}$ are jointly multivariate normal. For simplicity, we denote $\boldsymbol{x}_k$ by $\boldsymbol{x}$ and $\boldsymbol{z}_{1:k}$ by $\boldsymbol{z}$. Then the multivariate normal distribution of $\boldsymbol{x}$ and $\boldsymbol{z}$ is given by

$$f_{\boldsymbol{x},\boldsymbol{z}}\left(x,z\right) = \frac{1}{\sqrt{det\left(2\pi P\right)}} exp\left(-\frac{1}{2\pi}\left[\left(\boldsymbol{x}-\bar{x}\right)^T\left(\boldsymbol{z}-\bar{z}\right)^T\right] P^{-1} \begin{bmatrix} \boldsymbol{x}-\bar{x} \\ \boldsymbol{z}-\bar{z} \end{bmatrix}\right) \tag{8.12}$$

where $\bar{x}$ is the expectation of $\boldsymbol{x}$, $\bar{z}$ is the expectation of $\boldsymbol{z}$, and $P = \begin{bmatrix} P_{\boldsymbol{x}\boldsymbol{x}} & P_{\boldsymbol{x}\boldsymbol{z}} \\ P_{\boldsymbol{z}\boldsymbol{x}} & P_{\boldsymbol{z}\boldsymbol{z}} \end{bmatrix}$.

The following result (see [10], page 165) shows that the conditional distribution of jointly mul-

tivariate normal vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ is also multivariate normal with:

$$\mathbb{E}\left[\boldsymbol{x}|\boldsymbol{z}\right] = \bar{\boldsymbol{x}} + P_{\boldsymbol{xz}}P_{\boldsymbol{zz}}^{-1}\left(\boldsymbol{z} - \bar{z}\right) \tag{8.13}$$

$$\mathbb{C}ov\left[\boldsymbol{x}|\boldsymbol{z}\right] = P_{\boldsymbol{xx}|\boldsymbol{z}} = P_{\boldsymbol{xx}} - P_{\boldsymbol{xz}}P_{\boldsymbol{zz}}^{-1}P_{\boldsymbol{zx}} \tag{8.14}$$

In this Gaussian context, there are two points we need to note. Firstly, the conditional expectation of $\boldsymbol{x}$ given $\boldsymbol{z}$ is a linear function of the observations $\boldsymbol{z}$. Secondly, the covariance matrix $\mathbb{C}ov[\boldsymbol{x}|\boldsymbol{z}]$ is not a function of the observations $\boldsymbol{z}$ which may simplify the recursive computation of the Kalman filter recursions. These nice properties of the multivariate normal random variables then lead to the introduction of the Kalman filter which is a procedure for calculating the linear minimum MSE state estimator in linear state space models. The derivation and properties of the Kalman filter are discussed in the next section.

8.3 Filtering solution for Linear Gaussian case

In this section, we discuss the Kalman filter ([8], [10], [1]) and examine its important role for linear Gaussian models. Once a model is put in a state space form which is specified by equations (6.1) and (6.3), the Kalman filter emerges as an effective tool for computing the optimal estimator of the state vector at time k, based on the observations up to and including time k. The system matrices together with a_0 and P_0 are assumed to be known in all time periods and do not need to be explicitly included in the information set. Particularly, when the disturbances and the initial state vector are normally distributed, the Kalman filter enables the likelihood function to be calculated via the prediction error decomposition, and then this allows the estimation of any unknown parameters in the model. Moreover, it also provides the basis for statistical testing and model specification. The derivation of the Kalman filter presented below relies on the assumptions that the disturbances and initial state vector are normally distributed. A standard result on the multivariate normal distribution is then used to show how it is possible to calculate recursively the distribution of $\boldsymbol{x}_k$, conditional on the information set at time k, for all k from 1 to T. These conditional distributions are themselves normal and hence are completely specified by their means and covariance matrices. After deriving the Kalman filter, it is shown that the mean of the conditional distribution of $\boldsymbol{x}_k$ is an optimal estimator of $\boldsymbol{x}_k$ in the sense that it minimises the mean square error. When the normality assumption is omitted, it does not guarantee that the Kalman filter will give the conditional mean of the state vector. Nonetheless, it is still optimal in the sense that it minimises the mean square error within the class of all

linear estimators.

8.3.1 Derivation of the Kalman filter

Let $\boldsymbol{a}_{k-1}$ denote the mean of $\boldsymbol{x}_{k-1}$ conditional on the observations up to and including time k - 1, that is, $\boldsymbol{a}_{k-1} = \mathbb{E}(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$. Under the normality assumption, the initial state vector, $\boldsymbol{x}_0$, has a multivariate normal distribution with mean a_0 and covariance matrix P_0. The disturbances $\boldsymbol{\eta}_k$ and $\boldsymbol{\epsilon}_k$ also have multivariate normal distribution for k = 1, ..., T and are distributed independently of each other and of $\boldsymbol{x}_0$. The state vector at time k = 1 is given by

$$\boldsymbol{x}_1 = T_1\boldsymbol{x}_0 + c_1 + R_1\boldsymbol{\eta}_1. \tag{8.15}$$

Thus $\boldsymbol{x}_1$ is a linear combination of two vectors of random variables ($\boldsymbol{x}_0$ and $\boldsymbol{\eta}_1$), both with multivariate normal distributions, and a vector of constants (c_1). Hence it is itself multivariate normal with a mean of

$$a_{1|0} = T_1a_0 + c_1 \tag{8.16}$$

and a covariance matrix

$$P_{1|0} = T_1P_0T_1^T + R_1Q_1R_1^T. \tag{8.17}$$

The notation $a_{1|0}$ stands for the mean of the distribution of $\boldsymbol{x}_1$ conditional on the information at time k = 0. In order to obtain the distribution of $\boldsymbol{x}_1$ conditional on $\boldsymbol{z}_1$, we write

$$\boldsymbol{x}_1 = a_{1|0} + \left(\boldsymbol{x}_1 - a_{1|0}\right) \tag{8.18}$$

$$\boldsymbol{z}_1 = W_1a_{1|0} + d_1 + W_1\left(\boldsymbol{x}_1 - a_{1|0}\right) + \boldsymbol{\epsilon}_1. \tag{8.19}$$

Equation (8.19) is simply a rearrangement of the measurement equation. Furthermore, it can be seen that the vector $[\boldsymbol{x}_1^T \quad \boldsymbol{z}_1^T]^T$ has a multivariate normal distribution with a mean of $[a_{1|0}^T \ (W_1a_{1|0}+d_1)^T]^T$ and a covariance matrix $\mathbb{C}ov\left[\boldsymbol{x}_1^T \quad \boldsymbol{z}_1^T\right]^T = \begin{bmatrix} P_{1|0} & P_{1|0}W_1^T \\ W_1P_{1|0} & W_1P_{1|0}W_1^T + H_1 \end{bmatrix}$.

Now applying properties of multivariate normal random variables as described above, the distribution of $\boldsymbol{x}_1$, conditional on a particular value of $\boldsymbol{z}_1$, is multivariate normal with mean

$$a_1 = a_{1|0} + P_{1|0}W_1^TF_1^{-1}\left(\boldsymbol{z}_1 - W_1a_{1|0} - d_1\right) \tag{8.20}$$

and the covariance matrix

$$P_1 = P_{1|0} - P_{1|0}W_1^TF_1^{-1}W_1P_{1|0} \tag{8.21}$$

where

$$F_1 = W_1 P_{1|0} W_1^T + H_1 \tag{8.22}$$

and here we also assume that the inverse of F exists.

By repeating the above procedure to obtain equations (8.18)-(8.21) for k = 2, ..., T, we obtain the prediction and updating recursions for the Kalman filter as described in the next sub-section.

8.3.2 General form of the Kalman filter

In this section, we present the general form of the Kalman filter in the same construction as in [10]. The Kalman filter involves two recursions: prediction and updating. The *prediction equations* are given as follows

$$\boldsymbol{a}_{k|k-1} = T_k \boldsymbol{a}_{k-1} + c_k \tag{8.23}$$

$$P_{k|k-1} = T_k P_{k-1} T_k^T + R_k Q_k R_k^T \tag{8.24}$$

where $\boldsymbol{a}_{k|k-1}$ indicates the mean of the distribution of $\boldsymbol{x}_k$ conditional on the information up to time k-1 and $P_{k|k-1}$ is the covariance matrix of the estimation error.

Whenever the new observation, $\boldsymbol{z}_k$, becomes available, the estimator of $\boldsymbol{x}_k$, $\boldsymbol{a}_{k|k-1}$, can be updated. The *updating equations* are:

$$\boldsymbol{a}_k = \boldsymbol{a}_{k|k-1} + P_{k|k-1} W_k^T F_k^{-1} \left(\boldsymbol{z}_k - W_k \boldsymbol{a}_{k|k-1} - d_k \right) \tag{8.25}$$

and

$$P_k = P_{k|k-1} - P_{k|k-1} W_k^T F_k^{-1} W_k P_{k|k-1} \tag{8.26}$$

where[5]

$$F_k = W_k P_{k|k-1} W_k^T + H_k. \tag{8.27}$$

Taking equations (8.23)-(8.26) together forms the Kalman filter. Moreover, they can be written as recursions going directly from $\boldsymbol{a}_{k-1}$ to $\boldsymbol{a}_k$, or from $\boldsymbol{a}_{k|k-1}$, and this yields:

$$\boldsymbol{a}_{k+1|k} = \left(T_{k+1} - T_{k+1} K_k W_k \right) \boldsymbol{a}_{k|k-1} + T_{k+1} K_k \boldsymbol{z}_k + \left(c_{k+1} - T_{k+1} K_k d_k \right) \tag{8.28}$$

where the gain matrix K_k is given by

$$K_k = P_{k|k-1} W_k^T F_k^{-1}. \tag{8.29}$$

[5]It is assumed that the inverse of F exists. It can be replaced by a pseudo-inverse or a positive definite matrix.

Then the recursion for the error covariance matrix is

$$P_{k+1|k} = T_{k+1}\left(P_{k|k-1} - P_{k|k-1}W_k^T F_k^{-1} W_k P_{k|k-1}\right) T_{k+1}^T + R_{k+1}Q_{k+1}R_{k+1}^T \tag{8.30}$$

Here (8.30) is known as a *Riccati equation*.

The starting values for the Kalman filter may be specified in terms of $\boldsymbol{a}_0$ and P_0, or $\boldsymbol{a}_{1|0}$ and $P_{1|0}$. Given these initial conditions, the Kalman filter then provides the optimal estimator of the state vector once the new observation becomes available. Particularly, when all T observations have been processed, the filter then produces the optimal estimator of the current state vector, or the state vector in the next time period, based on the full information set. This estimator contains all information needed to make optimal predictions of future values of both the state and the observations.

8.3.3 Statistical properties of the Kalman filter

For Gaussian models, the Kalman filter produces the mean and covariance matrix of the distribution of $\boldsymbol{x}_k$ conditional on the information available at time k, and hence:

$$\boldsymbol{a}_k = \mathbb{E}\left[\boldsymbol{x}_k | \boldsymbol{z}_{1:k}\right] \tag{8.31}$$

and

$$P_k = \mathbb{E}\left[\left(\boldsymbol{x}_k - \boldsymbol{a}_k\right)\left(\boldsymbol{x}_k - \boldsymbol{a}_k\right)^T | \boldsymbol{z}_{1:k}\right]. \tag{8.32}$$

As was shown in the previous section, the conditional mean estimator $\boldsymbol{a}_k$ is the minimum mean square estimator of the state $\boldsymbol{x}_k$. The estimator is unbiased in the sense that the conditional expectation of the estimation error is zero. Moreover, this expectation can be taken over all variables in the observation set, and hence it is also unconditionally unbiased. Another point to note is that as pointed out in equation (8.14), the covariance matrix P_k is independent of the observations. Therefore, it can be referred to as an unconditional error covariance matrix associated with the conditional mean estimator. This implies that the right hand side of (8.32) can be written without being conditioned on the observation $\boldsymbol{z}_{1:k}$.

In general cases, when the normality assumption on the disturbances is dropped, it is no longer true that the Kalman filter produces the conditional mean of the state vector. This implies that (8.31) does not hold in these cases. Furthermore, the above points can be used to explain similarly for $\boldsymbol{a}_{k|k-1}$ and $P_{k|k-1}$. Let $\tilde{\boldsymbol{z}}_{k|k-1}$ denote the conditional mean of $\boldsymbol{z}_k$ at time k-1, that is:

$$\tilde{\boldsymbol{z}}_{k|k-1} = W_k \boldsymbol{a}_{k|k-1} + d_k. \tag{8.33}$$

Then $\tilde{\boldsymbol{z}}_{k|k-1}$ can be interpreted as the minimum mean square error (MMSE) of the observation $\boldsymbol{z}_k$ in a Gaussian model, and as the minimum mean square linear estimator in other cases.
Let $\boldsymbol{v}_k$ be the prediction errors of $\tilde{\boldsymbol{z}}_{k|k-1}$, that is:

$$\boldsymbol{v}_k = \boldsymbol{z}_k - \tilde{\boldsymbol{z}}_{k|k-1} = W_k \left(\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1}\right) + \boldsymbol{\epsilon}_k, \quad k = 1, ..., T. \tag{8.34}$$

Then $\boldsymbol{v}_k$ are known as the innovations, since they indicate the new information in the latest observation.
In a Gaussian model, it can be seen that the mean of $\boldsymbol{v}_k$ is a vector of zeros and its variance is F_k. Here F_k is given by (8.27). Moreover, these innovations are also independently and normally distributed, and therefore: $\boldsymbol{v}_k \sim NID(0, F_k)$.
In the cases without normality assumption, the expectation of the innovation vector is still a vector of zeros, and its covariance matrix at time k is F_k. Furthermore, it can be shown that the innovations in different time periods are uncorrelated, that is

$$\mathbb{E}\left[\boldsymbol{v}_k \boldsymbol{v}_s^T\right] = 0 \ \text{ for } k \neq s, \text{ and } k, \ s = 1, ..., T. \tag{8.35}$$

It should be noted that these properties on the distribution of the innovations only hold exactly if the system matrices are fixed and known. And hence, they are not true in general cases where these matrices contain unknown hyperparameters[6] which can be replaced by estimators.

Correlated measurement and transition equation disturbances

When the measurement and transition equation disturbances are correlated, we need to modify the Kalman filter. In this case, the modification of the Kalman filter depends on choosing the transition equation as in (6.3) or (6.8). The following results can be found in [10], page 112.
Consider the state space form given by (6.1) and (6.3), and suppose that

$$\mathbb{E}\left[\boldsymbol{\eta}_k \boldsymbol{\epsilon}_s^T\right] = \begin{cases} G_k, & \text{k = s} \\ 0, & \text{k} \neq \text{s} \end{cases} \tag{8.36}$$

where G_k is a known g x N matrix. The prediction equations (8.23) and (8.24) remain unchanged in the case of contemporaneous correlation, whereas the updating equations are modified as follows

$$\boldsymbol{a}_k = \boldsymbol{a}_{k|k-1} + \left(P_{k|k-1} W_k^T + R_k G_k\right) F_k^{-1} \left(\boldsymbol{z}_k - W_k \boldsymbol{a}_{k|k-1} - d_k\right) \tag{8.37}$$

$$\text{and} \quad P_k = P_{k|k-1} - \left(P_{k|k-1} W_k^T + R_k G_k\right) F_k^{-1} W_k P_{k|k-1}. \tag{8.38}$$

[6] In Bayesian statistics, a hyperparameter is a parameter of a prior distribution.

where

$$F_k = W_k P_{k|k-1} W_k^T + W_k R_k G_k + G_k^T R_k^T W_k^T + H_k. \tag{8.39}$$

The alternative state space form involves equations (6.1) and (6.8). By this construction, the correlation between the disturbances can be defined as in (8.36), however the model is a different one. The recursion for the state vector, (8.28), needs to be changed by redefining the gain matrix as

$$K_k = \left(P_{k|k-1} W_k^T + R_k G_k\right) F_k^{-1}. \tag{8.40}$$

The innovation covariance matrix, F_k, still remains as in (8.27). Whilst regarding the error covariance matrix, the recursion in (8.30) becomes

$$P_{k+1|k} = T_{k+1} P_{k|k-1} T_{k+1}^T - \left(T_{k+1} P_{k|k-1} W_k^T + R_k G_k\right) F_k^{-1} \left(T_{k+1} P_{k|k-1} W_k^T + R_k G_k\right)^T + R_k Q_k R_k^T. \tag{8.41}$$

Initial conditions and convergence of Kalman filter

In this sub-section, we consider the initial conditions for a time-invariant model. Here we restrict our attention to univariate series case, with the state space form given by

$$\boldsymbol{z}_k = w^T \boldsymbol{x}_k + d_k + \boldsymbol{\epsilon}_k \tag{8.42}$$

$$\boldsymbol{x}_k = T \boldsymbol{x}_{k-1} + c_k + R \boldsymbol{\eta}_k \tag{8.43}$$

$$\text{where} \quad \mathbb{V}ar\left[\boldsymbol{\epsilon}_k\right] = h \quad \text{and} \quad \mathbb{V}ar\left[\boldsymbol{\eta}_k\right] = Q. \tag{8.44}$$

In principle, the starting values for the Kalman filter are given by the mean and covariance matrix of the unconditional distribution of the state vector, that is,

$$\boldsymbol{a}_0 = \mathbb{E}\left[\boldsymbol{x}_0 | \boldsymbol{z}_0\right] = \mathbb{E}\left[\boldsymbol{x}_0\right] \tag{8.45}$$

$$P_0 = \mathbb{E}\left[\left(\boldsymbol{x}_0 - \boldsymbol{a}_0\right)\left(\boldsymbol{x}_0 - \boldsymbol{a}_0\right)^T | \boldsymbol{z}_0\right] = \mathbb{E}\left[\left(\boldsymbol{x}_0 - \boldsymbol{a}_0\right)\left(\boldsymbol{x}_0 - \boldsymbol{a}_0\right)^T\right]. \tag{8.46}$$

The state vector in (8.43) is *stationary* if $|\lambda_i(T)| < 1$ and c_k is time-invariant (that is $c_k = \mathrm{c}$ where c being constant). If this is the case, then it has mean $(I - T)^{-1} c$ and a covariance matrix P, which is the unique solution to the equation

$$P = T P T^T + R Q R^T. \tag{8.47}$$

The derivation of this equation can be found in chapter 8 of [10]. Since the unconditional distribution of $\boldsymbol{x}_1$ is equal to that of $\boldsymbol{x}_0$, we can choose the initial values for the Kalman filter

as $\boldsymbol{a}_0 = 0$, $P_0 = P$ or $\boldsymbol{a}_{1|0} = 0$, $P_{1|0} = P$. From equations (8.47) and (8.24), it can be seen that the covariance matrices P_0 and $P_{1|0}$ are consistent with each other.

When the state vector is non-stationary, its unconditional distribution is not easily determined. Unless genuine prior information is available, the initial distribution of $\boldsymbol{x}_0$ must be specified in terms of a diffuse or non-informative prior.

If we choose $P_0 = \kappa I$, where κ is a positive scalar, then the diffuse prior is obtained when $\kappa \to \infty$. This then leads to $P_0^{-1} = 0$. In addition, it is also possible to consider the diffuse prior applied to $\boldsymbol{x}_1$ by choosing $P_{1|0} = \kappa I$.

As was shown in [10] that the Kalman filter cannot be run if κ equals to infinity. However, by setting κ to be a large but finite number, a good approximation can be obtained.

It is also often the case that some of the elements in the state vector are stationary and some are non-stationary. If the non-stationary elements are taken to be the first d, where $d \leq m$, then the transition matrix must be of the form

$$T = \begin{bmatrix} T_1 & T_2 \\ 0 & T_4 \end{bmatrix} \tag{8.48}$$

where T_1 is a $d \times d$ matrix, T_2 is a $d \times (m-d)$ matrix and T_4 is a $(m-d) \times (m-d)$ matrix with $|\lambda(T_4)| < 1$. If $P_{1|0}$ is also partitioned conformably with (8.48), then the initial conditions can be chosen as

$$P_{1|0} = \begin{bmatrix} \kappa I & 0 \\ 0 & \mathrm{P} \end{bmatrix}. \tag{8.49}$$

In [10], it is shown that if w is partitioned conformably with T, that is $w^T = \begin{bmatrix} w_1^T & w_2^T \end{bmatrix}$, then a proper prior can be constructed from the first d observations provided that T_1 is non-singular and the first d elements of $\boldsymbol{x}_k$, namely the non-stationary ones, are observable, that is

$$\mathrm{Rank}\left[w_1, T_1^T w_1, \ldots, \left(T_1^T\right)^{d-1} w_1\right] = d. \tag{8.50}$$

We now turn to the issue on convergence of the filter. Given that the stationary part of a model is initialised in the Kalman filter by its unconditional mean and covariance matrix, and the non-stationary part is initialised with a diffuse prior. Assume that T_1 in (8.48) is non-singular and (8.50) holds. Convergence to the steady state is monotonic in the sense that the MSE matrix of $\boldsymbol{a}_{k|k-1}$ exceeds that of $\boldsymbol{a}_{k+1|k}$ by a positive semi-definite matrix, that is

$$P_{k|k-1} \geq P_{k+1|k},\ k = d+1, d+2, \ldots \tag{8.51}$$

This follows because there is no information at time k = 0 and so the estimator at k is based on more information than the estimator at k - 1, and $P_{k|k-1}$ does not depend on the actual observations. In addition, since $F_k = WP_{k|k-1}W^T + H$, W is of rank N and H is positive semi-definite, these together with (8.51) imply that

$$F_k \geq F_{k+1},\ k = d+1, d+2, \ldots. \tag{8.52}$$

From (8.52), it now follows that

$$|F_k| \geq |F_{k+1}| \geq |\Sigma| \geq 0 \tag{8.53}$$

where Σ is the steady state of F_k which is defined as $\lim_{k\to\infty} F_k = \Sigma = W\bar{P}W' + H$ (here $\bar{P}$ is such that $P_{k+1|k} = \bar{P}$ for all k) (see [10], page 120)

Result (Joseph form - see [3])

The subtraction in the covariance matrix $P_k = P_{k|k-1} - P_{k|k-1}W_k^T F_k^{-1} W_k P_{k|k-1}$ may lead to loss of symmetry and positive definiteness (for instance, negative eigenvalues may occur). The following form of Kalman filter recursion which is known as Joseph form can help to preserve symmetry and avoid negative eigenvalues for the covariance matrix:

$$P_k = (I - K_k W_k)\, P_{k|k-1}\, (I - K_k W_k)^T + K_k H_k K_k^T. \tag{8.54}$$

Proof

$$\begin{aligned}
P_k =& \mathbb{C}ov\left[\boldsymbol{x}_k - \boldsymbol{a}_k\right] \\
=& \mathbb{C}ov\left[\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1} - P_{k|k-1}W_k^T F_k^{-1}\left(\boldsymbol{z}_k - W_k\boldsymbol{a}_{k|k-1} - d_k\right)\right] \\
& \text{(using definition of } \boldsymbol{a}_k) \\
=& \mathbb{C}ov\left[\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1} - K_k\left(W_k\boldsymbol{x}_k + d_k + \boldsymbol{\epsilon}_k - W_k\boldsymbol{a}_{k|k-1} - d_k\right)\right] \\
& \text{(using definition of } K_k \text{ and } \boldsymbol{z}_k) \\
=& \mathbb{C}ov\left[\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1} - K_k\left(W_k\boldsymbol{x}_k - W_k\boldsymbol{a}_{k|k-1} + \boldsymbol{\epsilon}_k\right)\right] \\
=& \mathbb{C}ov\left[I\left(\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1}\right) - K_k W_k\left(\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1}\right) - K_k\boldsymbol{\epsilon}_k\right] \\
=& \mathbb{C}ov\left[\left(I - K_k W_k\right)\left(\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1}\right)\right] + \mathbb{C}ov\left[K_k\boldsymbol{\epsilon}_k\right] \\
=& \left(I - K_k W_k\right)\mathbb{C}ov\left[\boldsymbol{x}_k - \boldsymbol{a}_{k|k-1}\right]\left(I - K_k W_k\right)^T + K_k\mathbb{C}ov\left[\boldsymbol{\epsilon}_k\right]K_k^T \\
& \text{(using the properties of covariance)} \\
=& \left(I - K_k W_k\right) P_{k|k-1}\left(I - K_k W_k\right)^T + K_k H_k K_k^T.
\end{aligned}$$

8.3.4 Kalman filter Algorithm 1

: Assume that at time k - 1, P_{k-1} and $\boldsymbol{a}_{k-1}$ are known (initialised as P_0 and $\boldsymbol{a}_0$). The Kalman filter algorithm involves 6 steps as follows

1. $\boldsymbol{a}_{k|k-1} = T_k \boldsymbol{a}_{k-1} + c_k$.
2. $P_{k|k-1} = T_k P_{k-1} T_k^T + R_k Q_k R_k^T$.
3. $F_k = W_k P_{k|k-1} W_k^T + H_k$.
4. Receive the new observation $\boldsymbol{z}_k$.
5. $\boldsymbol{a}_k = \boldsymbol{a}_{k|k-1} + P_{k|k-1} W_k^T F_k^{-1} \left(\boldsymbol{z}_k - W_k \boldsymbol{a}_{k|k-1} - d_k\right)$.
6. $P_k = P_{k|k-1} - P_{k|k-1} W_k^T F_k^{-1} W_k P_{k|k-1}$.

8.4 Filtering solution for Non-linear Gaussian case

As described in the previous section, the Kalman filter provides an optimal solution to the filtering recursion problems when the functions of the state and the observation are linear and the disturbances are Gaussian. However, when one of these assumptions is dropped, it does not guarantee that we can obtain an optimal solution from the Kalman filter. In this section, we consider the case where the functions of the state and the observation are non-linear, while the normality conditions are still hold. From there on, we will present two extensions of the Kalman filter which can deal with the non-linear case (see [10]). The first one is called the Extended Kalman filter, and the second is the Unscented Kalman filter.

8.4.1 The Extended Kalman filter

Consider the non-linear state space model:

$$\boldsymbol{z}_k = f_k\left(\boldsymbol{x}_k\right) + \boldsymbol{\epsilon}_k \tag{8.55}$$

$$\boldsymbol{x}_k = h_k\left(\boldsymbol{x}_{k-1}\right) + R_k\left(\boldsymbol{x}_{k-1}\right)\boldsymbol{\eta}_k \tag{8.56}$$

where $f_k(\boldsymbol{x}_k)$ and $h_k(\boldsymbol{x}_{k-1})$ are $N \times 1$ and $m \times 1$ vectors respectively, and $R_k(\boldsymbol{x}_{k-1})$ is an $m \times g$ matrix. In addition, f_k and h_k are non-linear functions, and $R_k(\boldsymbol{x}_{k-1})$ may depend on the state vector.

