Latent state estimation in a class of nonlinear systems

Ksenia Ponomareva

A thesis submitted for the degree of Doctor of Philosophy

Department of Mathematical Sciences

Brunel University

February, 2012

Abstract

The problem of estimating latent or unobserved states of a dynamical system from observed data is studied in this thesis. Approximate filtering methods for discrete time series for a class of nonlinear systems are considered, which, in turn, require sampling from a partially specified discrete distribution. A new algorithm is proposed to sample from partially specified discrete distribution, where the specification is in terms of the first few moments of the distribution. This algorithm generates deterministic sigma points and corresponding probability weights, which match exactly a specified mean vector, a specified covariance matrix, the average of specified marginal skewness and the average of specified marginal kurtosis. Both the deterministic particles and the probability weights are given in closed form and no numerical optimization is required. This algorithm is then used in approximate Bayesian filtering for generation of particles and the associated probability weights which propagate higher order moment information about latent states. This method is extended to generate random sigma points (or particles) and corresponding probability weights that match the same moments. The algorithm is also shown to be useful in scenario generation for financial optimization. For a variety of important distributions, the proposed moment-matching algorithm for generating particles is shown to lead to approximation which is very close to maximum entropy approximation.

In a separate, but related contribution to the field of nonlinear state estimation, a closed-form linear minimum variance filter is derived for the systems with stochastic parameter uncertainties. The expressions for eigenvalues of the perturbed filter are derived for comparison with eigenvalues of the unperturbed Kalman filter. Moment-matching approximation is proposed for the nonlinear systems with multiplicative stochastic noise.

Acronyms

AvMRAE Average mean relative absolute error

AvRMSE Average root mean square error

CEV Constant elasticity of variance model

CIR Cox Ingersoll Ross interest rate model

CVaR Conditional value at risk

HOSPoF Higher order sigma point filter

PF-HOSPoF Particle filter with higher order sigma point filter proposal

PF-UKF Particle filter with unscented Kalman filter proposal

VaR Value at risk

Contents

List of Figures

Acknowledgments

I would like to thank my supervisor Dr Paresh Date for his encouragement and guidance. His experience and knowledge helped me through various challenges in my research.

I would also like to thank my family for support and encouragement throughout all these years.

Chapter 1

Introduction

1.1 Introduction

In this introductory chapter a motivation for studying the problem of latent state estimation is presented and the main contributions of the thesis are described in general terms. A short introduction for the chapters to follow is also given.

1.2 The problem of latent state estimation

The problem of estimating latent or unobserved states of a dynamical system from observed data often arises in many branches of science, such as engineering, econometrics, weather sciences and finance. The technique used to deal with this problem is called filtering. *Filtering* refers to any method for obtaining such state estimates, recursively in time, by combining model predictions with noisy observations. This technique can be applied to the problem of estimating current and future weather patterns from observed atmospheric data in weather forecasting. In mathematical finance, filtering can be used for pricing more complex financial instruments, like derivatives, when it is necessary to be able to estimate the "unseen" variables, such as the underlying interest rates implied by the observed bond prices or the underlying volatility implied by the stock prices etc, with a reasonable accuracy. In filtering, the state of the system may be estimated as an entire distribution (conditional on measurements up to that time) or its mean and variance conditional on current measurements

may only be estimated. For systems with linear dependence on the present state, a linear recursive estimator, called *the Kalman filter* after its inventor, is used. This filter is a popular state estimator, which is optimal in mean square sense, and it is simple to implement. For more complex nonlinear systems, filtering still remains a difficult problem. An exact nonlinear filter, with properties similar to that of Kalman filter in the linear case, is often unknown or impossible to implement. Many different approximate Bayesian filtering methods exist to deal with state estimation in nonlinear systems, such as extended Kalman filter (EKF), sigma point filter, ensemble filter and particle filter (PF). These would be examined in more detail in the next chapter.

Despite the increasing use of nonlinear filters in practice, some theoretical and computational problems remain unsolved. Two challenges considered in this thesis are:

1. Many different filtering methods exist with varying degrees of accuracy and computational complexity. Some nonlinear filters propagate only the information about the first two moments of the latent state; these include sigma point filters and ensemble filters. At the other extreme of the compromise between computation required and statistical information propagated, there are particle filters which are computationally a lot more intensive but can propagate full distributional information. The interest is in developing something which fits between these two in terms of the information propagated and still has the same computational effort as the sigma point filters.

2. Parameters of a nonlinear model are often estimated from data and are not known exactly. A small perturbation in the model parameters can lead to large estimation errors, which may grow over time in a recursive estimation procedure. Stability of the Kalman filter, where parameter perturbation is represented by a deterministic (unknown-but-bounded) multiplicative noise (e.g. multiplied by the state vector), has been extensively studied. The case is of particular interest when this multiplicative noise is a stochastic uncertainty and the covariance of transition noise is an affine function of the state vector, as is the case with a class of the financial time series models (Cox-Ingersoll-Ross type models described later in chapter 3).

A brief introduction to the rest of the thesis now follows.

Chapter 2 reviews the literature on latent state estimation problem from various disciplines including mathematical finance, weather sciences and engineering. The brief overview of time series analysis methods is also provided. The linear Gaussian filter (i.e. Kalman filter) is explored in some detail and also some of the other nonlinear filters are outlined which are used for comparison of the performance of the new filters later in the thesis. One of the nonlinear filters reviewed is unscented Kalman filter (UKF), whose advantages and disadvantages are discussed. Chapter 3 tackles the first challenge mentioned above and presents a new method for deterministic sigma point generation that aims to address some of the disadvantages of the traditional UKF. In particular, this new algorithm generates a set of deterministic sigma points and corresponding probability weights that match the given mean vector, the covariance matrix, the average marginal third and fourth moments exactly, without optimization. This allows the propagation of the higher order moment information without significant extra computation as compared to the traditional unscented Kalman filter. The filtering algorithm based on this new method is provided and it forms the new filter, called higher order sigma point filter (HOSPoF). Theoretical accuracy of the new sigma point generation method is also considered for conditional mean and covariance estimation. The performance of HOSPoF is compared to some of the existing nonlinear filters, described in chapter 2.

Even though HOSPoF, introduced in chapter 3, outperforms two other filters used for comparison in the numerical experiments, the number of sigma points generated still depends on the dimension of the state vector, as is the case with the traditional UKF. This makes the new method somewhat restricted as a larger number of sigma points could give a better representation of the posterior density. Chapter 4 addresses this issue and proposes a new method for generating random sigma points and corresponding probabilities that match the given mean vector, the covariance matrix, the average marginal third and fourth moments exactly, without the use of optimization. This method is significantly different from the one discussed in chapter 3, as not only it allows generation of any number of particles, but also almost all of the corresponding probabilities are generated randomly. The new filter, based on the moment matching proposal distribution, is called particle filter with higher order sigma point filter proposal (PF-HOSPoF). Its performance is compared with the particle filter with proposal generated by the traditional unscented Kalman filter (PF-UKF). Results show a significant

improvement achieved by PF-HOSPoF over PF-UKF in terms of errors and computational times.

So far, nonlinear fitlering methods for discrete time series have been considered, which involved sampling from partially specified discrete distribution, where specification is in terms of the first few moments of the distribution. The problem of determining the discrete probabilities of a set of events, conditioned upon the moment constraints, is quite common in probability and statistics. One of the methods to solve this problem is maximum entropy approach. In chapter 5, the new moment matching method, proposed in chapter 4, is compared to the method of entropy maximization. It is shown that the new method for random sigma point generation is a good alternative as it does not require optimization and yields a relative entropy which is remarkably close to the best approximation.

An application of the sigma point generation, introduced in chapter 4, in financial portfolio optimization is presented in chapter 6. One of the traditional approaches for decision making under uncertainty and risk is stochastic programming. This involves optimization problems in which some parameters are not certain, but are described by statistical distributions. In order for the stochastic programs to be numerically solvable, the distributions involved are approximated by discrete distributions with a finite number of scenarios (sigma points). The algorithm for sigma point generation from chapter 4 is used here for mean conditional value at risk (mean-CVaR) portfolio optimization model and is tested on financial market data. It is illustrated that desirable properties for a scenario generator are satisfied, including in-sample and out-of-sample stability. It is also shown that optimal solutions vary only marginally with increasing number of scenarios; thus, good solutions can be obtained with a relatively small number of scenarios.

Chapter 7 returns to the laten state estimation problem, discussed in chapters 2-4, and tackles the second challenge of nonlinear filtering. Unlike deterministic parameters used in models in the previous chapters, discrete time series filtering here is considered under random parameter perturbations of the state space model. The new closed-form minimum variance filter is derived for discrete systems with stochastic uncertainties in state parameters. Analysis of the sensitivity of the new filter to the size of the parameter perturbation is presented. Approximate moment-matching algorithm, inspired by the exact method introduced here, is also proposed for univariate time series that appears to work well for a wider case of nonlinear systems.

Finally, chapter 8 summarizes the main contributions of the thesis and suggests ideas for future work.

1.3 Notation

In this thesis upper case letters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{Q}$ and $\mathbf{R}$ represent constant matrices. However, in chapter 7 a perturbation matrix $\mathbf{\Delta_A}$ is introduced, where each element represents a noise source. $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ represent state and measurement vectors respectively in chapters 2-4 and 7. Lower case, boldface letters represent vector valued functions (e.g. **f**, **h** in chapter 2), vector valued random variables (e.g. **w**, **v**) or constant vectors (e.g. **d**). In chapters 2-4 and 7, **w** and **v** are uncorrelated Gaussian variables. Functions **f** and **h** are deterministic and can be linear or nonlinear, which is specified in the chapters. The definition of other symbols would be given in the chapter where they first appear.

1.4 Summary

In conclusion, it can be said that the thesis should be of interest to those working in both filtering theory and mathematical finance. New methods have been introduced for deterministic and random sigma point generation, a closed-form minimum variance filter has been derived for discrete time systems with parameter perturbations and a new scenario generation method has been introduced for portfolio optimization. Improved accuracy in nonlinear filtering and model calibration can help reduce pricing errors, improve risk measurement and help with stock investing strategy. Although the focus of this research has been on financial systems, the methods developed will be useful in other fields where filtering is applied, such as engineering, econometrics and weather sciences.

Chapter 2

Preliminaries

2.1 The discrete time filtering problem

Time series data occur naturally in many application areas, such as econometrics and finance, environmental and industrial processes. Time series analysis refers to problems in which observations are collected at regular time intervals and there are correlations among successive observations. These series can be decomposed into four distinct components, such as trend, seasonal effects, cycles and residuals. There are two main goals of time series analysis. First one is identifying the nature of the phenomenon represented by the sequence of observations. Second goal is forecasting or predicting future values of the time series variable. Both of these goals require that the pattern of observed time series data is identified and formally described. Once the pattern is established, it is possible to integrate it with other data and extrapolate the identified pattern to predict future events. The autoregressive moving average (ARIMA) methodology developed by Box & Jenkins [1976] allows user to uncover the hidden patterns in the data and also generate the forecasts. It is one of the main analytical systems currently in use for time series analysis and research practice confirms its power and flexibility. Please see Hoff [1983], Pankratz [1983] and Vandaele [1983] for more details. A different approach is to put individual component models into a single model, called a state space model, which provides the basis for analysis. The techniques that emerge from this approach are very flexible and are capable of handling a much wider range of problems than the Box-Jenkins ARIMA system, see Durbin & Koopman [2001]. From this point onwards, state space approach is employed

in the thesis. Also the scope of this thesis is limited to only discrete time series. For discussion and more detail on the continuous case, please see Kallianpur [1990].

Consider the following discrete state space form:

$$\boldsymbol{\mathcal{X}}(k+1) = \mathbf{f}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{Q}(\boldsymbol{\mathcal{X}}(k))\mathbf{w}(k+1), \tag{2.1a}$$

$$\boldsymbol{\mathcal{Y}}(k) = \mathbf{h}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{R}\ \mathbf{v}(k), \tag{2.1b}$$

where $\boldsymbol{\mathcal{X}}(k)$ and $\boldsymbol{\mathcal{Y}}(k)$ are the respective state vector and measurement vector at time t_k; $\mathbf{f}$, $\mathbf{h}$ are given vector-valued deterministic functions; $\mathbf{Q}$ is a matrix valued deterministic function; $\mathbf{R}$ is a deterministic matrix and $\mathbf{v}(k), \mathbf{w}(k)$ are vector-valued random variables. The time increment $t(k) - t(k-1)$ is assumed constant for all k. The latent state estimation problem is the problem of constructing an estimate of the random vector $\boldsymbol{\mathcal{X}}(k)$, $k \geq 1$, based on the noisy time series data $\boldsymbol{\mathcal{Y}}(1), \boldsymbol{\mathcal{Y}}(2), ..., \boldsymbol{\mathcal{Y}}(k)$. The filtering problem is the problem of finding the conditional distribution of $\mathbb{E}(\boldsymbol{\mathcal{X}}(k+1)|\boldsymbol{\mathcal{X}}(k), \boldsymbol{\mathcal{Y}}(k))$, or finding the samples of conditional moments of this random variable corresponding to the observed values of $\boldsymbol{\mathcal{Y}}(k)$ and estimated $\boldsymbol{\mathcal{X}}(k)$.

In the special case when $\mathbf{f}$, $\mathbf{h}$ are affine in $\boldsymbol{\mathcal{X}}(k)$, $\mathbf{Q}$ and $\mathbf{R}$ are identity matrices and $\mathbf{v}(k), \mathbf{w}(k)$ are Gaussian, the optimal recursive solution to the state estimation problem is given by the linear Kalman filter, as first outlined in Kalman [1960]. The Kalman filter and its generalizations have been the main tools for estimating the unobserved variables from the observed ones in econometrics and in engineering for several decades and their use is now becoming common in finance. The Kalman filter is a conditional moment estimator for linear Gaussian systems. It is used in calibration of time series models, forecasting of variables and also in data smoothing applications. In the next subsection a linear Kalman filter is described for a special case of a state space model in (2.1) before discussing the nonlinear filtering in section 2.2. The Kalman filter's application to maximum likelihood calibration of time series models is described in section 2.3.

2.1.1 The linear Kalman filter

Consider a discrete time, linear state space system:

$$\boldsymbol{\mathcal{X}}(k+1) = \mathbf{A}\boldsymbol{\mathcal{X}}(k) + \mathbf{b} + \mathbf{Q}\,\mathbf{w}(k+1), \tag{2.2a}$$

$$\boldsymbol{\mathcal{Y}}(k) = \mathbf{C}\boldsymbol{\mathcal{X}}(k) + \mathbf{d} + \mathbf{R}\,\mathbf{v}(k), \tag{2.2b}$$

where, $\mathbf{w}(k), \mathbf{v}(k)$ zero mean, unit variance, Gaussian and uncorrelated random variables at each time $t(k)$. $\mathbf{A}$, $\mathbf{b}$, $\mathbf{C}$, $\mathbf{d}$, $\mathbf{Q} > 0$ and $\mathbf{R} > 0$ are constants or are known functions of time. Only the real valued variable $\boldsymbol{\mathcal{Y}}(k)$ is measured or is observable; the variable $\boldsymbol{\mathcal{X}}(k)$ is of interest and needs to be estimated. Equations (2.2a)- (2.2b) are a special case of system (2.1) with linear functions $\mathbf{f}(\boldsymbol{\mathcal{X}}(k)) = \mathbf{A}\boldsymbol{\mathcal{X}}(k) + \mathbf{b}$ and $\mathbf{h}(\boldsymbol{\mathcal{X}}(k)) = \mathbf{C}\boldsymbol{\mathcal{X}}(k) + \mathbf{d}$.

The estimate of $\boldsymbol{\mathcal{X}}(k)$ based on information up to time $t(k-i)$ is denoted as $\hat{\boldsymbol{\mathcal{X}}}(k|k-i)$ for $i \geq 0$ and it is assumed that the initial estimate $\hat{\boldsymbol{\mathcal{X}}}(0|0)$ is known. The conditional variance of the estimate is denoted by $\mathbf{P}_{xx}(k|k-i)$ and $\mathbf{P}_{xx}(0|0) > 0$ is assumed to be known. With this notation, the following set of recursive equations is conventionally referred to as the *Kalman filter*:

$$\mathbf{z}(k) = \boldsymbol{\mathcal{Y}}(k) - (\mathbf{C}\hat{\boldsymbol{\mathcal{X}}}(k|k-1) + \mathbf{d}), \tag{2.3a}$$

$$\mathbf{S}(k) = \mathbf{C}\mathbf{P}_{xx}(k|k-1)\mathbf{C}^T + \mathbf{R}\mathbf{R}^T, \tag{2.3b}$$

$$\mathbf{K}(k) = \mathbf{P}_{xx}(k|k-1)\mathbf{C}^T\mathbf{S}(k)^{-1}, \tag{2.3c}$$

$$\hat{\boldsymbol{\mathcal{X}}}(k|k) = \hat{\boldsymbol{\mathcal{X}}}(k|k-1) + \mathbf{K}(k)\mathbf{z}(k), \tag{2.3d}$$

$$\mathbf{P}_{xx}(k|k) = \mathbf{P}_{xx}(k|k-1) - \mathbf{K}(k)\mathbf{C}\mathbf{P}_{xx}(k|k-1), \tag{2.3e}$$

$$\hat{\boldsymbol{\mathcal{X}}}(k+1|k) = \mathbf{A}\hat{\boldsymbol{\mathcal{X}}}(k|k) + \mathbf{b}, \tag{2.3f}$$

$$\mathbf{P}_{xx}(k+1|k) = \mathbf{A}\mathbf{P}_{xx}(k|k)\mathbf{A}^T + \mathbf{Q}\mathbf{Q}^T. \tag{2.3g}$$

Here, $\mathbf{z}(k)$ in (2.3a) represent information which could not have been derived from data up to time $t(k-1)$ and are called *innovations*. $\mathbf{S}(k)$ represents the covariance matrix of innovations. $\hat{\boldsymbol{\mathcal{X}}}(k+1|k)$

is in fact the sample of conditional expectation of $\boldsymbol{\mathcal{X}}(k+1)$ based on information up to time $t(k)$, determined by the realized value of $\boldsymbol{\mathcal{Y}}(k)$. This set of equations can be derived from the following relationship between conditional moments of jointly Gaussian variables (see, e.g. Grimmett & Stirzaker [2004]): if x, y are jointly Gaussian,

$$\mathbb{E}(\mathbf{x}|\mathbf{y}) \;=\; \mathbb{E}(\mathbf{x}) + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\mathbf{y} - \mathbb{E}(\mathbf{y})), \tag{2.4}$$

$$\mathbb{E}(\mathbf{x} - \mathbb{E}(\mathbf{x}|\mathbf{y}))(\mathbf{x} - \mathbb{E}(\mathbf{x}|\mathbf{y}))^T \;=\; \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}, \tag{2.5}$$

where $\boldsymbol{\Sigma}_{xy}$ *etc.* are variance terms. This has to be interpreted with care since $\hat{\boldsymbol{\mathcal{X}}}(k|k-1)$ is not *unconditional* mean; the reader is referred to specialist textbooks such as Durbin & Koopman [2001] for more details. Given $\boldsymbol{\mathcal{Y}}(k)$, $\hat{\boldsymbol{\mathcal{X}}}(k+1|k)$ and $(\mathbf{C}\hat{\boldsymbol{\mathcal{X}}}(k+1|k) + \mathbf{d})$ serve as one-step ahead forecasts of $\boldsymbol{\mathcal{X}}(k+1)$ and $\boldsymbol{\mathcal{Y}}(k+1)$ respectively.

2.2 Approximate nonlinear filtering

Now return to the system of equations (2.1) with nonlinear functions $\mathbf{f}$ and $\mathbf{h}$ and consider the filtering problem for nonlinear systems. Unlike system (2.2a)- (2.2b), the optimal recursive solution to the state estimation problem in nonlinear systems is usually not available in closed form. Current approaches to address the nonlinear filtering problems fall under one of the approximate Bayesian filtering methods. In the next subsections these methods are briefly described.

2.2.1 Extended Kalman filter

The first approach is *extended* Kalman filter. For scalar version of (2.1), one can expand the dynamics in Taylor series about $\hat{\boldsymbol{\mathcal{X}}}(k|k-1)$ as

$$\mathbf{f}(\boldsymbol{\mathcal{X}}(k)) \approx \mathbf{f}(\hat{\boldsymbol{\mathcal{X}}}(k|k-1)) + \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\mathcal{X}}}\right)(\boldsymbol{\mathcal{X}}(k) - \hat{\boldsymbol{\mathcal{X}}}(k|k-1)), \tag{2.6a}$$

$$\mathbf{h}(\boldsymbol{\mathcal{X}}(k)) \approx \mathbf{h}(\hat{\boldsymbol{\mathcal{X}}}(k|k-1)) + \left(\frac{\partial \mathbf{h}}{\partial \boldsymbol{\mathcal{X}}}\right)(\boldsymbol{\mathcal{X}}(k) - \hat{\boldsymbol{\mathcal{X}}}(k|k-1)), \tag{2.6b}$$

$$\mathbf{Q}(\boldsymbol{\mathcal{X}}(k)) \approx \mathbf{Q}(\hat{\boldsymbol{\mathcal{X}}}(k|k-1)), \tag{2.6c}$$

where the partial derivatives are evaluated at $\hat{\boldsymbol{\mathcal{X}}}(k|k-1)$. Equation (2.6) gives a linear approximation to the original nonlinear state space system, which, in turn allows to use the techniques from

sections 2.1.1 and 2.3 for model calibration and forecasting. This method works reasonably well in systems with smooth nonlinearities, however such an assumption is often not easy to validate. Standard textbooks such as Anderson & Moore [1979] carry an extensive discussion of its theoretical underpinnings and implementation; also see Jazwinski [1970] and Sorenson [1985]. This filter is referred to as a *local linearization filter* in Jimenez & Ozaki [2003] and a similar formulation has been used in parameter estimation for forward rate models in Chiarella *et al.* [2009]. Another problem with implementing the EKF is that it requires the calculation of the Jacobian matrix. The derivation of the Jacobian matrices is nontrivial in most applications and often can lead to significant implementation difficulties. More efficient alternatives to this method are introduced next.

2.2.2 Derivative free methods

Truncated Taylor series, used in EKF, provide an insufficiently accurate representation in many cases and significant bias or problems with convergence can arise due to the overly crude approximation. Methods that do not require computation of derivatives of function are particularly suitable for problems which the derivatives are not available or are extremely expensive to compute. Filters that resemble extended Kalman filter, but where estimators are based on the polynomial approximations of the nonlinear transformations using Stirling's interpolation formula instead of Taylor's linearization, have been proposed in Norgaard *et al* [2000]. These filters are simple to implement as no derivatives are needed, yet they provide excellent accuracy. Developed independently from Norgaard *et al* [2000], a similar approach, called a central difference filter, was introduced in Ito & Xiong [2000]. More recently a new derivative free approach was proposed in Arasaratnam & Haykin [2009]. In this paper a more accurate nonlinear filter was derived that could be applied to solve high-dimensional nonlinear filtering problems with minimal computational effort. The Bayesian filter solution in the Gaussian domain reduces to the problem of how to compute multi-dimensional integrals and cubature rule is used to solve this problem.

A different method, that also only uses functional evaluations instead of analytical Taylor series linearization, was proposed in Julier & *et al* [1995] and is described in more detail next.

2.2.3 Sigma point filters

An increasing popular alternative to EKF for signal processing in real time is using *unscented* or *sigma point* filters. Methods of this type have been developed independently in engineering (see, *e.g.* Julier & *et al* [1995], Julier & Uhlmann [1997] and references therein for examples) and in weather sciences, where they are referred to as *ensemble Kalman filters* (see, *e.g.* Evensen [1994] and Mitchell & Hotekamer [1998] for examples). In contrast with the extended Kalman filters described earlier, sigma point filters do not involve computation of derivatives. Augmentation method (see, *e.g.* Wu & Hu [2005]) incorporates noise into the augmented random state vector and from here onwards it will be assumed $\mathbf{f}$ and $\mathbf{h}$ to be augmented functions. The unscented filtering algorithm can be briefly described as follows.

Suppose that at time t_k, the mean $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ and the covariance $\mathbf{P}_{xx}(k|k)$ are available for the system in equation (2.1). Then $2n+1$ symmetric *sigma points* are chosen in the following way:

$$\begin{aligned} \boldsymbol{\mathcal{X}}^{(0)}(k|k) &= \hat{\boldsymbol{\mathcal{X}}}(k|k), \\ \boldsymbol{\mathcal{X}}^{(i)}(k|k) &= \hat{\boldsymbol{\mathcal{X}}}(k|k) \pm (\sqrt{(n+\kappa)\mathbf{P}_{xx}})_i, \end{aligned} \tag{2.7}$$

where $i = 1, 2, ..., n$, n is a dimension of the state vector, κ is a scaling parameter and $(\sqrt{\mathbf{P}_{xx}})_i$ is the i^{th} column of the matrix square root of $\mathbf{P}_{xx}$. The probability weights W_i associated with the i^{th} sigma point $\boldsymbol{\mathcal{X}}^{(i)}(k|k)$ are defined as:

$$W_0 = \frac{\kappa}{n+\kappa}, \; W_i = \frac{1}{2(n+\kappa)}, \;\; i = 1, 2, ..., 2n. \tag{2.8}$$

Usually $\kappa + n = 3$ is chosen for Gaussian systems. The predicted mean of $\boldsymbol{\mathcal{X}}(k+1|k)$ is computed using

$$\boldsymbol{\mathcal{X}}^{(i)}(k+1|k) = \mathbf{f}(\boldsymbol{\mathcal{X}}^{(i)}(k|k)), \tag{2.9}$$

$$\hat{\boldsymbol{\mathcal{X}}}(k+1|k) = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{X}}^{(i)}(k+1|k), \tag{2.10}$$

where W_i are defined in (2.8). Covariance matrices $\mathbf{P}_{xy}(k+1|k)$ and $\mathbf{P}_{yy}(k+1|k)$ are calculated as

$$\begin{aligned}\mathbf{P}_{xy}(k+1|k) &= \sum_{i=0}^{2n} W_i(\boldsymbol{\mathcal{X}}^{(i)}(k+1|k) - \hat{\boldsymbol{\mathcal{X}}}(k+1|k))\mathbf{v}^{(i)}(k)^T,\\ \mathbf{P}_{yy}(k+1|k) &= \sum_{i=0}^{2n} W_i\mathbf{v}^{(i)}(k)\mathbf{v}^{(i)}(k)^T,\end{aligned}$$

where

$$\mathbf{v}^{(i)}(k) = \boldsymbol{\mathcal{Y}}^{(i)}(k+1) - \hat{\boldsymbol{\mathcal{Y}}}(k+1),$$

$$\boldsymbol{\mathcal{Y}}^{(i)}(k+1) = \mathbf{h}(\boldsymbol{\mathcal{X}}^{(i)}(k+1|k))$$

and

$$\hat{\boldsymbol{\mathcal{Y}}}(k+1) = \sum_{i=0}^{2n} W_i\boldsymbol{\mathcal{Y}}^{(i)}(k+1).$$

$\mathbf{P}_{xx}(k+1|k)$ is computed similarly, using (2.5). Once the true measurement $\boldsymbol{\mathcal{Y}}_{k+1}$ becomes available, the mean estimate can be updated in (2.10) as

$$\hat{\boldsymbol{\mathcal{X}}}(k+1|k+1) = \hat{\boldsymbol{\mathcal{X}}}(k+1|k) + \mathbf{K}(k+1)(\boldsymbol{\mathcal{Y}}_{k+1} - \hat{\boldsymbol{\mathcal{Y}}}(k+1)),$$

where $\mathbf{K}(k+1) = \mathbf{P}_{xy}(k+1|k)\mathbf{P}_{yy}^{-1}(k+1|k)$. More details on this algorithm can be found in Julier & *et al* [1995]. More modifications of this algorithm have been proposed in Julier [1998] and Julier [2002].

It can be seen that the methods are based on constructing covariance matrices from the samples of distribution with correct first two moments, and then using the closed-form formulae (2.4)-(2.5) for the state and covariance update. The number of samples used tends to be significantly smaller than in particle filters, introduced in the next section. Ensemble filters use sampling from a Gaussian distribution and match sample mean and sample covariance, while sigma point filters use a deterministic algorithm to generate a discrete distribution (support points as well as possibly unequal probability weights) matching the specified two moments. Several applications of the unscented Kalman filter in communication, tracking and navigation are discussed in Farina *et al* [2002] and Julier & Uhlmann [2004]. UKF has been successfully used as an alternative to EKF in many applications; see Crassidis & Markley [2003], Evensen [1994], Merwe *et al* [2000], Wu & Hu [2005] and references therein.

Besides being used as a stand-alone filtering algorithm, it has also been used to produce a proposal distribution for the particle filter, see Merwe *et al* [2000]. The relations between EKF and UKF have been recently summarized in Gustafsson & Hendelby [2012].

Unless skewed approach to filtering is used as in Julier [1998], traditional UKF only propagates the mean vector and the covariance matrix, but not higher order moments. These could provide a better idea of the shape of the distribution and its departure from Gaussianity. This issue is addressed in chapter 3, where a new method for generating sigma points is proposed that allows higher order moment matching in closed-form and without the use of optimization.

2.2.4 Particle filters

Another group of approximate solutions are particle filters, which can be effectively applied to general nonlinear, non-Gaussian problems. One way to estimate the posterior distribution $p(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{Y}}(k))$ of the unobserved state $\boldsymbol{\mathcal{X}}(k)$ is by using particles drawn from it. Often posterior density is not known or it might not be easy to sample from it. One may instead choose to draw samples $\boldsymbol{\mathcal{X}}^{(i)}(k)$ from a known, easy-to-sample, *proposal* distribution $q(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{Y}}(k))$. Please see Gordon & *et al* [1993] and Isard & Blake [1996] for more details. Given samples $\boldsymbol{\mathcal{X}}^{(i)}(k)$, drawn from $q(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{X}}(k-1), \boldsymbol{\mathcal{Y}}(k))$, choosing the corresponding probability weights $W_i(k)$ such that

$$W_i(k) \propto \frac{p(\boldsymbol{\mathcal{X}}^{(i)}(k)|\boldsymbol{\mathcal{Y}}(k))}{q(\boldsymbol{\mathcal{X}}^{(i)}(k)|\boldsymbol{\mathcal{Y}}(k))}, \tag{2.11}$$

ensures that $\lim_{M\to\infty} \sum_{i=1}^{M} W_i(k)\mathbf{h}(\boldsymbol{\mathcal{X}}^{(i)}(k)) = \mathbb{E}(\mathbf{h}(\boldsymbol{\mathcal{X}}(k)))$, holds for any measurable function h for which $\mathbb{E}(\mathbf{h}(\boldsymbol{\mathcal{X}}))$ exists, where $\mathbb{E}(\cdot)$ is expectation with respect to probability measure $p(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{Y}}(k))$. The discrete distribution in particle filter is represented by a set of random particles and associated probability weights. The particles and weights are updated recursively as new measurements become available. In order to derive the recursive expression for updating the probability weights the following factorization is assumed:

$$q(\boldsymbol{\mathcal{X}}(k+1)|\boldsymbol{\mathcal{Y}}(k+1)) = q(\boldsymbol{\mathcal{X}}(k+1)|\boldsymbol{\mathcal{X}}(k), \boldsymbol{\mathcal{Y}}(k+1))q(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{Y}}(k)). \tag{2.12}$$

Remembering that state dynamics is a Markov process and observations are conditionally independent given states

$$p(\boldsymbol{\mathcal{X}}(k+1)|\boldsymbol{\mathcal{Y}}(k+1)) \propto p(\boldsymbol{\mathcal{Y}}(k+1)|\boldsymbol{\mathcal{X}}(k+1))p(\boldsymbol{\mathcal{X}}(k+1)|\boldsymbol{\mathcal{X}}(k))p(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{Y}}(k)),$$

when substituted together with (2.12) into (2.11) provides the recursive estimates for the probability weights

$$W_i(k+1) = W_i(k)\frac{p(\boldsymbol{\mathcal{Y}}(k+1)|\boldsymbol{\mathcal{X}}^{(i)}(k+1))p(\boldsymbol{\mathcal{X}}^{(i)}(k+1)|\boldsymbol{\mathcal{X}}^{(i)}(k))}{q(\boldsymbol{\mathcal{X}}^{(i)}(k+1)|\boldsymbol{\mathcal{X}}^{(i)}(k), \boldsymbol{\mathcal{Y}}(k+1))}. \tag{2.13}$$

Here $p(\boldsymbol{\mathcal{Y}}(k+1)|\boldsymbol{\mathcal{X}}^{(i)}(k+1))$ and $p(\boldsymbol{\mathcal{X}}^{(i)}(k+1)|\boldsymbol{\mathcal{X}}^{(i)}(k))$ denote the observation density and the state transition density respectively. Given $\boldsymbol{\mathcal{X}}^{(i)}(k)$, $i = 1, 2, ..., M$, $\boldsymbol{\mathcal{X}}^{(i)}(k+1)$ and $W_i(k+1)$ are obtained as follows:

(a) Sample particles $\boldsymbol{\mathcal{X}}^{(i)}(k+1)$ from the proposal density $q(\boldsymbol{\mathcal{X}}(k+1)|\boldsymbol{\mathcal{X}}(k), \boldsymbol{\mathcal{Y}}(k+1))$.