Derivation of the Extended Kalman filter

If the non-linear functions f_k and h_k are sufficiently smooth, they then can be expanded in a

Taylor series around conditional means, $\hat{\boldsymbol{a}}_{k|k-1}$ and $\hat{\boldsymbol{a}}_{k-1}$ as follows

$$f_k(\boldsymbol{x}_k) \approx f_k\left(\hat{\boldsymbol{a}}_{k|k-1}\right) + \hat{W}_k\left(\boldsymbol{x}_k - \hat{\boldsymbol{a}}_{k|k-1}\right) \tag{8.57}$$

$$h_k(\boldsymbol{x}_{k-1}) \approx h_k(\hat{\boldsymbol{a}}_{k-1}) + \hat{T}_k(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{a}}_{k-1}) \tag{8.58}$$

and

$$R_k(\boldsymbol{x}_{k-1}) \approx \hat{R}_k \tag{8.59}$$

where

$$\hat{W}_k = \frac{\partial f_k(x)}{\partial x}|_{x=\hat{\boldsymbol{a}}_{k|k-1}} \tag{8.60}$$

$$\hat{T}_k = \frac{\partial h_k(x)}{\partial x}|_{x=\hat{\boldsymbol{a}}_{k-1}} \tag{8.61}$$

and

$$\hat{R}_k = R_k(\hat{\boldsymbol{a}}_{k-1}). \tag{8.62}$$

Suppose that $\hat{\boldsymbol{a}}_{k|k-1}$ and $\hat{\boldsymbol{a}}_{k-1}$ are known and substitute (8.57)-(8.62) to (8.55) and (8.56), the nonlinear model then becomes

$$\boldsymbol{z}_k \approx \hat{W}_k\boldsymbol{x}_k + \hat{d}_k + \boldsymbol{\epsilon}_k \tag{8.63}$$

$$\boldsymbol{x}_k \approx \hat{T}_k\boldsymbol{x}_{k-1} + \hat{c}_k + \hat{R}_k\boldsymbol{\eta}_k \tag{8.64}$$

where

$$\hat{d}_k = f_k(\hat{\boldsymbol{a}}_{k|k-1}) - \hat{W}_k\hat{\boldsymbol{a}}_{k|k-1} \tag{8.65}$$

and

$$\hat{c}_k = h_k(\hat{\boldsymbol{a}}_{k-1}) - \hat{T}_k\hat{\boldsymbol{a}}_{k-1}. \tag{8.66}$$

Here the quantities $\hat{\boldsymbol{a}}_k$ and $\hat{\boldsymbol{a}}_{k|k-1}$ are computed by applying the Kalman filter to (8.63)-(8.66). The prediction equations (8.23) and (8.24) then become

$$\hat{\boldsymbol{a}}_{k|k-1} = h_k(\hat{\boldsymbol{a}}_{k-1}) \tag{8.67}$$

$$\hat{P}_{k|k-1} = \hat{T}_k\hat{P}_{k-1}\hat{T}_k^T + \hat{R}_kQ_k\hat{R}_k^T \tag{8.68}$$

while the updating equations are given by

$$\hat{\boldsymbol{a}}_k = \hat{\boldsymbol{a}}_{k|k-1} + \hat{P}_{k|k-1}\hat{W}_k^T\hat{F}_k^{-1}\left[\boldsymbol{z}_k - f_k\left(\hat{\boldsymbol{a}}_{k|k-1}\right)\right] \tag{8.69}$$

$$\hat{P}_k = \hat{P}_{k|k-1} - \hat{P}_{k|k-1}\hat{W}_k^T\hat{F}_k^{-1}\hat{W}_k\hat{P}_{k|k-1} \tag{8.70}$$

where

$$\hat{F}_k = \hat{W}_k \hat{P}_{k|k-1} \hat{W}_k^T + H_k. \tag{8.71}$$

Extended Kalman filter Algorithm 2:

Assume that at time k - 1, $\hat{P}_{k-1}$ and $\hat{\boldsymbol{a}}_{k-1}$ are known (initialised as $\hat{P}_0$ and $\hat{\boldsymbol{a}}_0$). The EKF algorithm consists of 8 steps as follows:

1. Using $\hat{\boldsymbol{a}}_{k-1}$, compute $\hat{T}_k$.
2. $\hat{R}_k = R_k(\hat{\boldsymbol{a}}_{k-1})$.
3. $\hat{\boldsymbol{a}}_{k|k-1} = h_k(\hat{\boldsymbol{a}}_{k-1})$.
4. $\hat{P}_{k|k-1} = \hat{T}_k \hat{P}_{k-1} \hat{T}_k^T + \hat{R}_k Q_k \hat{R}_k^T$.
5. Receive new observation $\boldsymbol{z}_k$.
6. $\hat{F}_k = \hat{W}_k \hat{P}_{k|k-1} \hat{W}_k^T + H_k$.
7. $\hat{\boldsymbol{a}}_k = \hat{\boldsymbol{a}}_{k|k-1} + \hat{P}_{k|k-1} \hat{W}_k^T \hat{F}_k^{-1} \left[\boldsymbol{z}_k - f_k\left(\hat{\boldsymbol{a}}_{k|k-1}\right)\right]$.
8. $\hat{P}_k = \hat{P}_{k|k-1} - \hat{P}_{k|k-1} \hat{W}_k^T \hat{F}_k^{-1} \hat{W}_k \hat{P}_{k|k-1}$.

8.4.2 The Unscented Kalman filter

As mentioned in the previous section, the EKF is an improvement of the Kalman filter since it can be applied in the nonlinear systems. However, the EKF also has some drawbacks:

- Linearisation can produce unstable filters if the assumptions of local linearity of the model is violated;
- The derivation of the Jacobian matrices in most applications are often difficult to implement.

In this section, we present a new tool called the Unscented Kalman filter [12] which avoids the two drawbacks above. This filter is mainly based on the principle of the unscented transformation which is presented below.

The Unscented Transformation

The Unscented Transformation is a new method for calculating the statistics of a random variable which undergoes a nonlinear transformation. Consider propagating a random variable $\boldsymbol{x}$ (dimension n) through a nonlinear function $\boldsymbol{y} = f(\boldsymbol{x})$. Assume $\boldsymbol{x}$ has mean $\bar{x}$ and covariance P_{xx}. We wish to calculate the mean $\bar{y}$ and covariance P_{yy} of $\boldsymbol{y}$.

The transformation includes a set of sigma points which are chosen so that their mean and covariance are $\bar{x}$ and P_{xx}. The nonlinear function is then applied to each sigma point to produce a set of transformed points. Assume $\boldsymbol{x}$ has dimension n, then it is approximated by 2n + 1 weighted points given as follows

$$\mathcal{X}_0 = \bar{x}, \qquad W_0 = \frac{\kappa}{n+\kappa} \tag{8.72}$$

$$\mathcal{X}_i = \bar{x} + \left(\sqrt{(n+\kappa)\,P_{xx}}\right)_i, \quad W_i = \frac{1}{2\,(n+\kappa)} \tag{8.73}$$

$$\mathcal{X}_{i+n} = \bar{x} - \left(\sqrt{(n+\kappa)\,P_{xx}}\right)_i, \quad W_{i+n} = \frac{1}{2\,(n+\kappa)} \tag{8.74}$$

where $\kappa \in \mathbb{R}$, $(\sqrt{(n+\kappa)P_{xx}})_i$ is the ith row or column of the matrix square root of $(n+\kappa)P_{xx}$ and W_i is the weight associated with the ith sigma point. The transformation procedure then consists of 3 following steps:

1. Transform each sigma point via the nonlinear function f(.),

$$\mathcal{Y}_i = f\left[\mathcal{X}_i\right]. \tag{8.75}$$

2. The mean is given by the weighted average of the transformed points,

$$\bar{y} = \sum_{i=0}^{2n} W_i \mathcal{Y}_i. \tag{8.76}$$

3. The covariance matrix is then given by,

$$P_{yy} = \sum_{i=0}^{2n} W_i \mathcal{Y}_i - \bar{y}\mathcal{Y}_i - \bar{y}^T. \tag{8.77}$$

The Unscented Kalman filter

The transformation processes in the UKF consist of three following steps:

- Predict the new state $\hat{\boldsymbol{x}}(k+1|k)$ and its associated covariance $P(k+1|k)$. This prediction must take into account the effects of process noise.
- Predict the expected observation $\hat{\boldsymbol{z}}(k+1|k)$ and the innovation covariance $P_{vv}(k+1|k)$. This prediction should include the effects of observation noise.
- Finally, predict the cross-correlation matrix $P_{xz}(k+1|k)$.

Before running these steps, the state vector is augmented with the noise terms to give $n^a = n+q$ dimensional vector:

$$\boldsymbol{x}^a(k) = \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{v}(k) \end{bmatrix}. \tag{8.78}$$

The process model[7] is then expressed as a function of $\boldsymbol{x}^a(k)$:

$$\boldsymbol{x}(k+1) = f\left[\boldsymbol{x}^a(k), \boldsymbol{u}(k), k\right]. \tag{8.79}$$

Now given the estimate mean and covariance of the state vector at time k based on the information up to time k as follows

$$\hat{x}^a(k|k) = \begin{pmatrix} \hat{x}(k|k) \\ 0_{q\times 1} \end{pmatrix} \tag{8.80}$$

$$\text{and} \quad P^a(k|k) = \begin{bmatrix} P(k|k) & P_{xv}(k|k) \\ P_{xv}(k|k) & Q(k) \end{bmatrix}. \tag{8.81}$$

Now apply the transformation equations (8.72)-(8.74) to (8.80)-(8.81), the UKF yields $2n^a + 1$ sigma points. The next steps in the UKF can be seen in the algorithm below

Unscented Kalman filter Algorithm 3:

The algorithm of the UKF includes the following main steps:

1. Applying equations (8.72)-(8.74) to (8.80)-(8.81), we first obtain $2n^a + 1$ sigma points.
2. The transformed set is then given by replacing each sigma points to the process model

$$\boldsymbol{\mathcal{X}}_i(k+1|k) = f\left[\boldsymbol{\mathcal{X}}_i^a(k|k), \boldsymbol{u}(k), k\right].$$

3. The predicted mean is calculated as

$$\hat{x}(k+1|k) = \sum_{i=0}^{2n^a} W_i \boldsymbol{\mathcal{X}}_i^a(k+1|k).$$

4. The predicted covariance is calculated as

$$P(k+1|k) = \sum_{i=0}^{2n^a} W_i \left\{\boldsymbol{\mathcal{X}}_i(k+1|k) - \hat{x}(k+1|k)\right\}\left\{\boldsymbol{\mathcal{X}}_i(k+1|k) - \hat{x}(k+1|k)\right\}^T.$$

5. Instantiate each prediction sigma point via the observation model

$$\boldsymbol{\mathcal{Z}}_i(k+1|k) = h\left[\boldsymbol{\mathcal{X}}_i(k+1|k), \boldsymbol{u}(k), k\right].$$

[7] Here for simplicity, we adopt the notation used in [12].

6. The predicted observation is then computed as

$$\hat{z}\left(k+1|k\right)=\sum_{i=1}^{2n^a}W_i\boldsymbol{\mathcal{Z}}_i\left(k+1|k\right).$$

7. Since the observation noise is additive and independent, the innovation covariance is

$$P_{vv}\left(k+1|k\right)=R\left(k+1\right)+\sum_{i=0}^{2n^a}W_i\left\{\boldsymbol{\mathcal{Z}}_i\left(k|k-1\right)-\hat{z}\left(k+1|k\right)\right\}\left\{\boldsymbol{\mathcal{Z}}_i\left(k|k-1\right)-\hat{z}\left(k+1|k\right)\right\}^T.$$

8. The cross correlation matrix is determined by

$$P_{xz}\left(k+1|k\right)=\sum_{i=0}^{2n^a}W_i\left\{\boldsymbol{\mathcal{X}}_i\left(k|k-1\right)-\hat{x}\left(k+1|k\right)\right\}\left\{\boldsymbol{\mathcal{Z}}_i\left(k|k-1\right)-\hat{z}\left(k+1|k\right)\right\}^T.$$

9. $\mathcal{K}=P_{xz}\left(k+1|k\right)P_{vv}^{-1}\left(k+1|k\right)^{-1}$.

10. Receive the new information $\boldsymbol{z}_{k+1}$.

11. $\hat{x}\left(k+1|k+1\right)=\hat{x}\left(k+1|k\right)+\mathcal{K}\left(\boldsymbol{z}_{k+1}-\hat{z}\left(k+1|k\right)\right)$.

12. $P\left(k+1|k+1\right)=P\left(k+1|k\right)-\mathcal{K}P_{vv}\mathcal{K}^T$.

8.5 Filtering solution for Non-linear or Non-Gaussian case

In the case of non-linear and non-Gaussian (state equation and observation equation) in state space models, one can also consider more accurate, though more computational filtering solutions based on random grids. These approaches are known as Particle filters or sequential Monte Carlo (see [1], [8]).

8.5.1 Sequential Importance Sampling (SIS) Particle Filter

The sequential importance sampling (SIS) [1] algorithm is a Monte Carlo (MC) method that forms the basis for most sequential MC filters developed over the past decades. The idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this MC characterization becomes an equivalent representation to the usual functional description of the posterior p.d.f., and the SIS approaches the optimal Bayesian estimate.

Let $\left\{\boldsymbol{x}_{0:k}^i,w_k^i\right\}_{i=1}^{N_s}$ denote a random measure that characterizes the posterior p.d.f. $p(\boldsymbol{x}_{0:k}|z_{1:k})$, where $\boldsymbol{x}_{0:k}^i,i=0,...,N_s$ is a set of support points with associated weights $w_k^i,i=1,...,N_s$ and

$\boldsymbol{x}_{0:k} = \boldsymbol{x}^j, j = 0, ..., k$ is the set of all states up to time k. The weights are normalized such that $\sum_i w_k^i = 1$. Then the posterior density at k can be approximated as in [1]

$$p\left(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k}\right) \approx \sum_{i=1}^{N_s} w_k^i \tilde{\delta}\left(\boldsymbol{x}_{0:k} - \boldsymbol{x}_{0:k}^i\right). \tag{8.82}$$

The weights are chosen using the principle of importance sampling. This principle relies on the following. Suppose $p(\boldsymbol{x}) \propto \pi(\boldsymbol{x})$ is a probability density from which it is difficult to draw samples but for which $\pi(\boldsymbol{x})$ can be evaluated. In addition, let $\boldsymbol{x}^i \sim q(\boldsymbol{x})$, i = 1, ..., N_s be samples that are easily generated from a proposal q(.) called an importance density. Then a weighted approximation for the density p(.) is provided in [1] as

$$p\left(\boldsymbol{x}\right) \approx \sum_{i=1}^{N_s} w^i \tilde{\delta}\left(\boldsymbol{x} - \boldsymbol{x}^i\right) \tag{8.83}$$

where

$$w^i \propto \frac{\pi\left(\boldsymbol{x}^i\right)}{q\left(\boldsymbol{x}^i\right)} \tag{8.84}$$

is the normalized weight of the ith particle.

If the samples $\boldsymbol{x}_{0:k}^i$ were drawn from an importance density $q(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k})$, then the weights in (8.82) are defined by (8.84) to be

$$w_k^i \propto \frac{p\left(\boldsymbol{x}_{0:k}^i|\boldsymbol{z}_{1:k}\right)}{q\left(\boldsymbol{x}_{0:k}^i|\boldsymbol{z}_{1:k}\right)}. \tag{8.85}$$

If the importance density is chosen to factorize such that

$$q\left(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k}\right) = q\left(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{1:k}\right) q\left(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{1:k-1}\right) \tag{8.86}$$

then one can obtain samples $\boldsymbol{x}_{0:k}^i \sim q(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k})$ by augmenting each of the existing samples $\boldsymbol{x}_{0:k-1}^i \sim q(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{1:k-1})$ with the new state $\boldsymbol{x}_k^i \sim q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{1:k})$. To derive the weight update equation, $p(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k})$ is first expressed in terms of $p(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{1:k-1})$, $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ [7]

$$p\left(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k}\right) \propto p\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right) p\left(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{1:k-1}\right). \tag{8.87}$$

Now by substituting (8.86) and (8.87) into (8.85), the weight update equation can be shown to be [4]

$$w_k^i \propto w_{k-1}^i \frac{p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^i\right) p\left(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i\right)}{q\left(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{z}_{1:k}\right)}. \tag{8.88}$$

Furthermore, if $q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{1:k}) = q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k)$, then the importance density only depends on $\boldsymbol{x}_{k-1}$ and $\boldsymbol{z}_k$. This is particularly useful when only a filtered estimate of $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is required at each time step. From this point on, only $\boldsymbol{x}_k^i$ need be stored, hence one can discard the path $\boldsymbol{x}_{0:k-1}^i$ and history of observations $\boldsymbol{z}_{1:k-1}$. The modified weight is then

$$w_k^i \propto w_{k-1}^i \frac{p\left(\boldsymbol{z}_k|\boldsymbol{x}_k^i\right) p\left(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i\right)}{q\left(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i, \boldsymbol{z}_k\right)} \tag{8.89}$$

and the posterior filtered density $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ can be approximated as [6]

$$p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}\right) \approx \sum_{i=1}^{N_s} w_k^i \tilde{\delta}\left(\boldsymbol{x}_k - \boldsymbol{x}_k^i\right) \tag{8.90}$$

where the weights w_k^i are defined in (8.89). In addition, as $N_s \to \infty$, the approximation (8.90) approaches the true posterior density $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$.

SIS Particle Filter Algorithm (4)

$[\{\boldsymbol{x}_k^i, w_k^i\}_{i=1}^{N_s}] = SIS[\{\boldsymbol{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \boldsymbol{z}_k]$

FOR $i = 1 : N_s$

- *Draw* $\boldsymbol{x}_k^i \sim q\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i, \boldsymbol{z}_k\right)$.

- *Assign the particle a weight,* w_k^i*, according to (8.89).*

END FOR

A common problem with the SIS particle filter is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. This implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is almost zero. A suitable measure of degeneracy of the algorthim is the effective sample size N_{eff} defined as [6]

$$N_{eff} = \frac{N_s}{1 + \mathbb{V}ar\left[w_k^{*i}\right]} \tag{8.91}$$

where $w_k^{*i} = \frac{p(\boldsymbol{x}_k^i|\boldsymbol{z}_{1:k})}{q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i, \boldsymbol{z}_k)}$ is referred to as the "true weight". This cannot be evaluated exactly, but an estimate $\widehat{N_{eff}}$ of N_{eff} can be obtained by [16]

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2} \tag{8.92}$$

where w_k^i is the normalized weight obtained using (8.88).

The degeneracy problem is an undesirable effect in particle filters. The brute force approach to reducing its effect is to use a very large N_s. This is often impractical, hence we rely on two other methods: computationally efficient choice of importance density and use of resampling.

Computationally efficient choice of Importance Density:

This is an optimal choice. It involves choosing the importance density $q(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}, \boldsymbol{z}_k)$ to minimize $\mathbb{V}ar[w^{*i}_k]$ so that N_{eff} is maximized. The optimal importance density function that minimizes the variance of the true weights w^{*i}_k conditioned on $\boldsymbol{x}^i_{k-1}$ and $\boldsymbol{z}_k$ is shown in [1] as

$$q\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}, \boldsymbol{z}_k\right)_{opt} = p\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}, \boldsymbol{z}_k\right) \tag{8.93}$$

$$= \frac{p\left(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{x}^i_{k-1}\right) p\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}\right)}{p\left(\boldsymbol{z}_k|\boldsymbol{x}^i_{k-1}\right)} \tag{8.94}$$

Substituting (8.94) into (8.89) yields

$$w^i_k \propto w^i_{k-1} \int p\left(\boldsymbol{z}_k|\boldsymbol{x}^{'}_k\right) p\left(\boldsymbol{x}^{'}_k|\boldsymbol{x}^i_{k-1}\right) d\boldsymbol{x}^{'}_k \tag{8.95}$$

A disadvantage of this optimal importance density is that it requires the ability to sample from $p(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}, \boldsymbol{z}_k)$ and to evaluate the integral over the new state. Finally, it is often convenient to choose the importance density to be the prior

$$q\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}, \boldsymbol{z}_k\right) = p\left(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}\right) \tag{8.96}$$

Substituting (8.96) into (8.89) yields

$$w^i_k \propto w^i_{k-1} p\left(\boldsymbol{z}_k|\boldsymbol{x}^i_k\right) \tag{8.97}$$

This appears to be the most common choice of importance density since it is intuitive and simple to implement.

8.5.2 Resampling Algorithm

As discussed, the choise of important density is important to control particle degeneracy on the path space. Another important process to help decrease particle degeneracy is to consider adaptive resampling. This is the second method (besides the importance density choice) by which the effects of degeneracy can be reduced whenever a significant degeneracy is observed, i.e., when N_{eff} falls below some threshold N_T. The idea of resampling is to eliminate particles that have small weights and to concentrate on particles with large weights.

The resampling step involves generating a new set $\left\{\boldsymbol{x}^{i*}_k\right\}^{N_s}_{i=1}$ by resampling N_s times from an approximate discrete representation of $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ given by [1]

$$p\left(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}\right) \approx \sum_{i=1}^{N_s} w^i_k \tilde{\delta}\left(\boldsymbol{x}_k - \boldsymbol{x}^i_k\right) \tag{8.98}$$

so that $Pr(\boldsymbol{x}_k^{i*} = \boldsymbol{x}_k^j) = w_k^j$. The resulting sample is in fact an i.i.d. sample from the discrete density (8.98); therefore, the weights are now reset to $w_k^i = \frac{1}{N_s}$.

Although the resampling step reduces the effects of the degeneracy problem, it introduces other practical problems. First, it limits the chance to parallelize since all particles must be combined. Second, the particles that have high weights w_k^i are statistically selected many times. This leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. Third, since the diversity of the paths of the particles is reduced, any smoothed estimates based on the particles' paths degenerate.

Resampling Algorithm 5

$[\left\{\boldsymbol{x}_k^{j*}, w_k^j, i^j\right\}_{j=1}^{N_s}] = RESAMPLE[\{\boldsymbol{x}_k^i, w_k^i\}_{i=1}^{N_s}]$

- *Initialize the CDF:* $c_1 = 0$.
- *FOR* $i = 2$: N_s
- *Construct CDF:* $c_i = c_{i-1} + w_k^i$.
- *END FOR*
- *Start at the bottom of the CDF:* $i = 1$.
- *Draw a starting point:* $u_1 \sim \mathbb{U}[0, N_s^{-1}]$.
- *FOR* $j = 1$: N_s
- *Move along the CDF:* $u_j = u_1 + N_s^{-1}(j-1)$.
- *WHILE* $u_j > c_i$
  - $i = i + 1$.
- *END WHILE*
- *Assign sample:* $\boldsymbol{x}_k^{j*} = \boldsymbol{x}_k^i$.
- *Assign weight:* $w_k^j = N_s^{-1}$.
- *Assign parent:* $i^j = i$.
- *END FOR*

8.5.3 Sequential Importance Resampling Particle Filter

Sequential Importance Resampling Algorithm 6

$[\{\boldsymbol{x}_k^i, w_k^i\}_{i=1}^{N_s}] = PF[\{\boldsymbol{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \boldsymbol{z}_k]$

- *FOR* $i = 1$: N_s
- *Draw* $\boldsymbol{x}_k^i \sim q(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^i, \boldsymbol{z}_k)$.
- *Assign the particle a weight,* w_k^i, *according to (8.89).*

- *END FOR*
- *Calculate total weight:* $t = SUM[\{w_k^i\}_{i=1}^{N_s}]$.
- *For* $i = 1: N_s$

- *Normalize:* $w_k^i = t^{-1} w_k^i$.
- *END FOR*
- *Calculate* $\widehat{N_{eff}}$ *using (8.92).*
- *IF* $\widehat{N_{eff}} < N_T$

- *Resampling using algorithm 2:*

$*[\{\boldsymbol{x}_k^i, w_k^i, -\}_{i=1}^{N_s}] = RESAMPLE[\{\boldsymbol{x}_k^i, w_k^i\}_{i=1}^{N_s}]$.

- *END IF*

The generic particle filter is an innovation of the SIS one since it uses the resampling step to reduce the effect of the degeneracy problem. However, it also has some problems regarding to the resampling step as described above.

9 Commodity model filtering results

In this section, we present empirical results achieved from implementing the Kalman filter (for model 2) and the Particle filter (for model 3) based on the observed futures contracts. The data used to test the models consist of daily observations of futures contracts in a range of 100 days. For the Kalman filter study (using model 2), we investigate the effects of noise (low noise and high noise) and different structures of futures contracts on the performance of the Kalman filter. Particularly, we use different numbers of the futures contract, and the contracts with different lengths and different initializations of time to maturity on a single contract.

The values for the parameters (in models 2 and 3) which were used to generate the true paths are summarized in the following tables

κ	μ	σ_1	σ_2	ρ
1.7	1.1	1	1	0.5

Table 15: Parameters used to generate the truth for model 2

κ	$\hat{\alpha}$	a	m^*	σ_1	σ_2	σ_3	ρ_1	ρ_2	ρ_3
1.4	-0.0071	1.1	0.0636	2	0.5	0.1	0.8	0.3	0.1

Table 16: Parameters used to generate the truth for model 3

As discussed earlier, under normality assumptions on the initial state vector and the disturbances, the Kalman filter produces optimal estimators for the mean and covariance of the state vector at each particular time. We shall see this via the figures in each sub-study below. Moreover, for all simulations in these studies, we set the initial state vector and the disturbances to be normally distributed. Throughout all figures in this section, the black line represents the true mean, whereas the blue line stands for the estimate mean. We also plot the error bars for the estimate, and these bars stand for the variance of the estimate.

9.1 Two-factor model Kalman filter study

In this section, the Kalman filter was implemented in Matlab for the two-factor Euler discretized state space model of section 6. In this first analysis, we consider three studies:

- The effects of using a single futures contract (9.1.1).
- The effects of the observation noise (9.1.2).
- The effects of the futures contracts structure (9.1.3).

For the last study, we focus on four different structures of futures contracts. Firstly, we observe the effect of changing the number of contracts while fixing the same maturity for both of these contracts. Secondly, we consider the effect of a single contract with different lengths. Thirdly, we use a single contract with different time to maturity on initialization and observe its impact on the performance of the Kalman filter. Finally, we consider different structures of the observation noise in terms of its correlation over the futures maturities (i.e. futures curve) per day.