(b) Compute the importance weights using (2.13).

(c) Normalize the importance weights to obtain the new probability weights

$$\widetilde{W}_i(k+1) = \frac{W_i(k+1)}{\sum_{i=1}^{M} W_i(k+1)}.$$

(d) Resample whenever a significant degeneracy is observed. Suitable criteria and algorithms for resampling could be found in Doucet *et al* [2001].

Various heuristics to improve performance of the particle filter (such as re-sampling) have been described in Doucet *et al* [2001] and more recently in Fu & Jia [2010].

Methods of this type have been used in a variety of areas including speech recognition, image processing, target tracking and financial modelling; see Doucet *et al.* [2001] for a review of applications while Arulampalam *et al.* [2002], Daum [2005], Kitagawa [1996] and Ristic *et al* [2004] provide comprehensive tutorials on various types of particle filters. One expects that the posterior density of the particle filter converges to the optimal conditional density as the number of particles becomes large. Crisan & Doucet [2002] bring together different asymptotic convergence results related to particle filters. In case the model contains a linear substructure, subject to Gaussian noise, marginalized particle filter (MPF) can be used. MPF is a combination of the standard particle filter and the Kalman filter. It has been shown that in some cases this filter provides better results than

the standard PF, see Doucet *et al* [2001], Gustafsson & *et al* [2002] and Karlsson *et al* [2005] for more details on complexity of MPF. Applications of particle filters in financial time series have been reported in Fearnhead [2005] and Pitt & Shepherd [1999], among others. These filters are difficult to calibrate due to the computation involved in computing the likelihood, since a closed-form expression is rarely available. Choosing the right proposal distribution is very important for a successful particle filter. Using the transition prior density $p(\boldsymbol{\mathcal{X}}^{(i)}(k+1)|\boldsymbol{\mathcal{X}}^{(i)}(k))$ as the proposal distribution offers easy implementation due to (2.13), but does not incorporate most recent observations. Usually, a more efficient choice is Gaussian posterior density generated by the extended Kalman filter for the same system, i.e. $q(\boldsymbol{\mathcal{X}}(k)|\boldsymbol{\mathcal{Y}}(k)) = N(\boldsymbol{\mathcal{X}}(k|k), \mathbf{P}_{xx}(k|k))$. Implementing posterior density obtained by unscented Kalman filter, described in 2.2.3, as the proposal distribution has been considered in Merwe *et al* [2000], which seems to outperform EKF-based proposal in terms of estimation accuracy. However, posterior densities obtained by either EKF or UKF only use the first two moments and not higher order moments for generating particles. This problem is further addressed in chapter 4, where a new moment-matching proposal is introduced.

2.2.5 Optimization based filters

Kalman filter and related methodologies are based on the premise that the optimal estimate is a conditional mean of the unobserved variable, given the measurements. Different heuristic methods for nonlinear filtering represent different ways of approximating this conditional mean. An entirely different approach is followed in Cortazar & Schwartz [2003]. To put the approach in Cortazar & Schwartz [2003] in a slightly general setting, consider a set of discretized equations

$$\begin{aligned} \boldsymbol{\mathcal{X}}(k+1) &= \mathbf{f}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{Q}(\boldsymbol{\mathcal{X}}(k))\mathbf{w}(k+1), \\ \boldsymbol{\mathcal{Y}}(k) &= \mathbf{h}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{R}\mathbf{v}(k), \end{aligned} \tag{2.14}$$

where $\mathbf{Q}$ is a deterministic matrix valued function and $\mathbf{R}$ is a deterministic matrix. Given $\boldsymbol{\mathcal{Y}}(k)$, $\boldsymbol{\mathcal{X}}(k)$ is found as a solution to the optimization problem of the following form:

$$\min_{\boldsymbol{\mathcal{X}}} J\left(\boldsymbol{\mathcal{Y}}(k) - \mathbf{h}(\boldsymbol{\mathcal{X}})\right), \tag{2.15}$$

where $J(\cdot)$ is a suitable non-negative cost function which is zero only at the origin. If $\hat{\boldsymbol{\mathcal{X}}}(k)$ is the argument minimising J in (2.15), the prediction $\hat{\boldsymbol{\mathcal{X}}}(k+1)$ is found using

$$\hat{\boldsymbol{\mathcal{X}}}(k+1) = \mathbf{f}(\hat{\boldsymbol{\mathcal{X}}}(k)).$$

This does away entirely with the need for knowing the statistical properties of the 'noise' terms $\mathbf{Q}(\boldsymbol{\mathcal{X}}(k))\mathbf{w}(k)$ and $\mathbf{R}\mathbf{v}(k)$ and can work well whenever the noise variances are not too large relative to the magnitude of the hidden states. A somewhat similar approach is followed in the *partially linearized sigma point filter* proposed in Date *et al.* [2010], where a set of linear programming problems are solved at each $t(k)$ to generate sigma points for a linearized measurement equation.

However, analysis of accuracy and convergence properties of optimization-based filters is somewhat more difficult and needs to be explored further. The unknown-but-bounded noise framework used in system identification may be useful for this purpose; see, e.g. Bravo *et al* [2006] and references therein.

2.2.6 Exact nonlinear filters

The Fokker-Planck equation is partial differential equation (PDE) that describes the evolution of the probability density of the state vector conditioned on the measurements. The main idea behind the exact nonlinear filters is to transform this PDE into a system of ordinary differential equations exactly, since these can be routinely solved in real time. For example, the Kalman filter is an exact filter for linear systems with Gaussian noise. Benes has derived an exact finite-dimensional filter for a special class of nonlinear problems in Benes [1981]. Unfortunately this particular filter didn't solve all the linear problems that Kalman filter solved exactly. Later Daum has extended Benes' theory to deal with a much wider class of nonlinear problems in Daum [1986a] and Daum [1988], covering all Kalman filter problems as well. More details about exact filters and the four progressively more general exact nonlinear filters derived since then can be found in Daum [2001] and Daum [1986b]. It is also worth noting that unlike Kalman filter that is based on Gaussian probability density, most general exact filters use the exponential family of multivariate probability densities. A method of approximating exact nonlinear filters with finite dimensional filters in case of exponential families is given in Brigo *et al.* [1998], where the differential geometric approach to statistics is utilized.

2.3 Time series calibration using maximum likelihood

Given (possibly vector-valued) measurements $\boldsymbol{\mathcal{Y}}(1), \boldsymbol{\mathcal{Y}}(2), \ldots, \boldsymbol{\mathcal{Y}}(N)$, one may use Kalman filter to calibrate a time series model with latent variables such as (2.2a)-(2.2b) as follows. Let $\mathcal{F}(k)$ denote all the measurements available until and including time $t(k)$. The probability density $p(\boldsymbol{\mathcal{Y}}(k+1)|\mathcal{F}(k))$ is Gaussian with

$$\mathbb{E}(\boldsymbol{\mathcal{Y}}(k+1)|\mathcal{F}(k)) = \mathbf{C}\hat{\boldsymbol{\mathcal{X}}}(k+1|k) + \mathbf{d} \quad \text{and}$$
$$Var(\boldsymbol{\mathcal{Y}}(k+1)|\mathcal{F}(k)) = \mathbf{S}(k+1).$$

The likelihood function (i.e., the joint probability function) can be written for the set of observations $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{\mathcal{Y}}(1), \boldsymbol{\mathcal{Y}}(2), \ldots, \boldsymbol{\mathcal{Y}}(N)\}$ as

$$L(\boldsymbol{\mathcal{Y}}) = p(\boldsymbol{\mathcal{Y}}(1)) \prod_{i=2}^{N} p(\boldsymbol{\mathcal{Y}}(i)|\mathcal{F}(i-1)).$$

It is usually simpler to work with the logarithm of likelihood, which is given by

$$\log L(\boldsymbol{\mathcal{Y}}) = \sum_{i=1}^{N} \log p(\boldsymbol{\mathcal{Y}}(i)|\mathcal{F}(i-1)) \; = \; -\frac{1}{2}\sum_{i=1}^{N} \left(\log|\mathbf{S}(i)| + \mathbf{z}(i)^{T}\mathbf{S}(i)^{-1}\mathbf{z}(i)\right),$$

when the constant terms are ignored. Given time series data $\boldsymbol{\mathcal{Y}}(1), \boldsymbol{\mathcal{Y}}(2), \ldots, \boldsymbol{\mathcal{Y}}(N)$, the quantities $\mathbf{z}(i)$ and $\mathbf{S}(i)$ are found through Kalman filter recursions outlined in the section 2.1.1. The above function can then be maximized to find the parameter vectors $\mathbf{b}, \mathbf{d}$ and matrices $\mathbf{A}, \mathbf{C}, \mathbf{Q}$ and $\mathbf{R}$ using an off-the-shelf nonlinear solver such as `fminsearch` in MATLAB. The initial state $\hat{\boldsymbol{\mathcal{X}}}(0|0)$ and its covariance $\mathbf{P}_{xx}(0|0)$ may be independently parameterized or it can be expressed in terms of other parameters. Harvey [1989] and Durbin & Koopman [2001] provide more details on maximum likelihood-based calibration, including parameter initialization issues in Kalman filtering framework. This method would be used for data calibration in numerical examples of chapters 3 and 4.

2.4 Summary

In this chapter the optimal recursive linear estimator for Gaussian systems, the Kalman filter, has been introduced together with the filtering algorithm. Unfortunately there are many nonlinear and

non-Gaussian models used in various industries. Also closed-form solution to latent state estimation problem is not available for nonlinear systems. Hence a number of suboptimal approximations have been proposed. Some of the approaches that address the nonlinear filtering problem and presented in this chapter are extended Kalman filter, unscented Kalman filter and particle filters. Extended Kalman filter approach has been popular in engineering for over three decades and works well if the system is approximately linear. However it can demonstrate flawed approximation and lead to filter divergence. Also derivation of Jacobian matrices can cause computational problems. Derivative free methods offer a more accurate alternative to EKF, with UKF being one of the proposed filters. Unscented transformation builds on the principle that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function, as in EKF. This approach employs a small set of deterministic sigma points and probability weights and is easy to implement. However UKF assumes conditional Gaussianity throughout filter recursions and can give misleading results if true density parts too far from assumed Gaussian density. Also this method propagates only the first two moments and this is the issue to be addressed in chapter 3. A different approach is particle filtering, where required conditional density of the state vector given measurements is also represented by a set of particles and associated probability weights. This set tends to be much larger than the one used in UKF. The success of PF depends on the proposal density used. The posterior density generated by EKF or UKF are considered to be an efficient choice. However these densities use only the information about the first two moments in generating the particles for PF. Propagating information about higher order moments might give a better representation of the posterior density and is the problem investigated in chapter 4. Finally, maximum likelihood method has been presented, which provides a powerful tool for data calibration.

Chapter 3

Higher order sigma point filter

3.1 Introduction

This chapter is concerned with the problem of the latent state estimation for nonlinear time series in discrete time. The material presented in this chapter has been published in Ponomareva *et al* [2010], Date & Ponomareva [2011] and submitted for publication in Ponomareva & Date [2012].
The following state space form for nonlinear time series is considered, first mentioned in the previous chapter:

$$\boldsymbol{\mathcal{X}}(k+1) = \mathbf{f}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{Q}\mathbf{w}(k+1),$$
$$\boldsymbol{\mathcal{Y}}(k) = \mathbf{h}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{R}\ \mathbf{v}(k).$$

In chapter 2, the algorithm for the unscented Kalman filter has been described. At time t_k, given the mean $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ and the covariance $\mathbf{P}_{xx}(k|k)$ are available for the system in equation (2.1), one can generate sigma points, $\boldsymbol{\mathcal{X}}^{(i)}$, and corresponding probability weights, W_i, using closed-form formulas in (2.7) and (2.8). The following result can then be verified by straightforward algebraic manipulation (see, e.g. Julier & Uhlmann [2004]):

Proposition 3.1

Sigma points and corresponding probability weights defined in (2.7) and (2.8) match the mean $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ and the covariance $\mathbf{P}_{xx}(k|k)$ exactly.

The UKF suffers from one major disadvantage, especially for systems with significant noise terms

in the transition equation (2.1). Even if the density of $\boldsymbol{\mathcal{X}}(k|k)$ at time $t(k)$ is Gaussian, a nonlinear function $\mathbf{f}$ will lead to a prediction $\boldsymbol{\mathcal{X}}(k+1|k)$ whose density is non-Gaussian in general. The unscented filter assumes conditional Gaussianity throughout the filter recursions and may lead to misleading results in case the density departs too far from the assumed Gaussian density.

Also, it is worth noting that while matching the first and the second moment accurately, traditional UKF does not propagate any information about higher order moments. This information may provide a better idea of the shape of the distribution and its departure from Gaussianity. More details on a modified algorithm for generating sigma points and probability weights that allows to match the first four moments of Gaussian distribution can be found in Julier & Uhlmann [1997] and Julier & Uhlmann [2004]. In Julier [2002] scaled unscented transformation preserves the first two moments of the sigma points, but allows third and higher order moments to be scaled by an arbitrary amount. Whereas Julier [1998] introduces a method for matching the first three moments of an arbitrary distribution. In this paper, expressions for sigma points and weights are given in closed form. However, there was very little improvement in the results of the new skewed filter with added third moment when compared to the traditional unscented transformation filter. One of the main reasons for that was the fact that the estimated skew was much smaller than the "true" skew and also that the linear state update does not incorporate any skew information. Other suggested algorithms, which try to match higher moments, either require optimization, as in Ledermann & Alexander [2011], or rely heavily on analytical solver, as in Tenne & Singh [2003]. In the next section a new modification of generation of sigma points and probability weights is proposed to partially address this issue.

3.2 A new algorithm for unscented Kalman filtering

3.2.1 Sigma point generation with higher order moment matching

In Date *et al.* [2008], a method was proposed to match the mean vector, the covariance matrix and the average marginal kurtosis of a multivariate distribution exactly, when the marginal densities are symmetric. In Ponomareva *et al* [2010], this algorithm was extended to asymmetric distributions, as outlined below. Additional parameters α and β are introduced in order to capture the 3^{rd} and the 4^{th} moments of $\boldsymbol{\mathcal{X}}(k+1|k+1)$ using augmented UKF. Here, an augmented method is considered, which

has been shown in Wu & Hu [2005] to give more accurate results compared to non-augmented UKF in the presence of significant noise terms. In this method a new state vector is used that incorporates the noise terms from measurement and transition equations,

$$\boldsymbol{\mathcal{X}}^a(k) = \begin{pmatrix} \boldsymbol{\mathcal{X}}(k) \\ \mathbf{w}(k) \\ \mathbf{v}(k) \end{pmatrix}.$$

The augmented functions $\mathbf{f}^a$ and $\mathbf{h}^a$ act as follows

$$\mathbf{f}^a(\boldsymbol{\mathcal{X}}^a(k)) = \mathbf{f}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{w}(k)$$

and

$$\mathbf{h}^a(\boldsymbol{\mathcal{X}}^a(k)) = \mathbf{h}(\boldsymbol{\mathcal{X}}(k)) + \mathbf{v}(k).$$

From here on it is assumed that the state vector and functions **f** and **h** are augmented and the index ^{a} will be dropped. Suppose that $\boldsymbol{\mathcal{X}}$ is a random n-vector with mean $\hat{\boldsymbol{\mathcal{X}}}$ and covariance $\mathbf{P}_{xx}$, and uncorrelated Gaussian noises **w** and **v**, with respective dimensions $m_1 \times 1$ and $m_2 \times 1$, as in equation (2.1). $n \neq m$ is allowed for generality, where $m = m_1 + m_2$. Matrix $\mathbf{P} > 0$ is such that $\mathbf{P}_{xx}$=$\mathbf{P}\mathbf{P}^T$, where $\mathbf{P}^T$ is the transpose of $\mathbf{P}$ and $\mathbf{P}_i$ is the i^{th} column of the matrix $\mathbf{P}$. The argument (k) will be dropped in the ensuing discussion in this section for notational simplicity. It is possible to create 2(n+m)+1 sigma points, $\boldsymbol{\mathcal{X}}^{(i)}$, and associated probability weights, W_i, for the augmented state vector as follows:

$$\boldsymbol{\mathcal{X}}^{(0)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} \\ \mathbf{0}_{m\times 1} \end{pmatrix} = \bar{\boldsymbol{\mathcal{X}}}, \quad W_0 = 1 - \sum_{i=1}^{2N} W_i,$$

$$\boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} + \alpha\sqrt{N}\mathbf{P}_i \\ \mathbf{0}_{m\times 1} \end{pmatrix}, W_i = \frac{1}{\alpha(\alpha+\beta)N},\ i = 1, 2, ..., n,$$

$$\boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} - \beta\sqrt{N}\mathbf{P}_i \\ \mathbf{0}_{m\times 1} \end{pmatrix}, W_i = \frac{1}{\beta(\alpha+\beta)N},\ i = n+1, ..., 2n,$$

$$\boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} \\ \sqrt{N}\mathbf{Q}_{i-2n} \end{pmatrix}, \quad W_i = \left(\frac{1}{2N}\right), \; i = 2n+1, ..., 2n+m_1,$$
$$\boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} \\ -\sqrt{N}\mathbf{Q}_{i-2n-m_1} \end{pmatrix}, \quad W_i = \left(\frac{1}{2N}\right), \; i = 2n+m_1+1, ..., 2n+2m_1,$$

$$\boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} \\ \sqrt{N}\mathbf{R}_{i-2n-2m_1} \end{pmatrix}, \quad W_i = \left(\frac{1}{2N}\right), \; i = 2n+2m_1+1, ..., 2n+2m_1+m_2,$$
$$\boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} \\ -\sqrt{N}\mathbf{R}_{i-2n-2m_1-m_2} \end{pmatrix}, \quad W_i = \left(\frac{1}{2N}\right), \; i = 2n+2m_1+m_2+1, ..., 2n+2m, \tag{3.1}$$

where $N = n + m$, the j^{th} element of a sigma point $\boldsymbol{\mathcal{X}}^{(i)}$ will be denoted as $\boldsymbol{\mathcal{X}}_j^{(i)}$. As there are only two degrees of freedom (*viz* α and β), it has been chosen to match the *average* third and fourth marginal moments of the state vector alone (i.e. ignoring the moments of noise terms). Note that it is possible to match average marginal moments of the augmented state vector, although this approach yields poor results when the state is non-Gaussian and the noise is Gaussian and hence is not followed here. Expressions for α and β are given next:

Definition 3.1

$$\alpha = \frac{1}{2}\phi_1 \pm \frac{1}{2}\sqrt{4\phi_2 - 3\phi_1^2},$$
$$\beta = -\frac{1}{2}\phi_1 \pm \frac{1}{2}\sqrt{4\phi_2 - 3\phi_1^2}, \tag{3.2}$$

where values of the same sign are taken,

$$\phi_1 = \frac{\sum_{j=1}^{n} \omega_j}{\sqrt{N}\sum_{l=1}^{n}\sum_{k=1}^{n} \mathbf{P}_{lk}^3}$$

and

$$\phi_2 = \frac{\sum_{j=1}^{n} \psi_j}{N\sum_{l=1}^{n}\sum_{k=1}^{n} \mathbf{P}_{lk}^4}.$$

Here $\omega_j = \sum_{i=0}^{2N} W_i(\boldsymbol{\mathcal{X}}_j^{(i)} - \bar{\boldsymbol{\mathcal{X}}}_j)^3$ and $\psi_j = \sum_{i=0}^{2N} W_i(\boldsymbol{\mathcal{X}}_j^{(i)} - \bar{\boldsymbol{\mathcal{X}}}_j)^4$ are the marginal 3^{rd} and 4^{th} central moments respectively. $\mathbf{P}_{ij}$ is entry in the i^{th} row and the j^{th} column of matrix $\mathbf{P}$, so that ϕ_1

and ϕ_2 are known from the data.

With the above choices of α, β, one can match the mean, covariance, the average central third and fourth marginal moments as it is stated in the next proposition:

Proposition 3.2

With $\boldsymbol{\mathcal{X}}^{(i)}, W_i, \alpha, \beta$ chosen as in (3.1)- (3.2), the following properties hold:

$$\sum_{i=0}^{2N} W_i = 1, \tag{3.3}$$

$$\sum_{i=0}^{2N} W_i \boldsymbol{\mathcal{X}}^{(i)} = \begin{pmatrix} \hat{\boldsymbol{\mathcal{X}}} \\ \mathbf{0}_{m_1 \times 1} \\ \mathbf{0}_{m_2 \times 1} \end{pmatrix}, \tag{3.4}$$

$$\sum_{i=0}^{2N} W_i (\boldsymbol{\mathcal{X}}^{(i)} - \bar{\boldsymbol{\mathcal{X}}})(\boldsymbol{\mathcal{X}}^{(i)} - \bar{\boldsymbol{\mathcal{X}}})^T = \begin{pmatrix} \mathbf{P}_{xx} & \mathbf{0}_{n \times m_1} & \mathbf{0}_{n \times m_2} \\ \mathbf{0}_{m_1 \times n} & \mathbf{Q}\mathbf{Q}^T & \mathbf{0}_{m_1 \times m_2} \\ \mathbf{0}_{m_2 \times n} & \mathbf{0}_{m_2 \times m_1} & \mathbf{R}\mathbf{R}^T \end{pmatrix}, \tag{3.5}$$

$$\frac{1}{n} \sum_{j=1}^{n} \sum_{i=0}^{2N} W_i (\boldsymbol{\mathcal{X}}_j^{(i)} - \bar{\boldsymbol{\mathcal{X}}}_j)^3 = \frac{1}{n} \sum_{j=1}^{n} \omega_j, \tag{3.6}$$

$$\frac{1}{n} \sum_{j=1}^{n} \sum_{i=0}^{2N} W_i (\boldsymbol{\mathcal{X}}_j^{(i)} - \bar{\boldsymbol{\mathcal{X}}}_j)^4 = \frac{1}{n} \sum_{j=1}^{n} \psi_j. \tag{3.7}$$

Proof: Equations (3.3)- (3.5) follow by straightforward algebraic manipulation. For the two last equations, substituting the expression for $\boldsymbol{\mathcal{X}}^{(i)}$ and W_i from (3.1) into the left-hand side of equations (3.6)- (3.7) one gets:

$$\begin{aligned} \alpha - \beta &= \frac{\sum_{j=1}^{n} \omega_j}{\sqrt{N} \sum_{l=1}^{n} \sum_{k=1}^{n} \mathbf{P}_{lk}^3}, \\ \alpha^2 - \alpha\beta + \beta^2 &= \frac{\sum_{j=1}^{n} \psi_j}{N \sum_{l=1}^{n} \sum_{k=1}^{n} \mathbf{P}_{lk}^4}. \end{aligned} \tag{3.8}$$

Using definitions of α and β from (3.2) in the equations (3.8) provides the required result. ∎

Note that $W_i \geq 0$ and $\sum_{i=0}^{2N} W_i = 1$ mean the set of probability weights and corresponding sigma points $\{W_i, \boldsymbol{\mathcal{X}}^{(i)}\}$ forms a valid probability distribution. This is *not* always the case in the unscented Kalman filter, since κ in (2.8) is not restricted to be positive. Provided $\phi_2 \geq \frac{3}{4}\phi_1^2$, (which is trivially

true for symmetric distributions), α and β allow to capture and propagate the marginal skewness and marginal kurtosis. Note also that the unscented filter presented in section 2 employs the same weights W_i for all sigma points $\boldsymbol{\mathcal{X}}^{(i)}$ for $i > 0$. In comparison, different expressions for probability weights depending on i are given in (3.1). A similar approach is taken in Tenne & Singh [2003]. However, the authors propose a method in Tenne & Singh [2003] where a set of nonlinear algebraic equations is solved to find support points and probability weights to match a given set of moments. Further, the probability weights found are held constant throughout the recursion and a closed form analytic solution is given only for the Gaussian case (in particular, with zero skewness). A different method that induces Pearson correlations among random vectors has been introduced in Ilich [2009]. This algorithm rearranges randomly generated variables with any statistical distribution functions, such that a desired product moment correlation matrix is achieved. More recently, a linear algebraic approach is taken in Ledermann & Alexander [2011], where the mean, covariance matrix, scalar multivariate skewness and scalar multivariate kurtosis are determined by a deterministic L-matrix. This matrix is then multiplied by arbitrary random orthogonal matrices to generate random samples with the same exact first four moments. However, in order to match the required skewness and kurtosis, numerical optimization must be applied to calibrate the parameters of the L-matrix, i.e. the solution is not available in the closed form. In contrast, new algorithm presented here provides matching of average higher order moments in roughly the same amount of numerical effort as the UKF with augmented states, and further allows to track changes in these higher order moments.

It is worth noting that while the aim is to improve the accuracy of the filter by matching higher moments, there may still be some uncertainty regarding the reliability and credibility of the approximations associated with using this method. In particular, Hochreiter& Pflug [2007] provides examples of four radically different distributions with the same first four moments. Nevertheless, matching moments is a simple and tractable way of generating approximate conditional distributions. It may also be seen as especially relevant in a filtering set-up, as the propagation of conditional moments (rather than the entire conditional distribution) is often of importance.

3.2.2 Filtering algorithm using higher order moments

The filtering algorithm for higher order sigma point filter (HOSPoF) can be described as follows.

1. Given $\boldsymbol{\mathcal{X}}^{(i)}(k|k)$, $\mathbf{P}_{xx}(k|k)$ and newly measured $\boldsymbol{\mathcal{Y}}_{k+1}$, propagate sigma points using (2.9) and update state estimates in the following way:

$$\begin{aligned}\boldsymbol{\mathcal{X}}^{(i)}(k+1|k+1) &= \boldsymbol{\mathcal{X}}^{(i)}(k+1|k) + \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}(\boldsymbol{\mathcal{Y}}_{k+1} - \mathbf{h}(\boldsymbol{\mathcal{X}}^{(i)}(k+1|k))),\\ \hat{\boldsymbol{\mathcal{X}}}(k+1|k+1) &= \sum_{i=0}^{2N} W_i \boldsymbol{\mathcal{X}}^{(i)}(k+1|k+1),\end{aligned}$$

   where $\boldsymbol{\mathcal{X}}^{(i)}(k+1|k) = \mathbf{f}(\boldsymbol{\mathcal{X}}^{(i)})$, W_i are the weights as defined in (3.1) and the covariance matrices $\mathbf{P}_{xy}(k+1|k)$ and $\mathbf{P}_{yy}(k+1|k)$ are calculated as described in section 2.

2. Calculating the average marginal skewness and the average marginal kurtosis of $\boldsymbol{\mathcal{X}}^{(i)}(k+1|k+1)$ provides with the updated values for α and β via (3.2). Now these values are used to generate a new set of sigma points and the corresponding weights at time $t(k+1)$.

The utility of this algorithm was illustrated with an example in Ponomareva *et al* [2010].

3.3 Numerical examples

In this section the performance of the new filter is tested on two examples.

3.3.1 Univariate non-stationary growth model

To test the accuracy of the new algorithm a univariate non-stationary growth model is considered, where this model is given by

$$\begin{aligned}\boldsymbol{\mathcal{X}}(k+1) &= a\boldsymbol{\mathcal{X}}(k) + b\frac{\boldsymbol{\mathcal{X}}(k)}{1+\boldsymbol{\mathcal{X}}^2(k)} + d\cos(1.2k) + \sigma_w \mathbf{w}(k+1),\\ \boldsymbol{\mathcal{Y}}(k) &= \boldsymbol{\mathcal{X}}^2(k)/20 + \sigma_v \mathbf{v}(k),\end{aligned}$$

where $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are i.i.d. $N(0,1)$ random variables. This model is very popular in econometrics and has been previously used in Date *et al.* [2008], Wu & Hu [2005], see also references there in. Parameters used in this numerical example are shown in Table 3.1.

Table 3.1. Parameter values

a	0.5
b	25
d	8
σ_w	0.1
σ_v	0.1

The performance of traditional unscented Kalman filter and HOSPoF are compared. In order to evaluate the efficiency of these filters the average of root mean squared error (AvRMSE) is evaluated in one step ahead prediction, where root mean squared error for sample path i is computed for F time-steps as

$$RMSE_{(i)} = \sqrt{\frac{1}{F}\sum_{k=1}^{F}(\boldsymbol{\mathcal{X}}^i(k) - \widehat{\boldsymbol{\mathcal{X}}}^i(k|k))^2}.$$

The average of sample mean of the relative absolute error (AvMRAE) is also considered, with MRAE defined as

$$MRAE_i = \frac{1}{F}\sum_{k=1}^{F}|\frac{\boldsymbol{\mathcal{X}}^i(k) - \hat{\boldsymbol{\mathcal{X}}}^i(k|k)}{\boldsymbol{\mathcal{X}}^i(k)}|.$$

This was computed over the relevant set of F observations. In this example $F = 100$ and average errors are based on 100 sample paths. Seven sigma points were used for traditional unscented Kalman filter and HOSPoF, proposed in this chapter. Tables 3.2-3.3 present both type of errors for both filters and extended Kalman filter.

Table 3.2. Average RMSE in 1 step ahead prediction

EKF	UKF	HOSPoF
3.97521	2.06909	1.75721

Table 3.3. Average MRAE of 1-step ahead prediction

EKF	UKF	HOSPoF
0.33848	0.20524	0.17793

It can be observed that HOSPoF outperforms both EKF and UKF for both types of errors, with average improvement over UKF of 14% for both types of error. This improvement has been achieved with very little extra computational effort.

3.3.2 Multi-factor CIR model

For the empirical, 'real-life' example, discretisation of a two-factor Cox-Ingersoll-Ross (CIR) model with a nonlinear measurement equation is considered. This is a multivariable extension of the model first proposed in Cox *et al.* [1985]; see Geyer & Pichler [1999] for more details on the use of this model in a filtering context. The state evolution is

$$\boldsymbol{\mathcal{X}}_j(k+1) = \kappa_j \epsilon_j \theta_j + (1 - \kappa_j \epsilon_j)\boldsymbol{\mathcal{X}}_j(k) + \mathbf{Q}_j(k+1)\mathbf{w}_j(k+1),$$

for $j = 1, 2$, where $\mathbf{w}_j(k)$ are zero mean, unit variance and uncorrelated Gaussian random variables. The standard deviation $\mathbf{Q}$ is given by

$$\mathbf{Q}_j(k+1) = \sigma_j \sqrt{\epsilon_j(\frac{1}{2}\theta_j \kappa_j \epsilon_j + (1 - \kappa_j \epsilon_j)\boldsymbol{\mathcal{X}}_j(k))},$$

where κ_j, σ_j and θ_j are constants and

$$\epsilon_j = \frac{(\,1 - e^{(-\kappa_j \Delta)})}{\kappa_j},$$

with $\Delta = t(k+1) - t(k)$. The observable variables are exponential in the latent states and are given by

$$\boldsymbol{\mathcal{Y}}_i(k) = \Pi_{j=1}^{2} \left(A_{i,j} \exp(-\sum_{j=1}^{2}(B_{i,j}\boldsymbol{\mathcal{X}}_j(k))) \right) + \mathbf{z}_i(k),$$

where

$$\begin{aligned} A_{i,j} &= \left(\frac{2\gamma_j \exp((\kappa_j + \gamma_j + \lambda_j)T_i/2)}{2\gamma_j + (\kappa_j + \lambda_j + \gamma_j)(exp(T_i\gamma_j) - 1)} \right)^{\frac{2\kappa_j\theta_j}{\sigma_j^2}}, \\ B_{i,j} &= \frac{2(exp(T_i\gamma_j) - 1)}{2\gamma_j + (\kappa_j + \lambda_j + \gamma_j)(exp(T_i\gamma_j) - 1)}, \\ \gamma_j &= \sqrt{(\kappa_j + \lambda_j)^2 + 2\sigma_j^2}. \end{aligned}$$

$\mathbf{z}_i(k)$ is observational noise with zero mean and is assumed to have a constant variance h^2 for each i and λ_i are constants. In practice, T_i represents the time to maturity and $\boldsymbol{\mathcal{Y}}_i(k)$ represents the price of

a zero coupon bond with a maturity $T_i + t(k)$, at time $t(k)$. Here, three maturities are used, $T_1 = 1$, $T_2 = 2$ and $T_3 = 4$.

For numerical experiments, weekly data from February 2001 to July 2005 for 3 different UK government bonds is used. Here 180 observations were used for calibration and 52 were used for out-of-sample validation. A 2-factor model was calibrated using the extended Kalman filter and the quasi-maximum likelihood method. In-built optimization routines from MATLAB were used for calibration. Table 3.4 presents the parameter values obtained as a result of calibration.

Table 3.4. Parameter values

θ_1	θ_2	σ_1	σ_2	κ_1	κ_2	λ_1	λ_2	h
0.0254	0.0175	0.0710	0.1870	0.0978	0.8035	-0.0350	-0.0490	0.001

After calibration, the sigma point generation method, described in the previous section, is used to generate sigma points at each $t(k)$, with initial values for mean θ_j and diagonal elements of covariance as $\frac{\theta_j \sigma_j^2}{2\kappa_j}$. Eleven sigma points are generated at each $t(k)$. Bearing in mind the nonnegativity restriction on state variables $\boldsymbol{\mathcal{X}}_j \geq 0$ any negative element of state estimate $\boldsymbol{\mathcal{X}}_j(k|k-1)$ are replaced with zero. This is a commonly used heuristic in nonlinear models with non-negative states, see, e.g. Geyer & Pichler [1999]. These points are then used to construct $\hat{\boldsymbol{\mathcal{X}}}_j(k|k)$, $j = 1, 2$ and $\mathbf{P}_{xx}(k|k)$, which are employed in order to obtain predictions of $\boldsymbol{\mathcal{Y}}_i(k)$, $i = 1, 2, 3$. MRSE and MRAE errors in this example are computed as

$$RMSE_{(i)} = \sqrt{\frac{1}{F}\sum_{j=1}^{F}(\boldsymbol{\mathcal{Y}}_i(j) - \widehat{\boldsymbol{\mathcal{Y}}}_i(j))^2},$$

$$MRAE_{(i)} = \frac{1}{F}\sum_{j=1}^{F}\frac{|\boldsymbol{\mathcal{Y}}_i(j) - \hat{\boldsymbol{\mathcal{Y}}}_{\boldsymbol{i}}(\boldsymbol{j})|}{\boldsymbol{\mathcal{Y}}_i(j)},$$

where the subscript i denotes the i^{th} time to maturity. Tables 3.5-3.8 list the errors computed for one step ahead prediction of yields for different types of filters for in-sample and out-of-sample data, where results are averaged over 100 runs. These filters are the EKF, traditional UKF and HOSPoF. It can be seen from these tables that HOSPoF outperforms both EKF and UKF for in-sample and out-of-sample data and for all times to maturity. In particular, the improvement for out-of-sample

predictions achieved with HOSPoF is over 10% for all the times to maturity, as compared to UKF. Prices of zero-coupon bonds are very important as they are used for calculations of the prices of interest rate derivatives. A 1% error in the price of the underlying bond may lead to a 25% error in an option price and this is justifiably unacceptable to many traders. Hence, it is essential to get the price as accurate as possible and HOSPoF enables user to obtain that extra degree of accuracy when compared to UKF. This improvement is obtained with very little extra computational effort, viz. computing the marginal 3^{rd} and 4^{th} moments and hence computing α, β using the closed-form expression in (3.2).

Table 3.5. Average RMSE for 1 step ahead in-sample prediction

τ_k	EKF	UKF	HOSPoF
$1Y$	0.00264	0.00114	0.00110
$2Y$	0.00432	0.00231	0.00224
$4Y$	0.00621	0.00375	0.00362

Table 3.6. Average MRAE for 1 step ahead in-sample prediction

τ_k	EKF	UKF	HOSPoF
$1Y$	0.00237	0.00079	0.00075
$2Y$	0.00388	0.00198	0.00190
$4Y$	0.00606	0.00365	0.00349

Table 3.7. Average RMSE for 1-step ahead out-of-sample prediction

τ_k	EKF	UKF	HOSPoF
$1Y$	0.007417	0.00087	0.00083
$2Y$	0.012736	0.00147	0.00140
$4Y$	0.022402	0.00283	0.00266

Table 3.8. Average MRAE for 1-step ahead out-of-sample prediction

τ_k	EKF	UKF	HOSPoF
$1Y$	0.004114	0.00075	0.00066
$2Y$	0.006832	0.00144	0.00125
$4Y$	0.011320	0.00300	0.00268

3.4 Theoretical accuracy of higher order moment matching filter

In this section, the theoretical accuracy of the sigma point generation algorithm used in HOSPoF is considered in the scalar case with smooth nonlinearities, in terms of estimating the conditional mean and the conditional variance.

3.4.1 Mean estimation

The univariate case is considered and as in section 3.2 one starts with a random variable $\boldsymbol{\mathcal{X}}$ with mean $\bar{\boldsymbol{\mathcal{X}}}$ and variance $\mathbf{P}_{xx}$. Suppose that the second random variable $\boldsymbol{\mathcal{Z}}$ is related to $\boldsymbol{\mathcal{X}}$ through the nonlinear transformation $\boldsymbol{\mathcal{Z}} = \mathbf{f}(\boldsymbol{\mathcal{X}})$. Then the expected value of $\boldsymbol{\mathcal{Z}}$ is:

$$\bar{\boldsymbol{\mathcal{Z}}} = \mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}) + \frac{1}{2!}\mathbf{f}''\mathbf{P}_{xx} + \frac{1}{3!}\mathbf{f}'''\mathbb{E}(\boldsymbol{\mathcal{X}} - \bar{\boldsymbol{\mathcal{X}}})^3 + \frac{1}{4!}\mathbf{f}''''\mathbb{E}(\boldsymbol{\mathcal{X}} - \bar{\boldsymbol{\mathcal{X}}})^4 + \cdots . \tag{3.9}$$

Now using the new sigma points, $\boldsymbol{\mathcal{X}}^{(i)}$, and the corresponding probability weights, W_i introduced in section 3.2, the accuracy in estimating the mean $\widehat{\boldsymbol{\mathcal{Z}}} = \sum_{i=0}^{2N} W_i \mathbf{f}(\boldsymbol{\mathcal{X}}^{(i)})$ is analyzed.

Proposition 3.3

If the function $\mathbf{f}$ is at least five times differentiable, then

$$\widehat{\boldsymbol{\mathcal{Z}}} = \mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}) + \frac{1}{2!}\mathbf{f}''\mathbf{P}_{xx} + \frac{1}{3!}\mathbf{f}'''\omega + \frac{1}{4!}\mathbf{f}''''\psi + \cdots , \tag{3.10}$$

where, as per section 3.2, ω and ψ are the marginal 3rd and 4th central moments respectively, the derivatives $\mathbf{f}''$ etc are computed at $\boldsymbol{\mathcal{X}} = \bar{\boldsymbol{\mathcal{X}}}$.

Proof: Expanding the expression for $\widehat{\boldsymbol{\mathcal{Z}}}$ for the minimum possible number of sigma points ($= 2N + 1 = 3$), one gets

$$\widehat{\boldsymbol{\mathcal{Z}}} = \sum_{i=0}^{2N} W_i \mathbf{f}(\boldsymbol{\mathcal{X}}^{(i)}) = W_0\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}) + W_1\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}} + \alpha\sqrt{N\mathbf{P}_{xx}}) + W_2\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}} - \beta\sqrt{N\mathbf{P}_{xx}}).$$

Further expanding $\mathbf{f}$ around the mean $\bar{\boldsymbol{\mathcal{X}}}$,

$$\begin{aligned}\widehat{\boldsymbol{\mathcal{Z}}} &= W_0\mathbf{f}(\bar{\mathcal{X}}) \\ &+ W_1\left(\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}) + \mathbf{f}'\alpha\sqrt{\mathbf{P}_{xx}} + \frac{1}{2!}\mathbf{f}''(\alpha\sqrt{\mathbf{P}_{xx}})^2 + \frac{1}{3!}\mathbf{f}'''(\alpha\sqrt{\mathbf{P}_{xx}})^3 + \frac{1}{4!}\mathbf{f}''''(\alpha\sqrt{\mathbf{P}_{xx}})^4 + \cdots\right) \\ &+ W_2\left(\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}) - \mathbf{f}'\beta\sqrt{\mathbf{P}_{xx}} + \frac{1}{2!}\mathbf{f}''(\beta\sqrt{\mathbf{P}_{xx}})^2 - \frac{1}{3!}\mathbf{f}'''(\beta\sqrt{\mathbf{P}_{xx}})^3 + \frac{1}{4!}\mathbf{f}''''(\beta\sqrt{\mathbf{P}_{xx}})^4 + \cdots\right).\end{aligned}$$

Collecting similar powers of $\mathbf{P}_{xx}$ together gives

$$\begin{aligned}\widehat{\boldsymbol{\mathcal{Z}}} &= \mathbf{f}'\sqrt{\mathbf{P}_{xx}}[W_1\alpha - W_2\beta] + \frac{1}{2!}\mathbf{f}''\mathbf{P}_{xx}[W_1\alpha^2 + W_2\beta^2] \\ &+ \frac{1}{3!}\mathbf{f}'''\sqrt{\mathbf{P}_{xx}{}^3}[W_1\alpha^3 - W_2\beta^3] + \frac{1}{4!}\mathbf{f}''''\mathbf{P}_{xx}{}^2[W_1\alpha^4 + W_2\beta^4] + \mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}),\end{aligned}$$

and using (3.1) simplifies $\widehat{\boldsymbol{\mathcal{Z}}}$ to

$$\widehat{\boldsymbol{\mathcal{Z}}} = \mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}) + \frac{1}{2!}\mathbf{f}''\mathbf{P}_{xx} + \frac{1}{3!}\mathbf{f}'''\sqrt{\mathbf{P}_{xx}{}^3}[\alpha - \beta] + \frac{1}{4!}\mathbf{f}''''\mathbf{P}_{xx}{}^2[\alpha^2 - \alpha\beta + \beta^2] + \ldots.$$

Expression (3.2) helps to eliminate α and β and obtain the required result. ∎

Comparing (3.9) with (3.10), it is concluded that the mean $\widehat{\boldsymbol{\mathcal{Z}}}$ is calculated accurate to the 4^{th} order using the new method for generating sigma points. This is a great improvement compared with the 2^{nd} order accuracy achieved in Julier & Uhlmann [2004].

This 4^{th} order accuracy, achieved in (3.10), is preserved in the univariate augmented case as well. This holds since with $n = 1$ and $m = 2$, there will be $2N + 1 = 7$ sigma points

$$\bar{\boldsymbol{\mathcal{X}}}, \bar{\boldsymbol{\mathcal{X}}} + \alpha\sqrt{N}\begin{pmatrix}\mathbf{P}\\0\\0\end{pmatrix}, \bar{\boldsymbol{\mathcal{X}}} - \beta\sqrt{N}\begin{pmatrix}\mathbf{P}\\0\\0\end{pmatrix}, \bar{\boldsymbol{\mathcal{X}}} \pm \sqrt{N}\begin{pmatrix}0\\\mathbf{Q}\\0\end{pmatrix}, \bar{\boldsymbol{\mathcal{X}}} \pm \sqrt{N}\begin{pmatrix}0\\0\\\mathbf{R}\end{pmatrix},$$

where $\bar{\boldsymbol{\mathcal{X}}} = \begin{pmatrix}\hat{\boldsymbol{\mathcal{X}}}\\0\\0\end{pmatrix}$. The corresponding probability weights are $\tilde{W}_0$, W_1, W_2, $\tilde{W}$, where $\tilde{W} = \frac{1}{2N}$ and $\tilde{W}_0 = 1 - W_1 - W_2 - 4W$. Then expanding $\sum_{i=0}^{2N} W_i\mathbf{f}(\boldsymbol{\mathcal{X}}^{(i)})$ and using the definition of an

augmented function **f** one obtains

$$\begin{aligned}\widehat{\boldsymbol{\mathcal{Z}}} &= \sum_{i=0}^{2N} W_i \mathbf{f}(\boldsymbol{\mathcal{X}}^{(i)}) \\ &= (\tilde{W}_0 + 4\tilde{W})\mathbf{f}(\hat{\boldsymbol{\mathcal{X}}}) + W_1\mathbf{f}(\hat{\boldsymbol{\mathcal{X}}} + \alpha\sqrt{N}P) + W_2\mathbf{f}(\hat{\boldsymbol{\mathcal{X}}} - \beta\sqrt{N}P) \\ &+ \tilde{W}(0 + \sqrt{N}\mathbf{Q}) + \tilde{W}(0 - \sqrt{N}\mathbf{Q}) \\ &= W_0\mathbf{f}(\hat{\boldsymbol{\mathcal{X}}}) + W_1\mathbf{f}(\hat{\boldsymbol{\mathcal{X}}} + \alpha\sqrt{N}P) + W_2\mathbf{f}(\hat{\boldsymbol{\mathcal{X}}} - \beta\sqrt{N}P),\end{aligned}$$

which is exactly the same as a starting point in the proof of *Proposition* 3.3.

3.4.2 Variance estimation

For accuracy in variance estimation for $\boldsymbol{\mathcal{Z}}$ first consider:

$$\mathbf{P}_{zz} = (\mathbf{f}')^2\mathbf{P}_{xx} - \frac{1}{4}(\mathbf{f}'')^2\mathbf{P}_{xx}{}^2 + \mathbf{f}'\mathbf{f}''\mathbb{E}(\boldsymbol{\mathcal{X}} - \bar{\boldsymbol{\mathcal{X}}})^3 + (\frac{1}{3}\mathbf{f}'\mathbf{f}''' + \frac{1}{4}\mathbf{f}''^2)\mathbb{E}(\boldsymbol{\mathcal{X}} - \bar{\boldsymbol{\mathcal{X}}})^4 + \dots . \quad (3.11)$$

Using the new sigma points and probability weights as in (3.1) one can write covariance $\mathbf{P}_{zz}$ as

Proposition 3.4

$$\mathbf{P}_{zz} = (\mathbf{f}')^2\mathbf{P}_{xx} - \frac{1}{4}(\mathbf{f}'')^2\mathbf{P}_{xx}^2 + \mathbf{f}'\mathbf{f}''\omega + (\frac{1}{4}(\mathbf{f}'')^2 + \frac{1}{3}\mathbf{f}'\mathbf{f}''')\psi + \dots . \quad (3.12)$$

Proof: Start with

$$\begin{aligned}\mathbf{P}_{zz} &= \sum_{i=0}^{2N} W_i(\boldsymbol{\mathcal{Z}}^{(i)} - \bar{\boldsymbol{\mathcal{Z}}})^2 \\ &= W_0(\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}}))^2 + W_1(\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}} + \alpha\sqrt{N\mathbf{P}_{xx}}))^2 + W_2(\mathbf{f}(\bar{\boldsymbol{\mathcal{X}}} - \beta\sqrt{N\mathbf{P}_{xx}}))^2 - \bar{\boldsymbol{\mathcal{Z}}}^2.\end{aligned}$$

Applying (3.10) for $\bar{\boldsymbol{\mathcal{Z}}}$ and after simplifying one obtains

$$\begin{aligned}\mathbf{P}_{zz} &= -\frac{1}{4}(\mathbf{f}'')^2\mathbf{P}_{xx}{}^2 + (\mathbf{f}')^2\mathbf{P}_{xx}[W_1\alpha^2 + W_2\beta^2] + \mathbf{f}'\mathbf{f}''\sqrt{\mathbf{P}_{xx}{}^3}[W_1\alpha^3 - W_2\beta^3] \\ &- \mathbf{f}'\mathbf{f}''\sqrt{\mathbf{P}_{xx}{}^3}[W_1\alpha - W_2\beta] + (\frac{1}{4}(\mathbf{f}'')^2 + \frac{1}{3}\mathbf{f}'\mathbf{f}''')\mathbf{P}_{xx}^2[W_1\alpha^4 + W_2\beta^4] + \dots .\end{aligned}$$

Getting rid off α and β through (3.2) will yield the required result. ∎

Comparing (3.12) with (3.11) it is concluded that the new sigma point filter estimates the variance correct up to the 4^{th} order for the univariate case. This is an improvement not only over accuracy to the 2^{nd} order of the traditional unscented transformation method in Julier & Uhlmann [2004], but also over the 2^{nd} order accuracy in the scaled unscented transformation in Julier [2002].

3.5 Summary

It is believed that propagating information about the higher order moments might give a better idea about the shape of the distribution and its departure from Gaussianity. This problem arises since the nonlinear function in the transition equation will often lead to a conditional density that is not Gaussian in general. Traditional UKF or scaled UKF only match the mean vector and the covariance matrix, while skewed approach matches the third moment, skew, as well. However, current methods that match higher order moments require optimization or heavily rely on the analytical solver. This questions suitability of these methods for online applications. In this chapter, a new filtering algorithm has been proposed, in which the sigma points and the corresponding probability weights are modified at each step to match exactly the predicted values of the average marginal skewness and the average marginal kurtosis, besides matching the mean vector and covariance matrix. The expressions for the new sigma points and the corresponding probability weights are given in closed form and do not require optimization. Since there are only two degrees of freedom, i.e. α and β, the average marginal skewness and the average marginal kurtosis of the state vector are chosen to be matched. It is understood, that this higher order moment matching might not add value when averages are taken over a very large state dimension. However, the class of applications where the dimension is five or less is still very large; in fact, it is unusual to find time series models with more than four latent states in econometrics and finance.

It has been shown that in the univariate case the HOSPoF predicts mean and variance up to fourth order of accuracy, which is a significant improvement on a similar result for the traditional UKF. Two numerical examples, one simulation and one based on the real financial market data illustrate the utility of the proposed algorithm. At the very least, HOSPoF is a worthy competitor to UKF and EKF as a filter of choice in many practical applications involving nonlinear filtering.

The dependence of the number of sigma points on the state dimension of the system in the HOSPoF has been mentioned previously. Also, introducing more randomness in sigma point generation might provide more flexibility in posterior density representation. This idea is explored further in the next chapter.

Chapter 4

A new method for generating random sigma points

4.1 Introduction

Chapter 3 dealt exclusively with a deterministic, approximate filter for the nonlinear system. Such filters are numerically inexpensive; however, re-constructing a conditional distribution using a particle filter can improve the accuracy significantly at the expense of increased computations. In this chapter, a modified version of HOSPoF, outlined in the previous section, is used as a proposal distribution for a particle type filter and will compare it with UKF and prior distribution as proposals. It is also of interest to use the information about higher order moments and not just the mean and the covariance (as in UKF or EKF-based proposals) for generating particles for the PF. Continuing with W_i and $\boldsymbol{\mathcal{X}}^i(k+1|k+1)$, computed in HOSPoF, will limit the user to a proposal with $2n+1$ *deterministic* points or particles. Since PF type filters work best for a large number of randomly generated particles, the aim is to find a non-Gaussian proposal which matches the given mean, covariance matrix, average marginal skewness and average marginal kurtosis. Further, the aim is to obtain the particles and the probability weights which achieve this moment matching in closed form, *e.g.* without needing an analytic solver or optimizer. A method to achieve this is described next and is a further modification of HOSPoF outlined in the previous chapter. The material presented in this

chapter forms the basis of Ponomareva & Date [2012].

4.2 A moment-matching proposal distribution

Start by using the results from Date *et al.* [2008], where the authors defined a multivariate discrete distribution over $2ns + 1$ points and associated probability weights as follows:

Definition 4.1

$$\boldsymbol{\mathcal{X}}^{(k)} = \hat{\boldsymbol{\mathcal{X}}} \pm \frac{1}{\sqrt{2sp_i}}\mathbf{L}_j,\ W_k = p_i, \tag{4.1}$$

$$\boldsymbol{\mathcal{X}}^{(0)} = \hat{\boldsymbol{\mathcal{X}}},\ W_0 = p_{s+1}, \tag{4.2}$$

where $i = 1, 2, ..., s$, $j = 1, 2, ..., n$, $k = 1, 2, ..., 2ns$, n is the dimension of the state vector $\boldsymbol{\mathcal{X}}$ as before, $2n\sum_{i=1}^{s} p_i + p_{s+1} = 1$ and $\mathbf{L}_j$ denotes the j^{th} column of a matrix $\mathbf{L}$.

By choosing the value of the parameter s the size of the set of probability weights is determined, p_i. If $\mathbf{L}$ is a symmetric positive definite matrix such that $\mathbf{L}\mathbf{L}^T = \mathbf{P}_{xx}$, then sigma point filter, based on (4.1)- (4.2), matches the mean vector, the covariance matrix and zero third marginal moment exactly, see Date *et al.* [2008] for the proofs. Extending this method to non-symmetric distributions, (4.2) is changed to introduce 2 new sigma points that would carry the information about average marginal third and fourth moments:

Definition 4.2

$$\begin{aligned}
\boldsymbol{\mathcal{X}}^{(k)} &= \hat{\boldsymbol{\mathcal{X}}} \pm \frac{1}{\sqrt{2sp_i}}\mathbf{L}_j,\ W_k = p_i,\\
\boldsymbol{\mathcal{X}}^{(0)} &= \hat{\boldsymbol{\mathcal{X}}}, \qquad\qquad W_0 = p_{s+1}\widetilde{W}_0,\\
\boldsymbol{\mathcal{X}}^{(2ns+1)} &= \hat{\boldsymbol{\mathcal{X}}} + \frac{\widetilde{\alpha}}{\sqrt{p_{s+1}}}\mathbf{Z},\ W_{2ns+1} = p_{s+1}\widetilde{W}_1,\\
\boldsymbol{\mathcal{X}}^{(2ns+2)} &= \hat{\boldsymbol{\mathcal{X}}} - \frac{\widetilde{\beta}}{\sqrt{p_{s+1}}}\mathbf{Z},\ W_{2ns+2} = p_{s+1}\widetilde{W}_2.
\end{aligned} \tag{4.3}$$

Here i, j, k are as in definition 4.1 and p_i satisfy $2n\sum_{i=1}^{s} p_i + p_{s+1} = 1$, $\mathbf{Z}$ is an arbitrary non-zero deterministic vector. However $\mathbf{Z}$ can also be randomly generated as long as it is non-zero and $\mathbf{P}_{xx} - \mathbf{Z}\mathbf{Z}^T > 0$. The matrix $\mathbf{L}$ is such that $\mathbf{L}\mathbf{L}^T + \mathbf{Z}\mathbf{Z}^T = \mathbf{P}_{xx}$ and $\widetilde{W}_i$ satisfy $\sum_{i=0}^{2} \widetilde{W}_i = 1$ and

are given by

$$\widetilde{W}_1 = \frac{1}{\widetilde{\alpha}(\widetilde{\alpha}+\widetilde{\beta})}, \widetilde{W}_2 = \frac{1}{\widetilde{\beta}(\widetilde{\alpha}+\widetilde{\beta})}, \widetilde{W}_0 = 1 - \frac{1}{\widetilde{\alpha}\widetilde{\beta}},$$

where the parameterization of $\widetilde{W}_i$ follows the structure used in (3.1).

Aim is to match the average marginal third and fourth moments, $\frac{1}{n}\sum_{j=1}^{n}\widetilde{\omega}_j$ and $\frac{1}{n}\sum_{j=1}^{n}\widetilde{\psi}_j$ respectively, which are defined in the same way as chapter 3. The closed-form expression for $\widetilde{\alpha}$ and $\widetilde{\beta}$ are:

Definition 4.3

$$\begin{aligned}\widetilde{\alpha} &= \frac{1}{2}\widetilde{\phi}_1 \pm \frac{1}{2}\sqrt{4\widetilde{\phi}_2 - 3\widetilde{\phi}_1^2},\\ \widetilde{\beta} &= -\frac{1}{2}\widetilde{\phi}_1 \pm \frac{1}{2}\sqrt{4\widetilde{\phi}_2 - 3\widetilde{\phi}_1^2},\end{aligned} \tag{4.4}$$

where the values of the same sign are taken and

$$\widetilde{\phi}_1 = \frac{\sum_{j=1}^{n}\widetilde{\omega}_j\sqrt{p_{s+1}}}{\sum_{k=1}^{n}\mathbf{Z}_k^3},$$

$$\widetilde{\phi}_2 = p_{s+1}\frac{\sum_{j=1}^{n}\widetilde{\psi}_j - \frac{1}{2s^2}\sum_{l=1}^{n}\sum_{k=1}^{n}\mathbf{L}_{lk}^4(\sum_{i=1}^{s}\frac{1}{p_i})}{\sum_{k=1}^{n}\mathbf{Z}_k^4}.$$

$2ns+3$ randomly generated sigma points and corresponding probability weights defined in (4.3) match exactly the given mean, the covariance matrix, the average marginal third and fourth moments, *irrespective* of the precise choice of p_i, as is stated in the next proposition.

Proposition 4.1

With $\boldsymbol{\mathcal{X}}^{(i)}, W_i, \widetilde{\alpha}, \widetilde{\beta}$ chosen as in (4.3)- (4.4), the following properties hold:

$$\sum_{i=0}^{2ns+2} W_i = 1, \tag{4.5}$$

$$\sum_{i=0}^{2ns+2} W_i\boldsymbol{\mathcal{X}}^{(i)} = \hat{\boldsymbol{\mathcal{X}}}, \tag{4.6}$$

$$\sum_{i=0}^{2ns+2} W_i(\boldsymbol{\mathcal{X}}^{(i)} - \hat{\boldsymbol{\mathcal{X}}})(\boldsymbol{\mathcal{X}}^{(i)} - \hat{\boldsymbol{\mathcal{X}}})^T = \mathbf{P}_{xx}, \tag{4.7}$$

$$\frac{1}{n}\sum_{j=1}^{n}\sum_{i=0}^{2ns+2} W_i(\boldsymbol{\mathcal{X}}_j^{(i)} - \hat{\boldsymbol{\mathcal{X}}}_j)^3 = \frac{1}{n}\sum_{j=1}^{n}\widetilde{\omega}_j, \tag{4.8}$$

$$\frac{1}{n}\sum_{j=1}^{n}\sum_{i=0}^{2ns+2} W_i(\boldsymbol{\mathcal{X}}_j^{(i)} - \hat{\boldsymbol{\mathcal{X}}}_j)^4 = \frac{1}{n}\sum_{j=1}^{n}\widetilde{\psi}_j, \tag{4.9}$$

Proof: Equations (4.5)- (4.7) can be verified by straightforward algebraic manipulation. Substituting expressions for $\boldsymbol{\mathcal{X}}^{(i)}$ and W_i from (4.3) into the left-hand side of equations (4.8)-(4.9)

$$\widetilde{\alpha} - \widetilde{\beta} = \frac{\sum_{j=1}^{n} \widetilde{\omega}_j \sqrt{p_{s+1}}}{\sum_{k=1}^{n} \mathbf{Z}_k^3},$$
$$\widetilde{\alpha}^2 - \widetilde{\alpha}\widetilde{\beta} + \widetilde{\beta}^2 = p_{s+1} \frac{\sum_{j=1}^{n} \widetilde{\psi}_j - \frac{1}{2s^2} \sum_{l=1}^{n} \sum_{k=1}^{n} \mathbf{L}_{lk}^4 (\sum_{i=1}^{s} \frac{1}{p_i})}{\sum_{k=1}^{n} \mathbf{Z}_k^4}.$$

By using the definitions for $\widetilde{\alpha}$ and $\widetilde{\beta}$ from (4.4), the required result is obtained. ∎

By imposing a constraint $\widetilde{\phi_2} \geq \frac{3}{4}\widetilde{\phi_1}^2$, an upper limit for $\sum_{i=1}^{s} \frac{1}{p_i}$ is obtained, namely ζ. The following proposition gives an expression for ζ.

Proposition 4.2

$$\zeta = 2s^2 \frac{\sum_{j=1}^{n} \widetilde{\psi}_j - \frac{3}{4} \sum_{k=1}^{n} \mathbf{Z}_k^4 (\frac{\sum_{j=1}^{n} \widetilde{\omega}_j}{\sum_{k=1}^{n} \mathbf{Z}_k^3})^2}{\sum_{l=1}^{n} \sum_{k=1}^{n} \mathbf{L}_{lk}^4} \tag{4.10}$$

Proof: The proof is easy to verify by a straightforward algebraic manipulation. ∎

4.3 Algorithm for generating sigma points

4.3.1 Basic algorithm

Once ζ is determined using (4.10), the algorithm for generating $2ns + 3$ sigma points, matching the given mean, the covariance matrix and the average marginal third and fourth moments exactly, is provided below.

1. Generate $p_i > 0$ for i=1,2,..., s such that $\sum_{i=1}^{s} \frac{1}{p_i} < \zeta$, $p_{s+1} = 1 - 2n \sum_{i=1}^{s} p_i$. Note that p_i can be generated using a random number generator, and s is an arbitrary positive integer. Hence the number of particles is independent of the state dimension. Note also that the distribution of p_i is of no consequence since the moment-matching is independent of how p_i are generated.

2. Calculate $\widetilde{\phi_1}$ and $\widetilde{\phi_2}$ for an arbitrary vector $\mathbf{Z}$ using definition 4.2.

3. Closed form solution for $\widetilde{\alpha}$ and $\widetilde{\beta}$ are given by (4.4).

4. Construct the $2ns + 3$ support points using (4.3).

It is worth outlining the differences between the traditional particle filter and the approach proposed here. First, while re-sampling is possible after moment matching scenario generation, the moments will no longer be matched after re-sampling. This limitation is not specific to the method and is shared by all particle filters which use moment matching to generate a proposal. The defects of several resampling algorithms are discussed in Fu & Jia [2010]. Secondly, the generated samples are not i.i.d but are "almost independent", with some dependence introduced through the constraints on p_i. For a large s, this dependence is minimal, and it may be preferable than generating a fixed number of points using a deterministic generator, as in the case of the UKF. This intuition is supported by the numerical experiments in the next section. This new method of using moment-matching proposal is denoted as PF-HOSPoF.

4.3.2 Stratified sampling

One simple way of ensuring that p_i form a valid measure is to choose p_i between 0 and $\frac{1}{2n}$. The problem that can arise using this method is that all p_i values are too small. This means that the scaling factors $\frac{1}{\sqrt{p_i}}$, used in the calculation of sigma points, become very large. As a result, samples can be non-uniformly spread, gathered in clusters far away from the mean value. An alternative around this is to use stratified sampling. For example, it is possible to generate $\frac{s}{3}$ values of p_i randomly between I_1 and I_2, I_2 and I_3, I_3 and I_4, where $0 < I_1 < I_2 < I_3 < I_4 < 1$. Using the constraints on p_i the appropriate interval bounds I_j can be selected for $j = 1, ..., 4$. Since for total $2ns$ random particles it is required that $I_4 \leq \frac{1}{2ns}$ and also $I_4 > I_3 > I_2$ holds, one can pick $I_4 = \frac{1}{2ns}$.
The probability weights p_i have to satisfy the following two inequalities:

$$0 < \sum_{i=1}^{s} p_i < \frac{1}{2n}, \tag{4.11}$$

$$0 < \sum_{i=1}^{s} \frac{1}{p_i} < \zeta. \tag{4.12}$$

If $\zeta > \frac{4}{3}ns^2$, then the suggested algorithm for stratified sampling is as follows:

1. Set $I_4 = \frac{1}{2ns}$.

2. Choose x and y such that $y > x > 2$ and $x + y < \frac{3\zeta}{ns^2}$. Set $I_3 = \frac{1}{xns}$ and $I_2 = \frac{1}{yns}$.

3. Choose I_1 such that $0 < \frac{1}{I_1} < \frac{1}{\frac{3\zeta}{s} - (x+y)ns}$.

4. Generate $\frac{s}{3}$ values of p_i randomly between I_1 and I_2, I_2 and I_3, I_3 and I_4 respectively.

Proposition 4.2

The probabilities p_i, generated using the above algorithm, would satisfy the required inequalities (4.11) and (4.12).

Proof: Rewriting (4.11) and (4.12) in terms of intervals I_j, the following inequalities that I_j have to satisfy can be obtained:

$$I_4 + I_3 + I_2 < \frac{3}{2ns}, \tag{4.13}$$

$$\frac{1}{I_3} + \frac{1}{I_2} + \frac{1}{I_1} < \frac{3\zeta}{s}. \tag{4.14}$$

Substituting expression for I_4, $I_3 = \frac{1}{xns}$ and $I_2 = \frac{1}{yns}$, where $y > x > 2$, into (4.13) and (4.14):

$$\frac{x+y}{xyns} < \frac{1}{ns},$$

$$\frac{1}{I_1} < \frac{3\zeta}{s} - (x+y)ns.$$

The first inequality holds for $y > x > 2$, that is already satisfied by the choice of I_3 and I_2. In the second inequality it is required that $I_1 > 0$, so $4 < x + y < \frac{3\zeta}{ns^2}$ must hold. Solutions x and y, that satisfy inequalities above, exist when $\frac{3\zeta}{ns^2} > 4$, i.e. $\zeta > \frac{4}{3}ns^2$. ∎

4.4 Numerical examples

It is of interest to test the performance of the new filter PF-HOSPoF on one numerical and one financial market data examples.