9.1.1 Single futures contract Kalman filter estimation study

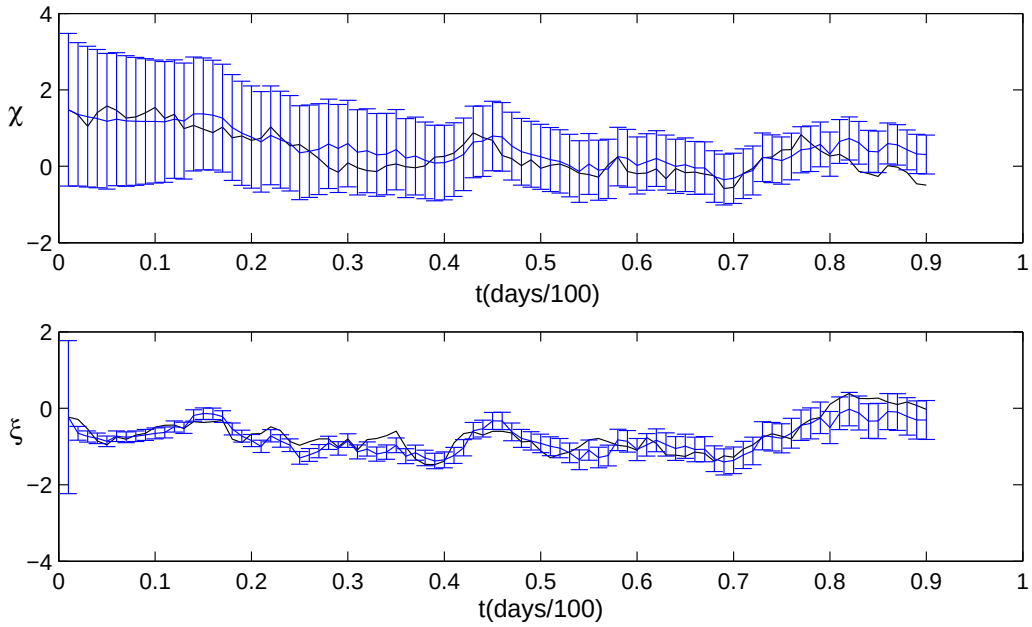

Figure 25: Model 2 - Single futures contract, length 90 days, estimated optimal Kalman filter states and precision at each day vs true latent process.

Figure 25 shows the result for the two factors in model 2 using the Kalman filter. Here we use only one futures contract with length 90 days and observe the performance of the Kalman filter over a period of 90 days. We can observe from this figure that the Kalman filter achieves very accurate results for both two factors in model 2 over the entire period. Particularly, for the

long-term factor ξ, over the first 30 days, the estimated mean almost matches the true mean. After the first 30 days, although the variance of the this factor expands, the estimated mean still remains very close to the truth.

9.1.2 Effects of the observation noise in futures contracts

For this sub-study, we use a single futures contract with length 90 days to investigate the effects of the observation noise on the performance of the Kalman filter. There are two types of noise ratios: low signal-to-noise ratio and high signal-to-noise ratio. Low signal-to-noise corresponds to a large value of the observation noise, whereas high signal-to-noise corresponds to a small value of the observation disturbance. Here we choose $\sigma_{obs} = 10$ for the first case, whereas $\sigma_{obs} = 0.1$ for the latter case.

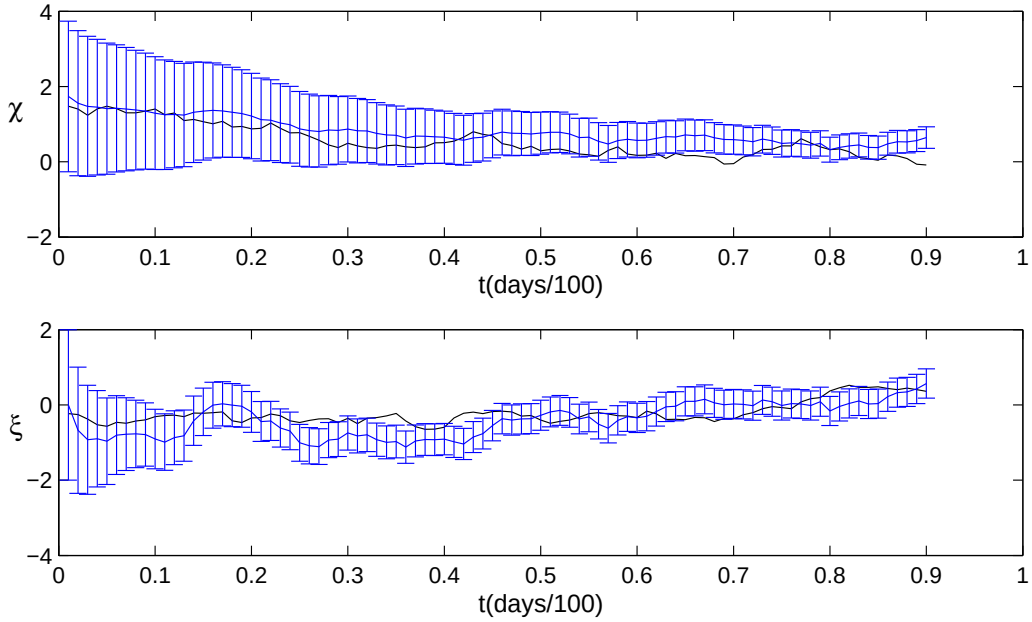

Figure 26: Model 2 - Low signal-to-noise ratio, single contract, length 90 days, estimated Kalman filter states and precision at each day vs true latent process.

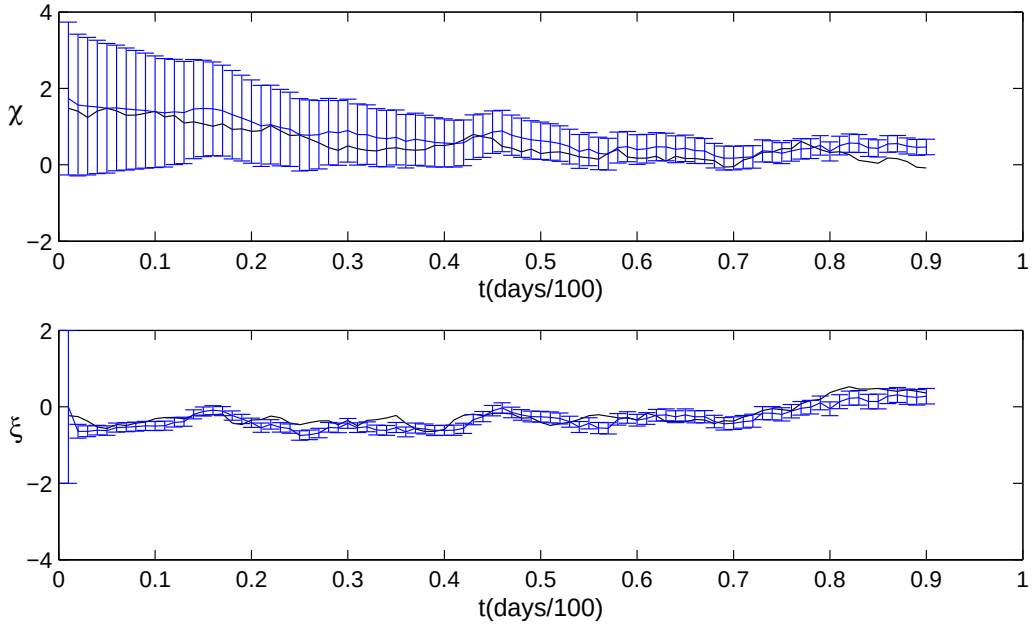

Figure 27: Model 2 - High signal-to-noise ratio, single contract, length 90 days, estimated Kalman filter states and precision at each day vs true latent process.

From figures 26 and 27, we first observe that the Kalman filter appears to give better results

in the high SNR case than in the low SNR case. In addition, the variance of the simulated mean tends to fall very fast in the high SNR case (especially for the long-term factor). This indicates that as the disturbances are small (according to high SNR case), then there appears less uncertainty in the short- and long-term factors. This together with the fact that the Kalman filter yields optimal variance at each time explain why the variance tends to be smaller very quickly in the high SNR case. As a result, the Kalman filter performs better in the high SNR case due to a fast reduction in variance and an optimal estimator for the mean at each time. In some sense, this implies that if we would like to get a good simulation for the true mean by using Kalman filter, then we need to adjust the disturbances in model 2 to be smaller. Otherwise, it may lead to a worse simulation for the factors in the model.

9.1.3 Effects of the futures contracts structure on Kalman filter estimation performance

In this sub-section, we investigate the effects of the futures contracts structure on the performance of the Kalman filter. In fact, there are three main structures of the futures contract which may affect the result of the Kalman filter: the number of futures contracts, the length of a single contract and the initialization of time to maturity on a single contract.

The number of futures contracts: For this study, we consider using three different numbers of futures contracts: 1, 3 and 5 respectively. All these contracts have the same maturity of 30 days, and we observe them in a range of 100 days.

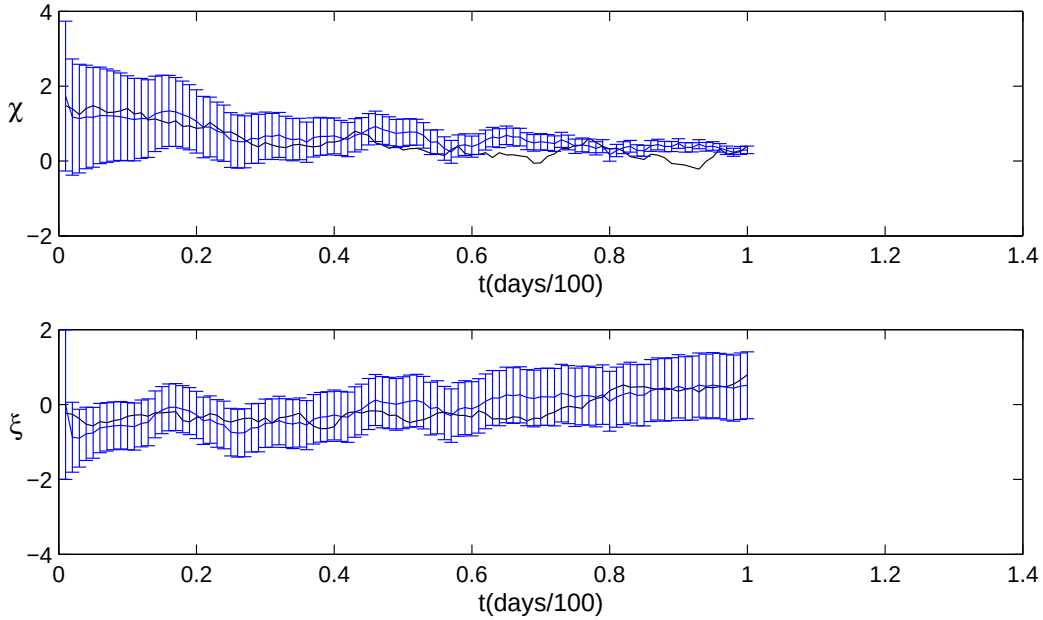

Figure 28: Model 2 - One futures contract, maturity 30 days, estimated Kalman filter states and precision at each day vs true latent process.

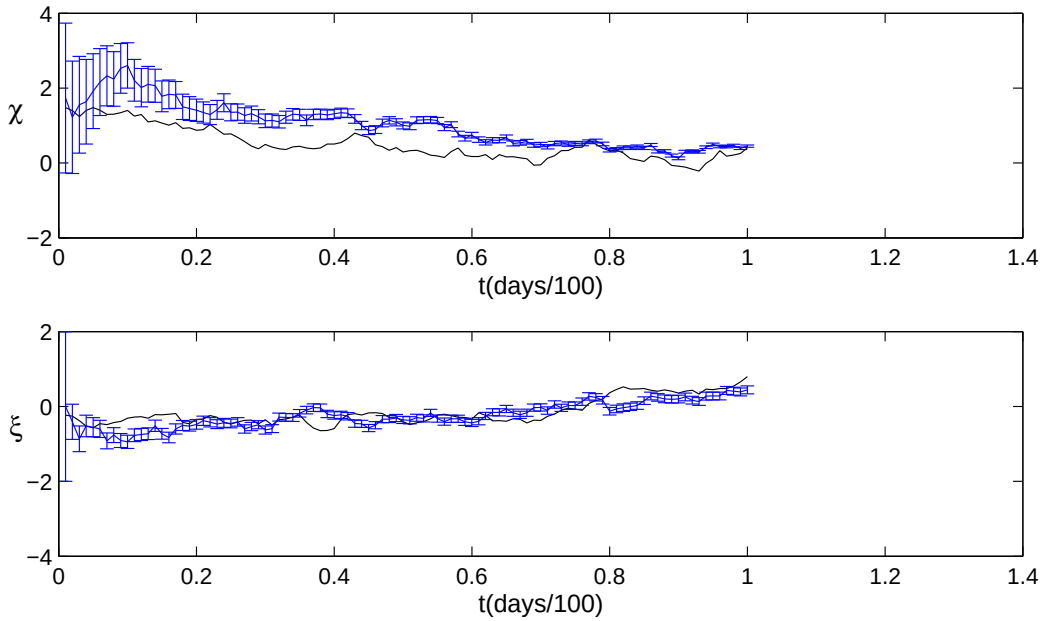

Figure 29: Model 2 - Three futures contracts, maturity 30 days, estimated Kalman filter states and precision at each day vs true latent process.

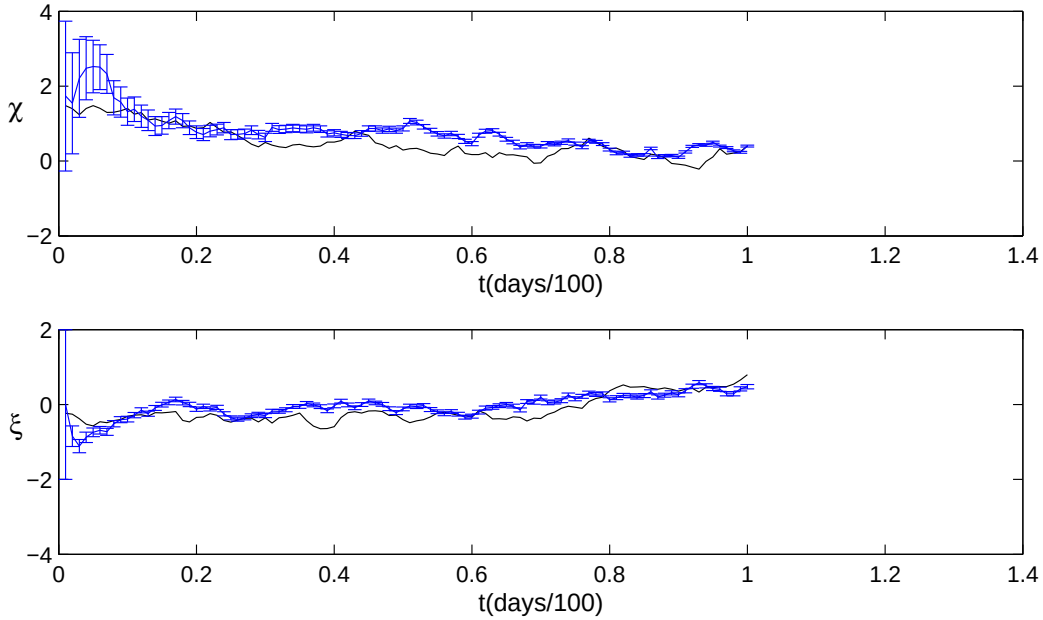

Figure 30: Model 2 - Five futures contracts, maturity 30 days, estimated Kalman filter states and precision at each day vs true latent process.

From figures 28, 29 and 30, we observe that:

- As the number of contracts increases, the accuracy of the Kalman filter improves with respect to precision in the state estimate.

- Moreover, as the number of contracts increases, we see that the mean estimate is not as accurate as in the single contract case. The reason for this is due to the model assumptions made. We assumed the correlation between observation noise on a given date t for all futures contracts was identical, that is $\sigma(F_{t,T}^{(1)}) = \sigma(F_{t,T}^{(2)}) = \ldots$. This assumption makes filtering easier but clearly affects the performance of the Kalman filter. In the last study of the futures contracts structure analysis, this assumption is removed and we shall see the performance of the mean estimate is improved as the number of contracts increases.

The length of futures contracts: For this study, we utilise three different lengths of a single futures contract: 30, 60 and 90 days to examine the effect of length of a futures contract on the performance of the Kalman filter. The range of the observed period for each case corresponds to the length of the contract, i.e. 30, 60 and 90 days respectively.

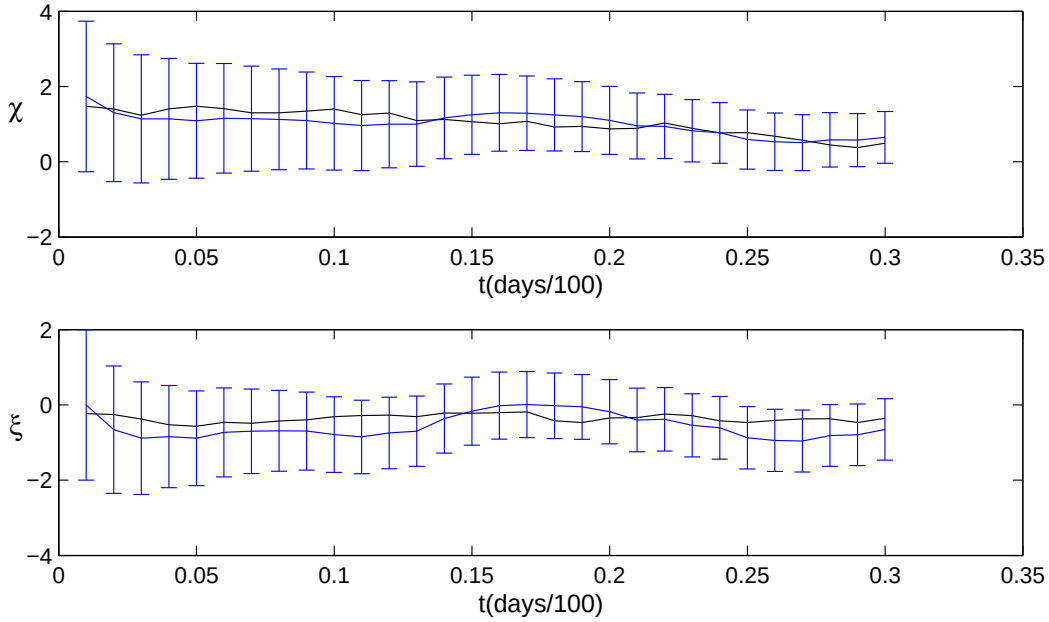

Figure 31: Model 2 - One futures contract, length 30 days, estimated Kalman filter states and precision at each day vs true latent process.

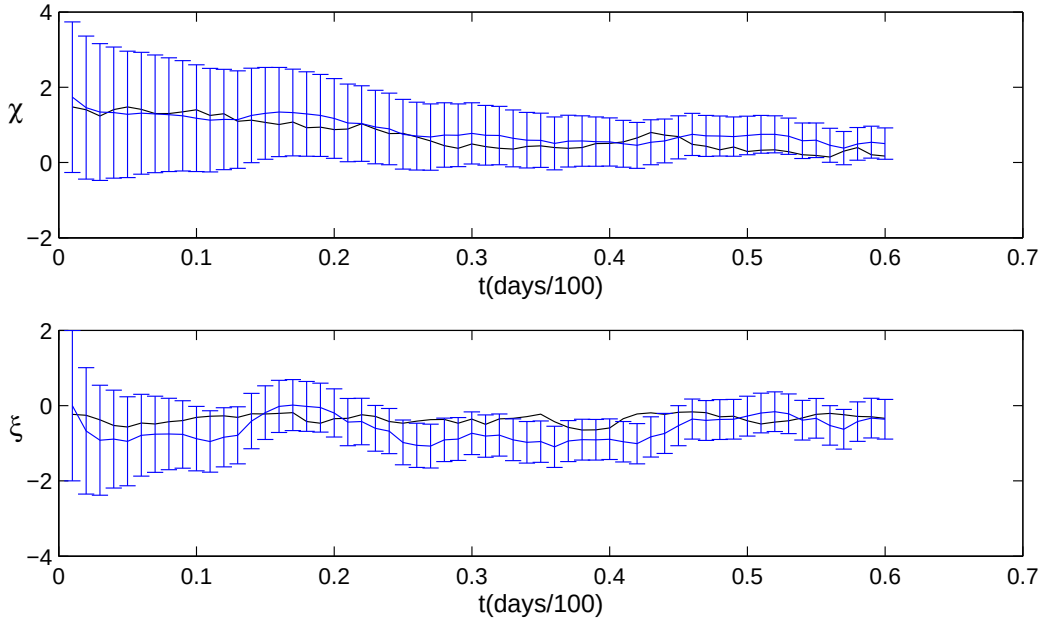

Figure 32: Model 2 - One futures contract, length 60 days, estimated Kalman filter states and precision at each day vs true latent process.

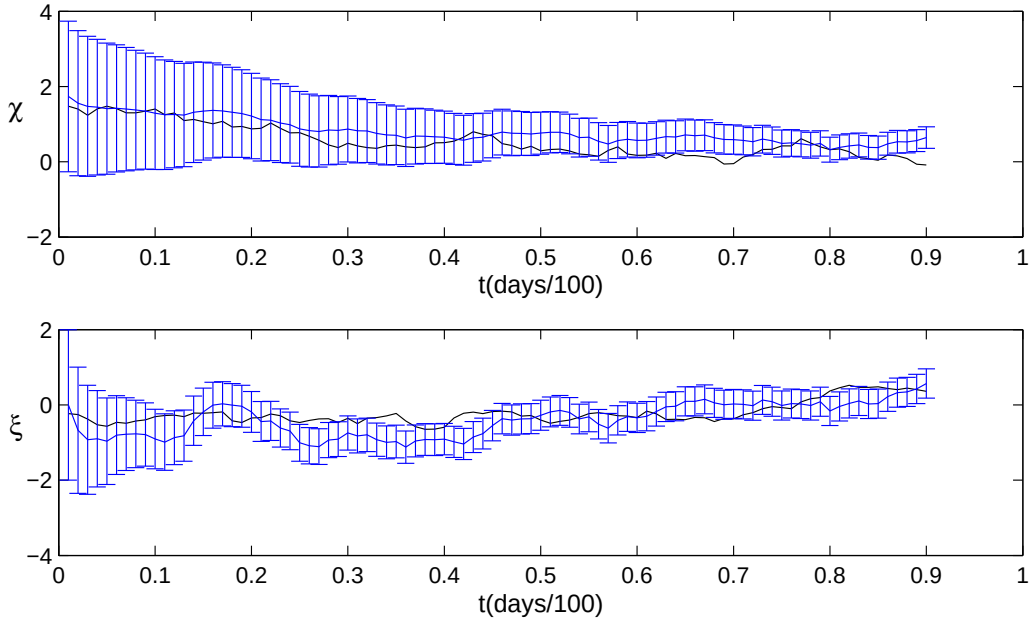

Figure 33: Model 2 - One futures contract, length 90 days, estimated Kalman filter states and precision at each day vs true latent process.

From figures 31, 32 and 33, we first see that the variance of the simulation (for both short- and long-term factors) tends to decrease as time increases. As a result, when we expand the observed period (or the length of the contract), then the Kalman filter tends to achieve better result at the end of the period (or at the maturity of the contract). In some sense, this implies that the estimated mean converges to the truth when the length of the contract is extended. This can be seen clearly from the result for the long-term factor (ξ) in figure 33 where the estimation and the truth are almost indistinguishable and are covered in a very small variance of the simulation at the maturity (day 90). However, it should also be noted that, the estimate mean of the short-term factor appears to be "far" away from the truth at the maturity. To some extent, this reveals that the Kalman filter estimate may not give good result as the length of the contract is extended. Nonetheless, the Kalman filter, in general, still works very well over the entire observed period for both two factors in model 2.

The effect of time to maturity on initialization of futures contracts on the Kalman filter performance: In this study, we examine the effect of time to maturity on initialization of a single futures contract. We utilise one futures contract with length 90 days. The initialization includes 5 days to maturity and 60 days to maturity respectively.

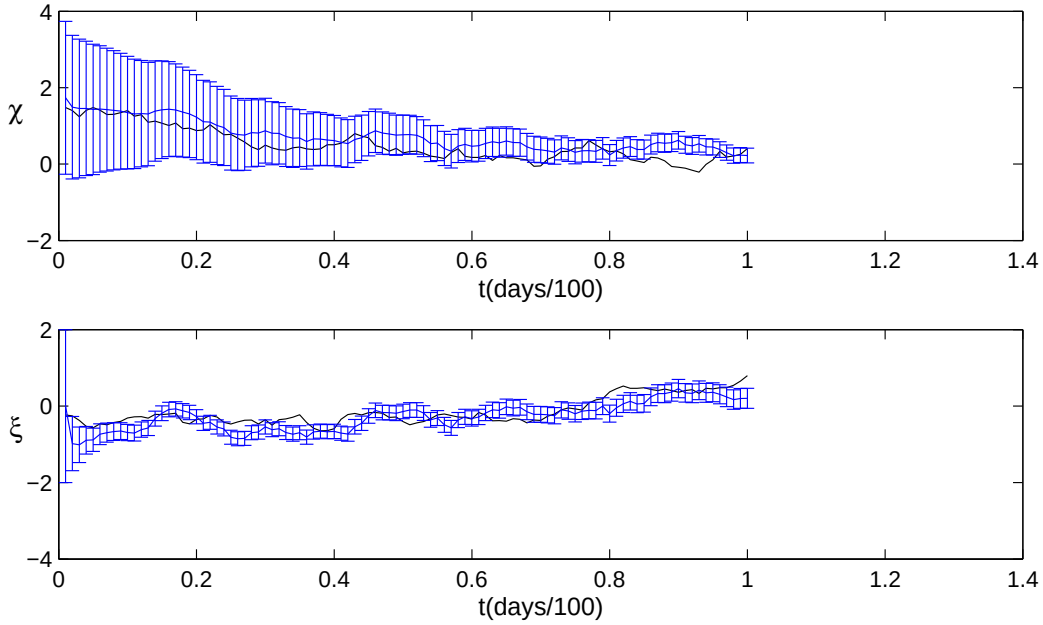

Figure 34: Model 2 - One futures contract, length 90 days, initialization of 5 days to maturity, estimated Kalman filter states and precision at each day vs true latent process.

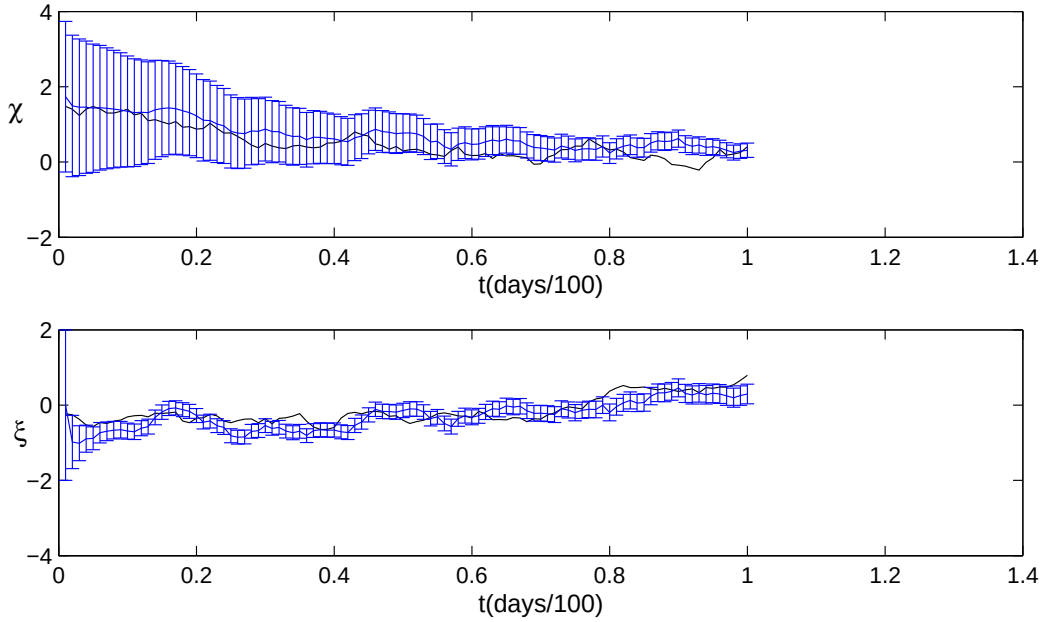

Figure 35: Model 2 - One futures contract, length 90 days, initialization of 60 days to maturity, estimated Kalman filter states and precision at each day vs true latent process.

From figures 34 and 35, we can see that although there are some differences at some points

between these cases, the results for both cases are almost indisguishable. This implies that the change in initialization with respect to time to maturity of a single contract does not lead to a significant change in the results of the Kalman filter.

The effect of correlation on the observation noise: For this study, we utilise five futures contracts (maturity of 30 days) and observe the effect of the correlation on the observation noise. Particularly, this study aims to examine the effect of different structure of the covariance matrix of the observation noise for a futures curve on a given date t, i.e. $\mathbb{C}ov(\widehat{F}(t,T_1),\widehat{F}(t,T_2),\ldots,\widehat{F}(t,T_5))$. We mainly focus on three cases of the covariance matrix of the observation noise but still diagonal in structure. The first case is when the correlations of noise are the same for all five futures contracts. The second case is when the correlations of noise increase from the first contract to the last contract. The third case is when the correlations of noise decrease from the first contract to the final contract. These studies now account for varying temporal dependence in the futures curves as well as dependence due to differing maturities on a given day t.

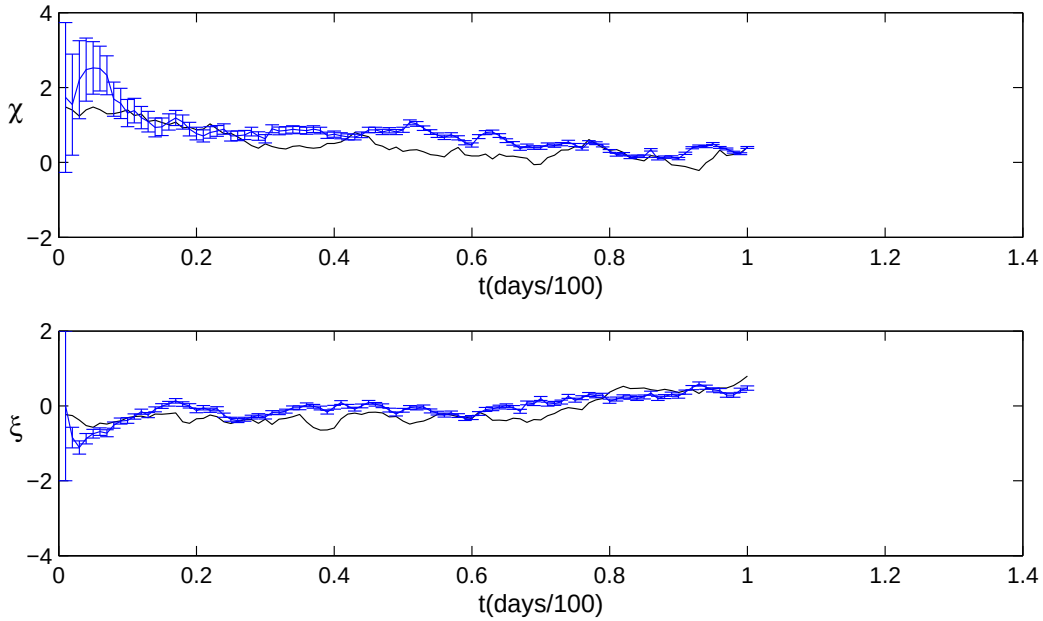

Figure 36: Model 2 - Five futures contract, maturity 30 days, constant correlation on the observation noise, estimated Kalman filter states and precision at each day vs true latent process.

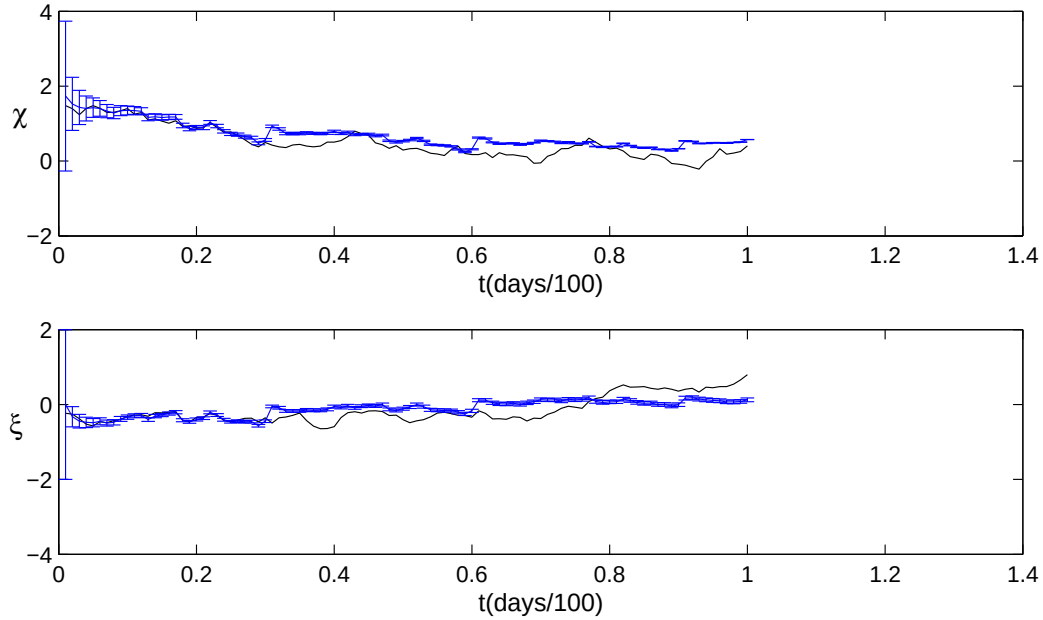

Figure 37: Model 2 - Five futures contract, maturity 30 days, increasing correlation on the observation noise, estimated Kalman filter states and precision at each day vs true latent process.

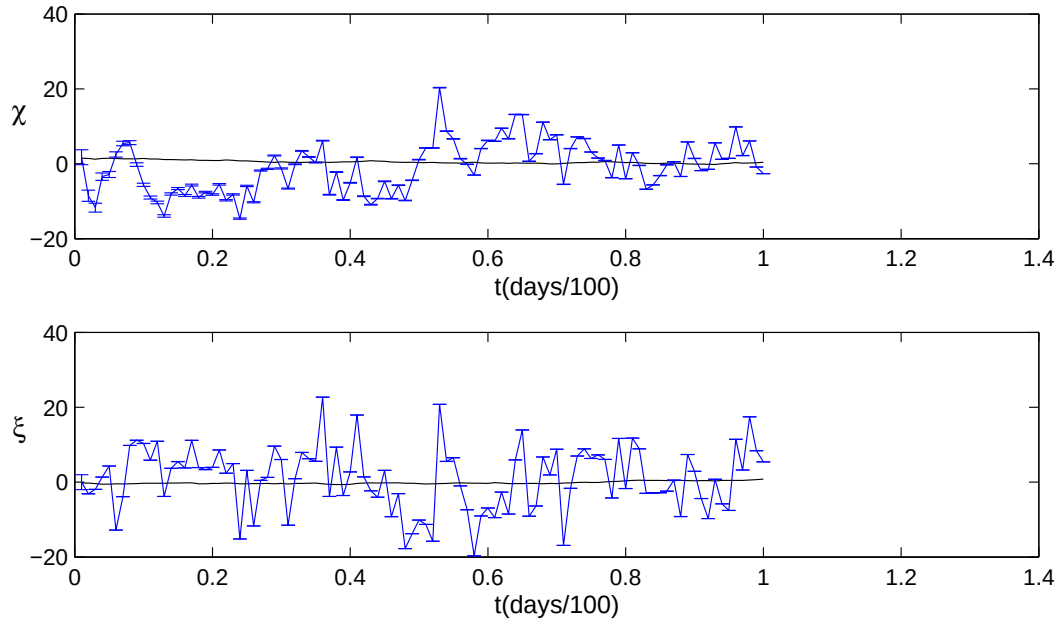

Figure 38: Model 2 - Five futures contract, maturity 30 days, decreasing correlation on the observation noise, estimated Kalman filter states and precision at each day vs true latent process.

From figures 36, 37 and 38, we first observe that there appear most uncertainty in the constant

correlation case, and for the last case (decreasing correlation), there is almost no uncertainty over the entire period for both two factors. In addition, it should be noted that the Kalman filter achieves very accurate and stable results for the constant and increasing cases during the first 30 days, and these results get worse as the observed period is expanded. On the other hand, although there is almost no uncertainty in the decreasing correlation case, the Kalman filter appears to produce much less accurate estimation for the truth in this third case. In some sense, these results imply that the Kalman filter performance gets worse as the correlations on the observation noise becomes descending from the first contract to the last contract. In addition, the Kalman filter performs well and stable as the correlations on the observation noise are constants (or increasing) across the futures contracts.

9.2 Three-factor model Extended Kalman filter study

We note that under the Euler scheme, the three-factor model still keeps the Gaussian property for both three factors. However, the spot price factor in this model is no longer linear for which the Kalman filter may not work well. In this second study, we apply the Extended Kalman filter to the non-linear case of the state space model of model 3. In addition, the data used to test model 3 using the Extended Kalman filter involve one and five futures contracts respectively with the same maturity of 30 days.

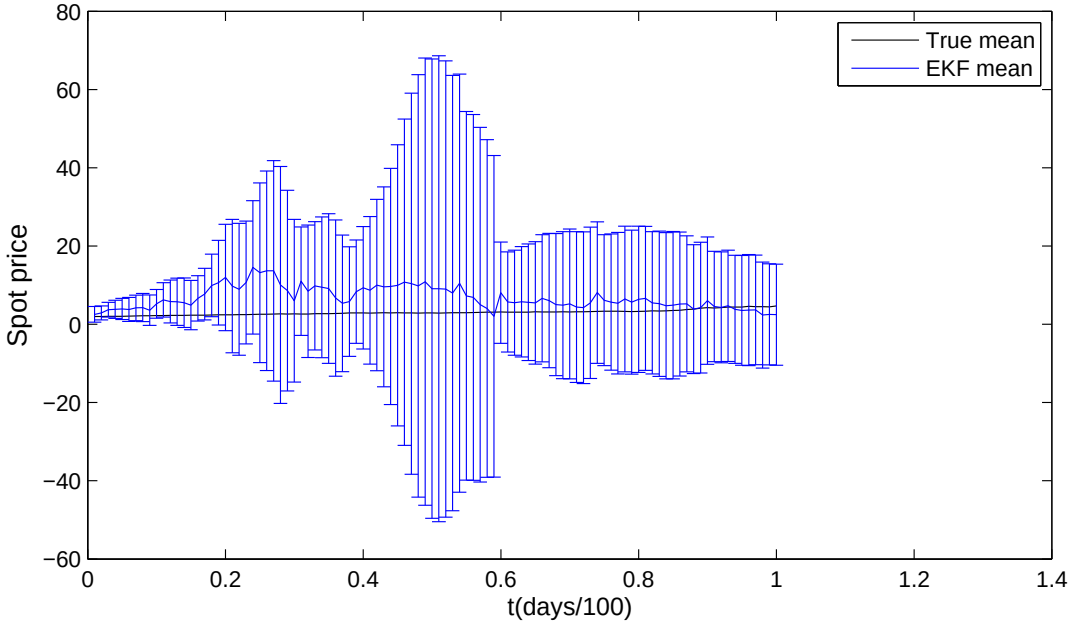

Figure 39: Spot price factor (model 3) - one futures contract, maturity 30 days, estimated Extended Kalman filter states and precision at each day vs true latent process.

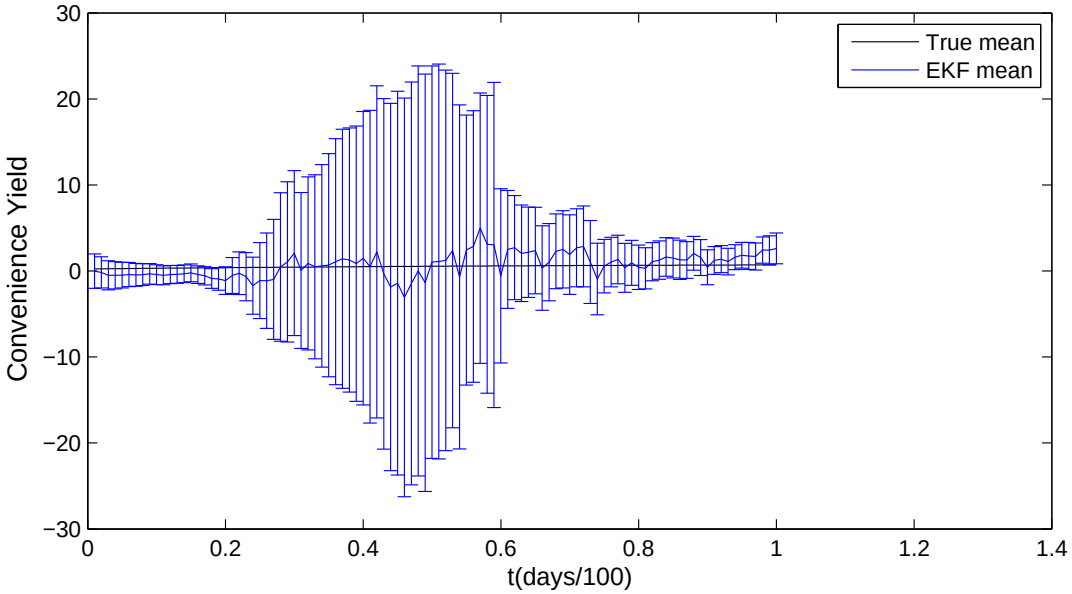

Figure 40: Convenience yield factor (model 3) - one futures contract, maturity 30 days, estimated Extended Kalman filter states and precision at each day vs true latent process.

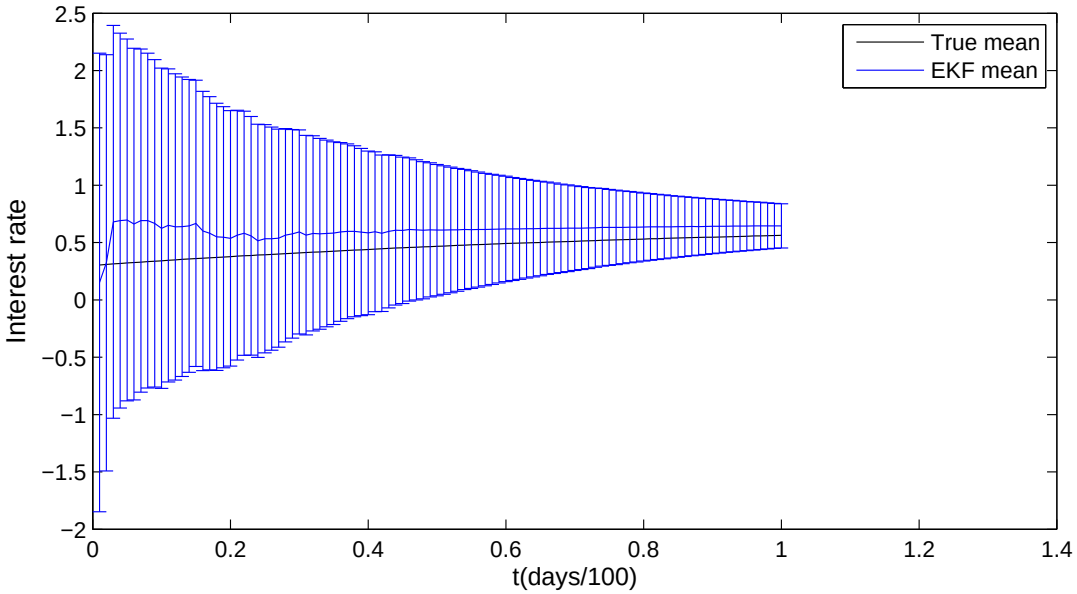

Figure 41: Interest rate factor (model 3) - one futures contract, maturity 30 days, estimated Extended Kalman filter states and precision at each day vs true latent process.

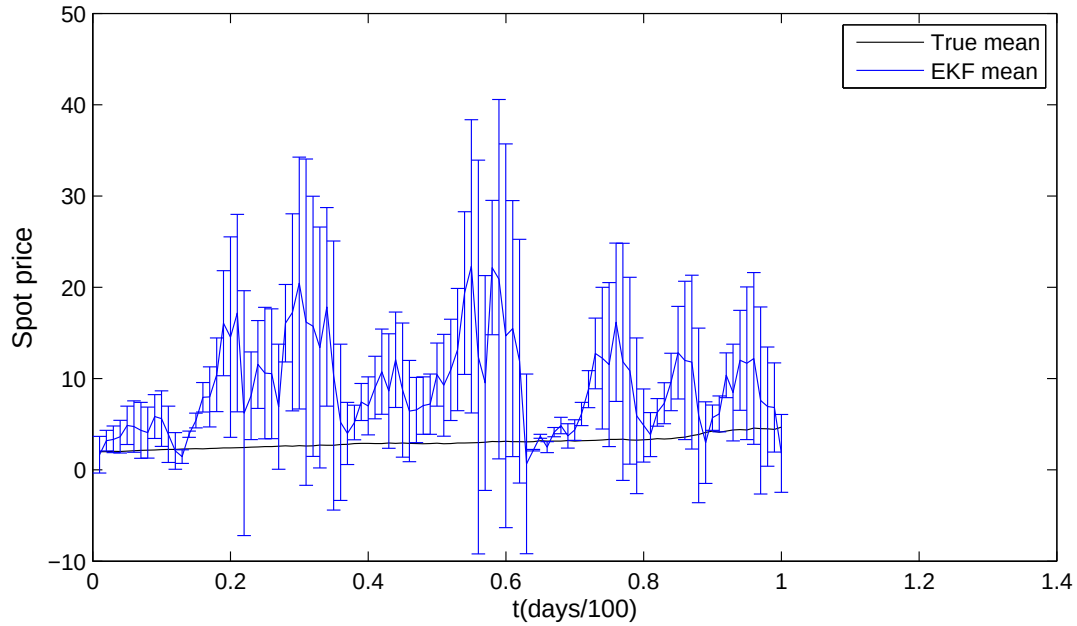

Figure 42: Spot price factor (model 3) - five futures contract, maturity 30 days, estimated Extended Kalman filter states and precision at each day vs true latent process.

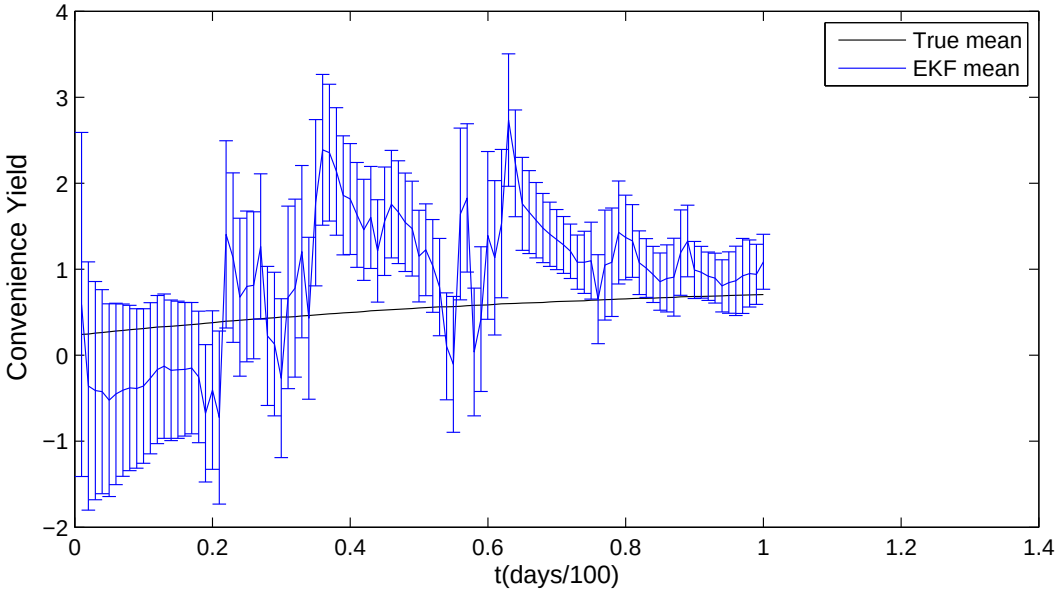

Figure 43: Convenience yield factor (model 3) - five futures contract, maturity 30 days, estimated Extended Kalman filter states and precision at each day vs true latent process.

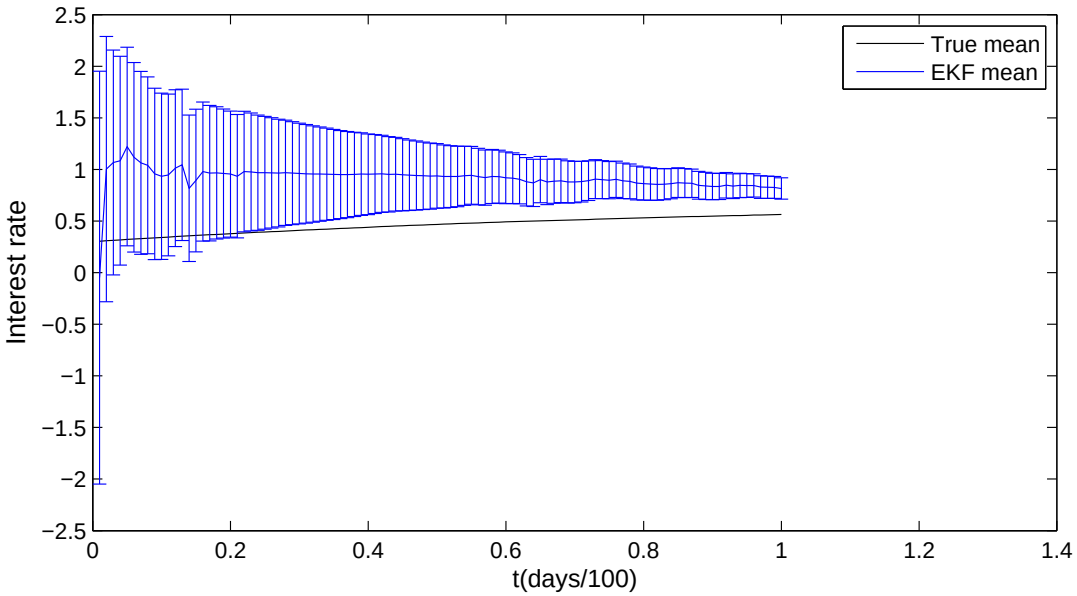

Figure 44: Interest rate factor (model 3) - five futures contract, maturity 30 days, estimated Extended Kalman filter states and precision at each day vs true latent process.

From figures 39, 40, 41, 42, 43, 44, we first observe that the Extended Kalman filter appears to give more accurate results for both three factors in the one contract case than in the five contracts case. Moreover, we could see from figures 39, 40 and 42, 43 that there is a strong

correlation between the spot price and the convenience yield, and this does affect the performance of the EKF. Particularly, for figures 39 and 40, a significant large variance in the spot price (at day 50) leads to a large variance in the convenience yield, and vice versa. In some sense, the correlation between the factors may affect considerably the performance of the EKF. In general, the Extended Kalman filter performance gets worse as the number of contracts increases. Otherwise, it achieves very accurate result in the one contract case, even though there appear more uncertainty in each factor in this case.

9.3 Three-factor model Particle filter study

9.3.1 One futures contract case

In this last study, we observe the performance of the Particle filter for the three-factor model. We shall in turn examine one and five futures contracts. For each particular number of contracts used, we consider utilising 1000 particles and 5000 particles and compare the performance of the Particle filter for these cases.