4.4.1 CEV type time series model

This numerical example has the following state and measurement equations:

$$\begin{aligned}\boldsymbol{\mathcal{X}}(k+1) &= a\boldsymbol{\mathcal{X}}(k) + \mathbf{b} + \sigma_w\{(\boldsymbol{\mathcal{X}}(k))^2\}^{\gamma/2}\mathbf{w}(k+1),\\ \boldsymbol{\mathcal{Y}}(k) &= c\boldsymbol{\mathcal{X}}(k) + \mathbf{d} + \sigma_v\mathbf{v}(k),\end{aligned}$$

where noise terms $w(k)$ and $v(k)$ are zero mean, unit variance and uncorrelated Gaussian random variables. $a, b, c, \sigma_w > 0$ and $\sigma_v > 0$ are constants. This type of time series includes the constant elasticity of variance (CEV) model described in Cox [1996] and exponential affine term structure models including the Cox, Ingersoll and Ross model Cox *et al.* [1985]. Table 4.1 provides the parameter values for the model. Three different values for γ have been used: 0.125, 0.25 and 0.375.

Table 4.1. Parameter values

a	0.9
b	0.1
c	1.0
d	0.1
σ_w	0.01
σ_v	0.01

In order to evaluate the efficiency of these filters the average of the root mean squared error (AvRMSE) is calculated in one step ahead prediction, where the root mean squared error for sample path i is computed for F time-steps as

$$RMSE_{(i)} = \sqrt{\frac{1}{F}\sum_{k=1}^{F}(\boldsymbol{\mathcal{X}}^{(i)}(k) - \widehat{\boldsymbol{\mathcal{X}}}^{(i)}(k|k))^2}.$$

The average of the sample mean of the relative absolute error (AvMRAE) is also considered, with MRAE defined as

$$MRAE_i = \frac{1}{F}\sum_{k=1}^{F}\left|\frac{\boldsymbol{\mathcal{X}}^{(i)}(k) - \hat{\boldsymbol{\mathcal{X}}}^{(i)}(k|k)}{\boldsymbol{\mathcal{X}}^{(i)}(k)}\right|.$$

This was computed over the relevant set of F observations. In this example $F = 100$ and the average errors are based on 100 sample paths. 1000 particles were used for the particle filter in all cases.

Tables 4.2-4.3 summarize results for these filters with four different choices of proposal densities, a particle filter with approximate transition density as proposal (PF-T), a particle filter with Gaussian proposal generated by the unscented Kalman filter (PF-UKF), a particle filter with proposal generated by the extended Kalman filter (PF-EKF) and finally a particle filter with a possibly non-Gaussian proposal using the new moment matching heuristic proposed here (PF-HOSPoF).

Table 4.2. Average root mean squared error in 1 step ahead prediction

γ	PF-T	PF-EKF	PF-UKF	PF-HOSPoF
0.125	0.022173	0.021121	0.010929	0.009559
0.250	0.022345	0.021273	0.011220	0.009635
0.375	0.022781	0.022055	0.011267	0.009733

Table 4.3. Average relative absolute error of 1-step ahead prediction

γ	PF-T	PF-EKF	PF-UKF	PF-HOSPoF
0.125	0.000701	0.000609	0.000214	0.000149
0.250	0.000712	0.000616	0.000218	0.000171
0.375	0.000739	0.000632	0.000239	0.000172

It can be observed from the tables that the PF-HOSPoF produces the smallest errors for three different values of γ when compared to the PF-UKF, PF-EKF and PF-T. It is also worth noting, that only 103 sigma points were used in PF-HOSPoF compared to three other particle filters. Due to unequal probability weights employed in PF-HOSPoF, it is possible to approximate the 'shape' of the density function by using a smaller number of points.The results are also illustrated in figures 4.1-4.2 for easy visual comparison of all four methods used in this simulation example.

4.4.2 Multi-factor CIR model

For the example with empirical data the multi-factor CIR model is implemented with data and parameters used in the previous chapter. Tables 4.4-4.7 list the errors computed for one step ahead prediction of zero coupon bond prices for different types of filters, where results are average over 100 runs. These filters are: a particle filter with approximate transition density as proposal (PF-T), a

Figure 4.1: AvRMSE for CEV model

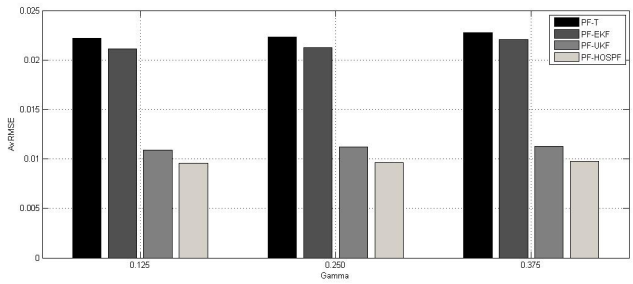
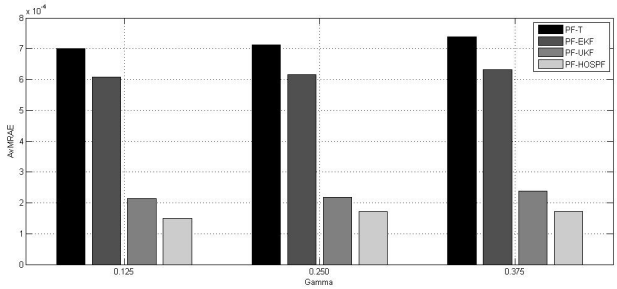


Figure 4.2: AvMRAE for CEV model



particle filter with Gaussian proposal generated by the unscented Kalman filter (PF-UKF), a particle filter with proposal generated by the extended Kalman filter (PF-EKF) and finally a particle filter with a possibly non-Gaussian proposal using the new moment matching heuristic proposed here (PF-HOSPoF). Note that, for PF-T, the mean and the variance of the transition density is known for this model structure; please see Cox *et al.* [1985] for an exact expression of density in terms of model parameters in the scalar case. Table 4.8 includes a summary of average (over 100 runs) computational times taken for one run of PF-UKF and PF-HOSPoF, and the average improvement for both types of errors achieved by PF-HOSPoF over PF-UKF.

Table 4.4. Average RMSE in 1 step ahead in-sample prediction

τ_k	PF-T	PF-EKF	PF-UKF	PF-HOSPoF
$1Y$	0.013416	0.001603	0.000179	0.000139
$2Y$	0.019821	0.001946	0.000238	0.000193
$4Y$	0.025710	0.002382	0.000271	0.000223

Table 4.5. Average MRAE of 1-step ahead in-sample prediction

τ_k	PF-T	PF-EKF	PF-UKF	PF-HOSPoF
$1Y$	0.012056	0.001449	0.000150	0.000117
$2Y$	0.018275	0.001629	0.000204	0.000159
$4Y$	0.025667	0.002277	0.000259	0.000208

Table 4.6. Average RMSE in 1 step ahead out-of-sample prediction

τ_k	PF-T	PF-EKF	PF-UKF	PF-HOSPoF
$1Y$	0.006950	0.000902	0.000162	0.000126
$2Y$	0.010574	0.001218	0.000220	0.000142
$4Y$	0.016215	0.002224	0.000257	0.000209

Table 4.7. Average MRAE of 1-step ahead out-of-sample prediction

τ_k	PF-T	PF-EKF	PF-UKF	PF-HOSPoF
$1Y$	0.006789	0.000863	0.000141	0.000105
$2Y$	0.010422	0.001158	0.000191	0.000126
$4Y$	0.017335	0.002237	0.000244	0.000195

Table 4.8. Comparison of accuracy and computational time for PF-UKF and PF-HOSPoF

Data Type	Time taken on average for one run, s		Time improvement of PF-HOSPoF over PF-UKF	Accuracy improvement of PF-HOSPoF over PF-UKF	
	PF-UKF	PF-HOSPoF		AvRMSE	AvMRAE
In-sample	101.0	53.7	46.8%	19.7%	21.2%
Out-of-sample	29.1	5.7	80.4%	25.5%	26.5%

From the tables, it can be seen that the PF-HOSPoF outperforms the PF-T, PF-EKF and PF-UKF for all the three times to maturity for out-of-sample data. The average improvement achieved with PF-HOSPoF is about 23% for all the maturities, as compared to PF-UKF. This improvement is obtained with very little extra computational effort, viz. computing the marginal 3^{rd} and 4^{th} moments and hence computing α, β using the closed-form expression in (4.4). The average computational times for PF-HOSPoF are also significantly smaller, than those required for PF-UKF. This is due to only

203 sigma points being used in PF-HOSPoF and 1000 particles in PF-UKF. Using unequal random probability weights allows better capturing of the shape of the distribution and offers computational advantage. Compared to numerical results for HOSPoF and UKF for the same model from chapter 3, the results in this chapter demonstrate a much better improvement of PF-HOSPoF over PF-UKF. It can be explained that a larger set of randomly generated sigma points and the corresponding probability weights, employed by PF-HOSPoF, allow an improved representation of the posterior density of the state vector. The results are also represented in figures 4.3-4.6 for easy visual comparison, with the difference between PF-UKF and PF-HOSPoF magnified in separate plots. All the numerical experiments were performed using Matlab 7.2 on a desktop with a dual core Pentium IV processor, 2.40GHz and 3.24Gb RAM.

4.5 Summary

In this chapter the heuristic, proposed in chapter 3, has been further extended to non-deterministic, moment matching sampling from the approximate posterior distribution in a particle like filter. The new method based on the random sigma point generation is called PF-HOSPoF.

This method is significantly different from the one discussed in chapter 3, as not only any number of particles can be generated, but also almost all of the corresponding probabilities are generated randomly. These sigma points and the probability weights match the given mean vector, the covariance matrix, the average marginal third and fourth moments exactly, without the use of optimization. This matching of higher order moments allows better representation of the posterior density and propagates more information about the shape of the distribution, than is achieved by the proposals generated by EKF and UKF. This intuition has been demonstrated by two numerical examples, one simulation and the other one based on real financial market data, where both illustrate the utility of the proposed algorithm. In the latter example, a much more significant improvement of PF-HOSPoF over PF-UKF has been achieved, compared to improvement of HOSPoF over UKF in chapter 3. This is due to introduction of randomness into the set of the new sigma points employed in PF-HOSPoF. The stratified sampling algorithm has also been introduced for random weights generation.

Figure 4.3: AvRMSE for in-sample data for CIR model

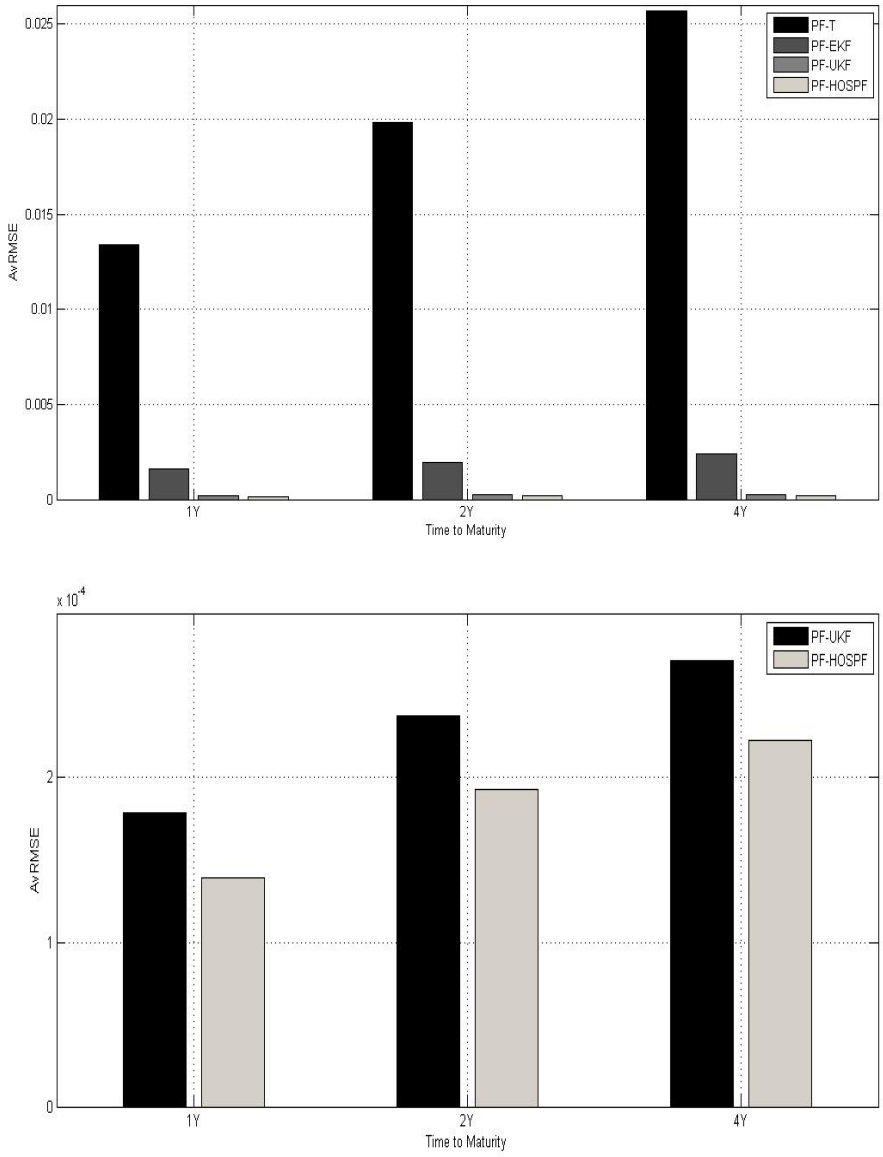


Figure 4.4: AvMRAE for in-sample data for CIR model

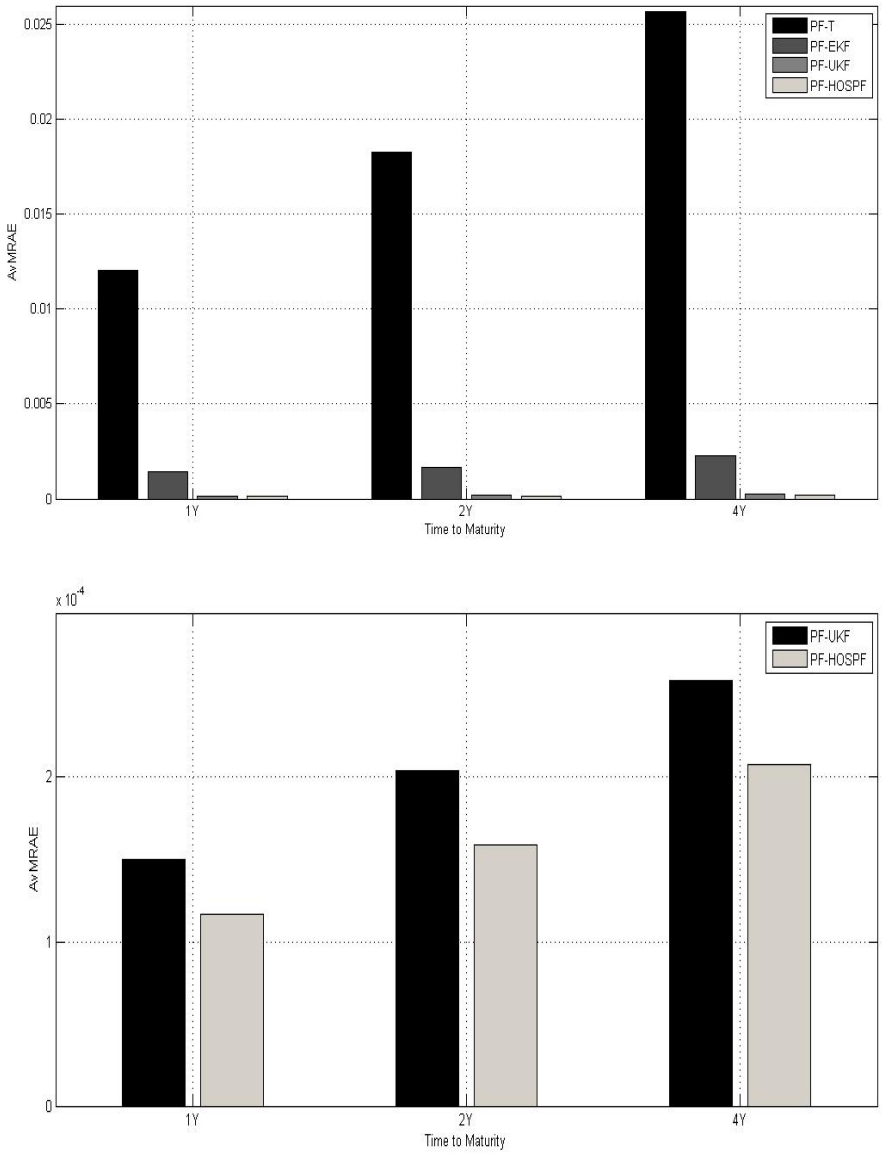


Figure 4.5: AvRMSE for out-of-sample data for CIR model

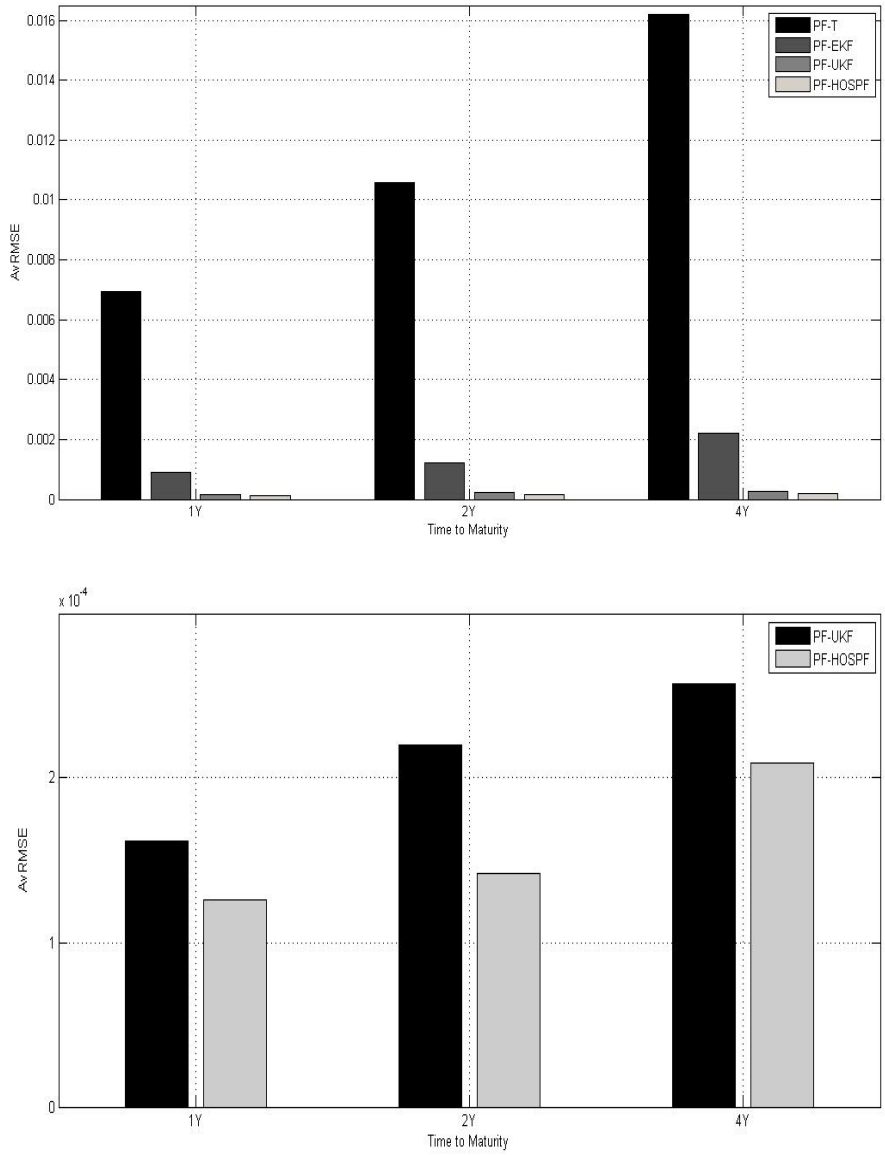
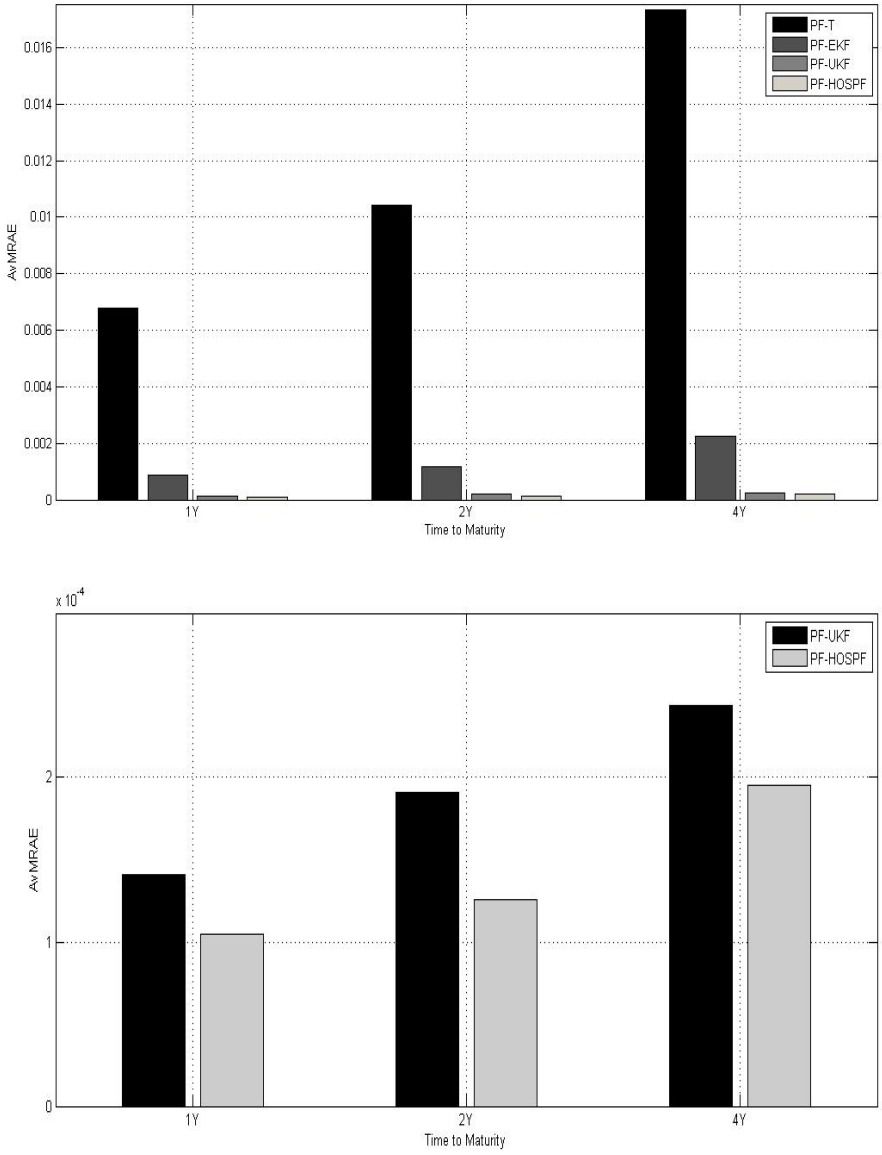


Figure 4.6: AvMRAE for out-of-sample data for CIR model



Chapter 5

Comparison with maximum entropy distributions

5.1 Introduction

In chapter 4 a method has been proposed for generation of random number of sigma points and corresponding probability weights that would match the mean vector, the covariance matrix, the average marginal third and fourth moments, without the use of optimization. In this chapter a comparison of the random moment matching method with entropy maximization is presented.

5.2 Maximum entropy distributions

The problem of determining the discrete probabilities of a set of events (or the continuous probability density function over an interval), conditioned upon the moment constraints, is quite common in probability and statistics. It might be possible to easily and accurately compute the first few moments from the data, but not know the probability distribution itself. There are several methods that solve this problem and in this section the maximum-entropy approach is considered. The focus is only on the discrete random variables, for continuous case please see Shannon [1948] and Smith [1993]. Consider discrete random variables $\mathcal{X}^{(i)}$ with corresponding probabilities p_i, where $i = 1, ..., n$. The

information about this probabilistic system is available in the form of constraint of probabilities

$$\sum_{i=1}^{n} p_i = 1,\ p_i \geq 0,\ i = 1, 2, ..., n, \tag{5.1}$$

and the moment constraints

$$\sum_{i=1}^{n} g_j(\mathcal{X}^{(i)}) p_i = a_j,\ j = 1, 2, ..., m, \tag{5.2}$$

where $g_j(\mathcal{X}^{(i)})$ and a_j are known constants.

The uncertainty measure of probability distribution, called entropy, was introduced in Shannon [1948] and is given by:

$$H(p) = -\sum_{i=1}^{n} p_i \ln p_i. \tag{5.3}$$

The use of the entropy measure in statistical problems was first proposed in Jaynes [1957]. In particular problem of determining p_i conditioned on (5.1) and (5.2), can be solved by maximizing the Shannon entropy in (5.3). The maximum entropy probability distribution is the most unbiased probability distribution conditioned upon the available information.

5.3 Computational issues

The problem of maximizing entropy subject to moment constraints is a convex optimization problem. The Lagrangian multiplier method in Fletcher [1991] can be used to determine p_i subject to (5.1) and (5.2) as:

$$p_i = exp\left(-\lambda_0 - \lambda_1 g_1(\mathcal{X}^{(i)}) - \lambda_2 g_2(\mathcal{X}^{(i)}) - ... - \lambda_j g_j(\mathcal{X}^{(i)})\right),\ i = 1, 2, ..., n, \tag{5.4}$$

where $\lambda_0, \lambda_1, ..., \lambda_m$ are the Lagrange multipliers corresponding to the $m+1$ constraints. Substituting (5.4) into (5.1) and (5.2) gives:

$$\lambda_0 = \ln\left(\sum_{i=1}^{n} exp(-\sum_{j=1}^{m} \lambda_j g_j(\mathcal{X}^{(i)}))\right), \tag{5.5}$$

$$a_k = \frac{\sum_{i=1}^{n} g_k(\mathcal{X}^{(i)}) exp(-\sum_{j=1}^{m} \lambda_j g_j(\mathcal{X}^{(i)}))}{\sum_{i=1}^{n} exp(-\sum_{j=1}^{m} \lambda_j g_j(\mathcal{X}^{(i)}))}, \tag{5.6}$$

where $k = 1, 2, ..., m$. Equations (5.6) represent m nonlinear equations with m unknowns. Rewriting these equations in the form of residuals makes them more convenient to find the numerical solution:

$$\widetilde{R}_k = 1 - \frac{\sum_{i=1}^{n} g_k(\mathcal{X}^{(i)}) exp(-\sum_{j=1}^{m} \lambda_j g_j(\mathcal{X}^{(i)}))}{a_k \sum_{i=1}^{n} exp(-\sum_{j=1}^{m} \lambda_j g_j(\mathcal{X}^{(i)}))}, \tag{5.7}$$

for $k = 1, 2, ..., m$. The least-squares method (see, e.g. Fletcher [1991]) is used to find the Lagrangian multipliers by minimizing the sum of the squares of the residuals:

$$R = \sum_{k=1}^{m} \widetilde{R}_k^2. \tag{5.8}$$

5.4 Methodology

It is desirable to compare probability weights, W_i, obtained by the new random sigma point generation method in chapter 4, section 4.2, with the p_i, obtained using the maximum entropy method. It is of interest to test these two methods on $n = 9$ univariate points, $\mathcal{X}^{(i)}$, given $m = 4$ moment constrains and with $g_j(\mathcal{X}^{(i)})$ defined as:

$$\begin{aligned} g_1(\mathcal{X}^{(i)}) &= \mathcal{X}^{(i)}, \\ g_2(\mathcal{X}^{(i)}) &= (\mathcal{X}^{(i)} - a_1)^2, \\ g_3(\mathcal{X}^{(i)}) &= (\mathcal{X}^{(i)} - a_1)^3, \\ g_4(\mathcal{X}^{(i)}) &= (\mathcal{X}^{(i)} - a_1)^4, \end{aligned} \tag{5.9}$$

where $i = 1, 2, ..., n$ and $a_1 = \sum_{i=1}^{n} g_1(\mathcal{X}^{(i)}) p_i$. Here a_1 is the mean, a_2 is the covariance, a_3 and a_4 are respectively the average marginal third and fourth moments. Two sets of probabilities will be calculated for samples generated from four known probability distributions. The idea is to use the moments of the sampled values to generate two sets of probability weights, using the new method and maximum entropy one.

For comparison the relative entropy or Kullback Leibler distance, see Cover & Thomas [1991] for more details, is used to determine which set of probabilities is closest to the original distribution. Relative entropy for two discrete distributions P and Q, $D(P||Q)$, is defined as:

$$D(P||Q) = \sum_{i=1}^{n} P_i \ln \frac{P_i}{Q_i}, \tag{5.10}$$

where $i = 1, 2, ..., n$. Here P is considered a "true" distribution and Q is an approximation of it. The relative entropy is a measure of the distance between two distributions, although it is not a metric. In this particular example, it is possible to interpret it as the set of probabilities with the smallest D that gives an accurate approximation of the "true" distribution.

For the discrete approximation P to each of the four distributions that are to be tested method in Hochreiter& Pflug [2007] will be used. It states that among all one-dimensional probability distributions $\tilde{G}$ which sit on the mass points $\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, ..., \mathcal{X}^{(n)}$, the one closest to G in the Wasserstein distance has the following probability weights:

$$P_i = G(\frac{\mathcal{X}^{(i)} + \mathcal{X}^{(i+1)}}{2}) - G(\frac{\mathcal{X}^{(i)} + \mathcal{X}^{(i-1)}}{2}), \tag{5.11}$$

where $\mathcal{X}^{(0)} = -\infty$ and $\mathcal{X}^{(n+1)} = +\infty$. Here G is a cumulative density function.

Finally, the algorithm for the comparison test for probability weights W_i and p_i would be as follows:

1. Draw samples from a known probability distribution, calculate the first four moments of these variables to determine a_j, $j = 1, ..., 4$.
2. Using these first four moments in the random sigma point generation method, obtain 9 sigma points, $\mathcal{X}^{(i)}$, and corresponding probability weights, W_i, $i = 1, ..., 9$.
3. Calculate values of $g_j(\mathcal{X}^{(i)})$ required to determine p_i in maximum-entropy method.
4. Determine p_i by constructing and minimizing R in (5.8).
5. Using $\mathcal{X}^{(i)}$, calculate the P_i in (5.11).
6. Find D_W and D_p using (5.10) with W_i and p_i respectively instead of Q_i, $i = 1, 2, ..., n$.

5.5 Examples

The test will be carried out for samples generated from four known probability distributions. These distributions are standard normal(Gaussian), Student's t-distribution, Laplace and Gumbel probability distributions. Matlab is used to generate random samples from these distributions and in-built optimization solver in Excel to find the minimum of R from (5.8). Figures 5.1-5.4 represent the

simulated densities along with two approximations, one with W_i generated by the random point generation method and the other with p_i obtained through entropy maximization.

5.5.1 Normal distribution

Standard norma distribution is well known probability distribution. 1000 samples are drawn from this distribution with zero mean and unit standard deviation. Then the mean(a_1), covariance(a_2), average marginal third(a_3) and fourth (a_4) moments are calculated from the data. These are shown in the table below.

Table 5.1. Values of a$_j$

a_1	a_2	a_3	a_4
0.0179	0.9563	0.0517	2.8874

The new random sigma point generation method gives $\mathcal{X}^{(i)}$, W_i and then probability weights p_i are obtained using the maximum-entropy method. These are shown in the table 5.2. Table 5.3 presents the minimized sum of residuals value, R, relative entropy for W_i and p_i.

Table 5.2. Values of sigma points and probability weights

$\mathcal{X}$	-2.5525	-1.2436	-1.1457	-0.9109	0.0179	0.9467	1.1815	1.2794	2.7687
W	0.021	0.0701	0.0824	0.1293	0.3957	0.1293	0.0824	0.0701	0.0196
p	0.0211	0.0662	0.0824	0.1361	0.3896	0.1378	0.082	0.0652	0.0197

Table 5.3. Relative Entropy values

R	$D(P\|\|W)$	$D(P\|\|p)$	Improvement of p over W
$5.83 * 10^{-7}$	0.02785	0.02723	2.22%

5.5.2 Students' t-distribution

For this distribution a smaller sample of 100 points is generated and the first four moments are calculated as before. These values are given in table 5.4, whereas table 5.5 shows the two sets of probability weights and finally table 5.6 presents the relative entropy values.

Table 5.4. Values of $\mathbf{a}_j$

a_1	a_2	a_3	a_4
0.0516	1.1129	0.2308	4.6196

This distribution has heavier tails than normal distribution, which is reflected in higher average marginal third and fourth moments.