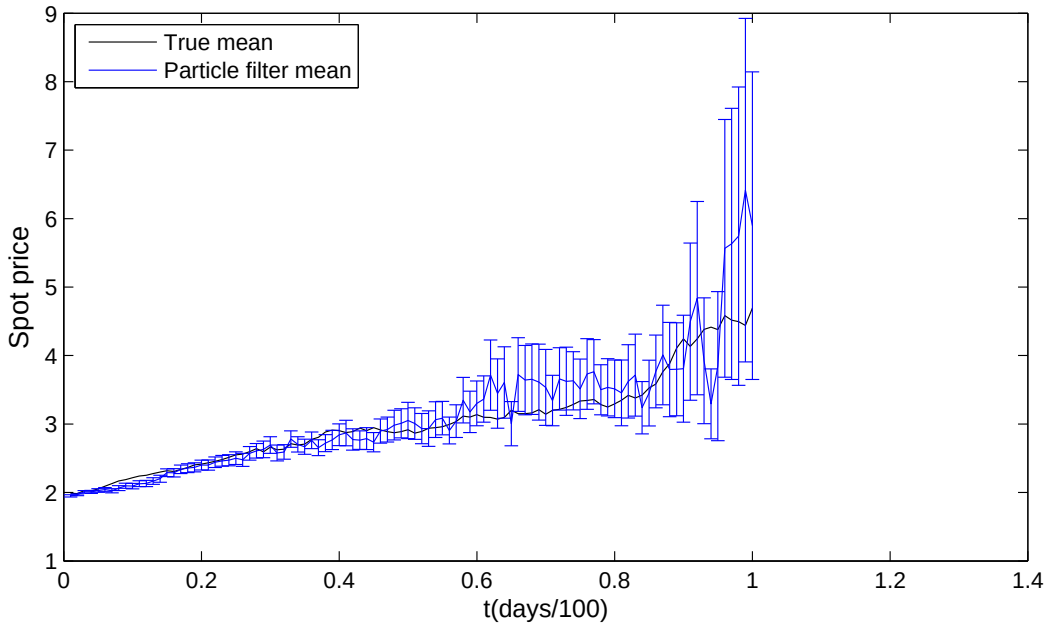

Figure 45: Spot price factor (model 3) - one futures contract, maturity 30 days, 1000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

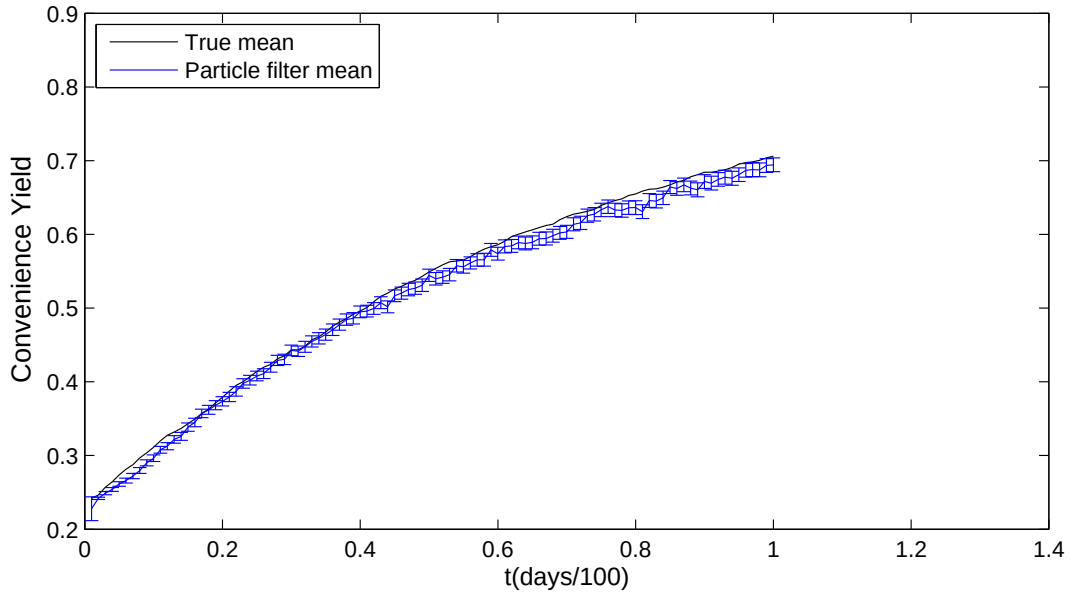

Figure 46: Convenience yield factor (model 3) - one futures contract, maturity 30 days, 1000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

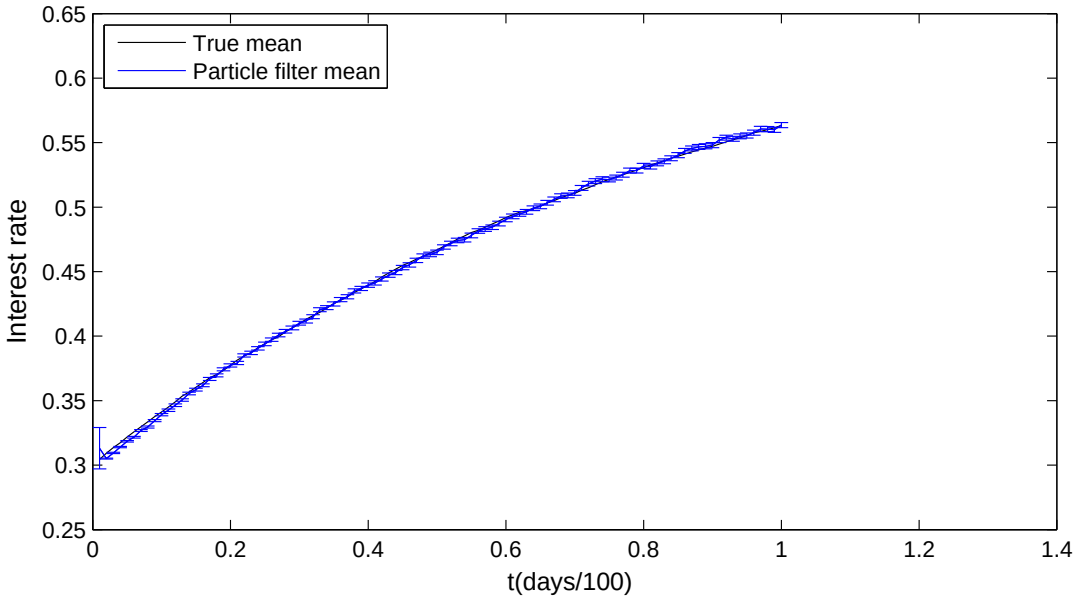

Figure 47: Interest rate factor (model 3) - one futures contract, maturity 30 days, 1000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

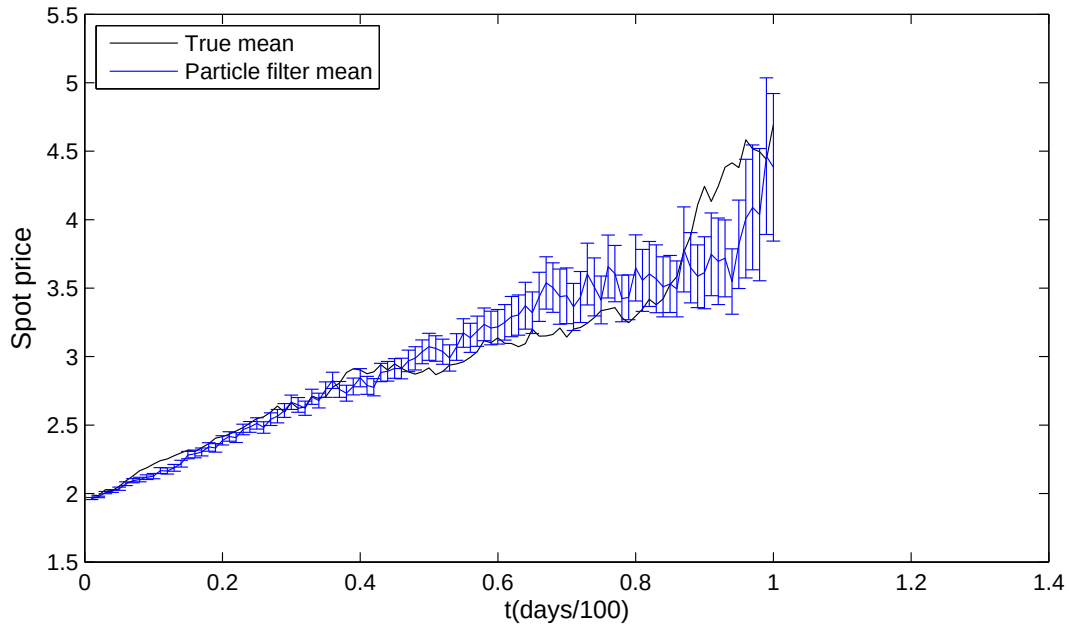

Figure 48: Spot price factor (model 3) - one futures contract, maturity 30 days, 5000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

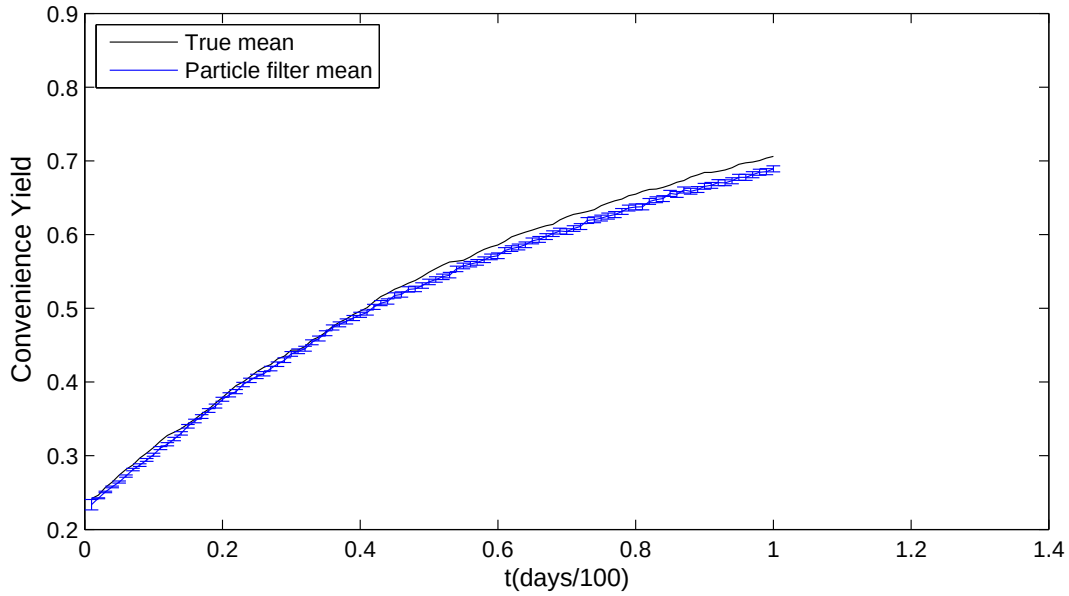

Figure 49: Convenience yield factor (model 3) - one futures contract, maturity 30 days, 5000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

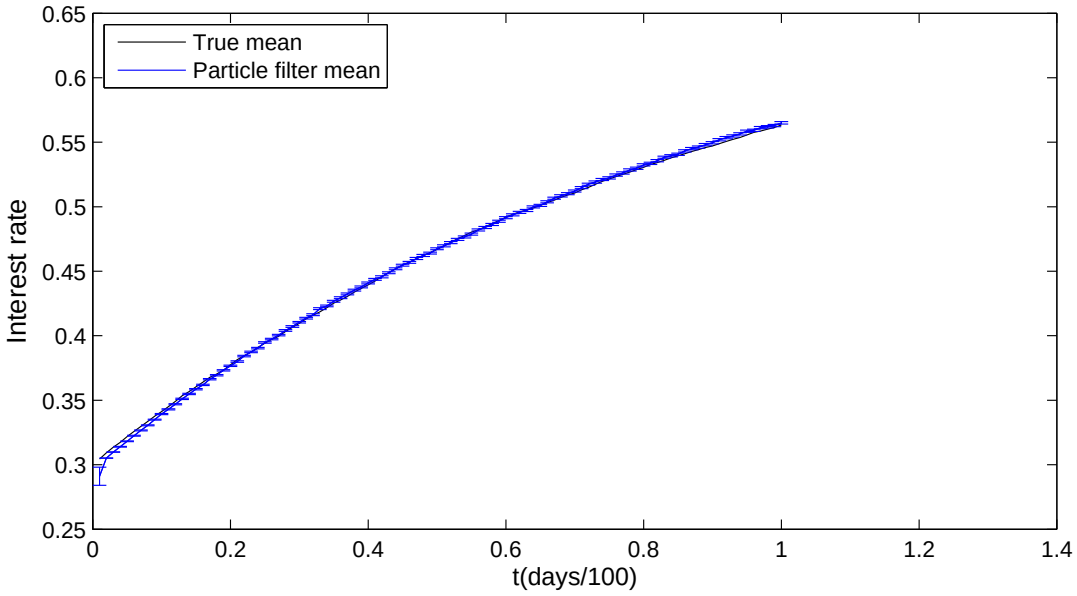

Figure 50: Interest rate factor (model 3) - one futures contract, maturity 30 days, 5000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

From figures 45, 46, 47, 48, 49, 50, it can be observed that as we utilise more particles, the Particle filter estimate becomes more accurate for both three factors in model 3. Moreover, increasing simulation particles also leads to a reduction in the variance of the estimate. Particularly, for the spot price factor, the Particle filter using 5000 particles produces very accurate simulation at the maturity (day 100) in the sense that the estimated mean gets very close to the truth as compared with the 1000 particles case. The variance of the estimate in the 5000 particles case is just a half of that in the 1000 case at maturity. It should also be noted that the Particle filter achieves very good results for both the convenience yield and interest rate. This may be due to the fact that the convenience yield and interest rate have special structures in the SDE where their diffusion terms are both constants.

9.3.2 Five futures contracts case

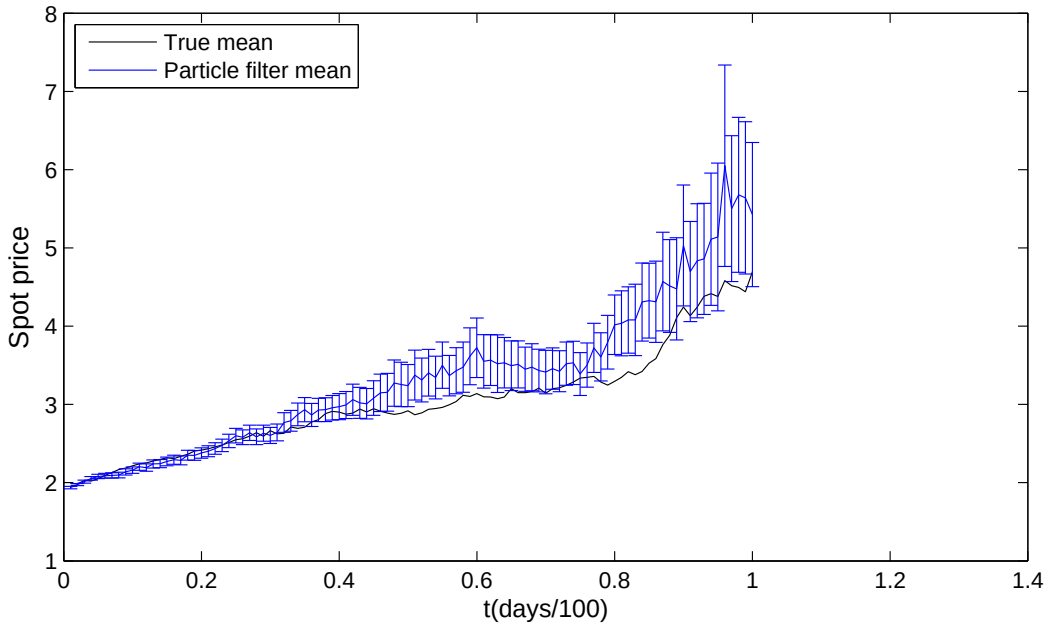

Figure 51: Spot price factor (model 3) - five futures contract, maturity 30 days, 1000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

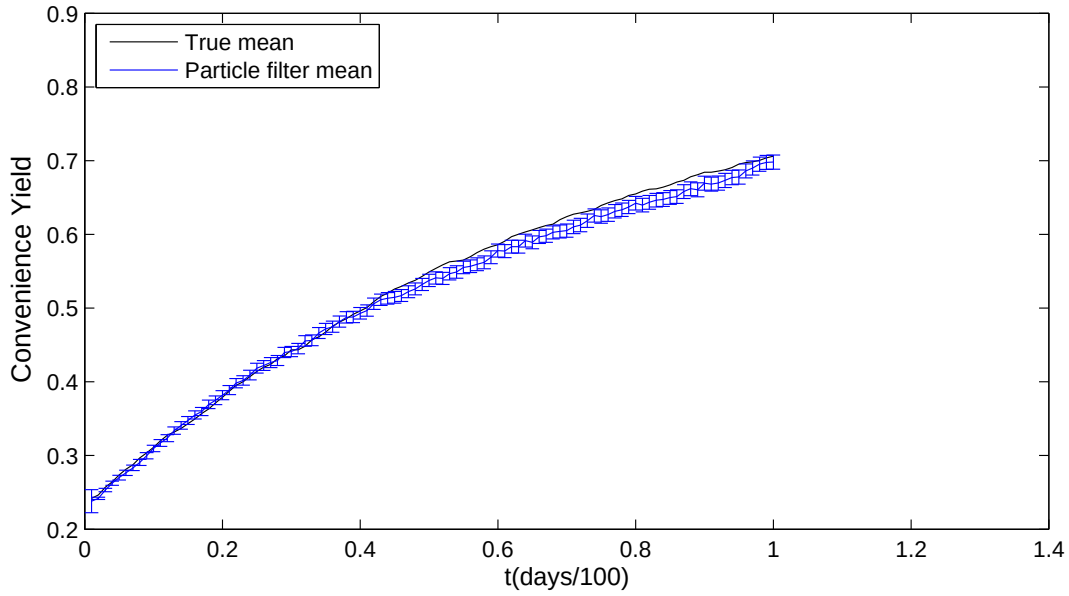

Figure 52: Convenience yield factor (model 3) - five futures contract, maturity 30 days, 1000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

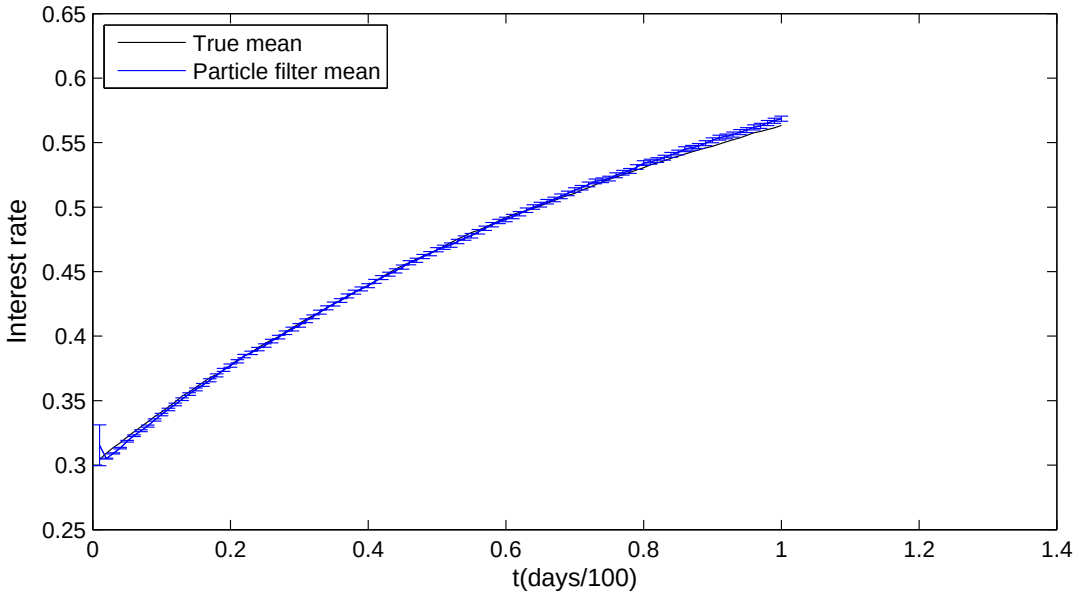

Figure 53: Interest rate factor (model 3) - five futures contract, maturity 30 days, 1000 particles are utilised, estimated Particle filter states and precision at each day vs true latent process.

From figures 51, 52, 53, we see that the Particle filter estimate in the five futures contracts

case is less accurate than that in the one contract case. This can be seen clearly from the result of the spot price factor. The estimate achieved in the five contracts tends to diverge as time increases (from day 40 to maturity), whereas the one contract estimate almost keeps close to the truth over the entire period. The results for the other factors (convenience yield and interest rate) in both these cases are not quite distinguishable, and very accurate. In some sense, this together with the results for the convenience yield and interest rate obtained in the one contract case imply that the Particle filter produces very good estimates for factors with special structures in their dynamics. For instance, in the three-factor model, the convenience yield and interest rate all have constant volatilities, and this leads to a more accurate Particle filter estimate for these factors than for the spot price where the diffusion term is stochastic.

9.4 Discussion

In the previous sections, we have implemented the three different filters for the two models 2 and 3. The data used involve futures contracts since in most commodity markets, the futures price is more easily observed than the spot price of a commodity. For the two-factor model, we utilise the Kalman filter since model 2 is both Gaussian and linear under its state space form. However, in the three-factor model, the linear property is dropped, and hence the Kalman filter does not work well for this model. We then use an extension of the Kalman filter, namely, the Extended Kalman filter to linearize the non-linear state space model of model 3. Once a model has been linearized, the Extended Kalman filter adopts the same technique as the Kalman filter for filtering the "new" state space model under Gaussian and linear conditions on the state and measurement equations. For model 3, we also implement the Particle filter to observe its performance with the Extended Kalman filter. The results as discussed in the previous section reveal that the Particle filter achieves much more accurate estimate than the Extended Kalman filter. Even though the initial state is chosen far away from that of the truth, the Particle filter quickly finds the accurate estimate within only a few first steps.

For the Kalman filter study, we particularly observe the effects of different factors: a single contract, high observation noise, low observation noise, and different structures of the futures prices. Especially, for the time to maturity study, we have found that the effects of changing the time to maturity on initialization is almost zero for both two factors in model 2. For the other studies, we see that using a single futures contract appears to give better result than using multi futures contracts. Furthermore, the length of a contract also has a noticeable impact on

the performance of the Kalman filter. As the length is expanded, the Kalman filter may achieve accurate result (as for the long-term factor ξ), but it may also achieve a less accurate estimate for the truth (as for the short-term factor χ). Moreover, the observation noise study implies that the high SNR case (according to a small noise) appears to give better simulation than the low SNR case (according to a large noise). Similar to the high SNR and low SNR case, correlation on the observation noise also has noticeable implications. Indeed, for a constant or increasing correlation on the observation noise, the Kalman filter produces very accurate estimate, but it does not for the decreasing correlation on the observation noise.

For the Extended Kalman filter study, utilising one futures contract seems to yield better estimate than using multi futures contracts. This result is similar the result we derived already in the Kalman filter implementation. Finally, for the Particle filter study, we observe that the number of particles utilised also has a strong effect on the performance of the Particle filter. Indeed, as we increase the number of simulation particles from 1000 to 5000, then the result is improved significantly as shown in figures 45 and 48. Moreover, the Particle filter is affected as the number of futures contracts is changed. As less futures contracts are utilised, the Particle filter produces more accurate estimate as seen in figures 48 and 51.

In general, the Kalman filter appears to be the optimal solution for the filtering recursion problem under Gaussian and linear assumptions on a state space model. However, when the Gaussian and linear assumptions are removed, then the Particle filter becomes an optimal choise for producing accurate estimate for the truth.

10 Conclusion and future research

In this thesis, we have presented three types of commodity models which take into account the mean reverting property of commodity prices. We basically attent our study on discretization and filtering aspects for the models 2 and 3 (since model 1 and model 2 are equivalent as discussed in section 3.3). For the discretization aspect, we study both theoretically and empirically the performance of the Euler and Milstein schemes. Moreover, we examine the effect of the discretization time interval and the maturity on these schemes. We note that the Euler and Milstein schemes only distinguish when applied to the spot price factor of model 3, since the spot price involves a stochastic diffusion term in its dynamics. All the results derived in section 5.1 imply that both these schemes achieve very good simulation results as the discretization time interval decreases. Moreover, as the maturity is expanded, then there appear more uncertainty arising in each simulation, and hence this leads to a less accuracy in the simulation obtained by the Euler or Milstein scheme.

Regarding filtering aspect, we utilise the state space model approach to facilitate the filtering techniques. We note that once a model has been put into a state space form, then the Kalman filter, Extended Kalman filter or the Particle filter can be implemented using futures contracts. The futures contract is more easily observed and handled than the spot price in most commodity markets. Therefore, it is used as a proxy for the spot price when implementing a filtering technique. Moreover, we also observe that the effects of futures contracts and different assumptions on the state space model to the performance of filtering techniques. Every different types of these factors have different implications on the filtering results.

In this thesis, we have considered two cases of a state space model. The first case is when the Gaussian and linear assumptions are put into the state space model (model 2). The second case is when the state space model still keeps Gaussian property, but now it is no longer linear in terms of its transition and measurement equations. Moreover, we also have just examined the effects of some particular factors based on the futures contracts and the observation noise. A further research in this regard can be made by considering the effect of other factors on the performance of each filtering technique. For instance, one may investigate changing the state noise and may try other different structures of correlation on the observation noise, etc.

11 Acknowledgements

I would like to extend a special thank to my supervisor, Gareth W. Peters, who has spent a lot of time reading over this thesis and given many comments that led to significant improvements. Thanks to the authors from the articles and books I have referenced to and used their results in this thesis. Thanks to my family for being supportive and understanding.

References

[1] MS Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and SA Adelaide. A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[2] M.J. Brennan. The price of convenience and the valuation of commodity contingent claims. *Stochastic models and option values*, 200:33–71, 1991.

[3] RG Brown and PY Hwang. *C. Introduction to random signals and applied kalman filtering.* Citeseer, 1992.

[4] O. Cappé, SJ Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[5] Investorwords dictionary. URL: $http://www.investorwords.com/975/commodity.html$.

[6] A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. *Sequential Monte Carlo methods in practice*, pages 3–14, 2001.

[7] A. Doucet and A.M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *The Oxford Handbook of Nonlinear Filtering. Oxford University Press. To appear*, 2009.

[8] J. Durbin and S.J. Koopman. *Time series analysis by state space methods.* Oxford Univ Pr, 2001.

[9] R. Gibson and E.S. Schwartz. Stochastic convenience yield and the pricing of oil contingent claims. *Journal of Finance*, 45(3):959–976, 1990.

[10] A.C. Harvey. *Forecasting, structural time series models and the Kalman filter.* Cambridge Univ Pr, 1991.

[11] Investopedia. URL: $http://www.investopedia.com/terms/c/commoditiesexchange.asp$.

[12] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, volume 3, page 26. Citeseer, 1997.

[13] F.C. Klebaner. *Introduction to stochastic calculus with applications.* Imperial College Pr, 2005.

[14] P.E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*. Springer Verlag, 1999.

[15] S. Kullback and RA Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86, 1951.

[16] JS Liu. Monte Carlo strategies in scientific computing (2001) New York. *NY: Springer-Verlag*.

[17] R. Lord, R. Koekkoek, and D.J.C. Van Dijk. A comparison of biased simulation schemes for stochastic volatility models.

[18] Marketswiki. URL: *http* : *//www.marketswiki.com/mwiki*.

[19] D.R. Ribeiro and S.D. Hodges. A two-factor model for commodity prices and futures valuation.

[20] E. Schwartz. Valuing long-term commodity assets. *Journal of Energy Finance & Development*, 3(2):85–99, 1998.

[21] E. Schwartz and J.E. Smith. Short-term variations and long-term dynamics in commodity prices. *Management Science*, 46(7):893–911, 2000.

[22] E.S. Schwartz. The stochastic behavior of commodity prices: Implications for valuation and hedging. *Journal of finance*, 52(3):923–973, 1997.

12 Appendix 1: Futures Price Derivation

12.1 Model 1

Let $X_t = \ln S_t$. Then from Ito's Lemma (see, for example, [13]), we obtain the dynamics of the process X as follows

$$dX = \left(r - c - \frac{1}{2}\sigma^2\right) dt + \sigma dZ^*$$

We denote the futures price at time t with maturity T by F(t, T). Since the futures price converges to the expected futures spot price, then we have

$$F(t,T) = \mathbb{E}^* [S_T | S_t] = \mathbb{E}^* \left[e^{X_T} | X_t\right]$$

where $\mathbb{E}^*$ denotes the expectation taken with respect to the risk neutral process.
In order to calculate this expectation, a natural way is to find the transition density $p(X_T, T|X_t, t)$ of the process X. This can be obtained by using the Kolmogorov backward equation (KBE) which is expressed as follows

$$\frac{\partial p(X_T, T|X_t, t)}{\partial t} + \left(r - c - \frac{1}{2}\sigma^2\right) \frac{p(X_T, T|X_t, t)}{\partial X} + \frac{1}{2}\sigma^2 \frac{\partial^2 p(X_T, T|X_t, t)}{\partial X^2} = 0$$

subject to the boundary condition: $p(X_T, T|X_t, t = T) = \tilde{\delta}(X_T - X_t)$.
To obtain the expression for the futures price, we multiply both sides by e^{X_t} and then integrate with respect to X_t. This then follows that

$$\frac{\partial F(t,T)}{\partial t} + \left(r - c - \frac{1}{2}\sigma^2\right) \frac{F(t,T)}{\partial X} + \frac{1}{2}\sigma^2 \frac{\partial^2 F(t,T)}{\partial X^2} = 0$$

subject to the boundary condition: $F(t = T, T) = e^{X_T}$.
We assume that the solution of this KBE has an exponential affine form:

$$F(t,T) = e^{A_0(t) + A_1(t) X_t}$$

Since $F(t = T, T) = e^{X_T}$, we must have: $A_0(T) = 0$ and $A_1(T) = 1$. The derivatives of the futures price can be found as follows

$$\begin{aligned}
\frac{\partial F(t,T)}{\partial t} &= \left(\frac{dA_0(t)}{dt} + X_t \frac{dA_1(t)}{dt}\right) F(t,T); \\
\frac{\partial F(t,T)}{\partial X} &= A_1(t) F(t,T); \\
\frac{\partial^2 F(t,T)}{\partial X^2} &= (A_1(t))^2 F(t,T)
\end{aligned}$$

Now, substitute these terms to the above KBE, we obtain:

$$\left(\frac{dA_0(t)}{dt} + X_t\frac{dA_1(t)}{dt}\right)F(t,T) + \left(r - c - \frac{1}{2}\sigma^2\right)A_1(t)F(t,T) + \frac{1}{2}\sigma^2(A_1(t))^2F(t,T) = 0$$

Divide both sides by $F(t,T)$:

$$\frac{dA_0(t)}{dt} + X_t\frac{dA_1(t)}{dt} + \left(r - c - \frac{1}{2}\sigma^2\right)A_1(t) + \frac{1}{2}\sigma^2(A_1(t))^2 = 0$$

This then follows that

$$\begin{cases} \frac{dA_1(t)}{dt} = 0 \\ \frac{dA_0(t)}{dt} + \left(r - c - \frac{1}{2}\sigma^2\right)A_1(t) + \frac{1}{2}\sigma^2(A_1(t))^2 = 0 \end{cases}$$

$$\Rightarrow \begin{cases} A_1(t) = 1 & (\text{since } A_1(T) = 1) \\ \frac{dA_0(t)}{dt} + r - c - \frac{1}{2}\sigma^2 + \frac{1}{2}\sigma^2 = 0 \end{cases}$$

$$\Rightarrow \begin{cases} A_1(t) = 1 & (\text{since } A_1(T) = 1) \\ A_0(t) = (c-r)(t-T) & (\text{since } A_0(T) = 0) \end{cases}$$

Thus, the futures price is given by

$$F(t,T) = e^{(c-r)(t-T)+X_t} = S_t e^{(c-r)(t-T)}$$

12.2 Model 2

Let $X_t = \ln S_t$. We denote the futures price at time t with maturity T by F(t, T). Since the futures price converges to the expected futures spot price, then we have

$$F(t,T) = \mathbb{E}^*[S_T|S_t] = \mathbb{E}^*\left[e^{X_T}|X_t\right] = \mathbb{E}^*\left[e^{\chi_T+\xi_T}|\chi_t,\xi_t\right]$$

where $\mathbb{E}^*$ denotes the expectation taken with respect to the risk neutral process.
In order to calculate this expectation, a natural way is to find the transition density $p(\chi_T + \xi_T, T|\chi_t, \xi_t, t)$. For simplicity, we denote $p(\chi_T + \xi_T, T|\chi_t, \xi_t, t)$ by $p(Y_T, T|Y_t, t)$. Then the Kolmogorov backward equation can be applied to obtain the expression for $p(Y_T, T|Y_t, t)$ as follows
$\frac{\partial p(Y_T,T|Y_t,t)}{\partial t} + (-\kappa\chi_t - \lambda_\chi)\frac{\partial p(Y_T,T|Y_t,t)}{\partial \chi} + (\mu_\xi - \lambda_\xi)\frac{\partial p(Y_T,T|Y_t,t)}{\partial \xi} + \frac{1}{2}\sigma_\chi^2\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial \chi^2} +$
$+ \frac{1}{2}\sigma_\xi^2\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial \xi^2} + \rho\sigma_\chi\sigma_\xi\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial \chi\partial\xi} = 0$
subject to the boundary condition: $p(Y_T, T|Y_t, t = T) = \tilde{\delta}(Y_T - Y_t)$.

To obtain the expression for the futures price, we multiply both sides by e^{X_t} and then integrate with respect to X_t. This then follows that

$$\frac{\partial F(t,T)}{\partial t}+(-\kappa\chi_t-\lambda_\chi)\frac{\partial F(t,T)}{\partial \chi}+(\mu_\xi-\lambda_\xi)\frac{\partial F(t,T)}{\partial \xi}+\frac{1}{2}\sigma_\chi^2\frac{\partial^2 F(t,T)}{\partial \chi^2}+\frac{1}{2}\sigma_\xi^2\frac{\partial^2 F(t,T)}{\partial \xi^2}+$$
$$+\rho\sigma_\chi\sigma_\xi\frac{\partial^2 F(t,T)}{\partial \chi\partial \xi}=0$$

subject to the boundary condition: $F(t=T,T)=e^{X_T}$.

We assume that the solution of this KBE has an exponential affine form:

$$F(t,T)=e^{A_0(t)+A_1(t)\chi_t+A_2(t)\xi_t}$$

Since $F(t=T,T)=e^{X_T}$, we must have: $A_0(T)=0$, $A_1(T)=1$ and $A_2(T)=1$. The derivatives of the futures price can be found as follows

$$\begin{aligned}
\frac{\partial F(t,T)}{\partial t}&=\left(\frac{dA_0(t)}{dt}+\chi_t\frac{dA_1(t)}{dt}+\xi_t\frac{dA_2(t)}{dt}\right)F(t,T);\\
\frac{\partial F(t,T)}{\partial \chi}&=A_1(t)F(t,T);\\
\frac{\partial F(t,T)}{\partial \xi}&=A_2(t)F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial \chi^2}&=(A_1(t))^2F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial \xi^2}&=(A_2(t))^2F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial \chi\partial \xi}&=A_1(t)A_2(t)F(t,T);
\end{aligned}$$

Now, substitute these terms to the above KBE, we obtain:

$\left(\frac{dA_0(t)}{dt}+\chi_t\frac{dA_1(t)}{dt}+\xi_t\frac{dA_2(t)}{dt}\right)F(t,T)+(-\kappa\chi_t-\lambda_\chi)A_1(t)F(t,T)+(\mu_\xi-\lambda_\xi)A_2(t)F(t,T)$
$+\frac{1}{2}\sigma_\chi^2(A_1(t))^2F(t,T)+\frac{1}{2}\sigma_\xi^2(A_2(t))^2F(t,T)+\rho\sigma_\chi\sigma_\xi A_1(t)A_2(t)F(t,T)=0$

Divide both sides by $F(t,T)$:

$\frac{dA_0(t)}{dt}+\chi_t\frac{dA_1(t)}{dt}+\xi_t\frac{dA_2(t)}{dt}+(-\kappa\chi_t-\lambda_\chi)A_1(t)+(\mu_\xi-\lambda_\xi)A_2(t)+\frac{1}{2}\sigma_\chi^2(A_1(t))^2+$
$+\frac{1}{2}\sigma_\xi^2(A_2(t))^2+\rho\sigma_\chi\sigma_\xi A_1(t)A_2(t)=0$
$\Rightarrow\chi_t\left(\frac{dA_1(t)}{dt}-\kappa A_1(t)\right)+\xi_t\frac{dA_2(t)}{dt}+\frac{dA_0(t)}{dt}-\lambda_\chi A_1(t)+(\mu_\xi-\lambda_\xi)A_2(t)+\frac{1}{2}\sigma_\chi^2(A_1(t))^2+$
$+\frac{1}{2}\sigma_\xi^2(A_2(t))^2+\rho\sigma_\chi\sigma_\xi A_1(t)A_2(t)=0$

This then follows that

$$\begin{cases}
\frac{dA_1(t)}{dt}=\kappa A_1(t)\\
\frac{dA_2(t)}{dt}=0\\
\frac{dA_0(t)}{dt}-\lambda_\chi A_1(t)+(\mu_\xi-\lambda_\xi)A_2(t)+\frac{1}{2}\sigma_\chi^2(A_1(t))^2+\frac{1}{2}\sigma_\xi^2(A_2(t))^2+\rho\sigma_\chi\sigma_\xi A_1(t)A_2(t)=0
\end{cases}$$

$$\Rightarrow \begin{cases} A_1(t) = e^{\kappa(t-T)} & (\text{since } A_1(T) = 1) \\ A_2(t) = 1 & (\text{since } A_2(T) = 1) \\ \frac{dA_0(t)}{dt} - \lambda_\chi e^{\kappa(t-T)} + \mu_\xi - \lambda_\xi + \frac{1}{2}\sigma_\chi^2 e^{2\kappa(t-T)} + \frac{1}{2}\sigma_\xi^2 + \rho\sigma_\chi\sigma_\xi e^{\kappa(t-T)} = 0 \end{cases}$$

$$\Rightarrow \begin{cases} A_1(t) = e^{\kappa(t-T)} \\ A_2(t) = 1 \\ A_0(t) = \frac{\lambda_\chi}{\kappa}\left(e^{\kappa(t-T)} - 1\right) - (\mu_\xi - \lambda_\xi)(t-T) - \frac{1}{4\kappa}\sigma_\chi^2\left(e^{2\kappa(t-T)} - 1\right) - \frac{1}{2}\sigma_\xi^2(t-T) - \frac{\rho\sigma_\chi\sigma_\xi}{\kappa}\left(e^{\kappa(t-T)} - 1\right) \end{cases}$$

Thus, the futures price is given by

$$F(t,T) = e^{e^{\kappa(t-T)}\chi_t + \xi_t + A_0(t)}$$

with $A_0(t)$ defined as above.

12.3 Model 3

Let $X_t = \ln S_t$. Then from Ito's Lemma (see, for example, [13]), we obtain the dynamics of the process X as follows

$$dX = \left(r - \delta - \frac{1}{2}\sigma_1^2\right)dt + \sigma_1 dZ_1^*$$

We denote the futures price at time t with maturity T by F(t, T). Since the futures price converges to the expected futures spot price, then we have

$$F(t,T) = \mathbb{E}^*[S_T|S_t] = \mathbb{E}^*\left[e^{X_T}|X_t\right]$$

where $\mathbb{E}^*$ denotes the expectation taken with respect to the risk neutral process.
In order to calculate this expectation, a natural way is to find the transition density of the state variables which is denoted by $p(X_T, \delta_T, r_T, T|X_t, \delta_t, r_t, t) = p(Y_T, T|Y_t, t)$. This can be obtained by using the Kolmogorov backward equation (KBE) which is expressed as follows
$\frac{\partial p(Y_T,T|Y_t,t)}{\partial t} + \left(r - \delta - \frac{1}{2}\sigma_1^2\right)\frac{\partial p(Y_T,T|Y_t,t)}{\partial X} + \kappa(\hat{\alpha} - \delta)\frac{\partial p(Y_T,T|Y_t,t)}{\partial \delta} + a(m^* - r)\frac{\partial p(Y_T,T|Y_t,t)}{\partial r} +$
$+ \frac{1}{2}\sigma_1^2\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial X^2} + \frac{1}{2}\sigma_2^2\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial \delta^2} + \frac{1}{2}\sigma_3^2\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial r^2} + \rho_1\sigma_1\sigma_2\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial X \partial \delta} +$
$+ \rho_2\sigma_2\sigma_3\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial \delta \partial r} + \rho_3\sigma_1\sigma_3\frac{\partial^2 p(Y_T,T|Y_t,t)}{\partial X \partial r} = 0$
subject to the boundary condition: $p(Y_T, T|Y_t, t = T) = \tilde{\delta}(Y_T - Y_t)$.
To obtain the expression for the futures price, we multiply both sides by e^{X_t} and then integrate with respect to X_t. This then follows that
$\frac{\partial F(t,T)}{\partial t} + \left(r - \delta - \frac{1}{2}\sigma_1^2\right)\frac{\partial F(t,T)}{\partial X} + \kappa(\hat{\alpha} - \delta)\frac{\partial F(t,T)}{\partial \delta} + a(m^* - r)\frac{\partial F(t,T)}{\partial r} + \frac{1}{2}\sigma_1^2\frac{\partial^2 F(t,T)}{\partial X^2} +$

$+\frac{1}{2}\sigma_2^2\frac{\partial^2 F(t,T)}{\partial\delta^2}+\frac{1}{2}\sigma_3^2\frac{\partial^2 F(t,T)}{\partial r^2}+\rho_1\sigma_1\sigma_2\frac{\partial^2 F(t,T)}{\partial X\partial\delta}+\rho_2\sigma_2\sigma_3\frac{\partial^2 F(t,T)}{\partial\delta\partial r}+\rho_3\sigma_1\sigma_3\frac{\partial^2 F(t,T)}{\partial X\partial r}=0$
subject to the boundary condition: $F(t=T,T)=e^{X_T}$.

We assume that the solution of this KBE has an exponential affine form:

$$F(t,T)=e^{A_0(t)+A_1(t)X_t+A_2(t)\delta_t+A_3(t)r_t}$$

Since $F(t=T,T)=e^{X_T}$, we must have: $A_0(T)=0$, $A_1(T)=1$, $A_2(T)=0$ and $A_3(T)=0$. The derivatives of the futures price can be found as follows

$$\begin{aligned}
\frac{\partial F(t,T)}{\partial t}&=\left(\frac{dA_0(t)}{dt}+X_t\frac{dA_1(t)}{dt}+\delta_t\frac{dA_2(t)}{dt}+r_t\frac{dA_3(t)}{dt}\right)F(t,T);\\
\frac{\partial F(t,T)}{\partial X}&=A_1(t)F(t,T);\\
\frac{\partial F(t,T)}{\partial \delta}&=A_2(t)F(t,T);\\
\frac{\partial F(t,T)}{\partial r}&=A_3(t)F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial X^2}&=(A_1(t))^2F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial \delta^2}&=(A_2(t))^2F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial r^2}&=(A_3(t))^2F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial X\partial\delta}&=A_1(t)A_2(t)F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial \delta\partial r}&=A_2(t)A_3(t)F(t,T);\\
\frac{\partial^2 F(t,T)}{\partial X\partial r}&=A_1(t)A_3(t)F(t,T);
\end{aligned}$$

Now, substitute these terms to the above KBE, we obtain:

$$\begin{aligned}
&\left(\frac{dA_0(t)}{dt}+X_t\frac{dA_1(t)}{dt}+\delta_t\frac{dA_2(t)}{dt}+r_t\frac{dA_3(t)}{dt}\right)F(t,T)+\left(r-\delta-\frac{1}{2}\sigma_1^2\right)A_1(t)F(t,T)+\\
&+\kappa(\hat{\alpha}-\delta)A_2(t)F(t,T)+a(m^*-r)A_3(t)F(t,T)+\frac{1}{2}\sigma_1^2(A_1(t))^2F(t,T)+\\
&+\frac{1}{2}\sigma_2^2(A_2(t))^2F(t,T)+\frac{1}{2}\sigma_3^2(A_3(t))^2F(t,T)+\rho_1\sigma_1\sigma_2A_1(t)A_2(t)F(t,T)+\\
&+\rho_2\sigma_2\sigma_3A_2(t)A_3(t)F(t,T)+\rho_3\sigma_1\sigma_3A_1(t)A_3(t)F(t,T)=0
\end{aligned}$$

Divide both sides by $F(t,T)$:

$$\frac{dA_0(t)}{dt}+X_t\frac{dA_1(t)}{dt}+\delta_t\frac{dA_2(t)}{dt}+r_t\frac{dA_3(t)}{dt}+\left(r-\delta-\frac{1}{2}\sigma_1^2\right)A_1(t)+\kappa(\hat{\alpha}-\delta)A_2(t)+$$
$$+a(m^*-r)A_3(t)+\frac{1}{2}\sigma_1^2(A_1(t))^2+\frac{1}{2}\sigma_2^2(A_2(t))^2+\frac{1}{2}\sigma_3^2(A_3(t))^2+\rho_1\sigma_1\sigma_2A_1(t)A_2(t)+$$
$$+\rho_2\sigma_2\sigma_3A_2(t)A_3(t)+\rho_3\sigma_1\sigma_3A_1(t)A_3(t)=0$$
$$\Rightarrow X_t\frac{dA_1(t)}{dt}+\delta_t\left(\frac{dA_2(t)}{dt}-A_1(t)-\kappa A_2(t)\right)+r_t\left(\frac{dA_3(t)}{dt}+A_1(t)-aA_3(t)\right)+\frac{dA_0(t)}{dt}-$$
$$-\frac{1}{2}\sigma_1^2A_1(t)+\kappa\hat{\alpha}A_2(t)+am^*A_3(t)+\frac{1}{2}\sigma_1^2(A_1(t))^2+\frac{1}{2}\sigma_2^2(A_2(t))^2+\frac{1}{2}\sigma_3^2(A_3(t))^2+$$
$$+\rho_1\sigma_1\sigma_2A_1(t)A_2(t)+\rho_2\sigma_2\sigma_3A_2(t)A_3(t)+\rho_3\sigma_1\sigma_3A_1(t)A_3(t)=0$$

This then follows that

$$\begin{cases}\frac{dA_1(t)}{dt}=0\\ \frac{dA_2(t)}{dt}-A_1(t)-\kappa A_2(t)=0\\ \frac{dA_3(t)}{dt}+A_1(t)-aA_3(t)=0\end{cases}$$

and

$$\frac{dA_0(t)}{dt}-\frac{1}{2}\sigma_1^2A_1(t)+\kappa\hat{\alpha}A_2(t)+am^*A_3(t)+\frac{1}{2}\sigma_1^2(A_1(t))^2+\frac{1}{2}\sigma_2^2(A_2(t))^2+\frac{1}{2}\sigma_3^2(A_3(t))^2+$$
$$+\rho_1\sigma_1\sigma_2A_1(t)A_2(t)+\rho_2\sigma_2\sigma_3A_2(t)A_3(t)+\rho_3\sigma_1\sigma_3A_1(t)A_3(t)=0$$

$$\Rightarrow\begin{cases}A_1(t)=1 & (\text{since } A_1(T)=1)\\ \frac{dA_2(t)}{dt}=1+\kappa A_2(t)\\ \frac{dA_3(t)}{dt}=-1+aA_3(t)\end{cases}$$

and

$$\frac{dA_0(t)}{dt}-\frac{1}{2}\sigma_1^2+\kappa\hat{\alpha}A_2(t)+am^*A_3(t)+\frac{1}{2}\sigma_1^2+\frac{1}{2}\sigma_2^2(A_2(t))^2+\frac{1}{2}\sigma_3^2(A_3(t))^2+\rho_1\sigma_1\sigma_2A_2(t)+$$
$$+\rho_2\sigma_2\sigma_3A_2(t)A_3(t)+\rho_3\sigma_1\sigma_3A_3(t)=0$$

$$\Rightarrow\begin{cases}A_1(t)=1\\ A_2(t)=\frac{1}{\kappa}\left(e^{\kappa(t-T)}-1\right) & (\text{since } A_2(T)=0)\\ A_3(t)=\frac{1}{a}\left(1-e^{a(t-T)}\right) & (\text{since } A_3(T)=0)\end{cases}$$

and

$$\begin{aligned}\frac{dA_0(t)}{dt} =& \frac{1}{2}\sigma_1^2 - \hat{\alpha}\left(e^{\kappa(t-T)}-1\right) + m^*\left(e^{a(t-T)}-1\right) - \frac{1}{2}\sigma_1^2 - \frac{\sigma_2^2}{2\kappa^2}\left(e^{\kappa(t-T)}-1\right)^2 - \\ & - \frac{\sigma_3^2}{2a^2}\left(e^{a(t-T)}-1\right)^2 - \frac{\rho_1\sigma_1\sigma_2}{\kappa}\left(e^{\kappa(t-T)}-1\right) + \frac{\rho_2\sigma_2\sigma_3}{\kappa a}\left(e^{\kappa(t-T)}-1\right)\left(e^{a(t-T)}-1\right) + \\ & + \frac{\rho_3\sigma_1\sigma_3}{a}\left(e^{a(t-T)}-1\right) \\ =& \frac{\rho_2\sigma_2\sigma_3}{\kappa a}e^{(\kappa+a)(t-T)} - \frac{\sigma_2^2}{2\kappa^2}e^{2\kappa(t-T)} - \frac{\sigma_3^2}{2a^2}e^{2a(t-T)} + \\ & + e^{\kappa(t-T)}\left(\frac{\sigma_2^2}{\kappa^2} - \hat{\alpha} - \frac{\rho_1\sigma_1\sigma_2}{\kappa} - \frac{\rho_2\sigma_2\sigma_3}{\kappa a}\right) + e^{a(t-T)}\left(m^* + \frac{\sigma_3^2}{a^2} - \frac{\rho_2\sigma_2\sigma_3}{\kappa a} + \frac{\rho_3\sigma_1\sigma_3}{a}\right) + \\ & + \hat{\alpha} - m^* - \frac{\sigma_2^2}{2\kappa^2} - \frac{\sigma_3^2}{2a^2} + \frac{\rho_1\sigma_1\sigma_2}{\kappa} + \frac{\rho_2\sigma_2\sigma_3}{\kappa a} - \frac{\rho_3\sigma_1\sigma_3}{a}\end{aligned}$$

Since $A_0(T) = 0$, it follows that

$$\begin{aligned}\Rightarrow A_0(t) =& \frac{\rho_2\sigma_2\sigma_3}{\kappa a(\kappa+a)}\left(e^{(\kappa+a)(t-T)}-1\right) - \frac{\sigma_2^2}{4\kappa^3}\left(e^{2\kappa(t-T)}-1\right) - \frac{\sigma_3^2}{4a^3}\left(e^{2a(t-T)}-1\right) + \\ & + \left(\frac{\sigma_2^2}{\kappa^3} - \frac{\hat{\alpha}}{\kappa} - \frac{\rho_1\sigma_1\sigma_2}{\kappa^2} - \frac{\rho_2\sigma_2\sigma_3}{\kappa^2 a}\right)\left(e^{\kappa(t-T)}-1\right) + \\ & + \left(\frac{m^*}{a} + \frac{\sigma_3^2}{a^3} - \frac{\rho_2\sigma_2\sigma_3}{\kappa a^2} + \frac{\rho_3\sigma_1\sigma_3}{a^2}\right)\left(e^{a(t-T)}-1\right) + \\ & + \left(\hat{\alpha} - m^* - \frac{\sigma_2^2}{2\kappa^2} - \frac{\sigma_3^2}{2a^2} + \frac{\rho_1\sigma_1\sigma_2}{\kappa} + \frac{\rho_2\sigma_2\sigma_3}{\kappa a} - \frac{\rho_3\sigma_1\sigma_3}{a}\right)(t-T)\end{aligned}$$

Thus, the futures price is given by

$$F(t,T) = e^{X_t + \frac{1}{\kappa}\left(e^{\kappa(t-T)}-1\right)\delta_t + \frac{1}{a}\left(1-e^{a(t-T)}\right)r_t + A_0(t)}$$

with $A_0(t)$ defined as above.