Table 5.5. Values of sigma points and probability weights

$\mathcal{X}$	-3.0991	-1.3555	-1.3095	-0.9641	0.0516	1.0673	1.4127	1.4588	4.2393
W	0.0096	0.0749	0.0801	0.1438	0.3854	0.1438	0.0801	0.0749	0.0072
p	0.0093	0.076	0.0828	0.1504	0.3644	0.1554	0.081	0.0735	0.0072

Table 5.6. Relative Entropy values

R	$D(P\|\|W)$	$D(P\|\|p)$	Improvement of p over W
$3.57 * 10^{-4}$	0.02599	0.02595	0.13%

5.5.3 Gumbel distribution

Here the standard Gumbel distribution is used that has zero location parameter, μ, and unit scale parameter, β. In this case, the cumulative distribution is $G(x) = exp(-exp(-x))$. 1000 samples are generated with $x = -\ln(-\ln(U))$, where U is drawn from a uniform distribution in the interval $[0, 1]$. The first four moments are generated from the data and are shown these in table 5.7. Table 5.8 shows the two sets of probability weights and finally table 5.9 presents the relative entropy values.

Table 5.7. Values of $\mathbf{a}_j$

a_1	a_2	a_3	a_4
0.5708	1.6239	2.2432	14.4907

Table 5.8. Values of sigma points and probability weights

$\mathcal{X}$	-1.4591	-0.9745	-0.5362	-0.46	0.5708	1.6779	2.1162	2.6008	5.5484
W	0.0427	0.0737	0.1435	0.0918	0.3694	0.1435	0.0737	0.0427	0.019
p	0.0016	0.0422	0.1913	0.2232	0.2655	0.0939	0.0793	0.0868	0.0161

Table 5.9. Relative Entropy values

R	$D(P\|\|W)$	$D(P\|\|p)$	Improvement of p over W
$2.56 * 10^{-2}$	0.0593	0.0574	3.22%

5.5.4 Laplace distribution

Here the Laplace distribution is used with zero location parameter, μ, and unit scale parameter, β. 1000 samples are generated, $x = -sgn(U)\ln(1 - 2|U|)$, where U is drawn from the uniform distribution in the interval $(-\frac{1}{2}, \frac{1}{2}]$. The $sgn(x)$ function is a sign function equal to -1 if x is negative, $+1$ if it is positive and 0 if $x = 0$. The cumulative distribution function is $G(x) = \frac{1}{2}(1 + sgn(x)(1 - exp(-|x|)))$. The first four moments of the data are shown in table 5.10 below. Table 5.11 shows the two sets of probability weights and finally table 5.12 presents the relative entropy values.

Table 5.10. Values of $\mathbf{a}_j$

a_1	a_2	a_3	a_4
-0.0101	2.0644	0.0954	33.3645

Table 5.11. Values of sigma points and probability weights

$\mathcal{X}$	-4.7713	-1.4577	-1.3742	-0.9856	-0.0101	0.9655	1.3541	1.4376	4.8192
W	0.0307	0.0525	0.0592	0.1157	0.4842	0.1157	0.0592	0.0525	0.0303
p	0.0290	0.0648	0.0770	0.1559	0.3469	0.1562	0.0769	0.06457	0.0287

Table 5.12. Relative Entropy values

R	$D(P\|\|W)$	$D(P\|\|p)$	Improvement of p over W
$5.61 * 10^{-3}$	0.01587	0.01500	5.45%

In Cover & Thomas [1991] it is shown that maximum entropy distribution based on the first few moments might not be always achievable. This is reflected in the non-zero values of R obtained. Even though the p_i are not always optimal, W_i, calculated using the new moment matching method in chapter 4, are very close to them. Any problem that requires optimization, has to consider two points: the existence of the solution and the amount of computational effort/time required to obtain

the solution. The computational effort can only grow as the number of variables, $\mathcal{X}^{(i)}$, increases. This is why the new method, proposed in chapter 4, is a suitable alternative in this case, since it does not require optimization and it yields a relative entropy which is remarkably close to the best approximation.

5.6 Summary

The problem of determining the discrete probabilities of a set of events, conditioned upon the moment constraints, is quite common in probability and statistics. One of the methods to solve this problem is to use maximum-entropy approach. This method requires the maximization of the Shannon's entropy, given the natural set of probability constraints and the set of the moment constraints. The probability density obtained this way is considered to be the most unbiased density conditioned on the given information. However, this maximum entropy distribution might not always be achievable and the discrete probability weights generated this way might match the given moments only approximately. In comparison, the random sigma point generation method, proposed in chapter 4, allows exact matching of the given mean, covariance, average marginal third and fourth moments. This matching of moments is in closed-form and does not require any optimization. These two methods have been tested on four well known probability densities: normal distribution, Students' t-distribution, Gumbel and Laplace distributions. The relative entropy or Kullback Leibler distance has been used to determine which set of probabilities is the closest to the original distribution. Note, that only the moments, obtained from the data sample, are given and no other information or knowledge of the original probability distribution is available. From the tables of the results and the figures, it can be concluded that the probability weights, generated using method from chapter 4, are remarkably close to the best approximation in many probability densities of practical interest. These probability weights also have an advantage of matching moments exactly and not requiring any optimization in the calculations.

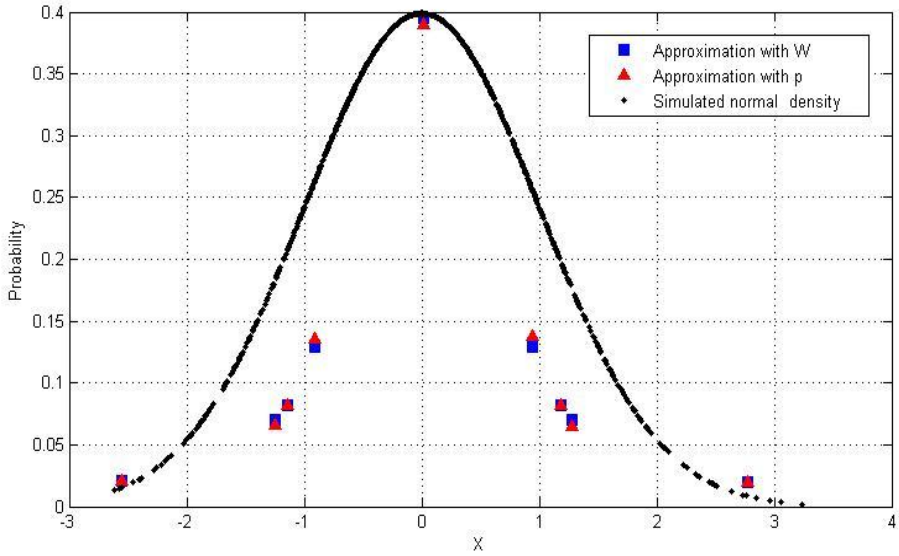


Figure 5.1: Simulated normal density and approximations

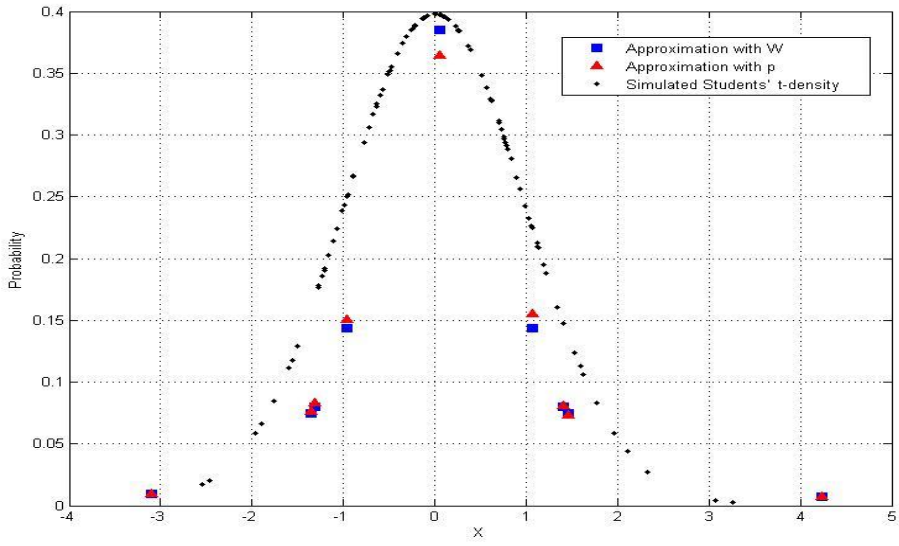


Figure 5.2: Simulated Students' t-density and approximations

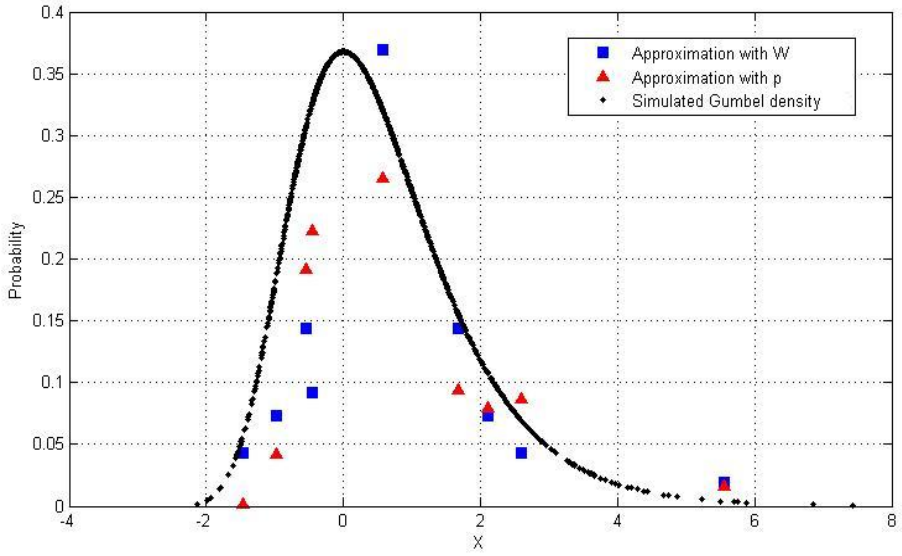


Figure 5.3: Simulated Gumbel density and approximations

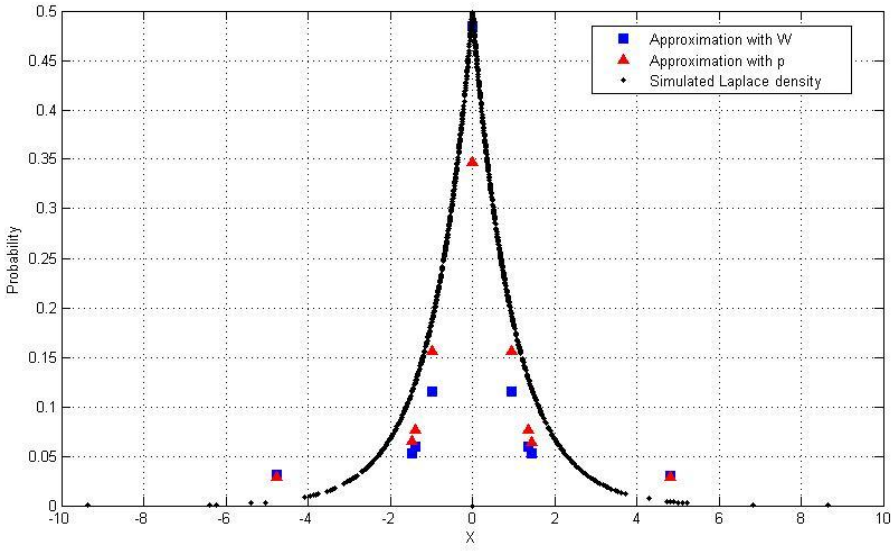


Figure 5.4: Simulated Laplace density and approximations

Chapter 6

Sigma point generation in financial portfolio optimization

6.1 Introduction

In chapter 4 the new random sigma point generation method was used in the context of nonlinear filtering. In this chapter this moment-matching sigma point generation algorithm will be used in the context of financial optimization. Recall that the proposed method for generating samples allows to match the given mean vector, the covariance matrix, the average marginal skewness as well as average marginal kurtosis (thus catering for asymmetric marginals), without needing an optimization procedure. In this chapter sigma points, proposed in (4.3)- (4.4) in chapter 4, would be referred to as scenarios in the context of financial optimization. The material presented in this chapter forms the basis of Ponomareva & *et al* [2012]. In the next section some motivation for moment-matching methods in the optimization problems is given.

6.2 Background

One of the traditional approaches for decision-making under uncertainty and risk is stochastic programming. It has wide ranging applications such as financial planning, energy systems management, supply chain logistics, agricultural planning. An extensive discussion of these methods can be found

in Wallace& Ziemba [2005].

Stochastic programming aims at solving optimization problems in which (some of) the parameters are not certain, but described by statistical distributions. In order for the stochastic programs to be numerically solved, the distributions involved are approximated by discrete distributions with a finite number of outcomes (scenarios). The approximation process, called "scenario generation", is important for the quality of the solution obtained: using "poor" scenarios could only result in obtaining a "poor" approximation of the true optimal solution.

One way to obtain scenarios is by sampling from an assumed distribution (or, simply, from historical data). As an example, future prices of financial assets may be assumed to follow a Geometric Brownian Motion, or a GARCH process (Bollerslev [1986]). Sampling methods have clear advantages. However, it may be argued that such distributional assumptions are too strong, or only applicable to one domain, viz. finance.

Other approaches to scenario generation with specific emphasis on operations research applications include principal component analysis-based simulation Topaloglou *et al* [2002] and stochastic approximation based on transportation metrics (Pflug [2001], Hochreiter& Pflug [2007]). A detailed survey of different scenario generation methods appears in Kaut & Wallace [2007].

Another class of scenario generation methods is based on matching a small set of statistical properties, e.g. moments (e.g., see Høyland & Wallace [2001]). These methods can be broadly divided into two groups.

Under the first approach, the statistical properties of the joint distribution are specified in terms of moments, usually including the covariance matrix. In Høyland *et al* [2003], cubic transformation of univariate, standard normal random variables and Cholesky factorization of covariance matrix are used to produce a multivariate distribution which approximately matches a given set of marginal central moments and the covariance matrix. Similar moment matching approach is employed to generate probability weights and support points using non-convex optimization in Gulpinar *et al* [2004]. In Smith [1993], the entropy maximization method is used to generate a discrete approximation to a given continuous distribution.

In the second group, specified (parametric) marginal distributions are sampled independently and the samples are then used along with Cholesky factorization of the covariance matrix to generate

the necessary multivariate distribution. An iterative procedure of this type in described in Lurie & Goldbers [1998], where specified marginal distributions and correlation matrix are used to produce the correlated vectors of random numbers.

The moment-matching scenario generation methods have been successful in the practical applications of stochastic optimization. They have several advantages: they are not specific to a particular field, they do not assume a specific parameterized family of distributions and allow for very different shapes of the marginals. It is only the moments that have to be specified - using historical data, predictions, specialist knowledge or a combination of these.
However, the moment matching approaches described in the above papers have certain limitations.

Firstly, they use non-convex optimization to generate scenarios which match a specified set of statistical properties, in addition to a needed factorization of the covariance matrix. Given a univariate random variable with known first 12 central moments, the approach used in Høyland *et al* [2003] and Høyland & Wallace [2001] finds a cubic polynomial function of this random variable which has the required four central moments. This requires a non-convex optimization in terms of the coefficients of the polynomial. The procedure has to be repeated iteratively for each marginal distribution. Similarly, the algorithm in Lurie & Goldbers [1998] requires a non-convex optimization over the space of lower triangular matrices.

Secondly, the achieved moments of the generated samples match the target moments only approximately. There are two sources of error in these moment matching methods: one is due to the fact that only local optima are found for the non-convex optimization problem and the other is the inexact starting moments of samples of univariate random variables. Since these procedures employ samples from a known, "simple" univariate distribution, the achieved moments usually depend on the sample moments of univariate random variables used.

6.3 Algorithm for scenario generation

In Date *et al.* [2008], a method was developed which uses *convex* optimization to match the given mean vector, covariance matrix exactly, and to minimize the mismatch between the marginal kurtosis across all variables. In chapter 4, this method has been extended to match the given mean vector, the

covariance matrix, the average marginal skewness as well as average marginal kurtosis, without the use of optimization. This new moment-matching algorithm may be used as a scenario generator on its own or its scalar version may be adopted to produce an initial guess for the optimization routines proposed by other authors. Being able to match a small set of statistical properties exactly, possibly with a very small set of scenarios, may be preferable to generating a very large number of scenarios to model the entire distribution. This is especially true when the scenarios are to be used in stochastic optimization procedures.

Specifically, the proposed moment-matching scenario generation algorithm has two major advantages over the existing methods:

1. It is computationally inexpensive - generating scenarios does not involve optimization. This is in contrast with all the other existing methods for moment matching which cater for moments higher than order two.

2. It generates scenarios (*i.e.* support points of the multivariate distribution) together with corresponding probability weights. This represents a big advantage not only because it eliminates the need to attach user-defined probabilities, but also because the computational difficulty and time for solving the stochastic program can be much decreased - a relatively modest number of *distinct* scenarios is needed to capture the spacial properties of multivariate density surface, such as marginal tail weights and asymmetry.

The significant reduction in computational complexity comes at the cost of matching only the *average* of third marginal moments and the *average* of fourth marginal moments exactly (instead of matching *all* third and fourth marginal moments approximately, as in Høyland *et al* [2003]). However, the computational simplicity as well as stability of results demonstrated in this paper arguably outweigh this shortcoming. If better moment-matching is needed for higher order marginals, the proposed method can provide an inexpensive initial guess for an algorithm such as the one proposed in Høyland *et al* [2003].

The quality of scenario generator, proposed in (4.3)- (4.4) in chapter 4, is tested in a financial portfolio optimization problem. Scenarios are generated for the future asset returns and use them as parameters in an optimization problem in which the portfolio's CVaR (Conditional Value-at-Risk)is

minimized (Rockafellar & Uryasev [2002]). In the next section, the performance of this scenario generator is evaluated for a real financial portfolio.

6.4 The mean-CVaR model for portfolio optimization

In the numerical experiments, scenarios are generated for a one-period portfolio optimization problem: given a set of n assets in which one may invest, how to divide *now* an amount of money W between these assets, such that, *after a specified period of time* T, to obtain a return on investment as high as possible?

The future returns of the assets are random variables; denote them by $R_1, \dots R_n$. Let x_j be the proportion of capital invested in asset j ($x_j = W_j/W$ where W_j is the capital invested in asset j), and let $x = (x_1, \dots, x_n)$ represent the portfolio resulting from this choice. This portfolio's return is the random variable: $\widetilde{R} = x_1 R_1 + \cdots + x_n R_n$; its distribution depends on the choice $x = (x_1, \dots, x_n)$ and on the distribution of $R_1, \dots R_n$.

With the popular "mean - risk" theory, a portfolio's return distribution is described by two statistics: the expected value and a "risk" value (desired to be kept low). The portfolio chosen for implementation should be "efficient", meaning, it should have the lowest risk value for a given expected return. An efficient portfolio is found by solving an optimization problem in which, for example, the risk of the portfolio is minimized, while a constraint on the expected return is imposed.

In most cases, there are no closed form solutions for these optimization problems. They have to be solved numerically, by approximating the distributions of the future asset returns with discrete ones with finite number of realizations; that is, by generating scenarios for the future asset returns.

Traditionally, risk is measured by variance (Markowitz [1952]). In the mean-variance optimization problem, there is no need scenarios for the future asset returns, but only their expected values and the covariance matrix. However, it has been pointed out that risk may be better quantified and several alternatives risk measures have been proposed.

More recently, portfolio optimization problems include more sophisticated risk measures, most notably those concerned with left tails of distributions. Risk measures in this category include Conditional Value-at-Risk (CVaR), which has good theoretical and computational properties and has

gained wide acceptance among academics and practitioners (Rockafellar & Uryasev [2002], Pflug [2000]).

In the numerical experiments, scenarios are generated for future asset returns in a mean-CVaR optimization model, which, unlike the mean-variance model, requires the full set of scenario returns. The scenarios and the corresponding probability weights are generated using equations (4.3)- (4.4) in chapter 4.

The definition of CVaR and with the model formulation, are presented below. Let $\widetilde{R}$ be a random variable representing the return of a portfolio x over a given holding period and $A\% = \alpha \in (0, 1)$ a percentage which represents a sample of "worst cases" for the outcomes of $\widetilde{R}$ (usually, $\alpha = 0.01 = 1\%$, $\alpha = 0.05 = 5\%$ or $\alpha = 0.10 = 10\%$).

The definition of CVaR at the specified level α is the mathematical transcription of the concept "average of losses in the worst $A\%$ of cases".

More formally, the CVaR at level α of $\widetilde{R}$ is defined as minus the mean of the α-tail distribution of $\widetilde{R}$, where the α-tail distribution is obtained by taking the lower $A\%$ part of the distribution of $\widetilde{R}$ (corresponding to extreme unfavourable outcomes) and rescaling it to span [0,1]. The α-tail distribution of $\widetilde{R}$ considers only losses above Value-at-Risk; for a detailed definition of CVaR, see Rockafellar & Uryasev [2002].

An important result is that CVaR can be computed and optimized by solving convex optimization problems. In Rockafellar & Uryasev [2002], an auxiliary function is used, $F : X \times \mathbb{R} \to \mathbb{R}$, where X is the set of feasible portfolios.

$$F_\alpha(x, v) = \frac{1}{\alpha}\mathbb{E}[-\widetilde{R} + v]^+ - v,$$

where $[u]^+ = u$ if $u \geq 0$ and 0 otherwise, $\mathbb{E}[]$ is the expected value operator. It was proved that minimizing CVaR over X can be done by minimizing F_α over $X \times \mathbb{R}$.

When the random asset returns are represented as discrete random variables (via scenarios), the CVaR optimization problem can be formulated as a linear program (LP). The algebraic formulation

of the mean-CVaR model is given below:

$$\min \quad -v + \frac{1}{\alpha} \sum_{i=1}^{S} p_i \cdot y_i \tag{M-CVaR}$$

Subject to:

$$v - \sum_{j=1}^{n} r_{ij} x_j \leq y_i, \quad \forall i \in \{1 \dots S\}$$

$$y_i \geq 0, \quad \forall i \in \{1 \dots S\}$$

$$\sum_{j=1}^{n} \mu_j x_j \geq d; \quad x \in X.$$

where the parameters of the model are:

- $S =$ the number of scenarios
- $n =$ the number of assets
- $p_i =$ the probability of scenario i occuring, $i = 1 \dots S$
- $r_{ij} =$ return of asset j under scenario i, $i = 1 \dots S$, $j = 1 \dots n$
- $\mu_j =$ the expected return of asset j, $j = 1 \dots n$ ($\mu_j = \frac{1}{S} \sum_{i=1}^{S} r_{ij}$)
- $d =$ the desired expected return of the portfolio

p_i and r_{ij} are obtained from scenario generation or sampling of historical data. μ_j are estimated prior to optmization (possibly from historical data). d is decided by the investor. By not imposing the last constraint on portfolio's expected return, the (absolute) minimum CVaR portfolio is obtained.
The decision variables of the model are:

- $x_j =$ the fraction of portfolio wealth invested in asset j, $j = 1 \dots n$
- $v =$ an α-quantile of the portfolio return distribution

- y_i = the magnitude of the negative deviations of the portfolio return from the α-quantile, for every scenario $i \in \{1 \dots S\}$ (they are 0 if the portfolio return $\sum_{j=1}^{n} r_{ij}x_j$ is higher than the α-quantile)

6.5 Numerical experiments

6.5.1 Scope and computational set-up

The behaviour of the proposed scenario generator is investigated, as used in conjunction with a mean-CVaR optimization problem (M-CVaR). The following are of interest:

(a) the stability of the scenario generator, both in-sample and out-of-sample;

(b) how the number of scenarios considered affects the optimal solution and the optimum;

(c) quality of the solution as compared to the solution obtained using historical data as scenarios.

A dataset drawn from FTSE 100 is used, with n=20 stocks and prices monitored weekly over the period of over 14 years from July 1997 until November 2011 (747 time periods of historical data). These 20 stocks were chosen in such a way that their combined market capitalization was roughly over 60% of FTSE 100.

The corresponding historical returns are computed and the moments for these series are calculatedf; they are displayed in Tables 6.1 - 6.3.

Table 6.1. Means for weekly returns of 20 assets, 10^{-4}

	1	2	3	4	5	6	7	8	9	10
Mean	6.27	25.51	8.9	13.79	7.06	38.6	37.67	36.97	13.08	19.25
	11	12	13	14	15	16	17	18	19	20
Mean	4.43	16.6	46.45	17.22	29.19	18.98	16.33	21.79	20.65	17.72

Table 6.2. Marginal 3rd and 4th moments for weekly returns of 20 assets, 10^{-6}

	1	2	3	4	5	6	7	8	9	10
Marginal 3rd moment	-8.36	25.34	1.74	5.29	9.27	74.39	86.66	26.93	-3.76	-3.81
Marginal 4th moment	26.5	24.22	11.6	14.93	10.32	44.83	133.18	23.55	15.04	7.93
	11	12	13	14	15	16	17	18	19	20
Marginal 3rd moment	-29.87	-63.57	191.96	-13.48	7.29	-8.86	3.33	285.29	157.21	42.79
Marginal 4th moment	21.5	29.4	121.67	12.31	31.28	9.97	6.96	459.01	87.35	23.84

The scenarios are generated for the future weekly returns of these stocks using the proposed moment-matching method, with the first four moments as above. Various number of scenarios are considered: $S = 123$, $S = 243$, $S = 363$, $S = 603$, $S = 723$, $S = 1083$ and $S = 5043$. (M-CVaR) is solved using the generated p_i and r_{ij}. d is fixed to 0.3%. CVaR is condsidered at confidence level $\alpha = 10\%$.

It is common practice to use for comparison a "benchmark" scenario generator with the same optimization model. In this study, the historical returns are used as "benchmark" scenarios. Thus, (M-CVaR) is also solved using p_i and r_{ij} as given by historical data, i.e. $p_i = 1/747$, r_{ij} =historical return of stock j at time period i.

The optimization problems are formulated in AMPL (Fourer *et al* [1989]) and solved with the FortMP solver (Ellison *et al* [2008]).

Table 6.3. Covariance matrix for weekly returns of 20 assets, 10^{-4}

	1	2	3	4	5	6	7	8	9	10
1	18.58	6.18	7.14	7.03	4.48	3.47	10.37	5.21	4.22	4.42
2	6.18	22.69	3.03	3.07	5.01	2.89	5.13	3.44	5.09	1.28
3	7.14	3.03	15.15	11.67	4.17	3.59	10.65	7.79	3.79	3.04
4	7.03	3.07	11.67	16.58	4.77	4.04	11.87	7.94	4.33	3.64
5	4.48	5.01	4.17	4.77	13.25	3.36	2.94	3.18	8.54	3.52
6	3.47	2.89	3.59	4.04	3.36	17.6	4.54	3.37	3.07	4.43
7	10.37	5.13	10.65	11.87	2.94	4.54	37.81	12.57	2.56	3.66
8	5.21	3.44	7.79	7.94	3.18	3.37	12.57	18.32	3.62	3.64
9	4.22	5.09	3.79	4.33	8.54	3.07	2.56	3.62	16.04	3.92
10	4.42	1.28	3.04	3.64	3.52	4.43	3.66	3.64	3.92	11.43
11	7.11	3.4	4.96	4.93	2.51	2.99	6.96	6.31	2.68	3.89
12	4.58	2.37	3.16	3.42	3.05	3.09	4.44	3.5	3.5	3.24
13	10.55	4.94	10.99	12.93	2.14	4.06	25.27	11.52	2.17	3.64
14	4.92	2.26	3.95	5.64	4.44	5.42	4.36	3.26	4.18	4.03
15	5.76	4.72	3.8	3.92	3.2	1.54	5.75	4.06	2.92	4.46
16	2.95	5.43	2.89	3.13	4.03	3.75	2.65	3.71	4.47	2.85
17	4.52	2.33	3.37	4.28	4.19	5.24	4.3	3.73	3.61	3.44
18	16.51	7.71	9.31	8.47	7.36	4.24	13.66	8.99	6.91	5.48
19	6.73	8.89	4.07	4.42	3.02	1.13	4.87	4.16	3.74	1.89
20	6.16	8.01	4.45	4.49	3.27	2.49	6.55	3.66	4.17	2.7

	11	12	13	14	15	16	17	18	19	20
1	7.11	4.58	10.55	4.93	5.76	2.95	4.52	16.51	6.73	6.16
2	3.4	2.37	4.94	2.26	4.72	5.43	2.33	7.71	8.89	8.01
3	4.96	3.16	10.99	3.95	3.8	2.89	3.37	9.31	4.08	4.45
4	4.93	3.42	12.93	5.64	3.92	3.13	4.28	8.47	4.42	4.49
5	2.51	3.05	2.14	4.44	3.2	4.03	4.19	7.36	3.02	3.27
6	2.99	3.09	4.06	5.42	1.54	3.75	5.24	4.24	1.13	2.49
7	6.96	4.44	25.27	4.36	5.75	2.65	4.3	13.66	4.87	6.55
8	6.31	3.5	11.52	3.26	4.06	3.71	3.73	8.99	4.16	3.66
9	2.68	3.5	2.17	4.18	2.92	4.47	3.61	6.91	3.74	4.17
10	3.89	3.24	3.64	4.03	4.46	2.85	3.44	5.48	1.89	2.7
11	14.97	4.24	7.4	3.05	6.09	3.41	2.28	11.38	4.54	4
12	4.24	19.85	4.71	4.98	4.02	2.5	4.36	7.32	0.96	2.4
13	7.4	4.71	34.81	4.25	5.79	2.19	4.51	13.44	6.9	5.63
14	3.05	4.98	4.25	14.12	4.34	2.46	6.24	6.3	2.29	2.49
15	6.09	4.02	5.79	4.34	22.62	2.84	3.28	9.1	5.02	4.01
16	3.41	2.5	2.19	2.46	2.84	10.65	3	5.11	4.23	3.79
17	2.28	4.36	4.51	6.24	3.28	3	11.76	6.9	1.36	1.77
18	11.38	7.32	13.44	6.3	9.1	5.11	6.9	46.21	10.27	8.75
19	4.54	0.96	6.9	2.29	5.02	4.23	1.36	10.27	28.29	12.14
20	4	2.4	5.63	2.49	4.01	3.79	1.77	8.75	12.14	19.22

6.5.2 Stability of the scenario generator

Stability is a basic and very important requirement. Since any scenario generator has an element of randomness, the final outcome depends on the values drawn from the corresponding distributions. Stability guarantees that the optimal solution of the optimization problem of interest does not vary (but, possibly, only to a small extent) with the specific scenario set chosen.

A scenario generator is said to manifest *in-sample stability* if, when generating several scenario sets of the same size and solving the optimization problem on each of these scenario sets, the optimal objectives are similar (see for example Kaut & Wallace [2007]).

A scenario generator is said to manifest *out-of-sample stability*, if, when generating several scenario sets of the same size and solving the optimization model on each of these scenario sets, the optimal solutions obtained yield similar "true" objective function values (i.e., the solutions obtained are evaluated on the "true" distribution of the random variables involved).

In Kaut & Wallace [2007] it is emphasized that, while it is straightforward to test in-sample stability (the scenario-based optimization problems are only solved), it is difficult to test the out-of-sample stability - that would mean knowing the "true" distribution of the random vector involved. What is usually done in practice is to use a "benchmark" scenario tree: a large scenario set obtained exogenously, that is known to be stable; this scenario set will stand for the "true" distribution.

In-sample stability does not imply the out-of sample one or vice versa. It is possible to have in-sample instability (of the objectives) but stability of the solutions - in this case, it is likely to have out-of-sample stability, since, in this case, all the solutions are tested on the same scenario set, representing the "true" distribution.

The in-sample stability of the new scenario generator is tested in the following way. For each number of scenarios S considered, 20 sets of scenarios are generated and used as an input in the mean-CVaR optimization problem (M-CVaR).

For each scenario size, a set of 20 optimal objective values is obtained; their statistics are displayed in Table 6.4 below.

Table 6.4. In-sample stability: Statistics of the sets of optimums for various scenario sizes

S	Mean	StDev	Min	Max	Range
123	0.032736	0.000336	0.032209	0.033004	0.000795
363	0.033326	0.000300	0.033053	0.033699	0.000646
723	0.036456	0.000276	0.036120	0.036751	0.000630
1083	0.036835	0.000234	0.036581	0.037163	0.000581
5043	0.037465	0.000128	0.037277	0.037702	0.000425

The in-sample stability is obvious; even for relatively small number of scenarios, the optimum CVaRs obtained have similar values, ranging over very small intervals. As expected, stability is further increased with increasing number of scenarios.

Also expected is the increase in the optimum values obtained with increasing number of scenarios. When considering 123 scenarios, the optimal CVaRs obtained range between 3.22% and 3.3%, while for 5043 scenarios, the optimums range between 3.73% and 3.77%. This is because when more scenarios are taken into account, the optimal CVaR can only get worse - and closer to the "true" optimal CVaR. However, this difference is small: for 123 scenarios, the average of the set of 20 optimal CVaRs is 3.27% while for 5043 scenarios, the average optimum is 3.75%.

For testing the out-of-sample stability, the historical data is used as a benchmark scenario set. Each of the optimal solutions obtained before are evaluated on the historical data, i.e. the portfolio weights previously obtained (using the new scenario generator) are used and the corresponding CVaRs are computed using historical scenarios. Thus, 20 "true" optimums ("historical CVaRs") are obtained for each scenario size. Their statistics are displayed in Table 6.5.

Table 6.5. Out-of-sample stability: Statistics of the sets of "true" optimum CVaRs -

S	Mean	StDev	Min	Max	Range
123	0.040995	0.000613	0.040315	0.041675	0.001360
363	0.040461	0.000554	0.039921	0.041088	0.001167
723	0.040277	0.000506	0.039700	0.040855	0.001155
1083	0.040221	0.000342	0.039397	0.040514	0.001117
5043	0.039621	0.000283	0.039211	0.040264	0.001053

The out-of-sample stability is also obvious, even for small scenario sets; for example, in the case of 123 scenarios, the "true" CVaRs range between 4.03% and 4.16%. As expected, better solutions are

obtained with increasing number of (in-sample) scenarios: not only even more stable, but also resulting in a better (i.e. smaller) out-of-sample CVaRs. Again notice that, when increasing the number of (in-sample) scenarios, the difference between the out-of-sample CVaRs is small: an average "true" CVaR of 4.01% is obtained for 123 in-sample scenarios; the average true CVaR for 5043 in-sample scenarios is 3.96%.

Also note that, for 5043 in-sample scenarios, the in-sample CVaRs (obtained using scenarios generated with the proposed method) and out-of-sample CVaRs (the same optimal solutions but the objective function evaluated on historical data) have very close values: 3.75% and 3.96%, respectively.

6.5.3 Optimal solutions

Note that the proposed scenario generator is not only stable in-sample and out-of-sample (with respect to the optimum CVaR), but also manifests "stability of optimal solutions". With a specified number of in-sample scenarios, the optimal portfolio weights obtained are very similar under different runs. This is valid even in the case of smallest scenario sets (123 scenarios). Table 5.6 below displays the optimal portfolio weights obtained in each of the 20 runs, when the scenario size is 5043.

Moreover, with increasing number of scenarios, the optimal portfolio weights change only *marginally*, i.e. the portfolio weights obtained by using 123 scenarios are similar to the portfolio weights obtained by using 5043 scenarios (please see Table 6.7).

This supports the idea that stable solutions can be obtained with relatively small number of scenarios and increasing the number of scenarios leads only to a marginal change in the optimal solution. A natural question to ask is: how do these solutions compare with the portfolio solution obtained by using the historical data as *in-sample* scenarios?

Table 6.7 displays a summary of optimal solutions obtained by using in-sample: (a) the proposed scenario generator with various scenario set sizes and (b) historical data. For a given scenario set size, the weights displayed are obtained by averaging the (very similar) optimal weights obtained for the 20 different runs. "HD" signifies the portfolio weights obtained using historical data as in-sample scenarios (747 scenarios equally probable scenarios).

Table 6.6. Optimal portfolio weights for S= 5043 scenarios

CVaR	W2	W6	W7	W8	W9	W10	W12	W13	W14	W15	W16	W17	W19	W20
0.040264	4.63%	36.00%	0.82%	9.76%	5.16%	5.06%	5.18%	5.47%	2.21%	7.80%	6.09%	2.59%	4.95%	4.28%
0.039535	5.07%	32.67%	0.81%	9.37%	4.62%	6.16%	5.25%	5.76%	2.87%	7.89%	7.13%	3.22%	4.90%	4.27%
0.039442	5.11%	32.10%	0.90%	9.58%	4.70%	6.24%	5.27%	5.79%	2.98%	7.68%	7.08%	3.35%	4.85%	4.38%
0.039774	5.09%	33.65%	0.70%	9.54%	4.46%	5.99%	5.06%	6.00%	2.63%	7.84%	7.16%	2.99%	4.86%	4.02%
0.039512	5.02%	32.36%	0.95%	9.09%	4.88%	6.14%	5.31%	5.90%	2.95%	7.78%	7.08%	3.27%	4.87%	4.39%
0.039266	5.10%	30.83%	0.89%	9.68%	4.57%	6.34%	5.26%	6.24%	3.13%	7.84%	7.44%	3.45%	4.84%	4.40%
0.039261	5.14%	30.87%	0.99%	9.60%	4.67%	6.31%	5.23%	6.07%	3.14%	7.96%	7.24%	3.54%	4.91%	4.33%
0.040007	4.81%	34.80%	0.59%	9.40%	4.89%	5.59%	5.17%	5.99%	2.64%	7.65%	6.52%	2.83%	4.87%	4.27%
0.039491	5.08%	32.08%	0.95%	9.75%	4.39%	6.20%	5.25%	5.81%	2.96%	7.85%	7.01%	3.41%	4.91%	4.34%
0.039575	4.95%	32.53%	0.74%	9.98%	4.53%	5.94%	5.28%	5.94%	2.89%	7.73%	6.77%	3.35%	4.95%	4.41%
0.039644	5.18%	33.00%	0.76%	9.40%	4.79%	6.20%	5.01%	6.15%	2.95%	7.89%	7.06%	2.84%	4.72%	4.06%
0.039828	4.88%	33.83%	0.96%	9.24%	4.62%	5.82%	5.29%	5.64%	2.68%	7.88%	6.67%	3.15%	4.95%	4.39%
0.039679	5.07%	33.16%	0.98%	9.29%	4.53%	6.04%	5.24%	5.67%	2.89%	7.84%	6.93%	3.15%	4.91%	4.29%
0.039211	5.28%	30.75%	0.92%	9.63%	4.79%	6.58%	5.06%	6.16%	3.13%	7.92%	7.41%	3.40%	4.83%	4.14%
0.039463	5.10%	32.34%	0.97%	9.20%	4.80%	6.23%	5.17%	5.64%	2.95%	8.02%	7.10%	3.30%	4.90%	4.28%
0.039384	5.10%	31.40%	0.83%	9.82%	4.63%	6.28%	5.13%	6.33%	2.93%	8.03%	7.37%	3.18%	4.72%	4.26%
0.039395	5.20%	31.85%	0.84%	9.68%	4.74%	6.39%	4.99%	6.05%	2.99%	7.90%	7.54%	3.02%	4.70%	4.11%
0.040039	4.75%	35.02%	0.97%	9.04%	4.72%	5.52%	5.22%	5.41%	2.71%	7.73%	6.76%	2.91%	4.98%	4.27%
0.039810	4.77%	33.74%	1.10%	8.87%	4.74%	5.70%	5.35%	5.52%	2.85%	7.78%	6.74%	3.31%	5.07%	4.46%
0.039845	4.73%	33.94%	0.73%	9.39%	4.68%	5.61%	5.19%	5.96%	2.75%	7.72%	7.04%	3.03%	4.91%	4.33%

Another encouraging result is that the "true" CVaR of the portfolio obtained using historical data in-sample (that is, the optimization *and* evaluation of CVaR is done on historical data) is 3.87%. Thus, as measured on the benchmark scenario generator, a solution obtained using the scenario generator in-sample gives a "true" CVaR of 3.96%, while the solution obtained using historical data in-sample gives a value only 0.09% lower.

Table 6.7. Optimal portfolio weights obtained using (a) the proposed method with various number of scenarios and (b) historical data(HD) as scenarios

S	CVaR(in)	CVaR(out)	W2	W6	W7	W8	W9	W10
HD	0.038712		13.83%	24.50%	0.00%	16.37%	3.55%	8.33%
123	0.032736	0.040995	4.20%	37.92%	0.92%	7.99%	5.66%	4.62%
243	0.032828	0.040559	4.28%	36.24%	0.51%	8.58%	5.48%	4.76%
363	0.033326	0.040461	3.70%	34.92%	0.36%	9.21%	5.82%	3.97%
603	0.033704	0.040449	4.38%	35.27%	1.18%	8.90%	2.88%	6.00%
723	0.036456	0.040277	4.41%	35.62%	0.61%	9.66%	5.09%	4.72%
1083	0.036835	0.040221	4.62%	35.87%	0.65%	9.45%	5.05%	5.27%
5043	0.037465	0.039621	5.00%	32.85%	0.87%	9.46%	4.69%	6.02%
W12	W13	W14	W15	W16	W17	W18	W19	W20
5.69%	6.50%	0.60%	9.35%	9.64%	1.64%	0.00%	0.00%	0.00%
5.37%	5.40%	2.26%	7.09%	5.38%	2.74%	0.84%	5.11%	4.50%
5.18%	5.71%	2.11%	8.95%	5.63%	2.61%	0.79%	4.85%	4.30%
5.23%	6.33%	2.40%	9.75%	4.97%	2.98%	0.86%	4.93%	4.57%
5.90%	5.24%	3.39%	8.12%	6.72%	1.64%	0.31%	5.34%	4.72%
5.32%	6.05%	2.62%	7.24%	6.18%	2.92%	0.00%	5.07%	4.49%
5.11%	5.67%	2.39%	7.77%	6.16%	2.72%	0.00%	5.02%	4.24%
5.20%	5.87%	2.86%	7.84%	7.01%	3.17%	0.00%	4.88%	4.28%

6.6 Summary

The random sigma point generation method, proposed in chapter 4, has been used in the context of financial optimization. The method presents several advantages over the existing approaches for scenario generation in financial portfolio optimization. First, it is computationally cheap; there is no optimization involved in generating scenarios. Secondly, due to the unequal generated probabilities of the scenarios, this method may perform well even with a relatively small number of scenarios. In contrast, methods that would assume by default equal probabilities would need a larger number of scenarios.

These assertions are supported by the numerical results. The quality of the proposed scenario generator has been tested in a mean-CVaR portfolio optimization model. Several observations were made. First, the method is remarkably stable, both in-sample and out-of-sample. It also manifests

stability of optimal solutions, i.e. not only the optimums are similar, but also the optimal portfolio weights, representing the solution to implement. Secondly, the optimal solutions vary only marginally with increasing number of scenarios. That means, by using only a small number of scenarios, a reasonably good solution is obtained.

In the numerical case presented here, direct comparison with solutions obtained via using historical data could be made. The proposed scenario generation method is applicable to cases where there is no (or not enough) historical data available, but only expert opinion on the statistical properties involved.

This method has also a big advantage when used with a computationally difficult optimization model, requiring only a limited number of scenarios. Stable and good quality solutions can be obtained with a relatively small number of scenarios, which presents a great computational advantage. Intuitively, this may be attributed to the fact that unequal and random probability weights are generated along with the support points. The creation of non-equally weighted scenarios is especially important, as this approach ensures that extreme events are considered even if the size of the approximated set is small.

The numerical results are very encouraging. This scenario generation method may work well for multi-stage stochastic optimization, where generally only a limited number of scenarios can be considered and there are scale-up issues.

Mean-CVaR optimization is used widely in financial industry to maximize the portfolio return and minimize the average losses. The new scenario generation method, based on random sigma point generation of chapter 4, would be a very useful tool for improving risk management and can help with stock investing strategy.

Chapter 7

Time series filtering under parameter perturbations

7.1 Introduction

In this chapter, the problems in filtering context and once again discrete state space systems, mentioned in chapters 3 and 4, are considered. In these two chapters, the state parameters were considered constant, either known or calibrated from data. Here, however, the case when these parameters are under random perturbation is investigated. An exact, closed-form minimum variance filter is designed for a class of perturbed discrete time systems which allows for both multiplicative and additive noise sources. The multiplicative noise model includes a popular class of models (Cox-Ingersoll-Ross type models) in econometrics. The parameters of the system under consideration which describe the state transition are assumed to be subject to stochastic uncertainties. The problem addressed is the design of a filter that minimizes the trace of the estimation error variance. Sensitivity of the new filter to the size of parameter uncertainty, in terms of the variance of parameter perturbations, is also considered. The new filter is referred to as the 'perturbed Kalman filter' since it reduces to the traditional (or unperturbed) Kalman filter as the size of stochastic perturbation approaches zero. A related approximate filtering heuristic for univariate time series is also considered and the filter based on this heuristic is referred to as approximate perturbed Kalman filter. The performance

of the new filters is tested on three simulated numerical examples and the results are compared with the unperturbed Kalman filter that ignores the uncertainty in the transition equation. The material presented in this chapter forms the basis of Ponomareva & Date [2012].

7.2 Background

The estimation of the state variables given the noisy measurements is one of the fundamental problems in control and signal processing, as mentioned in chapter 1. It is well known that the Kalman filter requires an accurate model with a Gaussian noise. See, for example, Anderson & Moore [1979]. However, precise modeling of systems is usually difficult or impossible and system parameters may vary with time or be affected by disturbances. This has motivated studies on robust Kalman filter design that will guarantee an upper bound on the filtering error covariance for any parameter uncertainty. Filtering with guaranteed error for uncertain systems was first considered in Jain [1975]. In recent years, several results have been derived on the design of robust estimators that give an upper bound on the error variance for any allowed modeling uncertainty, see Jain [1975], Shaked & de Souza [1995] and references therein. The idea of seeking the upper bound on the error variance has been recently applied to discrete-time Markovian jump systems with parameter and noise uncertainty in Zhu *et al* [2007].

A Riccati equation based approach was adopted in Jain [1975], Zhu *et al* [2002], Xie & Soh [1994], Shaked & de Souza [1995] and Yang *et al* [2002] to deal with parameter uncertainty of norm-bounded type. More recently, in Suoto *et al* [2009], this method has been extended to uncertain discrete-time nonlinear systems where nonlinear functions are assumed to be unknown but within a conic region, characterized as a Lipschitz condition on the system state and control signal residuals. The Riccati equation approach involves searching for a scaling parameter that insures that the associated Riccati equation has a solution and the guaranteed error is minimized. The parameterized Riccati differential equation is used for the finite horizon case ($\mathcal{H}_2$) and an algebraic Riccati equation is for the infinite horizon ($\mathcal{H}_\infty$).

Note that the coefficient matrices in the algebraic Riccati equation, both in continuous (CARE) and discrete time (DARE), can themselves be affected by perturbations. These perturbations arise due to

the various errors in the formulation of the problems or in the computer solutions. It is interesting to know how the Hermitian positive semidefinite solution to algebraic Riccati equation is affected in this case. Many perturbation results can be found in the literature. See Byers [1985] for the first-order perturbation bounds for the solution to CARE, whereas global perturbation bounds for the solution were derived in Chen [1988] and Konstantinov & *et al* [1990]. More details on the computational methods involved can be found in Petkov & *et al* [1991]. As a continuation of these results, new perturbation bounds for the Hermitian positive semidefinite solution to the CARE and DARE were derived in a uniform manner in Sun [1998]. More recently the improved perturbation bounds for a specific matrix expression have been derived. This specific matrix expression arises in DARE problems. The new bounds application to the sensitivity analysis and the solution of fractional-affine matrix equations have been considered in Konstantinov & *et al* [2009].

An alternative approach to modeling uncertainty is based on the linear matrix inequality (LMI) method and involves interior point algorithm for convex optimization. This interior-point method for convex optimization has also been used to recursively compute the minimal confidence ellipsoid for the state in Ghaoui & Calafiore [2001]. The LMI approach has been applied to solve the robust $\mathcal{H}_2$ and $\mathcal{H}_\infty$ filtering for systems with norm-bounded uncertainty, integral quadratic constraints and polytopic uncertainty, see Barbosa *et al* [2002], Geromel [1999], de Souza *et al* [2008] and Xie *et al* [2003]. When the model under study is nonlinear, the uncertainty set to be characterized may be nonconvex and may even consist of several disconnected components. The nonlinear parameter and state estimation can be formulated in a bounded-error context as problems of characterization of sets. Then obtaining approximations of these sets is possible with the tools provided by interval analysis, that first appeared in Moore [1959] to assess the numerical errors resulting form the use of a finite-precision arithmetic. Kieffer & Walter [2005] provides more details on the application of the interval analysis nonlinear parameter and state estimation. A problem with both deterministic (unknown-but-bounded) and stochastic uncertainties has been considered in Yang *et al* [2002], Petersen & James [1996], Gershon *et al* [2001], Ghaoui [1995] and more recently in Wang *et al* [2006]. In these papers this stochastic uncertainty is expressed as a multiplicative noise. Unlike the case of the additive noise, the second order statistics of the multiplicative noise is usually unknown, as it depends on the real state of the system.

The approach taken in this chapter differs from the contributions mentioned above in several crucial aspects. First, discrete time-varying uncertain systems are considered with both multiplicative and additive noise sources. The system under consideration is assumed to be subject to stochastic uncertainties in the parameters of the transition equation. This class of systems includes square-root affine systems, *i.e.* Cox-Ingersoll-Ross type models, first introduced in Cox *et al.* [1985], which are frequently used in econometrics and finance literature, and also models that arise in the area of stochastically sampled digital control systems, see de Koning [1980] for more details. Secondly, the problem addressed is the design of a filter that minimizes the estimation error variance. The optimal linear one-stage predictors for the discrete-time systems with different types of multiplicative noises were derived by Nahi [1969], Rajasekaran *et al* [1971], while conditions for uniform asymptotic stability of the optimal linear filter was considered in Tugnait [1981]. In de Koning [1984], an optimal linear estimator for linear discrete-time systems with stochastic parameters that are statistically independent in time was derived. In this chapter an independent derivation of the results in de Koning [1984]is provided for a class of perturbed systems. In contrast with the optimization based worst case approaches, an exact, closed form expression for this variance minimizing filter is derived here. For univariate time series, some results on the deviation of eigenvalues of the state transition matrix of the new filter from those of the state transition matrix of Kalman estimator for the same system, when the uncertainty is ignored, are presented. This analysis of eigenvalues of the perturbed system complements the results on asymptotic stability in de Koning [1984]. It is shown that the new perturbed filter is a well-behaved function of the model parameters, in the sense that it converges to the traditional Kalman filter as the variance of stochastic uncertainty tends to zero. Approximate filtering heuristic is considered for univariate time series, inspired by the exact method proposed here, which appears to work well for a wider case of nonlinear systems. It is also demonstrated through extensive numerical experiments that the new filters perform better than the unperturbed Kalman filter even when the size of uncertainty is poorly identified.

7.3 Derivation of the new filter

Consider the following discrete-time state-space system:

$$\boldsymbol{\mathcal{X}}(k+1) = \mathbf{A}\boldsymbol{\mathcal{X}}(k) + \Delta\mathbf{A}(k)\boldsymbol{\mathcal{X}}^{\gamma}(k) + \mathbf{B}\mathbf{w}(k+1), \tag{7.1}$$

$$\boldsymbol{\mathcal{Y}}(k) = \mathbf{C}\boldsymbol{\mathcal{X}}(k) + \mathbf{D}\mathbf{v}(k), \tag{7.2}$$

where $\gamma \in \{0, 0.5, 1\}$. $\boldsymbol{\mathcal{X}}(k)$ and $\boldsymbol{\mathcal{Y}}(k)$ are the respective state vector and measurement vector at time $t(k)$; $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ are given deterministic matrices; and $\mathbf{v}(k), \mathbf{w}(k)$ are uncorrelated zero mean Gaussian random variables with identity covariance matrix. $\boldsymbol{\mathcal{X}}^{\gamma}(k)$ indicates a vector whose each element is the corresponding element of $\boldsymbol{\mathcal{X}}(k)$ raised to the power γ and a positive value is chosen when $\gamma = 0.5$. The time increment $t(k) - t(k-1)$ is assumed constant for all k. Matrix $\mathbf{A}$ in transition equation is perturbed by a random $n \times n$ matrix $\Delta\mathbf{A}(k)$, where the elements $\Delta\mathbf{A}_{ij}(k)$ are uncorrelated stochastic variables with zero mean and variance $\mathbf{P}_{\Delta\mathbf{A}_{ij}}$. This allows for a potentially large number of uncorrelated noise sources, potentially $O(n^2)$, instead of $O(n)$ if the noise is purely additive. Obviously, some of the entries in $\Delta\mathbf{A}(k)$ can be identically zero. $\Delta\mathbf{A}(k)$ and the noise sources $\mathbf{w}(k)$, $\mathbf{v}(k)$ are assumed to be uncorrelated for all k.

Instead of applying the standard linear estimation theory to the new systems of equations and writing the optimal estimator using Anderson & Moore [1979], as in de Koning [1984], the aim is to derive the equations for a recursive linear minimum variance filter for state $\boldsymbol{\mathcal{X}}(k)$ given the measurements $\boldsymbol{\mathcal{Y}}(k), \boldsymbol{\mathcal{Y}}(k-1), \cdots, \boldsymbol{\mathcal{Y}}(0)$, which will reduce to Kalman filter if $\mathbf{P}_{\Delta\mathbf{A}}$ is 0. As in the case of any recursive estimation algorithm, start by assuming that $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ is known. The predicted mean, i.e. $\hat{\boldsymbol{\mathcal{X}}}(k+1|k)$, and the updated mean, $\hat{\boldsymbol{\mathcal{X}}}(k+1|k+1)$, are written in the same form as the Kalman filter. Then the covariance at the time step $t(k+1)$ is worked out using equation (7.1). The standard prediction and update equations for a linear filter and the system described by (7.1) are

$$\hat{\boldsymbol{\mathcal{X}}}(k+1|k) = \mathbf{A}\hat{\boldsymbol{\mathcal{X}}}(k|k), \tag{7.3}$$

$$\hat{\boldsymbol{\mathcal{X}}}(k+1|k+1) = \hat{\boldsymbol{\mathcal{X}}}(k+1|k) + \bar{\mathbf{K}}(k+1)(\boldsymbol{\mathcal{Y}}(k+1) - \mathbf{C}\hat{\boldsymbol{\mathcal{X}}}(k+1|k)). \tag{7.4}$$

The aim is to find a filter gain $\bar{\boldsymbol{K}}(k+1)$ that would minimize the state covariance, which will be denoted by $\bar{\mathbf{P}}(k+1|k+1)$. The equations (7.3) and (7.4) are combined and the fact that $\Delta\mathbf{A}(k)$

and $\boldsymbol{\mathcal{X}}(k)$ are uncorrelated due to the former being a sequence of uncorrelated stochastic variables is used. Then the expression for the covariance can be easily shown to be

$$\begin{aligned}\bar{\mathbf{P}}(k+1|k+1) = \bar{\boldsymbol{K}}(k+1)\mathbf{D}\mathbf{D}^T\bar{\boldsymbol{K}}(k+1)+ \\ (\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C})\bar{\mathbf{P}}(k+1|k)(\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C})^T,\end{aligned} \tag{7.5}$$

where

$$\bar{\mathbf{P}}(k+1|k) = \mathbf{A}\bar{\mathbf{P}}(k|k)\mathbf{A}^T + \mathbf{B}\mathbf{B}^T + \widetilde{\mathbf{P}(k|k)} \tag{7.6}$$

and $\widetilde{\mathbf{P}(k|k)}$ is a diagonal $n \times n$ matrix that is different for each value of $\gamma \in \{0, 0.5, 1\}$:

$$\begin{aligned}\widetilde{\mathbf{P}_{ii}(k|k)} &= \sum_{j=1}^{n} \mathbf{P}_{\Delta\mathbf{A}_{ij}} \left(\bar{\mathbf{P}}_{jj}(k|k) + (\hat{\boldsymbol{\mathcal{X}}}_j(k|k))^2\right) \text{ if } \gamma = 1, \\ &= \sum_{j=1}^{n} \mathbf{P}_{\Delta\mathbf{A}_{ij}} \hat{\boldsymbol{\mathcal{X}}}_j(k|k) \text{ if } \gamma = 0.5, \\ &= \sum_{j=1}^{n} \mathbf{P}_{\Delta\mathbf{A}_{ij}} \text{ if } \gamma = 0.\end{aligned} \tag{7.7}$$

Here $\hat{\boldsymbol{\mathcal{X}}}_j(k|k)$ is the j^{th} element of vector $\hat{\boldsymbol{\mathcal{X}}}(k|k)$. It is necessary to find $\bar{\boldsymbol{K}}(k+1)$ that would minimize trace of the covariance, i.e. $\text{tr}\,\bar{\mathbf{P}}(k+1|k+1)$. Now, the partial derivative of trace with respect to the filter gain matrix is

$$\frac{\partial\,\text{tr}\,\bar{\mathbf{P}}(k+1|k+1)}{\partial\bar{\boldsymbol{K}}(k+1)} = -2\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T + 2\bar{\boldsymbol{K}}(k+1)(\mathbf{C}\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T + \mathbf{D}\mathbf{D}^T). \tag{7.8}$$

Setting this partial derivative to zero leads to the following expression for $\bar{\boldsymbol{K}}(k+1)$:

$$\bar{\boldsymbol{K}}(k+1) = \bar{\mathbf{P}}(k+1|k)\mathbf{C}^T[\mathbf{C}\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T + \mathbf{D}\mathbf{D}^T]^{-1}. \tag{7.9}$$

It can be verified that this is indeed a minimum by examining the Hessian of the $\text{tr}\,\bar{\mathbf{P}}(k+1|k+1)$, please see Gelb [1986] for details of differentiation of a scalar function of a matrix. Given $\hat{\boldsymbol{\mathcal{X}}}(k|k), \bar{\mathbf{P}}(k|k)$ are specified, the sequence of operations specified by equations (7.7), (7.6), (7.9), (7.5), (7.3) and (7.4) (in this order) completely define the recursive filter, with equation (7.7) determining a $\gamma-$dependent term.

Comparing this equation to that of the unperturbed Kalman filter, note that there is an extra term, namely $\widetilde{\mathbf{P}(k|k)}$, in the expression for $\bar{\mathbf{P}}(k+1|k)$ and this is due to the random parameter perturbation

in equation (7.1). If $\Delta\mathbf{A}(k)$ was zero for all k, the expression for the predicted covariance $\bar{\mathbf{P}}(k+1|k)$ would be the same as one in the original Kalman filter equations for an unperturbed state space system (with $\Delta\mathbf{A}(k) = 0$ in (7.1)). The same would happen to the equations for the updated covariance $\bar{\mathbf{P}}(k+1|k+1)$ and the filter gain $\bar{\boldsymbol{K}}(k+1)$. Equation (7.7) indicates why three specific values of γ are used, since any other values of γ would mean having to compute other moments of $\boldsymbol{\mathcal{X}}(k|k)$ recursively. In section 7.5, an approximate filtering heuristic will be considered, that is inspired by the exact method for the three values of γ proposed here, which appears to work well for certain other values of γ. The model structure employed here is still quite flexible and includes traditional linear state space systems ($\gamma = 0$), Cox-Ingersoll-Ross type models used in econometrics Cox *et al.* [1985] ($\gamma = 0.5$) and multiplicative noise models ($\gamma = 1$) that are sometimes useful in digital control systems.

7.4 Stability and sensitivity

Start by rewriting the equation (7.4) in terms of $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ for the unperturbed case:

$$\hat{\boldsymbol{\mathcal{X}}}(k+1|k+1) \quad = \quad (\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{A}\hat{\boldsymbol{\mathcal{X}}}(k|k) + \mathbf{K}(k+1)\boldsymbol{\mathcal{Y}}(k+1). \tag{7.10}$$

It is known that the above system is stable in the case of unperturbed Kalman filter ((7.10), as mentioned before), i.e. all the eigenvalues of $(\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{A}$ are inside the unit circle if the pair ($\mathbf{C}$, $\mathbf{A}$) is observable, see Simon [2006] and references therein. As seen in the previous section, the filter gain of the perturbed Kalman filter differs from the unperturbed Kalman filter by an additive factor which depends on the variance of perturbations. It is of interest to see what happens to the stability of this matrix, which maps $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ to $\hat{\boldsymbol{\mathcal{X}}}(k+1|k+1)$, under parameter perturbations.
In this section, the eigenvalues of the matrix $(\mathbf{I}-\mathbf{K}(k+1)\mathbf{C})\mathbf{A}$ are considered, with $\mathbf{K}(k+1)$ from the traditional (or unperturbed) Kalman filter and compare it with the eigenvalues of $(\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A}$, with $\bar{\boldsymbol{K}}(k+1)$ from the perturbed Kalman filter, introduced above. The scalar measurement case is presented first and a more general result is given in the further subsection.

7.4.1 Scalar measurement case

A case when the measurement equation (7.1) is scalar is considered, i.e. $\mathbf{C}$ is a $1 \times n$ matrix and $\mathbf{D}$ is a scalar, whereas $\mathbf{A}$ is $n \times n$ matrix. To start with, write down the equations for the Kalman gain and the predicted covariance for the unperturbed Kalman filter for the system in (7.1) with $\Delta\mathbf{A}(k) = 0$ for all k:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^T[\mathbf{C}\mathbf{P}(k+1|k)\mathbf{C}^T + \mathbf{D}\mathbf{D}^T]^{-1},$$
$$\mathbf{P}(k+1|k) = \mathbf{A}\mathbf{P}(k|k)\mathbf{A}^T + \mathbf{B}\mathbf{B}^T,$$

and similar expressions for the perturbed Kalman filter from the previous section are repeated for convenience:

$$\bar{\boldsymbol{K}}(k+1) = \bar{\mathbf{P}}(k+1|k)\mathbf{C}^T[\mathbf{C}\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T + \mathbf{D}\mathbf{D}^T]^{-1}, \tag{7.11}$$
$$\bar{\mathbf{P}}(k+1|k) = \mathbf{A}\bar{\mathbf{P}}(k|k)\mathbf{A}^T + \mathbf{B}\mathbf{B}^T + \widetilde{\bar{\mathbf{P}}(k|k)}. \tag{7.12}$$

Suppose that both filters start with the same initial mean and covariance, $\hat{\boldsymbol{\mathcal{X}}}(0|0)$ and $\mathbf{P}(0|0)$. The idea is to keep track of the differences between $\bar{\mathbf{P}}(k+1|k)$ and $\mathbf{P}(k+1|k)$, so that at each time step $t(k)$ it is possible to express eigenvalues of the perturbed KF in terms of the ones from the unperturbed KF.

Defining

$$\mathbf{S}(k+1) = \mathbf{C}\mathbf{P}(k+1|k)\mathbf{C}^T + \mathbf{D}\mathbf{D}^T, \tag{7.13}$$

in case of the unperturbed Kalman filter

$$(\mathbf{I} - \mathbf{K}(k+1)\mathbf{C})\mathbf{A} = (\mathbf{I} - \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C})\mathbf{A}. \tag{7.14}$$

The following result gives an exact expression for $\bar{\mathbf{P}}(k|k)$ in terms of $\mathbf{P}(k|k)$, which proves to be useful in establishing the necessary relationship in perturbed and unperturbed eigenvalues:

Proposition 7.1:

$$\bar{\mathbf{P}}(k|k) = \mathbf{P}(k|k) + \Delta\bar{\mathbf{P}}(k|k), \tag{7.15}$$

where the recursion for $\Delta\bar{\mathbf{P}}(k|k)$ is as follows:

$$\begin{aligned}
&\Delta\bar{\mathbf{P}}(0|0) = 0, \\
&\Delta\bar{\mathbf{P}}(k|k) = \Delta\bar{\mathbf{P}}(k|k-1) + \mathbf{P}(k|k-1)\mathbf{C}^T\mathbf{S}(k)^{-1}\mathbf{C}\mathbf{P}(k|k-1) \\
&- \left(\mathbf{P}(k|k-1) + \Delta\bar{\mathbf{P}}(k|k-1)\right)\phi(k)\mathbf{C}\left(\mathbf{P}(k|k-1) + \Delta\bar{\mathbf{P}}(k|k-1)\right),
\end{aligned}$$

and where

$$\begin{aligned}
\bar{\mathbf{P}}(k|k-1) &= \mathbf{P}(k|k-1) + \Delta\bar{\mathbf{P}}(k|k-1), \\
\Delta\bar{\mathbf{P}}(k|k-1) &= \widetilde{\bar{\mathbf{P}}(k-1|k-1)} + \mathbf{A}\Delta\bar{\mathbf{P}}(k-1|k-1)\mathbf{A}^T.
\end{aligned} \tag{7.16}$$

Finally, $\phi(k) = \mathbf{C}^T\mathbf{S}(k)^{-1}[\mathbf{I} + \mathbf{C}\Delta\bar{\mathbf{P}}(k|k-1)\mathbf{C}^T\mathbf{S}(k)^{-1}]^{-1}$.