13 Appendix 2: Derivation of Statistical Properties for Model 2 (Equations (3.28) and (3.29))

The discrete-time approximation of the process with time step $\triangle t = \frac{t}{n}$ can be written as

$$\boldsymbol{x}_t = c + Q\boldsymbol{x}_{t-1} + \boldsymbol{\eta}_t$$

where $\boldsymbol{x}_t \equiv [\chi_t, \xi_t]$, $c \equiv [0, \mu\triangle t]$, $Q \equiv \begin{bmatrix} \Phi & 0 \\ 0 & 1 \end{bmatrix}$,
$\Phi \equiv 1 - \kappa\triangle t$, and $\boldsymbol{\eta}_t$ is a 2 x 1 vector of serially uncorrelated disturbances with $\mathbb{E}[\boldsymbol{\eta}_t] = 0$, and

$$\mathbb{V}ar\left[\boldsymbol{\eta}_t\right] = W \equiv \begin{bmatrix} \sigma_1^2 \triangle t & \rho\sigma_1\sigma_2 \triangle t \\ \rho\sigma_1\sigma_2 \triangle t & \sigma_2^2 \triangle t \end{bmatrix}$$

The n-step ahead mean vector (m_n) and covariance matrix (V_n) are given recursively by $m_n = c + Qm_{n-1}$ and $V_n = QV_{n-1}Q^T + W$, with $m_0 = \boldsymbol{x}_0 \equiv [\chi_0, \xi_0]$ and $V_0 = 0$ (see, for instance, Harvey 1989). Then applying this recursion, we find that

$$m_n = [\Phi^n \chi_0, \xi_0 + \mu n \triangle t],$$

$$V_n = \begin{bmatrix} \sigma_1^2 \triangle t \sum_{i=0}^{n-1} \Phi^{2i} & \rho\sigma_1\sigma_2 \triangle t \sum_{i=0}^{n-1} \Phi^i \\ \rho\sigma_1\sigma_2 \triangle t \sum_{i=0}^{n-1} \Phi^i & n \triangle t \sigma_1^2 \end{bmatrix}$$

(These recursive calculations can be checked easily by inductive arguments). Now we can rewrite the geometric series in m_n and V_n as follows

$$\sum_{i=0}^{n-1} \Phi^i = \frac{1 - \Phi^{n-1}}{1 - \Phi} \quad \text{and} \quad \sum_{i=0}^{n-1} \Phi^{2i} = \frac{1 - \Phi^{2(n-1)}}{1 - \Phi^2}$$

As $n \to \infty$, $\triangle t = \frac{t}{n}$ reaches 0, then $\Phi^n = (1 - \frac{\kappa t}{n})^n$ approaches $e^{-\kappa t}$, Φ^{2n} approaches $e^{-2\kappa t}$, and

$$\frac{1 - \Phi^{n-1}}{1 - \Phi} \triangle t \to \frac{1 - e^{-\kappa t}}{\kappa} \quad \text{and} \quad \frac{1 - \Phi^{2(n-1)}}{1 - \Phi^2} \triangle t \to \frac{1 - e^{-2\kappa t}}{2\kappa}$$

Now substituting these limits into the expressions for m_n and V_n, we obtain the mean vector and covariance matrix given in equations (3.28) and (3.29).

14 Appendix 3: Discretization for Model 3 Using The Milstein Scheme

Model 3 will first be recast with respect to independent Wiener processes dW_1, dW_2 and dW_3 as follows

$$\begin{aligned} dS_t &= (\mu - \delta_t) S_t dt + \sigma_1 S_t dW_1 \\ d\delta_t &= \kappa (\alpha - \delta_t) dt + \sigma_2 \left(\rho_1 dW_1 + \sqrt{1 - \rho_1^2} dW_2 \right) \\ dr_t &= a (m - r_t) dt + \sigma_3 \left(\rho_3 dW_1 + \sqrt{1 - \rho_3^2} dW_3 \right) \end{aligned}$$

This will result in the following specifications:

$$a^1(t,S_t) = (\mu - \delta_t) S_t; a^2(t,\delta_t) = \kappa(\alpha - \delta_t); a^3(t,r_t) = a(m - r_t)$$
$$b^{1,1}(t,S_t) = \sigma_1 S_t; b^{1,2}(t,S_t) = 0; b^{1,3}(t,S_t) = 0$$
$$b^{2,1}(t,\delta_t) = \sigma_2\rho_1; b^{2,2}(t,\delta_t) = \sigma_2\sqrt{1-\rho_1^2}; b^{2,3}(t,\delta_t) = 0$$
$$b^{3,1}(t,r_t) = \sigma_3\rho_3; b^{3,2}(t,r_t) = 0; b^{3,3}(t,r_t) = \sigma_3\sqrt{1-\rho_3^2}$$
$$L^1 b^{1,1}(t,S_t) = b^{1,1}(t,S_t)\frac{\partial}{\partial S_t}b^{1,1}(t,S_t) + b^{2,1}(t,\delta_t)\frac{\partial}{\partial \delta_t}b^{1,1}(t,S_t) + b^{3,1}(t,r_t)\frac{\partial}{\partial r_t}b^{1,1}(t,S_t) = \sigma_1^2 S_t$$

It can easily be seen that $L^k b^{i,j} = 0$ for all k, i, j such that they do not equal to 1 simultaneously. Now we arrive at the trivariate Milstein discretization scheme as follows

$$S_t = S_{t-1} + (\mu - \delta_{t-1}) S_{t-1}\triangle t + \sigma_1 S_{t-1}\sqrt{\triangle t} n_{S,t-1} + \sigma_1^2 S_{t-1}\frac{1}{2}\left(\triangle t n_{S,t-1}^2 - \triangle t\right)$$
$$\delta_t = \delta_{t-1} + \kappa(\alpha - \delta_{t-1})\triangle t + \sigma_2\rho_1\sqrt{\triangle t} n_{S,t-1} + \sigma_2\sqrt{1-\rho_1^2}\sqrt{\triangle t} n_{\delta,t-1}$$
$$r_t = r_{t-1} + a(m - r_{t-1})\triangle t + \sigma_3\rho_3\sqrt{\triangle t} n_{S,t-1} + \sigma_3\sqrt{1-\rho_3^2}\sqrt{\triangle t} n_{r,t-1}$$

where $n_{S,t-1}$, $n_{\delta,t-1}$ and $n_{r,t-1}$ are i.i.d. standard normal random variables.

15 Appendix 4: Derivation of The Transition Equation for Model 3

As mentioned earlier, model 3 can be expressed in terms of the log of the spot price, the convenience yield and the interest rate as

$$dX = \left(\mu - \delta - \frac{1}{2}\sigma_1^2\right)dt + \sigma_1 dZ_1$$
$$d\delta = \kappa(\alpha - \delta)dt + \sigma_2 dZ_2$$
$$dr = a(m - r)dt + \sigma_3 dZ_3$$

$$dZ_1 dZ_2 = \rho_1 dt, \quad dZ_2 dZ_3 = \rho_2 dt, \quad dZ_1 dZ_3 = \rho_3 dt.$$

To derive the transition equation for model 3, we first need to convert the joint stochastic processes of the spot price and the convenience yield into the discretized forms.
Model 3 will first be recast with respect to independent Wiener processes dW_1 and dW_2 as

follows

$$dX = \left(\mu - \delta - \frac{1}{2}\sigma_1^2\right) dt + \sigma_1 dW_1$$
$$d\delta = \kappa\left(\alpha - \delta\right) dt + \sigma_2\rho_1 dW_1 + \sigma_2\sqrt{1-\rho_1^2}dW_2$$

Then by using the Euler scheme, the joint stochastic processes of the spot price and the convenience yield can be discretized as follows

$$\begin{aligned} X_k &= X_{k-1} + \left(\mu - \delta_{k-1} - \frac{1}{2}\sigma_1^2\right)\triangle k + \sigma_1\sqrt{\triangle k}n_{X,k-1} \\ &= X_{k-1} - \delta_{k-1}\triangle k + \left(\mu - \frac{1}{2}\sigma_1^2\right)\triangle k + \sigma_1\sqrt{\triangle k}n_{X,k-1} \\ \delta_k &= \delta_{k-1} + \kappa\left(\alpha - \delta_{k-1}\right)\triangle k + \sigma_2\sqrt{\triangle k}n_{\delta,k-1} \\ &= \left(1 - \kappa\triangle k\right)\delta_{k-1} + \kappa\alpha\triangle k + \sigma_2\sqrt{\triangle k}n_{\delta,k-1} \end{aligned}$$

where $n_{X,k-1}$ and $n_{\delta,k-1}$ are i.i.d. standard normal random variables.
Thus, from these equations, the transition equation for model 3 can be obtained as

$$\boldsymbol{x}_k = T_k\boldsymbol{x}_{k-1} + c_k + R_k\boldsymbol{\eta}_k, \quad k = 1, ..., n_T$$

where:

$\boldsymbol{x}_k = [\boldsymbol{X}_k, \boldsymbol{\delta}_k]$ is a 2×1 vector of state variables;

$T_k = \begin{bmatrix} 1 & -\triangle k \\ 0 & 1 - \kappa\triangle k \end{bmatrix}$ is a 2×2 matrix;

$c_k = [(\mu - \frac{1}{2}\sigma_1^2)\triangle k, \kappa\alpha\triangle k]$ is a 2×1 vector;

R_k is a 2×2 identity matrix;

$\boldsymbol{\eta}_k$ is a 2×1 vector of serially uncorrelated, normally distributed disturbances with $\mathbb{E}\left[\boldsymbol{\eta}_k\right] = 0$ and $\mathbb{V}ar\left[\boldsymbol{\eta}_k\right] = Q_k = \mathbb{C}ov[(\boldsymbol{X}_k, \boldsymbol{\delta}_k)] = \begin{bmatrix} \sigma_1^2\triangle k & \rho_1\sigma_1\sigma_2\triangle k \\ \rho_1\sigma_1\sigma_2\triangle k & \sigma_1^2\triangle k \end{bmatrix}$.

16 Appendix 5: Matlab Code - Generating The Three-factor Model

```
% The SDE of the 3-factor model (under the risk neutral framework) is:
%   dS = (r-Delta).S.dt + sigma1.S.dZ1
%   dDelta = kappa.(alpha-Delta).dt + sigma2.dZ2  (here alpha implies
%   alpha_hat)
%   dr = a.(m-r).dt + sigma3.dZ3  (here m implies m*)
%   dZ1.dZ2 = rho12.dt,  dZ1.dZ3 = rho13.dt,  dZ2.dZ3 = rho23.dt
clear all
clc
randn('seed',1)
rand('seed',1)
T = 1;
sigma1 = 0.25; sigma2 = 0.15; sigma3 = 0.1;
m = 0.76; kappa = 0.3; a = 0.18; alpha = 1;
rho12 = 0.24; rho23 = 0.3; rho13 = 0.08;
S0 = 2*randn+0.7; Delta0 = randn; r0 = 2*randn+0.4;
Delta = [0.001 0.1];
g = 0;
Deta = 0.0001; t0 = [0:Deta:T];
for d = 1:500
    XM1(1) = S0; XE2(1) = Delta0; XE3(1) = r0;
    for j = 1:length(t0)-1
        dW1 = sqrt(Deta)*randn;
        dW2 = sqrt(Deta)*randn;
        dW3 = sqrt(Deta)*randn;
        XM1(j+1) = XM1(j) + (XE3(j) - XE2(j))*XM1(j)*Deta ...
        + sigma1*XM1(j)*dW1 + 0.5*(sigma1^2)*XM1(j)*((dW1^2)-Deta);
        XE2(j+1)=XE2(j)+kappa*(alpha-XE2(j))*Deta+sigma2*rho12*dW1 ...
        + sigma2*sqrt(1-rho12^2)*dW2;
        XE3(j+1)=XE3(j)+a*(m-XE3(j))*Deta+sigma3*rho13*dW1 ...
        + sigma3*sqrt(1-rho13^2)*dW3;
    end
    pat1(d,:) = XM1(:);
    pat2(d,:) = XE2(:);
    pat3(d,:) = XE3(:);
    XM1=[]; XE2=[]; XE3=[];
end
truemean1 = mean(pat1(:,1:length(t0)));
```

```
truemean2 = mean(pat2(:,1:length(t0)));
truemean3 = mean(pat3(:,1:length(t0)));
pat1=[]; pat2=[]; pat3=[];
for  k = 1:length(Delta)
    t = [];
    Xe1 = []; X1 = []; X2 = []; X3 = [];
    X1(1) = S0; Xe1(1) = S0;
    X2(1) = Delta0; X3(1) = r0;
    t = [0:Delta(k):T];
    if Delta(k) == 0.001
        d = 1000;
    elseif Delta(k) == 0.1
        d = 100;
    end
    for u = 1:d
        Xe1 = []; X1 = []; X2 = []; X3 = [];
        Xe1(1) = S0; X1(1) = S0; X2(1) = Delta0; X3(1) = r0;
        for j = 1:length(t)-1
            dW1 = sqrt(Delta(k))*randn;
            dW2 = sqrt(Delta(k))*randn;
            dW3 = sqrt(Delta(k))*randn;
            X1(j+1)=X1(j)+(X3(j)-X2(j))*X1(j)*Delta(k)+sigma1*X1(j)*dW1 ...
            + 0.5*(sigma1^2)*X1(j)*((dW1^2)-Delta(k));
            X2(j+1)=X2(j)+kappa*(alpha-X2(j))*Delta(k)+sigma2*rho12*dW1 ...
            + sigma2*sqrt(1-rho12^2)*dW2;
            X3(j+1)=X3(j)+a*(m-X3(j))*Delta(k)+sigma3*rho13*dW1 ...
            + sigma3*sqrt(1-rho13^2)*dW3;
        end
        if Delta(k) == 0.001
            figure(1)
            hold on
            subplot(3,1,1),title('Spot price'),plot(t,X1)
            hold on
            subplot(3,1,2),title('Convenience yield'),plot(t,X2)
            hold on
            subplot(3,1,3),title('Interest rate'),plot(t,X3)
        elseif Delta(k) == 0.1
            figure(2)
            hold on
            subplot(3,1,1),title('Spot price'), plot(t,X1)
```

```
            hold on
            subplot(3,1,2),title('Convenience yield'), plot(t,X2)
            hold on
            subplot(3,1,3),title('Interest rate'), plot(t,X3)
        end
        g = g + 1
    end
    if Delta(k) == 0.001
       figure(1)
       hold on
       subplot(3,1,1),title('Spot price'),plot(t0,truemean1,'k')
       hold on
       subplot(3,1,2),title('Convenience yield'),plot(t0,truemean2,'k')
       hold on
       subplot(3,1,3),title('Interest rate'),plot(t0,truemean3,'k')
    elseif Delta(k) == 0.1
       figure(2)
       hold on
       subplot(3,1,1),title('Spot price'),plot(t0,truemean1,'k')
       hold on
       subplot(3,1,2),title('Convenience yield'),plot(t0,truemean2,'k')
       hold on
       subplot(3,1,3),title('Interest rate'),plot(t0,truemean3,'k')
    end
end
save threeMod -v7.3
```

17 Appendix 6: Matlab Code - Kalman Filter for Single Contract with Length 90 Days (Model 2)

```
% Implementation for the short-/long-term model using the Kalman filter
% The SDE of the short-term/long-term model under the real-world framework is:
%     dchi = -kappa.chi.dt + sigma1.dZ1
%     dxi  = mu.dt + sigma2.dZ2
%     dZ1.dZ2 = rho.dt
clear all
clc
randn('seed',1)
rand('seed',1)
```

```
mydata = importdata('1Contract90.xls');
ID = mydata.data(:,1);
C1 = mydata.data(:,2);  % Single futures contract
g = 1;
while C1(g)~=0
    g = g + 1;
end
H = 0.1*eye(1);
% Step 1: Generate the latent process under the real-world framework
T = 100;
kappa = 1.7; sigma1 = 2; mu = 1.1; sigma2 = 2; rho = 0.5;
chi0 = randn+0.5; xi0 = randn*2+0.3;  % random initial values
X1(1) = chi0; X2(1) = xi0;
Delta = 0.01; t = [0.01:Delta:1];
mean1 = chi0 * exp(-kappa*t); mean2 = xi0 + mu*t;
for j = 1:length(t)-1
    dW1 = sqrt(Delta)*randn; dW2 = sqrt(Delta)*randn;
    X1(j+1) = (1 - kappa*Delta)*X1(j) + sigma1*dW1;
    X2(j+1) = X2(j)+ mu*Delta+sigma2*rho*dW1+sigma2*sqrt(1-rho^2)*dW2;
end
XX1(1) = chi0; XX2(1) = xi0;  t1 = [0.0001:0.0001:1];
for j = 1:length(t1)-1
    dW1 = sqrt(0.0001)*randn; dW2 = sqrt(0.0001)*randn;
    XX1(j+1) = (1 - kappa*0.0001)*XX1(j) + sigma1*dW1;
    XX2(j+1) = XX2(j)+mu*0.0001+sigma2*rho*dW1+sigma2*sqrt(1-rho^2)*dW2;
end
% Step 2: Calculate the futures price and the observed futures price
lambda1 = 1.2;   % The risk premium of chi
lambda2 = 0.7;   % The risk premium of xi
A=@(t,T) (lambda1/kappa)*(exp(kappa*(t-T)/100)-1)-((mu-lambda2)*(t-T)/100) ...
-(0.25/kappa)*(sigma1^2)*(exp(2*kappa*(t-T)/100)-1) ...
-0.5*(sigma2^2)*((t-T)/100)-(rho*sigma1*sigma2/kappa)*(exp(kappa*(t-T)/100)-1);
LNP1 = @(tau) exp(kappa*(-tau/100))*X1(90-tau)+X2(90-tau)+A(90-tau,90);
s = 1;
for i = 1:100
    ObsNoise = mvnrnd(zeros(1),H);
    if i <= 100-g
        LNF1(i) = LNP1(C1(i+g));  LNF1o(i) = LNF1(i) + ObsNoise(1);
    else
        LNF1(i) = LNP1(C1(100)-s);  LNF1o(i) = LNF1(i) + ObsNoise(1);
```

```
        s = s + 1;
    end
end
% Step 3: Using the Kalman filter to estimate the mean and covariance
% Firstly, establish the system matrices
W = @(k) [exp(kappa*(k-90)/100) 1];
d = @(k) [A(k,90)];
TT = [exp(-kappa*Delta) 0; 0 1];
c = [0; mu*Delta];
R = eye(2);
Q = @(k) [
(1-exp(-2*kappa*Delta))*(sigma1^2/(2*kappa)) (1-exp(-kappa*Delta))*(rho*sigma1*sigma2/kappa)
(1-exp(-kappa*Delta))*(rho*sigma1*sigma2/kappa) (sigma2^2)*Delta];
% Run the Kalman filter algorithm
a(:,1) = [chi0; xi0];       % Initial value of the mean
P(:,:,1) = [2 0; 0 2];      % Initial value of the covariance
for i = 2:90
    aa(:,i) = TT*a(:,i-1) + c;
    PP(:,:,i) = TT*P(:,:,i-1)*TT' + R*Q(i)*R';
    z(i) = [LNF1o(i)];
    F(i) = W(i)*PP(:,:,i)*(W(i))' + H;
    a(:,i) = aa(:,i) + PP(:,:,i)*(W(i))'*inv(F(i))*(z(:,i) - W(i)*aa(:,i) - d(i));
    P(:,:,i) = PP(:,:,i) - PP(:,:,i)*(W(i))'*inv(F(i))*W(i)*PP(:,:,i);
end
figure(1)
subplot(2,1,1),plot(t(1:90),X1(1:90),'k')
hold on
errorbar(t(1:90),a(1,:),P(1,1,:))
xlabel t(days/100)
ylabel Chi
subplot(2,1,2),plot(t(1:90),X2(1:90),'k')
hold on
errorbar(t(1:90),a(2,:),P(2,2,:))
xlabel t(days/100)
ylabel Xi
save shortlongFsim90 -v7.3
```

18 Appendix 7: Matlab Code - Kalman Filter for The Number of Contracts Study and Correlation on The Observation Noise Study (Model 2)

```
% Implementation for the short-/long-term model using the Kalman filter
% The SDE of the short-term/long-term model under the real-world framework is:
%     dchi = -kappa.chi.dt + sigma1.dZ1
%     dxi  = mu.dt + sigma2.dZ2
%     dZ1.dZ2 = rho.dt
clear all
clc
randn('seed',1)
rand('seed',1)
mydata = importdata('ContractDates_Synth.xls');
ID = mydata.data(:,1);
C1 = mydata.data(:,2);
C2 = mydata.data(:,3);
C3 = mydata.data(:,4);
C4 = mydata.data(:,5);
C5 = mydata.data(:,6);
g=1;
while C1(g)~=0
    g = g + 1;
end

H = 1*eye(5);  % Case: constant correlation on the observation noise

% Case: increasing correlation on the observation noise
% H(1,1) = 0.002;
% H(2,2) = 0.4;
% H(3,3) = 0.6;
% H(4,4) = 0.8;
% H(5,5) = 100;

% Case: decreasing correlation on the observation noise
%H(1,1) = 100;
%H(2,2) = 0.8;
%H(3,3) = 0.6;
%H(4,4) = 0.4;
```

```
%H(5,5) = 0.002;

% Step 1: Generate the latent process in the real-world parameters
T = 100;
kappa = 1.7; sigma1 = 1; mu = 1.1; sigma2 = 1; rho = 0.5;
chi0 = randn+0.5; xi0 = randn*2+0.3;  % random initial values
X1 = []; X2 = []; X1(1) = chi0; X2(1) = xi0;
Delta = 0.01; t = [0.01:Delta:1];
mean1 = chi0 * exp(-kappa*t);  mean2 = xi0 + mu*t;
for j = 1:length(t)-1
    dW1 = sqrt(Delta)*randn; dW2 = sqrt(Delta)*randn;
    X1(j+1) = (1 - kappa*Delta)*X1(j) + sigma1*dW1;
    X2(j+1) = X2(j) +  mu*Delta + sigma2*rho*dW1 + sigma2*sqrt(1-rho^2)*dW2;
end
% Step 2: Calculate the futures price and the observed futures price
lambda1 = 1.2;   % The risk premium of chi
lambda2 = 0.7;   % The risk premium of xi
A = @(t,T) (lambda1/kappa)*(exp(kappa*(t-T)/100)-1) - ((mu-lambda2)*(t-T)/100) ...
    -(0.25/kappa)*(sigma1^2)*(exp(2*kappa*(t-T)/100)-1) ...
    - 0.5*(sigma2^2)*((t-T)/100)-(rho*sigma1*sigma2/kappa)*(exp(kappa*(t-T)/100)-1);
LNP1 = @(tau) exp(kappa*(-tau/100))*X1(30-tau)+X2(30-tau)+A(30-tau,30);
LNP2 = @(tau) exp(kappa*(-tau/100))*X1(60-tau)+X2(60-tau)+A(60-tau,60);
LNP3 = @(tau) exp(kappa*(-tau/100))*X1(90-tau)+X2(90-tau)+A(90-tau,90);
LNP4 = @(tau) exp(kappa*(-tau/100))*X1(120-tau)+X2(120-tau)+A(120-tau,120);
LNP5 = @(tau) exp(kappa*(-tau/100))*X1(150-tau)+X2(150-tau)+A(150-tau,150);
s = 1;
for i=1:100
    ObsNoise = mvnrnd(zeros(5),H);
    if i <= 100-g
        LNF1(i) = LNP1(C1(i+g));  LNF1o(i) = LNF1(i) + ObsNoise(1);
        LNF2(i) = LNP2(C2(i+g));  LNF2o(i) = LNF2(i) + ObsNoise(2);
        LNF3(i) = LNP3(C3(i+g));  LNF3o(i) = LNF3(i) + ObsNoise(3);
        LNF4(i) = LNP4(C4(i+g));  LNF4o(i) = LNF4(i) + ObsNoise(4);
        LNF5(i) = LNP5(C5(i+g));  LNF5o(i) = LNF5(i) + ObsNoise(5);
    else
        LNF1(i) = LNP1(C1(100)-s);  LNF1o(i) = LNF1(i) + ObsNoise(1);
        LNF2(i) = LNP2(C2(100)-s);  LNF2o(i) = LNF2(i) + ObsNoise(2);
        LNF3(i) = LNP3(C3(100)-s);  LNF3o(i) = LNF3(i) + ObsNoise(3);
        LNF4(i) = LNP4(C4(100)-s);  LNF4o(i) = LNF4(i) + ObsNoise(4);
        LNF5(i) = LNP5(C5(100)-s);  LNF5o(i) = LNF5(i) + ObsNoise(5);
```

```
        s = s + 1;
    end
end
% Step 3: Using the Kalman filter to estimate the mean and covariance
% Firstly, establish the system matrices
W = @(k) [exp(kappa*(k-30)/100) 1; exp(kappa*(k-60)/100) 1; exp(kappa*(k-90)/100) 1;
          exp(kappa*(k-120)/100) 1; exp(kappa*(k-150)/100) 1];
d = @(k) [A(k,30); A(k,60); A(k,90); A(k,120); A(k,150)];
TT = [exp(-kappa*Delta) 0; 0 1];
c = [0; mu*Delta];
R = eye(2);
Q = @(k) [
(1-exp(-2*kappa*Delta))*(sigma1^2/(2*kappa)) (1-exp(-kappa*Delta))*(rho*sigma1*sigma2/kappa)
(1-exp(-kappa*Delta))*(rho*sigma1*sigma2/kappa) (sigma2^2)*Delta];
% Run the Kalman filter algorithm
a(:,1) = [0.5*rand+chi0; 0.5*rand+xi0];   % Initial value of the mean
P(:,:,1) = [2 0; 0 2];                    % Initial value of the covariance
for i = 2:100
    aa(:,i) = TT*a(:,i-1) + c;
    PP(:,:,i) = TT*P(:,:,i-1)*TT' + R*Q(i)*R';
    z(:,i) = [LNF1o(i) ; LNF2o(i); LNF3o(i); LNF4o(i); LNF5o(i)];
    F(:,:,i) = W(i)*PP(:,:,i)*(W(i))' + H;
    a(:,i) = aa(:,i) + PP(:,:,i)*(W(i))'*inv(F(:,:,i))*(z(:,i) - W(i)*aa(:,i) - d(i));
    P(:,:,i) = PP(:,:,i) - PP(:,:,i)*(W(i))'*inv(F(:,:,i))*W(i)*PP(:,:,i);
end
figure(1)
subplot(2,1,1),plot(t,X1,'k')
hold on
errorbar(t,a(1,:),P(1,1,:))
xlabel t(days/100)
ylabel Chi
subplot(2,1,2),plot(t,X2,'k')
hold on
errorbar(t,a(2,:),P(2,2,:))
xlabel t(days/100)
ylabel Xi
save shortlongF -v7.3
```