Proof: This proposition will be proved using a simple mathematical induction argument. Suppose that both filters have the same initial covariance $\mathbf{P}(0|0) = \bar{\mathbf{P}}(0|0)$. Also let $\Delta\bar{\mathbf{P}}(0|0) = 0$, so that the proposition holds for $k = 0$. Assume that (7.15) holds for some $k = m$ with $m \geq 0$, i.e.

$$\bar{\mathbf{P}}(m|m) = \mathbf{P}(m|m) + \Delta\bar{\mathbf{P}}(m|m),$$

where $\Delta\bar{\mathbf{P}}(m|m) = \Delta\bar{\mathbf{P}}(m|m-1) + \mathbf{P}(m|m-1)\mathbf{C}^T\mathbf{S}(m)^{-1} - (\mathbf{P}(m|m-1) + \Delta\bar{\mathbf{P}}(m|m-1))\phi(m)$. Considering the expression for $\bar{\mathbf{P}}(m+1|m)$ first

$$\begin{aligned}
\bar{\mathbf{P}}(m+1|m) &= \mathbf{A}\bar{\mathbf{P}}(m|m)\mathbf{A}^T + \mathbf{B}\mathbf{B}^T + \widetilde{\bar{\mathbf{P}}(m|m)} \\
&= \mathbf{A}\mathbf{P}(m|m)\mathbf{A}^T + \mathbf{B}\mathbf{B}^T + \mathbf{A}\Delta\bar{\mathbf{P}}(m|m)\mathbf{A}^T + \widetilde{\bar{\mathbf{P}}(m|m)} \\
&= \mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m).
\end{aligned}$$

Then the expression for $\bar{\mathbf{P}}(m+1|m+1)$ becomes

$$\begin{aligned}
&\bar{\mathbf{P}}(m+1|m+1) = \bar{\mathbf{P}}(m+1|m) - \bar{\mathbf{P}}(m+1|m)\mathbf{C}'[\mathbf{C}\bar{\mathbf{P}}(m+1|m)\mathbf{C}' + DD')^{-1}\mathbf{C}\bar{\mathbf{P}}(m+1|m) \\
&= \mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m) - (\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m))\mathbf{C}^T\mathbf{S}(m+1)^{-1} \\
&* [I + \mathbf{C}\Delta\bar{\mathbf{P}}(m+1|m)\mathbf{C}^T\mathbf{S}(m+1)^{-1}]^{-1}\mathbf{C}(\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m)) \\
&= \mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m) - (\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m))\phi(m+1) \\
&* \mathbf{C}(\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m))
\end{aligned}$$

Rearranging the terms as before, the following expression is obtained

$$\begin{aligned}\bar{\mathbf{P}}(m+1|m+1) &= \mathbf{P}(m+1|m) - \mathbf{P}(m+1|m)\mathbf{C}^T\mathbf{S}(m+1)^{-1}\mathbf{C}\mathbf{P}(m+1|m) \\ &+ \Delta\bar{\mathbf{P}}(m+1|m) + \mathbf{P}(m+1|m)\mathbf{C}^T\mathbf{S}(m+1)^{-1}\mathbf{C}\mathbf{P}(m+1|m) \\ &- (\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m))\phi(m+1)\mathbf{C}(\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m)) \\ &= \mathbf{P}(m+1|m+1) + \Delta\bar{\mathbf{P}}(m+1|m+1),\end{aligned}$$

where

$$\begin{aligned}\Delta\bar{\mathbf{P}}(m+1|m+1) &= \Delta\bar{\mathbf{P}}(m+1|m) + \mathbf{P}(m+1|m)\mathbf{C}^T\mathbf{S}(m+1)^{-1}\mathbf{C}\mathbf{P}(m+1|m) \\ &-(\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m))\phi(m+1)\mathbf{C}(\mathbf{P}(m+1|m) + \Delta\bar{\mathbf{P}}(m+1|m))\end{aligned}$$

as required.

Hence by induction *Proposition* 7.1 has been proven to hold for all $k \geq 0$. ∎

Define

$$\alpha(k+1) = \mathbf{C}\Delta\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}. \tag{7.17}$$

Given the dimensions of $\mathbf{C}$, $\mathbf{S}(k+1)$ is a positive scalar and hence $\alpha(k+1)$ is a scalar as well. As the covariance of the perturbation matrix $\Delta\mathbf{A}$, i.e. $\mathbf{P}_{\Delta\mathbf{A}} \to 0$ and $\alpha(k+1) \to 0$, $\Delta\bar{\mathbf{P}}(k|k-1) \to 0$ and $\Delta\bar{\mathbf{P}}(k|k) \to 0$. Hence $\bar{\mathbf{P}}(k+1|k) \to \mathbf{P}(k+1|k)$. Using this relationship, the following expression is obtained for the perturbed case for small enough perturbation, i.e. for $0 < \alpha(k+1) < 1$.

Proposition 7.2:

$$(\mathbf{I} - \bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A} = \left(\mathbf{I} - \frac{(\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k))}{(1+\alpha(k+1))}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right)\mathbf{A}. \tag{7.18}$$

Proof: From equation (7.16) the following relationship between $\bar{\mathbf{P}}(k+1|k)$ and $\mathbf{P}(k+1|k)$ exists:

$$\bar{\mathbf{P}}(k+1|k) \quad = \quad \mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k).$$

Here $\Delta\bar{\mathbf{P}}(k+1|k) = \widetilde{\bar{\mathbf{P}}(k|k)} + \mathbf{A}\Delta\bar{\mathbf{P}}(k|k)\mathbf{A}^T$ and the recursion for $\bar{\mathbf{P}}(k|k)$ is as in *Proposition* 7.1.

Rewriting equation (7.11) and using definition of $\mathbf{S}(k+1)$ from (7.13),

$$\begin{aligned}\bar{\boldsymbol{K}}(k+1) &= (\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k)) * \mathbf{C}^T(\mathbf{S}(k+1) + \mathbf{C}\Delta\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T)^{-1} \\ &= (\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k))\phi(k+1) \\ &= \frac{(\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k))\mathbf{C}^T\mathbf{S}(k+1)^{-1}}{1+\alpha(k+1)}.\end{aligned}$$

For small enough perturbation, i.e. $0 < \alpha(k+1) < 1$, one can expand $(1+\alpha(k+1))^{-1}$ as a power series and then write:

$$\begin{aligned}\bar{\boldsymbol{K}}(k+1) &= \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\left(1 - \frac{\alpha(k+1)}{1+\alpha(k+1)}\right) + \Delta\bar{\mathbf{P}}(k+1|k)\phi(k+1) \\ &= \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1} + \Delta\bar{\mathbf{P}}(k+1|k)\phi(k+1) - \alpha(k+1)\mathbf{P}(k+1|k)\phi(k+1) \\ &= \mathbf{K}(k+1) + \Delta\bar{\boldsymbol{K}}(k+1),\end{aligned}$$

where the perturbation in the Kalman gain, to account for the random perturbation in the model parameters, is given by

$$\Delta\bar{\boldsymbol{K}}(k+1) = \frac{\Delta\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}}{1+\alpha(k+1)} - \alpha(k+1)\frac{\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}}{1+\alpha(k+1)}.$$

Rearranging one can get the desired result:

$$\begin{aligned}(\mathbf{I} - \bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A} &= (\mathbf{I} - (\mathbf{K}(k+1) + \Delta\bar{\boldsymbol{K}}(k+1))\mathbf{C})\mathbf{A} \\ &= \left(\mathbf{I} - \frac{(\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k))}{(1+\alpha(k+1))}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right)\mathbf{A}. \quad \blacksquare\end{aligned}$$

It is of interest to compare the eigenvalues of the perturbed KF and the eigenvalues of the traditional KF. It is also desirable to try to express the size of the eigenvalue perturbation as a function of the size of parameter perturbation. Suppose that matrix $\mathbf{A}$ has eigenvalues $\mu_1, \mu_2, ..., \mu_n$; $\lambda_1(k+1), \lambda_2(k+1), ..., \lambda_n(k+1)$ and $\bar{\lambda}_1(k+1), \bar{\lambda}_2(k+1), ..., \bar{\lambda}_n(k+1)$ are eigenvalues of (7.14) and (7.18) respectively.

Consider relationship between $\lambda_i(k+1)$ and μ_i first. It is known that $\det(\mathbf{A}) = \prod_{i=1}^n \mu_i$ and

$\text{tr}\,(\mathbf{A}) = \sum_{i=1}^{n} \mu_i$. Considering the determinant and the trace of (7.14) gives the following:

$$\prod_{i=1}^{n} \lambda_i(k+1) = \det\left(\mathbf{I} - \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right) \prod_{i=1}^{n} \mu_i \,, \tag{7.19}$$

$$= \mathbf{D}\mathbf{D}^T\mathbf{S}(k+1)^{-1} \prod_{i=1}^{n} \mu_i \text{ if } n = 2,$$

$$\sum_{i=1}^{n} \lambda_i(k+1) = \text{tr}\,(\mathbf{A}) - \text{tr}\,\left(\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A}\right). \tag{7.20}$$

Considering the determinant and the trace of (7.18), one can obtain the main result for the eigenvalues of the perturbed filter.

Proposition 7.3:

$$(a) \prod_{i=1}^{n} \bar{\lambda}_i(k+1) = \det\left(\mathbf{I} - (\mathbf{P}(k+1|k) + \Delta\bar{\boldsymbol{P}}(k+1|k))\psi(k+1)\right) \prod_{i=1}^{n} \mu_i, \tag{7.21}$$

$$= \frac{\mathbf{D}\mathbf{D}^T\mathbf{S}(k+1)^{-1}}{(1+\alpha(k+1))} \prod_{i=1}^{n} \mu_i \text{ if } n = 2,$$

$$(b) \sum_{i=1}^{n} \bar{\lambda}_i(k+1) = \text{tr}\,(\mathbf{A}) - \text{tr}\,\left((\mathbf{P}(k+1|k) + \Delta\bar{\boldsymbol{P}}(k+1|k))\psi(k+1)\mathbf{A}\right), \tag{7.22}$$

where $\psi(k+1) = \frac{\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}}{1+\alpha(k+1)}$ and $\alpha(k+1)$ is as in (7.17).

Proof: To prove equation (a), start with the determinant of (7.18)

$$\det\left(\mathbf{I} - \bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\mathbf{A} = \det\left(\mathbf{A}\right) \det\left(\mathbf{I} - \frac{(\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k))}{(1+\alpha(k+1))}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right).$$

When $n = 2$, one can use the fact that for any matrix $\mathbf{B}$, $\det(\mathbf{I}-\mathbf{B}) = 1-\text{tr}(\mathbf{B})+\det(\mathbf{B})$. Remember that $\alpha(k+1) = \mathbf{C}\Delta\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}$. Also note that $\det\left(\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right) = 0$ and from the definition of $\mathbf{S}(k+1)$ in (7.13)one has $\text{tr}\,(\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}) = 1 - \mathbf{D}\mathbf{D}^T\mathbf{S}(k+1)^{-1}$.

This leads to

$$\begin{aligned}
\det\left(\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\mathbf{A} &= \det\left(\mathbf{A}\right)\left(1-\ \mathrm{tr}\ \left(\frac{\left(\mathbf{P}(k+1|k)+\Delta\bar{\mathbf{P}}(k+1|k)\right)}{\left(1+\alpha(k+1)\right)}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right)\right),\\
&= \det\left(\mathbf{A}\right)\left(1-\frac{\mathrm{tr}\left(\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right)}{1+\alpha(k+1)}\right)\\
&- \det\left(\mathbf{A}\right)\left(\frac{\mathrm{tr}\left(\mathbf{C}\Delta\bar{\mathbf{P}}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\right)}{1+\alpha(k+1)}\right),\\
&= \det\left(\mathbf{A}\right)\left(1-\frac{1-\mathbf{D}\mathbf{D}^T\mathbf{S}(k+1)^{-1}}{1+\alpha(k+1)}-\frac{\alpha(k+1)}{1+\alpha(k+1)}\right),\\
&= \det\left(\mathbf{A}\right)\frac{\mathbf{D}\mathbf{D}^T\mathbf{S}(k+1)^{-1}}{\left(1+\alpha(k+1)\right)}.
\end{aligned}$$

To prove equation (b), the trace of (7.18) is taken

$$\mathrm{tr}\left(\left(\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\mathbf{A}\right)=\mathrm{tr}(\mathbf{A})-\mathrm{tr}\left(\mathbf{I}-\frac{\left(\mathbf{P}(k+1|k)+\Delta\bar{\mathbf{P}}(k+1|k)\right)}{\left(1+\alpha(k+1)\right)}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right),$$

which leads to the required result. ∎

Note that (7.21) and (7.22) are equalities and *not* upper nor lower bounds. Consider (7.19) and (7.21):

$$\begin{aligned}
\prod_{i=1}^{n}\lambda_i(k+1) &= \det\ \left(\mathbf{I}-\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right)\prod_{i=1}^{n}\mu_i,\\
\prod_{i=1}^{n}\bar{\lambda}_i(k+1) &= \det\ \left(\mathbf{I}-\frac{\mathbf{P}(k+1|k)+\Delta\bar{\boldsymbol{P}}(k+1|k)}{1+\alpha(k+1)}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\right)\prod_{i=1}^{n}\mu_i.
\end{aligned}$$

Defining $\widetilde{\mathbf{P}}(k+1|k)=\frac{\left(\mathbf{P}(k+1|k)+\Delta\bar{\mathbf{P}}(k+1|k)\right)}{\left(1+\alpha(k+1)\right)}$ for simplicity, note as $\Delta\bar{\mathbf{P}}(k+1|k)\rightarrow 0$, one has $\alpha(k+1)\rightarrow 0$, $\widetilde{\mathbf{P}}(k+1|k)\rightarrow\mathbf{P}(k+1|k)$ and $\mathrm{tr}\left(\widetilde{\mathbf{P}}(k+1|k)\right)\rightarrow\mathrm{tr}\left(\mathbf{P}(k+1|k)\right)$. Hence $\prod_{i=1}^{n}\bar{\lambda}_i(k+1)$ is bounded in the same way as $\prod_{i=1}^{n}\lambda_i(k+1)$. Now consider (7.20) and (7.22):

$$\begin{aligned}
\sum_{i=1}^{n}\lambda_i(k+1) &= \mathrm{tr}\left(\mathbf{A}\right)-\ \mathrm{tr}\ \left(\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A}\right),\\
\sum_{i=1}^{n}\bar{\lambda}_i(k+1) &= \mathrm{tr}\left(\mathbf{A}\right)-\ \mathrm{tr}\ \left(\frac{\mathbf{P}(k+1|k)+\Delta\bar{\mathbf{P}}(k+1|k)}{\left(1+\alpha(k+1)\right)}\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A}\right).
\end{aligned}$$

Observe also that covariances $\mathbf{P}(k+1|k)$ and $\widetilde{\mathbf{P}}(k+1|k)$ are both symmetric and positive semidef-

inite. Hence

$$\gamma_{min}\,\mathrm{tr}\,(\mathbf{P}(k+1|k)) \leq \mathrm{tr}\,(\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A}) \leq \gamma_{max}\,\mathrm{tr}\,(\mathbf{P}(k+1|k)),$$
$$\gamma_{min}\,\mathrm{tr}\,(\widetilde{\mathbf{P}}(k+1|k)) \leq \mathrm{tr}\,(\widetilde{\mathbf{P}}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A}) \leq \gamma_{max}\,\mathrm{tr}\,(\widetilde{\mathbf{P}}(k+1|k)),$$

where γ_{min} and γ_{max} are respectively the smallest and the largest eigenvalues of the matrix $\frac{1}{2}(\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A} + (\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}\mathbf{A})^T)$, see Mori [1988] for more details. Thus $\sum_{i=1}^{n}\bar{\lambda}_i(k+1)$ is bounded in the same way as $\sum_{i=1}^{n}\lambda_i(k+1)$. For the case $n=2$, $\bar{\lambda}_i$ are fully defined by their sum and product because of the characteristic equation $\bar{\lambda}^2 - (\bar{\lambda}_1+\bar{\lambda}_2)\bar{\lambda} + \bar{\lambda}_1\bar{\lambda}_2 = 0$. Therefore, it is possible to conclude that for a small enough perturbation, $\bar{\lambda}_i$ will not differ by too much from λ_i. Hence it is clear that the perturbed Kalman filter will have a transient behaviour very similar to an unperturbed Kalman filter provided the random perturbations are small. Even though this equality result is only for scalar systems with two (or less) eigenvalues, it gives a useful qualitative insight into the impact of perturbations on the transient behaviour of the filter.

7.4.2 Multivariate measurement case

Note that *Proposition* 7.2 and *Proposition* 7.3 can be both extended to cover the case when $\mathbf{C}$ is a $n \times n$ matrix and $\mathbf{D}$ is a $n \times 1$ vector. In this case $\alpha(k+1)$ given in (7.17) is a matrix. Provided that $|||\alpha(k+1)||| < 1$, where $|||*|||$ is a matrix norm, from Horn & Johnson [1985]:

$$(\mathbf{I}+\alpha(k+1))^{-1} = \sum_{l=0}^{\infty}(-\alpha(k+1))^l.$$

In this case *Proposition* 7.2 becomes:

Proposition 7.4:

$$(\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A} = \left(\mathbf{I}-\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C} - \Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\mathbf{A}, \tag{7.23}$$

where

$$\Delta\bar{\boldsymbol{K}}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\sum_{l=1}^{\infty}(-\alpha(k+1))^l + \Delta\bar{\mathbf{P}}(k+1|k)\phi(k+1)$$

and $\phi(k+1)$ is as in *Proposition* 7.1.

Proof: Using power series for $\alpha(k+1)$ it is possible to rewrite $\bar{\boldsymbol{K}}(k+1)$ as:

$$\begin{aligned}\bar{\boldsymbol{K}}(k+1) &= (\mathbf{P}(k+1|k) + \Delta\bar{\mathbf{P}}(k+1|k))\phi(k+1) \\ &= \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\left(\mathbf{I} + \sum_{l=1}^{\infty}(-\alpha(k+1))^l\right) + \Delta\bar{\mathbf{P}}(k+1|k)\phi(k+1) \\ &= \mathbf{K}(k+1) + \Delta\bar{\boldsymbol{K}}(k+1),\end{aligned}$$

where

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}$$

and

$$\Delta\bar{\boldsymbol{K}}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\sum_{l=1}^{\infty}(-\alpha(k+1))^l + \Delta\bar{\mathbf{P}}(k+1|k)\phi(k+1). \quad (7.24)$$

Then using $\Delta\bar{\boldsymbol{K}}(k+1)$ it is possible to obtain

$$\begin{aligned}(\mathbf{I} - \bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A} &= \left(\mathbf{I} - (\mathbf{K}(k+1) + \Delta\bar{\boldsymbol{K}}(k+1))\mathbf{C}\right)\mathbf{A}, \\ &= \left(\mathbf{I} - \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C} - \Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\mathbf{A},\end{aligned}$$

as required. ∎

Taking the trace and determinant of (7.23) will provide with the sum and product of eigenvalues $\bar{\lambda}_i$ for the general case.

Proposition 7.5:

$$(a)\prod_{i=1}^{n}\bar{\lambda}_i(k+1) = \det\left(\mathbf{I} - \mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C} - \Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\prod_{i=1}^{n}\mu_i, \quad (7.25)$$

$$(b)\sum_{i=1}^{n}\bar{\lambda}_i(k+1) = \mathrm{tr}\,(\mathbf{A}) - \mathrm{tr}\left((\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C} + \Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A}\right). \quad (7.26)$$

Proof: In order to proof (a) one can start by taking the determinant of (7.23) and use expression in *Proposition* 7.4:

$$\begin{aligned}\prod_{i=1}^{n}\bar{\lambda}_i(k+1) &= \det\left((\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A}\right)\\ &= \det\left(\mathbf{I}-\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}-\Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\det(\mathbf{A})\\ &= \det\left(\mathbf{I}-\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}-\Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C}\right)\prod_{i=1}^{n}\mu_i.\end{aligned}$$

For the proof of (b) one can take the trace of (7.23) to obtain the required result:

$$\begin{aligned}\sum_{i=1}^{n}\bar{\lambda}_i(k+1) &= \text{tr}\left((\mathbf{I}-\bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A}\right)\\ &= \text{tr}(\mathbf{A})-\text{tr}\left((\mathbf{P}(k+1|k)\mathbf{C}^T\mathbf{S}(k+1)^{-1}\mathbf{C}+\Delta\bar{\boldsymbol{K}}(k+1)\mathbf{C})\mathbf{A}\right).\end{aligned}$$

■

Note as $\Delta\bar{\mathbf{P}}(k+1|k)\rightarrow 0$, one has $\alpha(k+1)\rightarrow 0$ and hence $\Delta\bar{\boldsymbol{K}}(k+1)\rightarrow 0$. Comparing (7.25) with (7.19) and (7.26) with (7.20), it can be observed that $\prod_{i=1}^{n}\bar{\lambda}_i(k+1)$ is bounded in the same way as $\prod_{i=1}^{n}\lambda_i(k+1)$ and $\sum_{i=1}^{n}\bar{\lambda}_i(k+1)$ is bounded in the same way as $\sum_{i=1}^{n}\lambda_i(k+1)$. Therefore, one can conclude that for a small enough perturbation, $\bar{\lambda}_i$ won't differ by too much from λ_i in general case as well.

7.5 Approximate filtering for $\gamma \geq 1.5$

In this section a univariate state-space system in considered as in (7.1) with $\gamma=\frac{l}{2}$ with $l=3,4,....$ These values of γ come up in CEV type models, see Cox [1996] for more details. In this particular case, $\mathbf{P}(\widetilde{k-1|k-1})=\mathbf{P}_{\Delta\mathbf{A}}\mathbb{E}(\boldsymbol{\mathcal{X}}^{2\gamma}(k-1))$, where positive values of $\boldsymbol{\mathcal{X}}$ are chosen whenever there are two roots. Hence in order to find the covariance $\bar{\mathbf{P}}(k|k-1)$, optimal gain $\bar{\boldsymbol{K}}(k)$ and the updated state $\hat{\mathcal{X}}(k|k)$ at time step $t(k)$, it is necessary to know $\mathbb{E}(\boldsymbol{\mathcal{X}}^{2\gamma}(k-1))$ from $t(k-1)$. In general there is no closed form solution for this for $l\geq 3$ as only the first two moments of $\boldsymbol{\mathcal{X}}$ are propagated throughout the filter recursions. However one can carry out moment matching approximation of $\Delta\mathbf{A}(k-1)\boldsymbol{\mathcal{X}}^{\gamma}(k-1)$ by rewriting it as a noise term with the same first two moments as $\mathbf{F}(k-1)\mathbf{u}(k-1)$. Here $\mathbf{u}(k-1)$ is a Gaussian random variable, uncorrelated with the

state and noise terms $\mathbf{w}(k)$ and $\mathbf{v}(k)$, and $\mathbf{F}(k-1)^2 = \mathbf{P}_{\Delta\mathbf{A}}\mathbb{E}\left(\boldsymbol{\mathcal{X}}^{2\gamma}(k-1)\right)$. This way it is possible to match the first two moments of term $\Delta\mathbf{A}(k-1)\boldsymbol{\mathcal{X}}^{\gamma}(k-1)$ exactly, and higher order moments would be proportional to those of $\mathbf{u}(k-1)$, in particular, the odd moments would be zero.
Using (7.1) it is possible to rewrite $\boldsymbol{\mathcal{X}}(k) - \hat{\boldsymbol{\mathcal{X}}}(k|k)$ as an expression depending on the state, Δ_s, and an expression consisting solely of noise terms, Δ_n :
Proposition 7.6

$$\boldsymbol{\mathcal{X}}(k) - \hat{\boldsymbol{\mathcal{X}}}(k|k) = \Delta_s + \Delta_n,$$

where $\Delta_s = \left(\mathbf{I} - \bar{\boldsymbol{K}}(k)\mathbf{C}\right)\mathbf{A}\left(\boldsymbol{\mathcal{X}}(k-1) - \hat{\boldsymbol{\mathcal{X}}}(k-1|k-1)\right)$ and
$\Delta_n = \left(\mathbf{I} - \bar{\boldsymbol{K}}(k)\mathbf{C}\right)\left(\mathbf{B}\mathbf{w}(k) + \mathbf{F}(k-1)\mathbf{u}(k-1)\right) - \bar{\boldsymbol{K}}(k)\mathbf{D}\mathbf{v}(k)$.
Proof: In order to prove *Proposition 7.6* start by using (7.1) while rewriting $\Delta\mathbf{A}(k-1)\boldsymbol{\mathcal{X}}^{\gamma}(k-1)$ as $\mathbf{F}(k-1)\mathbf{u}(k-1)$. On expanding of $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ using (7.10) and rearranging the terms:

$$\begin{aligned}
\boldsymbol{\mathcal{X}}(k) - \hat{\boldsymbol{\mathcal{X}}}(k|k) &= \mathbf{A}\boldsymbol{\mathcal{X}}(k-1) + \mathbf{F}(k-1)\mathbf{u}(k-1) + \mathbf{B}\mathbf{w}(k) \\
&\quad - \left(\mathbf{A}\hat{\boldsymbol{\mathcal{X}}}(k-1|k-1) + \bar{\boldsymbol{K}}(k)(\boldsymbol{\mathcal{Y}}(k) - \mathbf{C}\mathbf{A}\hat{\boldsymbol{\mathcal{X}}}(k-1|k-1))\right) \\
&= \left(\mathbf{I} - \bar{\boldsymbol{K}}(k)\mathbf{C}\right)\mathbf{A}\left(\boldsymbol{\mathcal{X}}(k-1) - \hat{\boldsymbol{\mathcal{X}}}(k-1|k-1)\right) \\
&\quad + \left(\mathbf{I} - \bar{\boldsymbol{K}}(k)\mathbf{C}\right)\left(\mathbf{B}\mathbf{w}(k) + \mathbf{F}(k-1)\mathbf{u}(k-1)\right) - \bar{\boldsymbol{K}}(k)\mathbf{D}\mathbf{v}(k). \qquad (7.27)
\end{aligned}$$

■

Since $\mathbf{u}(k), \mathbf{w}(k)$ and $\mathbf{v}(k)$ are zero mean uncorrelated Gaussian random variables, $\mathbb{E}(w^j(k)) = \mathbb{E}(v^j(k)) = \mathbb{E}(u^j(k)) = 0$ hold for any odd integer j, thus $\mathbb{E}(\Delta_n)^j = 0$. In the case of $l = 3$ and $\gamma = 1.5$, one can raise equation (7.27) to the power 3, and take expectations of both sides. This would allow to find the expression for $\mathbb{E}(\boldsymbol{\mathcal{X}}(k))^3$:
Proposition 7.7

$$\begin{aligned}
\mathbb{E}(\boldsymbol{\mathcal{X}}^3(k)) &= 3\mathbf{P}(k|k)\hat{\boldsymbol{\mathcal{X}}}(k|k) + \hat{\boldsymbol{\mathcal{X}}}^3(k|k) + (\mathbf{I} - \bar{\boldsymbol{K}}(k)\mathbf{C})^3\mathbf{A}^3 * \\
&\quad * \left(\mathbb{E}(\boldsymbol{\mathcal{X}}^3(k-1)) - 3\mathbf{P}(k-1|k-1)\hat{\boldsymbol{\mathcal{X}}}(k-1|k-1) - \hat{\boldsymbol{\mathcal{X}}}^3(k-1|k-1)\right).
\end{aligned}$$

Proof: Taking expectations of both sides of (7.27) raised to the power 3:

$$\begin{aligned}\mathbb{E}\left(\boldsymbol{\mathcal{X}}(k)-\hat{\boldsymbol{\mathcal{X}}}(k|k)\right)^3 &= \mathbb{E}(\Delta_s+\Delta_n)^3\\ &= \mathbb{E}(\Delta_s^3)+3\mathbb{E}(\Delta_s^2\Delta_n)+3\mathbb{E}(\Delta_s\Delta_n^2)+\mathbb{E}(\Delta_n^3)\\ &= \mathbb{E}(\Delta_s^3)\\ &= \mathbb{E}\left((\mathbf{I}-\bar{\mathbf{K}}(k)\mathbf{C})\mathbf{A}\left(\boldsymbol{\mathcal{X}}(k-1)-\hat{\boldsymbol{\mathcal{X}}}(k-1|k-1)\right)\right)^3\\ &= (\mathbf{I}-\bar{\mathbf{K}}(k)\mathbf{C})^3\mathbf{A}^3\mathbb{E}\left(\boldsymbol{\mathcal{X}}(k-1)-\hat{\boldsymbol{\mathcal{X}}}(k-1|k-1)\right)^3. \qquad (7.28)\end{aligned}$$

On expanding both sides of (7.28)

$$\begin{aligned}&\mathbb{E}\left(\boldsymbol{\mathcal{X}}^3(k)-3\boldsymbol{\mathcal{X}}^2(k)\hat{\boldsymbol{\mathcal{X}}}(k|k)+3\boldsymbol{\mathcal{X}}(k)\hat{\boldsymbol{\mathcal{X}}}^2(k|k)-\hat{\boldsymbol{\mathcal{X}}}^3(k|k)\right)\\ &= \mathbb{E}\left(\boldsymbol{\mathcal{X}}^3(k-1)-3\boldsymbol{\mathcal{X}}^2(k-1)\hat{\boldsymbol{\mathcal{X}}}(k-1|k-1)+3\boldsymbol{\mathcal{X}}(k-1)\hat{\boldsymbol{\mathcal{X}}}^2(k-1|k-1)-\hat{\boldsymbol{\mathcal{X}}}^3(k-1|k-1)\right)*\\ &*(\mathbf{I}-\bar{\mathbf{K}}(k)\mathbf{C})^3\mathbf{A}^3. \qquad (7.29)\end{aligned}$$

Rearranging equation (7.29) the required result is obtained for $\mathbb{E}(\boldsymbol{\mathcal{X}}(k))^3$. ∎

In general

$$\mathbb{E}\left(\boldsymbol{\mathcal{X}}(k)-\hat{\boldsymbol{\mathcal{X}}}(k|k)\right)^q = \mathbb{E}(\Delta_s+\Delta_n)^q, \qquad (7.30)$$

for some integer power q. Provided at time step $t(k)$ $\mathbb{E}(\boldsymbol{\mathcal{X}}(k-1))^i$ are known for $i=1,2,...,2\gamma$, one can recursively find $\mathbb{E}(\boldsymbol{\mathcal{X}}(k))^j$ for $j=3,...,2\gamma$ using the following proposition.

Proposition 7.8

$$\begin{aligned}\sum_{i=0}^{q}\binom{q}{i}\mathbb{E}(\boldsymbol{\mathcal{X}}(k))^i(\hat{\boldsymbol{\mathcal{X}}}(k|k))^{q-i} &= \sum_{j=0,2,...,q}\binom{q}{j}\mathbb{E}(\Delta_n)^j\mathbb{E}(\Delta_s)^{q-j} \text{ if q is even,}\\ &= \sum_{j=0,2,...,q-3}\binom{q}{j}\mathbb{E}(\Delta_n)^j\mathbb{E}(\Delta_s)^{q-j} \text{ if q is odd,} \qquad (7.31)\end{aligned}$$

for $q=3,...,2\gamma$.

Proof: Left hand side of equation (7.30) simplifies to

$$\mathbb{E}\left(\boldsymbol{\mathcal{X}}(k)-\hat{\boldsymbol{\mathcal{X}}}(k|k)\right)^q = \sum_{i=0}^{q}\binom{q}{i}\mathbb{E}(\boldsymbol{\mathcal{X}}(k))^i(\hat{\boldsymbol{\mathcal{X}}}(k|k))^{q-i}. \qquad (7.32)$$

When $i \in \{0, 1, 2\}$, expectations $\mathbb{E}(\boldsymbol{\mathcal{X}}(k))^i$ in (7.32) are given by filter update equations in section 7.3. However the problem arises for $i \geq 3$. It has been shown in *Proposition* 7.7 how to find the expectation for $i = 3$. Similarly to find these expectations for $i \geq 3$, one needs to raise $\mathbb{E}\left(\boldsymbol{\mathcal{X}}(k) - \hat{\boldsymbol{\mathcal{X}}}(k|k)\right)$ to the power q, where $q = 3, 4, ..., 2\gamma$. Right hand side of equation (7.30) can be expressed as

$$\mathbb{E}(\Delta_s + \Delta_n)^q = \sum_{j=0}^{q} \binom{q}{j} \mathbb{E}(\Delta_n)^j \mathbb{E}(\Delta_s)^{q-j}. \tag{7.33}$$

It is known that for odd values of j $\mathbb{E}(\Delta_n)^j = 0$. Also $\mathbb{E}(\Delta_s) = 0$ from the definition. Hence equation (7.33) reduces to

$$\begin{aligned} \mathbb{E}(\Delta_s + \Delta_n)^q &= \sum_{j=0,2,...,q} \binom{q}{j} \mathbb{E}(\Delta_n)^j \mathbb{E}(\Delta_s)^{q-j} \text{ if q is even,} \\ &= \sum_{j=0,2,...,q-3} \binom{q}{j} \mathbb{E}(\Delta_n)^j \mathbb{E}(\Delta_s)^{q-j} \text{ if q is odd.} \end{aligned} \tag{7.34}$$

Putting equations (7.32) and (7.34) together will give the required result. ∎

Note that minimum variance derivation of the equations for $\bar{\boldsymbol{K}}(k)$ in (7.9) still holds for these values of γ as well. This filter is denoted as approximate perturbed Kalman filter (APKF) as it is a linear filter providing a solution by approximating the perturbation term by a noise term.