19 Appendix 8: Matlab Code - Kalman Filter for The Length of A Contract Study (Model 2)

```
% Implementation for the short-/long-term model using the Kalman filter
% The SDE of the short-term/long-term model under the real-world framework is:
%     dchi = -kappa.chi.dt + sigma1.dZ1
%     dxi  = mu.dt + sigma2.dZ2
%     dZ1.dZ2 = rho.dt
clear all
clc
randn('seed',1)
rand('seed',1)
mydata = importdata('ContractDates_Synth_LengthStudy_90.xls');
ID = mydata.data(:,1);
C1 = mydata.data(:,2);     % 1 futures contract
g=1;
while C1(g)~=0
    g = g + 1;
end
H = 10*eye(1);

% Step 1: Generate the latent process under the real-world framework
T = 100;
kappa = 1.7; mu = 1.1; sigma1 = 1; sigma2 = 1; rho = 0.5;
chi0 = randn+0.5; xi0 = randn*2+0.3;  % random initial values
X1(1) = chi0;  X2(1) = xi0;
Delta = 0.01;  t = [0.01:Delta:1];
mean1 = chi0 * exp(-kappa*t);
mean2 = xi0 + mu*t;
for j = 1:length(t)-1
    dW1 = sqrt(Delta)*randn;
    dW2 = sqrt(Delta)*randn;
    X1(j+1) = (1 - kappa*Delta)*X1(j) + sigma1*dW1;
    X2(j+1) = X2(j) +  mu*Delta + sigma2*rho*dW1 + sigma2*sqrt(1-rho^2)*dW2;
end
XX1(1) = chi0; XX2(1) = xi0;
t1 = [0.0001:0.0001:1];
for j = 1:length(t1)-1
    dW1 = sqrt(0.0001)*randn;
    dW2 = sqrt(0.0001)*randn;
```

```
    XX1(j+1) = (1 - kappa*0.0001)*XX1(j) + sigma1*dW1;
    XX2(j+1) = XX2(j) +  mu*0.0001 + sigma2*rho*dW1 + sigma2*sqrt(1-rho^2)*dW2;
end
% Step 2: Calculate the futures price and the observed futures price
lambda1 = 1.2;   % The risk premium of chi
lambda2 = 0.7;   % The risk premium of xi
A = @(t,T) (lambda1/kappa)*(exp(kappa*(t-T)/100)-1)-((mu-lambda2)*(t-T)/100) ...
- (0.25/kappa)*(sigma1^2)*(exp(2*kappa*(t-T)/100)-1) ...
-0.5*(sigma2^2)*((t-T)/100)-(rho*sigma1*sigma2/kappa)*(exp(kappa*(t-T)/100)-1);
LNP1 = @(tau) exp(kappa*(-tau/100))*X1(90-tau)+X2(90-tau)+A(90-tau,90);
s = 1;
for i = 1:100
    ObsNoise = mvnrnd(zeros(1),H);
    if i <= 100-g
        LNF1(i) = LNP1(C1(i+g));
        LNF1o(i) = LNF1(i) + ObsNoise(1);
    else
        LNF1(i) = LNP1(C1(100)-s);
        LNF1o(i) = LNF1(i) + ObsNoise(1);
        s = s + 1;
    end
end
% Step 3: Using the Kalman filter to estimate the mean and covariance
% Firstly, establish the system matrices
W = @(k) [exp(kappa*(k-90)/100) 1];
d = @(k) [A(k,90)];
TT = [exp(-kappa*Delta) 0; 0 1];
c = [0; mu*Delta];
R = eye(2);
Q = @(k) [
(1-exp(-2*kappa*Delta))*(sigma1^2/(2*kappa)) (1-exp(-kappa*Delta))*(rho*sigma1*sigma2/kappa)
(1-exp(-kappa*Delta))*(rho*sigma1*sigma2/kappa) (sigma2^2)*Delta];
% Run the Kalman filter algorithm
a(:,1) = [0.5*rand+chi0; 0.5*rand+xi0];   % Initial value of the mean
P(:,:,1) = [2 0; 0 2];                    % Initial value of the covariance
for i = 2:90
    aa(:,i) = TT*a(:,i-1) + c;
    PP(:,:,i) = TT*P(:,:,i-1)*TT' + R*Q(i)*R';
    z(i) = [LNF1o(i)];
    F(i) = W(i)*PP(:,:,i)*(W(i))' + H;
```

```
    a(:,i) = aa(:,i) + PP(:,:,i)*(W(i))'*inv(F(i))*(z(:,i)-W(i)*aa(:,i)-d(i));
    P(:,:,i) = PP(:,:,i) - PP(:,:,i)*(W(i))'*inv(F(i))*W(i)*PP(:,:,i);
end
figure(1)
subplot(2,1,1),plot(t(1:90),X1(1:90),'k')
hold on
errorbar(t(1:90),a(1,:),P(1,1,:))
xlabel t(days/100)
ylabel Chi
subplot(2,1,2),plot(t(1:90),X2(1:90),'k')
hold on
errorbar(t(1:90),a(2,:),P(2,2,:))
xlabel t(days/100)
ylabel Xi
save shortlongFsim90 -v7.3
```

20 Appendix 9: Matlab Code - Extended Kalman Filter for Five Futures Contracts with maturity of 30 Days (Model 3)

```
% Implementation for the 3 factor model using the Extended Kalman filter
% The SDE of the 3-factor model (under the risk neutral framework) is:
%   dS = (r-Delta).S.dt + sigma1.S.dZ1
%   dDelta = kappa.(alpha-Delta).dt + sigma2.dZ2  (here alpha implies
%   alpha_hat)
%   dr = a.(m-r).dt + sigma3.dZ3  (here m implies m*)
%   dZ1.dZ2 = rho12.dt,  dZ1.dZ3 = rho13.dt,  dZ2.dZ3 = rho23.dt
clear all
clc
randn('seed',1)
rand('seed',1)
mydata = importdata('ContractDates_Synth.xls');
ID = mydata.data(:,1);
C1 = mydata.data(:,2);
C2 = mydata.data(:,3);
C3 = mydata.data(:,4);
C4 = mydata.data(:,5);
C5 = mydata.data(:,6);
g=1;
while C1(g)~=0
```

```
    g = g + 1;
end
g = g - 1;

H = 1*eye(5);

T = 100;
mu = 1.5; sigma1 = 2; sigma2 = 0.5; sigma3 = 0.1;
kappa = 1.4; a = 1.1; alpha = 0.85; m = 0.7;
rho12 = 0.8; rho23 = 0.3; rho13 = 0.1;
lambda1 = 1.2;   % The risk premium of Delta
lambda2 = 0.7;   % The risk premium of r
alpha1 = alpha - (lambda1/kappa); m1 = m - (lambda2/a);
S0 = 2*randn; Delta0 = randn+0.5; r0 = randn^2;
Delta = 0.01;  t = [0.01:Delta:1];
Deta = 0.001;  t0 = [0.001:Deta:1];
v = 1;
truth1 = []; truth2 = []; truth3 = [];
for u = 1:1000
    XE1 = []; XE2 = []; XE3 = [];
    XE1(1) = S0; XE2(1) = Delta0; XE3(1) = r0;
    for j = 1:length(t0)-1
        dW1 = sqrt(Deta)*randn;
        dW2 = sqrt(Deta)*randn;
        dW3 = sqrt(Deta)*randn;
        XE1(j+1) = XE1(j) + (mu-XE2(j))*XE1(j)*Deta + sigma1*XE1(j)*dW1;
        XE2(j+1) = XE2(j) + kappa*(alpha-XE2(j))*Deta + sigma2*rho12*dW1 ...
        + sigma2*sqrt(1-rho12^2)*dW2;
        XE3(j+1) = XE3(j) + a*(m-XE3(j))*Deta + sigma3*rho13*dW1 ...
        + sigma3*sqrt(1-rho13^2)*dW3;
    end
    for k = 1:100
        truth1(u,k) = XE1(k*10);
        truth2(u,k) = XE2(k*10);
        truth3(u,k) = XE3(k*10);
    end
end
for k = 1:100
    truemean1(k) = mean(truth1(:,k));
    truemean2(k) = mean(truth2(:,k));
```

```
    truemean3(k) = mean(truth3(:,k));
end
truth1=[]; truth2=[]; truth3=[];
XE1=[]; XE2=[]; XE3=[];
LNF1 = []; LNF1o = [];
% Calculate the futures price and the observed futures price
A = @(t,T) (rho23*sigma2*sigma3/(kappa*a*(kappa+a)))*(exp((kappa+a)*(t-T)/100)-1) ...
- (sigma2^2/(4*(kappa^3)))*(exp(2*kappa*(t-T)/100)-1) ...
- (sigma3^2/(4*(a^3)))*(exp(2*a*(t-T)/100)-1) ...
+ ((sigma2^2/(kappa^3))-(alpha1/kappa)-(rho12*sigma1*sigma2/(kappa^2)) ...
- (rho23*sigma2*sigma3/((kappa^2)*a)))*(exp(kappa*(t-T)/100)-1) ...
+ ((m1/a)+(sigma3^2/(a^3))-(rho23*sigma2*sigma3/(kappa*(a^2))) ...
+ (rho13*sigma1*sigma3/(a^2)))*(exp(a*(t-T)/100)-1) ...
+ (alpha1-m1-(sigma2^2/(2*(kappa^2)))-(sigma3^2/(2*(a^2))) ...
+ (rho12*sigma1*sigma2/kappa)+(rho23*sigma2*sigma3/(kappa*a)) ...
- (rho13*sigma1*sigma3/a))*((t-T)/100);
LNP1 = @(tau) truemean1(30-tau)+(1/kappa)*(exp(-kappa*tau/100)-1)*truemean2(30-tau) ...
+ (1/a)*(1-exp(-a*tau/100))*truemean3(30-tau)+A(30-tau,30);
LNP2 = @(tau) truemean1(60-tau)+(1/kappa)*(exp(-kappa*tau/100)-1)*truemean2(60-tau) ...
+ (1/a)*(1-exp(-a*tau/100))*truemean3(60-tau)+A(60-tau,60);
LNP3 = @(tau) truemean1(90-tau)+(1/kappa)*(exp(-kappa*tau/100)-1)*truemean2(90-tau) ...
+ (1/a)*(1-exp(-a*tau/100))*truemean3(90-tau)+A(90-tau,90);
LNP4 = @(tau) truemean1(120-tau)+(1/kappa)*(exp(-kappa*tau/100)-1)...
* truemean2(120-tau)+(1/a)*(1-exp(-a*tau/100))*truemean3(120-tau) + A(120-tau,120);
LNP5 = @(tau) truemean1(150-tau)+(1/kappa)*(exp(-kappa*tau/100)-1)*truemean2(150-tau) ...
+ (1/a)*(1-exp(-a*tau/100))*truemean3(150-tau)+A(150-tau,150);
s = 1;
for k = 1:100
    ObsNoise = mvnrnd(zeros(5),H);
    if k <= 100-g
        LNF1(k) = LNP1(C1(k+g));
        LNF1o(k) = LNF1(k) + ObsNoise(1);
        LNF2(k) = LNP2(C2(k+g));
        LNF2o(k) = LNF2(k) + ObsNoise(2);
        LNF3(k) = LNP3(C3(k+g));
        LNF3o(k) = LNF3(k) + ObsNoise(3);
        LNF4(k) = LNP4(C4(k+g));
        LNF4o(k) = LNF4(k) + ObsNoise(4);
        LNF5(k) = LNP5(C5(k+g));
        LNF5o(k) = LNF5(k) + ObsNoise(5);
```

```
    else
        LNF1(k) = LNP1(C1(100)-s);
        LNF1o(k) = LNF1(k) + ObsNoise(1);
        LNF2(k) = LNP2(C2(100)-s);
        LNF2o(k) = LNF2(k) + ObsNoise(2);
        LNF3(k) = LNP3(C3(100)-s);
        LNF3o(k) = LNF3(k) + ObsNoise(3);
        LNF4(k) = LNP4(C4(100)-s);
        LNF4o(k) = LNF4(k) + ObsNoise(4);
        LNF5(k) = LNP5(C5(100)-s);
        LNF5o(k) = LNF5(k) + ObsNoise(5);
        s = s + 1;
    end
end
% Using the Extended Kalman filter to generate the mean and covariance
% Firstly, establish the system matrices
f = @(S,delta,r,k) [
A(k,30)+log(S)+(1/kappa)*(exp(kappa*(k-30)/100)-1)*delta+(1/a)*(1-exp(a*(k-30)/100))*r;
A(k,60)+log(S)+(1/kappa)*(exp(kappa*(k-60)/100)-1)*delta+(1/a)*(1-exp(a*(k-60)/100))*r;
A(k,90)+log(S)+(1/kappa)*(exp(kappa*(k-90)/100)-1)*delta+(1/a)*(1-exp(a*(k-90)/100))*r;
A(k,120)+log(S)+(1/kappa)*(exp(kappa*(k-120)/100)-1)*delta+(1/a)*(1-exp(a*(k-120)/100))*r;
A(k,150)+log(S)+(1/kappa)*(exp(kappa*(k-150)/100)-1)*delta+(1/a)*(1-exp(a*(k-150)/100))*r];
h = @(S,delta,r) [S + (mu - delta)*S*Delta
                  delta + kappa*(alpha - delta)*Delta
                  r + a*(m - r)*Delta];
W = @(S,k) [
1/S  (1/kappa)*(exp(kappa*(k-30)/100)-1) (1/a)*(1 - exp(a*(k-30)/100));
1/S   (1/kappa)*(exp(kappa*(k-60)/100)-1) (1/a)*(1 - exp(a*(k-60)/100));
1/S   (1/kappa)*(exp(kappa*(k-90)/100)-1) (1/a)*(1 - exp(a*(k-90)/100));
1/S   (1/kappa)*(exp(kappa*(k-120)/100)-1) (1/a)*(1 - exp(a*(k-120)/100));
1/S   (1/kappa)*(exp(kappa*(k-150)/100)-1) (1/a)*(1 - exp(a*(k-150)/100));];
TT = @(S,delta) [1+(mu-delta)*Delta     0                 0
                 -S*Delta               1-kappa*Delta     0
                 0                      0                 1-a*Delta];
R = @(S) [
sigma1*S*sqrt(Delta)        0                                    0
sigma2*rho12*sqrt(Delta)    sigma2*sqrt(Delta*(1-(rho12^2)))     0
sigma3*rho13*sqrt(Delta)    0       sigma3*sqrt(Delta*(1-(rho13^2)))];
Q = eye(3);
% Run the Extended Kalman filter algorithm
```

```
aa(:,1) = [0.5*randn+S0; 0.5*randn+Delta0; 0.5*randn + r0];  %initial mean
P(:,:,1) = [2 0 0; 0 2 0; 0 0 2];        % initial covariance
for k = 2:100
    Tk_hat = TT(aa(1,k-1),aa(2,k-1));
    Rk_hat = R(aa(1,k-1));
    am(:,k) = h(aa(1,k-1),aa(2,k-1),aa(3,k-1));
    Pm(:,:,k) = Tk_hat*P(:,:,k-1)*(Tk_hat)' + Rk_hat*Q*(Rk_hat)';
    z(k) = [LNF1o(k)];
    F(:,:,:,:,:,k) = W(am(1,k),k)*Pm(:,:,k)*(W(am(1,k),k))' + H;
    aa(:,k) = am(:,k) + Pm(:,:,k)*(W(am(1,k),k))'*inv(F(:,:,:,:,:,k))*(z(k) ...
    - f(am(1,k),am(2,k),am(3,k),k));
    P(:,:,k) = Pm(:,:,k) - Pm(:,:,k)*(W(am(1,k),k))'*inv(F(:,:,:,:,:,k)) ...
    *W(am(1,k),k)*Pm(:,:,k);
    Tk_hat = []; Rk_hat = [];
end
figure(1)
plot(t,truemean1,'k')
hold on
errorbar(t,aa(1,:),P(1,1,:))
legend('True mean','EKF mean')
%title('Spot price (3 factor model)')
xlabel t(days/100)
ylabel S
hold off
figure(2)
plot(t,truemean2,'k')
hold on
errorbar(t,aa(2,:),P(2,2,:))
legend('True mean','EKF mean')
%title('Convenience yield (3 factor model)')
xlabel t(days/100)
ylabel delta
hold off
figure(3)
plot(t,truemean3,'k')
hold on
errorbar(t,aa(3,:),P(3,3,:))
legend('True mean','EKF mean')
%title('Interest rate (3 factor model)')
xlabel t(days/100)
```

```
ylabel r
hold off
save EKFMod3_5Contracts -v7.3
```

21 Appendix 10: Matlab Code - Particle Filter for One Futures Contract with Maturity of 30 Days (Model 3)

```
function [XX, WW] = GenericPF(X,W,LNFo,LNF,path,Ns,NT,k)
% Generic Particle Filter algorithm
for i = 1:Ns
    XX(i) = path(i,k); % + noise-state!!!
    WW(i) = normpdf(LNFo(i),LNF(i),1);
end
h = sum(WW(:));
for i = 1:Ns
    WW(i) = WW(i)/h;
end
Neff = 1/(sum(WW(:).^2));
if Neff < NT
    % Resampling Algorithm
    c(1) = 0;
    for i = 2:Ns
        c(i) = c(i-1) + WW(i);
    end
    i = 1;
    u(1) = (1/Ns)*rand;
    for j = 1:Ns
        u(j) = u(1) + (1/Ns)*(j-1);
        while i <= Ns && u(j) > c(i)
            i = i + 1;
        end
        if i > Ns
            i = i - 1;
        end
        XX(j,k) = XX(i);
        WW(j,k) = 1/Ns;
    end
end
end
```

```
% Implementation for the 3 factor model using the Particle filter
% The SDE of the 3-factor model (under the risk neutral framework) is:
%   dS = (r-Delta).S.dt + sigma1.S.dZ1
%   dDelta = kappa.(alpha-Delta).dt + sigma2.dZ2  (here alpha implies
%   alpha_hat)
%   dr = a.(m-r).dt + sigma3.dZ3  (here m implies m*)
%   dZ1.dZ2 = rho12.dt,  dZ1.dZ3 = rho13.dt,  dZ2.dZ3 = rho23.dt
clear all
clc
randn('seed',1)
rand('seed',1)
mydata = importdata('ContractDates_Synth.xls');
ID = mydata.data(:,1);
C1 = mydata.data(:,2);
g=1;
while C1(g)~=0
    g = g + 1;
end
H = eye(1);
% Step 1: Generate the latent process in the real-world parameters
%Ns = 200; NT = 100;   % Case: choosing 200 particles, and a threshold of 100
%Ns = 5000; NT = 200;  % Case: choosing 5000 particles, and a threshold of 200
Ns = 1000; NT = 200;   % Case: choosing 1000 particles, and a threshold of 200
T = 100;
mu = 1.5; sigma1 = 2; sigma2 = 0.5; sigma3 = 0.1;
kappa = 1.4; a = 1.1; alpha = 0.85; m = 0.7;
rho12 = 0.8; rho23 = 0.3; rho13 = 0.1;
lambda1 = 1.2;   % The risk premium of Delta
lambda2 = 0.7;   % The risk premium of r
alpha1 = alpha - (lambda1/kappa); m1 = m - (lambda2/a);
S0 = 2*randn; Delta0 = randn+0.5; r0 = randn^2;
Delta = 0.01; t = [0.01:Delta:1];
Deta = 0.001; t0 = [0.001:Deta:1];
v = 1;
truth1 = []; truth2 = []; truth3 = [];
for u = 1:1000
    XE1 = []; XE2 = []; XE3 = [];
    XE1(1) = S0; XE2(1) = Delta0; XE3(1) = r0;
```

```
    for j = 1:length(t0)-1
        dW1 = sqrt(Deta)*randn;
        dW2 = sqrt(Deta)*randn;
        dW3 = sqrt(Deta)*randn;
        XE1(j+1) = XE1(j) + (mu-XE2(j))*XE1(j)*Deta + sigma1*XE1(j)*dW1;
        XE2(j+1) = XE2(j) + kappa*(alpha-XE2(j))*Deta + sigma2*rho12*dW1 ...
        + sigma2*sqrt(1-rho12^2)*dW2;
        XE3(j+1) = XE3(j) + a*(m-XE3(j))*Deta + sigma3*rho13*dW1 ...
        + sigma3*sqrt(1-rho13^2)*dW3;
    end
    for k = 1:100
        truth1(u,k) = XE1(k*10);
        truth2(u,k) = XE2(k*10);
        truth3(u,k) = XE3(k*10);
    end
end
for k = 1:100
    truemean1(k) = mean(truth1(:,k));
    truemean2(k) = mean(truth2(:,k));
    truemean3(k) = mean(truth3(:,k));
end
truth1=[]; truth2=[]; truth3=[];
XE1=[]; XE2=[]; XE3=[];
LNF1 = []; LNF1o = [];
for i = 1:Ns
    X1 = []; X2 = []; X3 = [];
    X1(1) = S0; X2(1) = Delta0; X3(1) = r0;
    for k = 2:length(t)
        dW1 = sqrt(Delta)*randn;
        dW2 = sqrt(Delta)*randn;
        dW3 = sqrt(Delta)*randn;
        X1(k) = X1(k-1) + (mu-X2(k-1))*X1(k-1)*Delta + sigma1*X1(k-1)*dW1 ...
        + 0.5*(sigma1^2)*X1(k-1)*((dW1^2)-Delta);
        X2(k) = X2(k-1) + kappa*(alpha-X2(k-1))*Delta + sigma2*rho12*dW1 ...
        + sigma2*sqrt(1-rho12^2)*dW2;
        X3(k) = X3(k-1) + a*(m-X3(k-1))*Delta + sigma3*rho13*dW1 ...
        + sigma3*sqrt(1-rho13^2)*dW3;
    end
    for k = 1:length(t)
        path1(i,k) = X1(k);
```

```
        path2(i,k) = X2(k);
        path3(i,k) = X3(k);
    end
    % Calculate the futures price and the observed futures price
    A = @(t,T) (rho23*sigma2*sigma3/(kappa*a*(kappa+a)))*(exp((kappa+a)*(t-T)/100)-1) ...
     -(sigma2^2/(4*(kappa^3)))*(exp(2*kappa*(t-T)/100)-1) ...
     -(sigma3^2/(4*(a^3)))*(exp(2*a*(t-T)/100)-1) ...
     +((sigma2^2/(kappa^3))-(alpha1/kappa)-(rho12*sigma1*sigma2/(kappa^2)) ...
     -(rho23*sigma2*sigma3/((kappa^2)*a)))*(exp(kappa*(t-T)/100)-1) ...
     +((m1/a)+(sigma3^2/(a^3))-(rho23*sigma2*sigma3/(kappa*(a^2))) ...
     +(rho13*sigma1*sigma3/(a^2)))*(exp(a*(t-T)/100)-1) ...
     +(alpha1-m1-(sigma2^2/(2*(kappa^2)))-(sigma3^2/(2*(a^2))) ...
     +(rho12*sigma1*sigma2/kappa)+(rho23*sigma2*sigma3/(kappa*a)) ...
     -(rho13*sigma1*sigma3/a))*((t-T)/100);
    LNP1 = @(tau) X1(30-tau)+(1/kappa)*(exp(-kappa*tau/100)-1)*X2(30-tau) ...
    +(1/a)*(1-exp(-a*tau/100))*X3(30-tau)+A(30-tau,30);
    s = 1;
    for k = 1:100
        ObsNoise = mvnrnd(zeros(1),H);
        if k <= 100-g
            LNF1(i,k) = LNP1(C1(k+g));
            LNF1o(i,k) = LNF1(i,k) + ObsNoise(1);
        else
            LNF1(i,k) = LNP1(C1(100)-s);
            LNF1o(i,k) = LNF1(i,k) + ObsNoise(1);
            s = s + 1;
        end
    end
end
% Using the Particle filter
Xp1 = []; Xp2 = []; Xp3 = [];
Wp1 = []; Wp2 = []; Wp3 = [];
for i = 1:Ns
    Xp1(i,1) = 0.5*randn + S0;
    Xp2(i,1) = 0.5*randn + Delta0;
    Xp3(i,1) = 0.5*randn + r0;
    Wp1(i,1) = 1/Ns;
    Wp2(i,1) = 1/Ns;
    Wp3(i,1) = 1/Ns;
end
```

```
for k = 2:length(t)
    [Xo1, Wo1] = GenericPF(Xp1(:,k-1),Wp1(:,k-1),LNF1o(:,k),LNF1(:,k),path1,Ns,NT,k);
    [Xo2, Wo2] = GenericPF(Xp2(:,k-1),Wp2(:,k-1),LNF1o(:,k),LNF1(:,k),path2,Ns,NT,k);
    [Xo3, Wo3] = GenericPF(Xp3(:,k-1),Wp3(:,k-1),LNF1o(:,k),LNF1(:,k),path3,Ns,NT,k);
    for i = 1:Ns
        Xp1(i,k) = Xo1(i);
        Xp2(i,k) = Xo2(i);
        Xp3(i,k) = Xo3(i);
        Wp1(i,k) = Wo1(i);
        Wp2(i,k) = Wo2(i);
        Wp3(i,k) = Wo3(i);
    end
    Xo1 = []; Xo2 = []; Xo3 = []; Wo1 = []; Wo2 = []; Wo3 = [];
end
% Calculate the mean obtained by the Particle filter
for k = 1:length(t)
    XP1(k) = sum(Xp1(:,k).*Wp1(:,k));
    XP2(k) = sum(Xp2(:,k).*Wp2(:,k));
    XP3(k) = sum(Xp3(:,k).*Wp3(:,k));
end
% Calculate the standard deviations of the estimate points generated by the Particle filter
for k = 1:length(t)
    Std_SpotPrice(k) = sqrt(mean(Wp1(:,k).*((Xp1(:,k)-truemean1(k)).^2)));
    Std_ConvenienceYield(k) = sqrt(mean(Wp2(:,k).*((Xp2(:,k)-truemean2(k)).^2)));
    Std_InterestRate(k) = sqrt(mean(Wp3(:,k).*((Xp3(:,k)-truemean3(k)).^2)));
end
figure(1)
plot(t,truemean1,'k')
hold on
errorbar(t,XP1,Std_SpotPrice)
legend('True mean','Particle filter mean','Location','NorthWest')
%title('Spot price factor (3 factor model)')
xlabel t(days/100)
ylabel Spotprice
hold off
figure(2)
plot(t,truemean2,'k')
hold on
errorbar(t,XP2,Std_ConvenienceYield)
legend('True mean','Particle filter mean','Location','NorthWest')
```

```
%title('Convenience yield factor (3 factor model)')
xlabel t(days/100)
ylabel Convenienceyield
hold off
figure(3)
plot(t,truemean3,'k',t,XP3)
hold on
errorbar(t,XP3,Std_InterestRate)
legend('True mean','Particle filter mean','Location','NorthWest')
%title('Interest rate factor (3 factor model)')
xlabel t(days/100)
ylabel Interestrate
hold off
save Mod3ParticleFilter1Contract -v7.3
```