7.6 Numerical examples

Consider two different cases for two allowable values of γ to illustrate and contrast the performance of the exact minimum variance filter, PKF, with that of the Kalman filter. Also included here is one example for the approximate perturbed filter for the discrete univariate system with $\gamma = 1.5$.

7.6.1 Case when $\gamma = 1$

In this section a numerical example is considered with the following state-space equations.

$$\begin{aligned} \boldsymbol{\mathcal{X}}(k+1) &= \mathbf{A}\boldsymbol{\mathcal{X}}(k) + \Delta\mathbf{A}(k)\boldsymbol{\mathcal{X}}^{\gamma}(k) + \mathbf{B}\mathbf{w}(k+1), \\ \boldsymbol{\mathcal{Y}}(k) &= \mathbf{C}\boldsymbol{\mathcal{X}}(k) + \mathbf{D}\mathbf{v}(k), \end{aligned}$$

where $\gamma = 1$,

$$\mathbf{A} = \begin{pmatrix} 0 & -0.5 \\ 1 & 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -6 \\ 1 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} -100 & 10 \end{pmatrix}$$

and $\mathbf{D} = 1$. $\mathbf{v}(k), \mathbf{w}(k)$ are uncorrelated Gaussian random variables. Perturbation matrix $\Delta\mathbf{A}(k)$ has zero mean, the matrix elements have the following covariance matrix

$$\mathbf{P}_{\Delta\mathbf{A}} = \begin{pmatrix} 0.12 & 0.02 \\ 0.15 & 0.1 \end{pmatrix}.$$

Initial conditions are $\boldsymbol{\mathcal{X}}(0) = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$, $\hat{\boldsymbol{\mathcal{X}}}(0|0) = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ and

$$\mathbf{P}(0|0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In order to compare the performance of both the perturbed and the unperturbed Kalman filters , the average root mean square error (AvRMSE) for the state is considered, where RMSE for a sample path l is given by

$$RMSE_l = \sqrt{\frac{1}{2F}\sum_{j=1}^{F}((\boldsymbol{\mathcal{X}}_1(j) - \hat{\boldsymbol{\mathcal{X}}}_1(j))^2 + (\boldsymbol{\mathcal{X}}_2(j) - \hat{\boldsymbol{\mathcal{X}}}_2(j))^2)},$$

and AvRMSE over a given number of sample paths L is defined by $\frac{1}{L}\sum_{l=1}^{L} RMSE_I$. Here $F = 100$ and $L = 100$ are used. Comparison of the variance of RMSE (VAR) for both filters is also shown in tables 7.1-7.3 and is calculated as $\frac{1}{L}\sum_{l=1}^{L}\left(RMSE_l - AvRMSE\right)^2$. Three cases are investigated: when the real covariance of perturbation matrix $\Delta\mathbf{A}$ is known exactly and when two different cases where it is incorrectly estimated. If the model is calibrated from data, it is likely that the parameters are imperfectly known and even the size of uncertainty (in terms of its covariance matrix) is not known exactly. It is of interest to see whether the filter performs well if this is indeed the case. $\widetilde{\mathbf{P}}_{\Delta\mathbf{A}1} = \begin{pmatrix} 0.2 & 0.1 \\ 0.05 & 0.15 \end{pmatrix}$ and $\widetilde{\mathbf{P}}_{\Delta\mathbf{A}2} = \begin{pmatrix} 0.25 & 0.15 \\ 0.05 & 0.2 \end{pmatrix}$ are used, but $\mathbf{P}_{\Delta\mathbf{A}}$ is kept in the equations of PKF (*i.e.* $\mathbf{P}_{\Delta\mathbf{A}}$ is the *incorrect* covariance matrix used when the correct covariances are $\widetilde{\mathbf{P}}_{\Delta\mathbf{A}1}$, $\widetilde{\mathbf{P}}_{\Delta\mathbf{A}2}$). These results are presented in tables 7.2 and 7.3 respectively. A comparison of a simulated path with the paths generated by PKF and KF for $\boldsymbol{\mathcal{X}}(1)$, with $\widetilde{\mathbf{P}}_{\Delta\mathbf{A}1}$ as the real covariance of the perturbation matrix $\Delta\mathbf{A}$, is shown in Fig. 7.1.

Table 7.1. Comparison of AvRMSE when $\mathbf{P}_{\Delta\mathbf{A}}$ is estimated correctly

	KF	PKF	Improvement
AvRMSE	14.750	5.927	59.8%
VAR	54.061	2.285	95.8%

Table 7.2. Comparison of AvRMSE when $\mathbf{P}_{\Delta\mathbf{A}1}$ is not estimated correctly

	KF	PKF	Improvement
AvRMSE	18.775	8.102	56.9%
VAR	472.414	22.214	95.3%

Table 7.3. Comparison of AvRMSE when $\mathbf{P}_{\Delta\mathbf{A}2}$ is not estimated correctly

	KF	PKF	Improvement
AvRMSE	37.557	16.088	57.1%
VAR	1465.857	158.557	89.2%

It can be seen from all the three tables that PKF provides more accurate state estimates in terms of both the measures of error when compared to unperturbed Kalman filter, in both the cases when $\mathbf{P}_{\Delta\mathbf{A}}$ is estimated correctly and when it is not. Note that acknowledging that the model is not precise and accounting for the random parameter uncertainties, makes the filter more robust even to poor estimates of the parameter uncertainty (in terms of variance of $\Delta\mathbf{A}$). This observation is in line with the intuition that unperturbed filter is highly *tuned* to the system parameters and the addition of $\widetilde{\bar{\mathbf{P}}(k|k)}$ provides a *de-tuning* effect, thereby making the filter more robust.

7.6.2 Case when $\gamma = 0.5$

As another example, consider a nonlinear system

$$\begin{aligned} \boldsymbol{\mathcal{X}}(k+1) &= \mathbf{A}\boldsymbol{\mathcal{X}}(k) + \Delta\mathbf{A}(k)\boldsymbol{\mathcal{X}}^{\gamma}(k) + \mathbf{B}\mathbf{w}(k+1), \\ \boldsymbol{\mathcal{Y}}(k) &= \mathbf{C}\boldsymbol{\mathcal{X}}(k) + \mathbf{D}\mathbf{v}(k), \end{aligned}$$

with $\gamma = 0.5$, $\mathbf{A} = 0.9$, $\mathbf{B} = 0.1, \mathbf{C} = 1$ and $D = 0.01$. $\mathbf{v}(k), \mathbf{w}(k)$ are uncorrelated Gaussian random variables. Initial conditions are assumed to be $\boldsymbol{\mathcal{X}}(0) = 0.1$, $\hat{\boldsymbol{\mathcal{X}}}(0|0) = 0$ and $\mathbf{P}(0|0) = 1$. The performance of the new perturbed Kalman filter will be compared with unperturbed Kalman filter. Average root mean square error is calculated for both filters as follows:

$$RMSE = \sqrt{\frac{1}{F}\sum_{j=1}^{F}((\boldsymbol{\mathcal{X}}(j) - \widehat{\boldsymbol{\mathcal{X}}}(j))^2)},$$

$F = 100$ and AvRMSE is calculated over 100 paths, as before. **A**s in the previous example, these errors are also calculated for the case when covariance $\mathbf{P}_{\triangle\mathbf{A}}$ is not estimated correctly. Results are summarized in tables 7.4-7.5.

Table 7.4. Comparison of AvRMSE when $\mathbf{P}_{\triangle\mathbf{A}}$ is estimated correctly

$\mathbf{P}_{\triangle\mathbf{A}}$	KF	PKF	Improvement
0.2	0.009418	0.009131	3.1%
0.3	0.009422	0.008954	5.0%
0.4	0.009661	0.008921	10.1%

Table 7.5. Comparison of AvRMSE when $\mathbf{P}_{\triangle\mathbf{A}}$ is not estimated correctly

True $\mathbf{P}_{\triangle\mathbf{A}}$	Assumed $\mathbf{P}_{\triangle\mathbf{A}}$	KF	PKF	Improvement
0.4	0.2	0.009965	0.009219	7.5%
0.3	0.2	0.009491	0.009052	4.6%
0.2	0.3	0.009355	0.009078	3.0%
0.2	0.4	0.009418	0.009129	2.8%

It can be seen from these tables that perturbed Kalman filter provides better accuracy when compared to unperturbed Kalman filter for different values of the perturbation matrix variance. The improvement is more pronounced when the variance of the uncertainty is larger, as can be expected.

7.6.3 Case when $\gamma = 1.5$

As an example for the APKF, consider a nonlinear system

$$\begin{aligned} \boldsymbol{\mathcal{X}}(k+1) &= \mathbf{A}\boldsymbol{\mathcal{X}}(k) + \Delta\mathbf{A}(k)\boldsymbol{\mathcal{X}}^{\gamma}(k) + \mathbf{B}\mathbf{w}(k+1), \\ \boldsymbol{\mathcal{Y}}(k) &= \mathbf{C}\boldsymbol{\mathcal{X}}(k) + \mathbf{D}\mathbf{v}(k), \end{aligned}$$

with $\gamma = 1.5$, $\mathbf{A} = 0.9$, $\mathbf{B} = 0.1, \mathbf{C} = 1$ and $\mathbf{D} = 0.01$. $\mathbf{v}(k), \mathbf{w}(k)$ are uncorrelated Gaussian random variables. Initial conditions are assumed to be $\boldsymbol{\mathcal{X}}(0) = 0.1$, $\hat{\boldsymbol{\mathcal{X}}}(0|0) = 0$ and $\mathbf{P}(0|0) = 1$, as per example in the previous subsection.

The performance of the new approximate perturbed Kalman filter will be compared with unperturbed Kalman filter. Average root mean square error is calculated for both filters as in the previous subsection. As in the previous example, these errors are also calculated for the case when covariance $\mathbf{P}_{\Delta\mathbf{A}}$ is not estimated correctly. Results are summarized in tables 7.6-7.7.

Table 7.6. Comparison of AvRMSE when $\mathbf{P}_{\Delta\mathbf{A}}$ is estimated correctly

$\mathbf{P}_{\Delta\mathbf{A}}$	KF	APKF	Improvement
0.2	0.009542	0.009204	3.5%
0.3	0.010312	0.009219	10.6%
0.4	0.012105	0.009235	23.7%

Table 7.7. Comparison of AvRMSE when $\mathbf{P}_{\Delta\mathbf{A}}$ is not estimated correctly

True $\mathbf{P}_{\Delta\mathbf{A}}$	Assumed $\mathbf{P}_{\Delta\mathbf{A}}$	KF	APKF	Improvement
0.4	0.2	0.012087	0.009232	23.6%
0.3	0.2	0.011253	0.009329	17.1%
0.2	0.3	0.009527	0.009214	3.2%
0.2	0.4	0.009583	0.009297	3.0%

It can be observed from these tables that approximate perturbed Kalman filter provides better accuracy when compared to unperturbed Kalman filter for different values of the perturbation matrix variance. The improvement is more pronounced when the variance of the uncertainty is larger, as can be expected.

7.7 Summary

In this chapter a class of systems with parameters of the transition equation under stochastic perturbation is considered. These perturbed discrete time systems allow for both multiplicative and additive noise sources. The class of these systems includes square-root affine systems, i.e. Cox-Ingersoll-Ross type models, first introduced in Cox *et al.* [1985], which are frequently used in econometrics and finance literature, and also models that arise in the area of stochastically sampled digital control systems, see de Koning [1980] for more details.

The problem addressed is the design of a filter that minimizes the estimation error variance and reduces to the Kalman filter if the covariance of the perturbation matrix is zero. The recursive filtering equations for this new perturbed Kalman filter have been derived for this class of systems. Comparing these equations to those of the traditional unperturbed Kalman filter, it is observed that there is an extra term in the expression for the filter covariance. This is a diagonal matrix that takes different values depending on γ in the transition equation and is due to the random parameter perturbation.

It is of interest to see what happens to the stability of the matrix, which maps $\hat{\boldsymbol{\mathcal{X}}}(k|k)$ to $\hat{\boldsymbol{\mathcal{X}}}(k+1|k+1)$, under parameter perturbations. The eigenvalues of the matrix $(\mathbf{I}-\mathbf{K}(k+1)\mathbf{C})\mathbf{A}$ are considered, with $\mathbf{K}(k+1)$ from the traditional (or unperturbed) Kalman filter and compare it with the eigenvalues of $(\mathbf{I}-\bar{\mathbf{K}}(k+1)\mathbf{C})\mathbf{A}$, with $\bar{\mathbf{K}}(k+1)$ from the perturbed Kalman filter. The sensitivity of the new filter to the size of the parameter perturbation has also been analyzed in the case of a scalar measurement equation and have provided results for the product and sum of the eigenvalues of the new perturbed filter in terms of the perturbation parameter. It has been deduced that the perturbed Kalman filter will have a transient behaviour very similar to an unperturbed Kalman filter, provided the random perturbations are small. These results have been extended to the multivariate case as well.

An approximate perturbed Kalman filter has also been considered for the special univariate case of $\gamma = \frac{l}{2}$ for any positive integer $l \geq 3$. These particular values of γ come up in CEV type models, described in Cox [1996], and frequently used in economics and finance. The need for this approximate moment matching approach arises since only the first two moments of the state are propagated throughout the linear filter recursion. The approximate perturbed Kalman filter was inspired by the

exact method, proposed earlier in this chapter, and appears to work well for a wider case of nonlinear systems.

Three numerical examples illustrate the improved accuracy achieved by the new filters when compared to the traditional (or unperturbed) Kalman filter. Two types of cases are investigated, when the real covariance of perturbation matrix $\Delta\mathbf{A}$ is known exactly and when it is incorrectly estimated. The improvement is more pronounced when the variance of the uncertainty is larger, as can be expected. Crucially, the examples indicate that the perturbed filter and the approximate perturbed filter perform better than the unperturbed Kalman filter even when the size of uncertainty is poorly identified. This has important implications in cases where the model is calibrated from data.

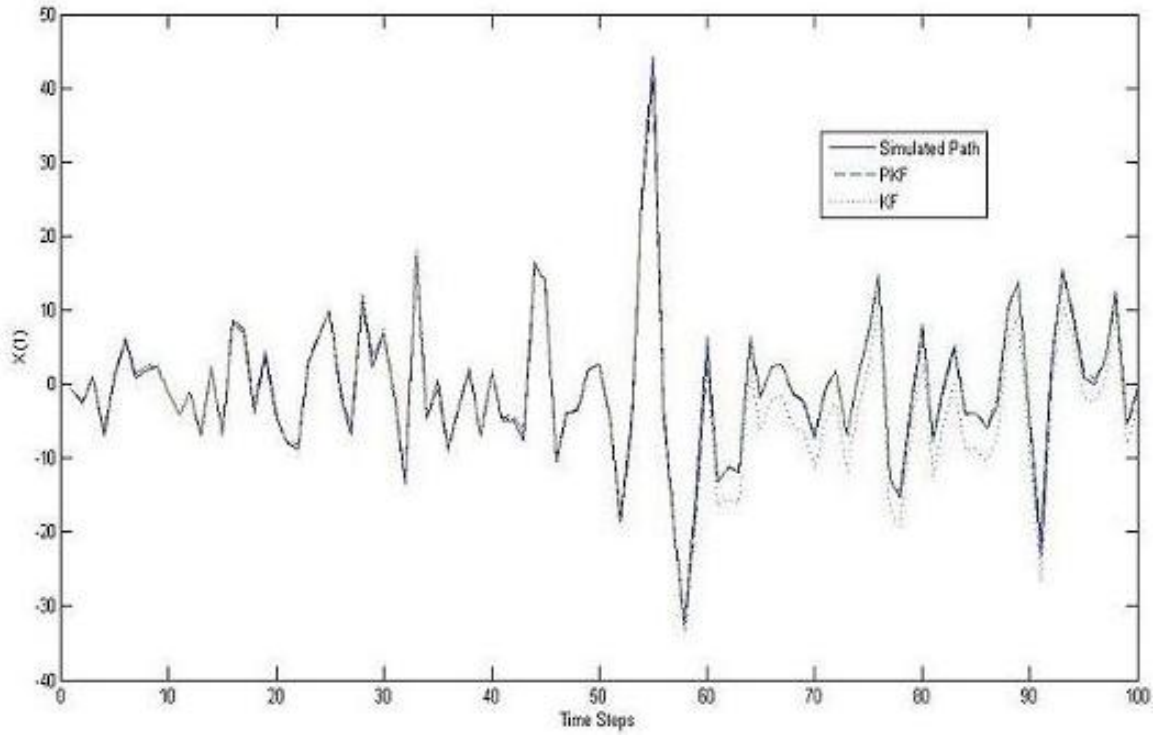


Figure 7.1: Simulated paths

Chapter 8

Conclusions and Summary of Contributions

In this chapter the main contributions of the thesis are summarized with some suggestions for further research.

In chapter 3 a new method has been proposed for generating deterministic sigma points and corresponding probabilities that match the given mean vector, the covariance, the average marginal third and fourth moments exactly, without the use of optimization. The information about higher order moments can give a better idea of the shape of the posterior distribution of the latent state and in particular its partition from assumed Gaussianity in the traditional UKF. The new algorithm for sigma point generation is used as a part of the new filter, HOSPoF. The structure of the deterministic sigma points and corresponding probability weights allows HOSPoF to propagate information about the higher order moments (not just mean and covariance as with the traditional unscented KF) throughout the filter recursions. The utility of the new algorithm has been tested on two examples, one simulation and one based on the empirical financial market data. It has also been shown that the new method improves on predictions of mean and covariance of the state, when compared to the traditional and scaled unscented transformations used in unscented Kalman filters. This improvement is achieved with very little extra computational help, as compared to the traditional UKF.

However, the deterministic generation of sigma points and dependence of the number of points on the

state dimension, inherited from UKF, can be considered too restrictive in a filter. It is believed that adding some randomness to the process of sigma point generation can improve the representation of the required posterior density. This idea has been investigated in chapter 4. A new method for generating random sigma points and corresponding probabilities that match the given mean vector, the covariance, the average marginal third and fourth moments exactly, without the use of optimization, has been introduced. This allows generation of a moment-matching proposal distribution with any number of particles and almost all of the corresponding probabilities are generated randomly. Together with HOSPoF this new algorithm is used as a part of the new filter inspired by the particle filter and is called here PF-HOSPoF. For a particle filter the right choice of a proposal distribution is very important. Two numerical examples, one simulation and the other one based on real financial market data, illustrate the utility of the proposed algorithm in comparison with particle filter with proposals generated by EKF and UKF. PF-HOSPoF has shown to outperform the particle filter with proposal distribution generated by UKF (and EKF) in average improvement for both types of errors and computational times. This can be explained as using unequal random probability weights allows better capturing of the shape of the distribution and offers computational advantage.

The new algorithm for generating random sigma points and corresponding probability weights has been compared with the maximum entropy method in chapter 5. Entropy maximization allows determining of the probability weights that satisfy the set of the moment constraints, which are usually given or can be obtained from a data sample. This method makes use of convex optimization and the probability density achieved is considered to be the most unbiased conditioned upon the available information. Comparison is carried out using relative entropy for four well known probability distributions. The results show that the probability weights generated using the new method, introduced in chapter 4, are very close to the optimal ones in accuracy and have an advantage of not requiring any optimization in the calculations. The other advantage is that sigma points generated using the new method match the mean vector, covariance matrix, the average marginal third and fourth moments exactly. This is often not the case with the points and probability weights achieved using maximum entropy method.

In chapter 6, the new random sigma point generation method, proposed in chapter 4, is considered in the context of financial optimization. The method presents several advantages over the existing

approaches for scenario generation in financial portfolio optimization. First, it is computationally cheap; there is no optimization involved in generating scenarios. Secondly, due the (unequal) generated probabilities of the scenarios, this method may perform well even with a relatively small number of scenarios. In contrast, methods that would assume by default equal probabilities would need a larger number of scenarios. The quality of the proposed scenario generator has been tested in a mean-CVaR portfolio optimization model. Several observations were made.First, the method is remarkably stable, both in-sample and out-of-sample. It also manifests stability of optimal solutions, i.e. not only the optimums are similar, but also the optimal portfolio weights, representing the solution to implement. Secondly, the optimal solutions vary only marginally with increasing number of scenarios. That means, by using only a small number of scenarios, it is possible to obtain a reasonably good solution. In the numerical case presented here, direct comparison with solutions obtained via using historical data could be made. It has been noted that the proposed scenario generation method is applicable to cases where there is no (or not enough) historical data available, but only expert opinion on the statistical properties involved. This method may also present a big advantage when used with a computationally difficult optimization model, requiring only a limited number of scenarios; stable and good quality solutions can be obtained with a relatively small number of scenarios.

An independent derivation of a closed-form minimum variance filter for discrete systems with stochastic uncertainties in state parameters has been provided in chapter 7. If the model is calibrated from data, it is likely that the parameters are imperfectly known and even the size of uncertainty (in terms of its covariance matrix) is not known exactly. Analysis of the sensitivity of the new filter to the size of the parameter perturbation (in terms of its covariance matrix) in the case of a scalar measurement equation has been presented and results for the product and sum of the eigenvalues of the new perturbed filter in terms of the perturbation parameter have been provided. These results have also been extended to a general case for any dimension of the measurement vector. A new way of approximate moment matching in the univariate case for $\gamma = \frac{l}{2}$ for any integer $l \geq 3$ has also been introduced in this chapter. This approach has been expired by the exact method and the need for it arises from parameter perturbations occurring in the CEV models. Filter, based on the approximate moment matching method, is called the approximate perturbed Kalman filter. The performance of the new

perturbed Kalman filter and the approximate perturbed Kalman filter has been tested on three numerical examples. Results illustrate the improved accuracy achieved by the new filters when compared to traditional(or unperturbed) Kalman filter. Importantly, the examples indicate that the perturbed filter and the approximate filter perform better than the unperturbed Kalman filter even when the size of uncertainty is poorly identified. Note that acknowledging that the model is not precise and accounting for the random parameter uncertainties, makes the filter more robust even to poor estimates of the parameter uncertainty (in terms of variance of $\Delta\mathbf{A}$). This has important implications in cases where the model is calibrated from data.

Ideas for future research

Besides a good choice of proposal, another important issue for a successful particle filter is maintaining the diversity of the particles and avoiding the sample impoverishment. The recent trend is to use particles with high enough importance weight to generate the new particles, instead of copying. The new method for generating random sigma points and corresponding probabilities, proposed in chapter 4, can be used in the resampling step in order to enrich the representation of the posterior distribution. In chapter 5, the comparison was done for the univariate data between the maximum entropy approach and the new method, introduced in chapter 4. It might be of interest to extend the comparison of these two methods to the multivariate case and increase the number of sigma points involved. It might also be of interest to use the new method for generating univariate random variables with specified moments matched exactly. These can be used in a cubic transformation or similar and a matrix transformation can then be applied to transform a multivariate distribution to obtain a given correlation matrix, as, for example, is done in Høyland *et al* [2003].

In chapter 7, the parameters in the transition equation of a discrete time state space form have been affected by the stochastic perturbation. It is of interest to extend the perturbations to the parameters in the measurement equations as well. The linear minimum variance filter has been derived in chapter 7 and the question is whether this method for dealing with parameter stochastic perturbations can be extended to the EKF and to the correlated perturbations.

Summary

To conclude, a suite of approximate filtering methods has been developed and has been tested on several numerical examples. At the moment, there is no single best approximation to the nonlin-

ear filtering problem, which works well in all cases. It is felt that the methods proposed here are a valuable addition to the tool-kit to address nonlinear or non-Gaussian filtering problems in various branches of science. The utility of the new algorithm for generating samples from a partially specified distribution has been demonstrated in a completely different application domain, viz., financial portfolio optimization.

Bibliography

B.D.O. ANDERSON AND J.B. MOORE (1979) Optimal filtering, *Prentice Hall, Englewood Cliffs*.

I. ARASARATNAM AND S. HAYKIN (2009) Cubature Kalman filters, *IEEE Transactions on Automatic Control*, **54(6)**, 1254–1269.

M.S. ARULAMPALAM, S. MASKELL, N. GORDON AND T. CLAPP (2002) A tutorial on particle filters for online nonlinear /non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing*, **50(2)**, 174–188.

K.A. BARBOSA, C.E. DE SOUZA AND A. TROFINO (2002) Robust H_2 filtering for discrete-time uncertain linear systems using parameter-dependent Lyapunov functions, *Proceedings of American Control Conference, Anchorage, Alaska, USA*, 3224–3229.

V.E. BENES (1981) Exact finite-dimensional filters for certain diffusions with nonlinear drift, *Stochastics*, **5(1-2)**, 65–92.

J.M. BRAVO, T. ALAMO AND E.F. CAMANCHO (2006) Bounded error identification of systems with time varying parameters, *IEEE Transactions on Automatic Control*, **51(7)**, 1144–1150.

D. BRIGO, B. HANZON AND F. LEGLAND (1998) A differential geometric approach to nonlinear filtering: the projection filter, *IEEE Transactions on Automatic Control*, **43(2)**, 247–252.

T. BOLLERSLEV (1987) A conditionally heteroskedastic time series model for speculative prices and rates of return, *Review of Economics and Statistics*, **69(3)**, 542–547.

T. BOLLERSLEV (1986) Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics* **31(3)**, 307–327.

G.E.P. BOX AND G.M. JENKINS (1976) Time series analysis: forecasting and control, *San Francisco: Holden Day*.

R. BYERS (1985) Numerical condition of the algebraic Riccati equation, *Linear Algebra and its Role in System Theory, Contemporary Mathematics, Providence, RI, American Mathematical Society*, **47**, 35–49.

C.H. CHEN (1988) Perturbation analysis for solutions of algebraic Riccati equations, *Journal of Comupational Mathematics*, **6**, 336–347.

C. CHIARELLA, H. HUNG AMD TÔ THUY-DUONG (2009) The volatility structure of the fixed income market under the HJM framework: A nonlinear filtering approach, *Journal of Computational Statistics and Data Analysis*, **53(6)**, 2075–2088.

G. CORTAZAR AND E.S. SCHWARTZ (2003) Implementing a stochastic model for oil futures price, *Journal of Energy Economics*, **25(3)**, 215–238.

T.M. COVER AND J.A. THOMAS (1991) Information theory, John Wiley and Sons, Inc.

J.C. COX, J.E. INGERSOLL AND S.A. ROSS (1985) A theory of the term structure of interest rates, *Econometrica*, **53(2)**, 385–407.

J.C. COX(1996) The constant elasticity of variance option pricing model, *Journal of Portfolio Management*, **Special Issue**, 15–17.

J.L. CRASSIDIS AND F.L.MARKLEY (2003) Unscented filtering for spacecraft attitude estimation, *AIAA Journal on Guidance, Control and Dynamics*, **26(4)**, 536–542.

D. CRISAN AND A. DOUCET (2002) A survey of convergence results on particle filtering methods for ractitioners, *IEEE Transactions on Signal Processing*, **50(3)**, 736–746.

P. DATE, L. JALEN AND R. MAMON (2008a) A new algorithm for latent state estimation in nonlinear time series models, *Applied Mathematics and Computation* **203(1)**, 224–232.

P. DATE, R. MAMON AND L. JALEN (2008b) A new moment matching algorithm for sampling from partially specified symmetric distributions, *Operations Research Letters* **36(6)**, 669–672.

P. DATE, R. MAMON AND L. JALEN (2010) A partially linearized sigma point filter for latent state estimation in nonlinear time series models, *Journal of Computational and Applied Mathematics*, **233(10)**, 2675–2682.

P. DATE AND K. PONOMAREVA (2011) Linear and non-linear filtering in mathematical finance: a review, *IMA Journal of Management Mathematics*, **22(3)**, 195–211.

F.E. DAUM (2005) Nonlinear filters: beyond the Kalman filter, *IEEE Aerospac and Electronic Systems Maragazine*, **20(8)**, 57–69.

F.E. DAUM (1986) Exact finite dimensional nonlinear filters, *IEEE Transactions on Automatic Control*, **31(7)**, 616–622.

F.E. DAUM (1988) New exact nonlinear filters, *Bayesian Analysis of Time Series and Dynamic Models*, New York: Marcel Dekker, **Chapter 8**.

F.E. DAUM (2001) Industrial strength nonlinear filters, *Proceedings of Workshop on Estimation, Tracking and Fusion: a tribute to Yaakov Bar-Shalom, Monterey, CA*.

F.E. DAUM (1986) New nonlinear filters and exact soluation of the Fokker-Planck equation, *Proceedings of IEEE American Control Conference, Seattle, WA*, 884–888.

W.L. DE KONING (1984) Optimal estimation of linear discrete-time systems with stochastic parameters, *Automatica*, **20(1)**, 113–115.

W.L. DE KONING (1980) Equivalent discrete optimal control problem for randomly sampled digital control systems, *International Journal of Systems Science*, **11(7)**, 841–850.

C.E. DE SOUZA, M. FU AND K.A. BARBOSA (2008) Robust filtering for uncertain linear discrete-time descriptor systems, *Automatica*, **44(3)**, 792–798.

A. DOUCET , A. DE FRIETAS AND N. GORDON (EDITORS) (2001) *Sequential Monte Carlo Methods in Practice*, Springer.

A. DOUCET, N. DE FREITAS, AND N. GORDON (2001) *Sequential monte carlo methods in practice*, Springer.

A. DOUCET, N. GORDON AND V.KRISHNAMURTHY (2001) Particle filters for state estimation of jump Markov linear systems, *IEEE Transactions on Signal Processing*, **49(3)**, 613–624.

J. DURBIN AND S.J. KOOPMAN (2001) *Time series analysis by state space methods*, Oxford University Press.

L. EL GHAOUI (1995) State-feedback control of systems with multiplicative noise via linear matrix inequalities, *Systems and Control Letters*, **24(3)**, 223–228.

L. EL GHAOUI AND G. CALAFIORE (2001) Robust filtering for discrete-time systems with bounded noise and parametric uncertainty, *IEEE Transactions on Automatic Control*, **46(7)**, 1084–1089.

E.F.D. ELLISON, M. HAJIAN, H. JONES, R. LEVKOVITZ, I. MAROS, G. MITRA AND D. SAYERS (2008) FortMP Manual, Brunel University, London and Numerical Algorithms Group, Oxford, link=http://www.optirisk-systems.com/manuals/FortmpManual.pdf.

G. EVENSEN (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Research*, **99(C5)**, 143–162.

A. FARINA, B. RISTIC, AND D. BENVENUTI (2002) Tracking a ballistic target: comparison of several nonlinear filters, *IEEE Transactions on Aerospace and Electronic Systems*, **38(3)**, 854–867.

F. FEARNHEAD (2005) Using random quasi-Monte-Carlo within particle filters, with application to financial time series, *Journal of Computational and Graphical Statistics*, **14(4)**, 751–769.

R. FLETCHER(1991) Practical Methods of Optimization, John Wiley.

R. FOURER, AND D. M. GAY AND B. KERNIGHAN (1989) AMPL: A Mathematical Programming Language.

X. FU AND Y. JIA (2010) An improvement on resampling algorithm of particle filters, *IEEE Transactions on Signal Processing*, **58(10)**, 5414–5420.

A. GELB (1986) Applied optimal estimation, *M.I.T. Press, Cambridge, M.A.*.

E. GERSHON, U. SHAKED, I. YAESH (2001) $\mathcal{H}_\infty$ control and filtering of discrete-time stochastic systems with multiplicative noise, *Automatica*, **37(3)**, 409–417.

J.C. GEROMEL (1999) Optimal linear filtering under parameter uncertainty, *IEEE Transactions on Signal Processing*, **47(1)**, 168–175.

A. GEYER AND S. PICHLER (1999) A state-space approach to estimate and test multifactor Cox-Ingersoll-Ross models of the term structure, *Journal of Financical Research*, **22(1)**, 107–130.

N.J. GORDON, D.J. SALMOND AND A.F.M. SMITH (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proceedings-F, **140(2)**, 107-113.*

G. GRIMMETT AND D. STIRZAKER *(2004)* Probability and random processes*, Oxford University Press.*

N. GULPINAR AND B. RUSTEM AND R. SETTERGREN *(2004) Simulation and optimization approaches to scenario tree generation,* Journal of Economic Dynamics and Control*, **28(7)**, 1291–1315.*

F. GUSTAFSSON, F. GUNNARSSON, N. BERGMAN, U. FORSSELL, J. JANSSON, R. KARLSSON AND P. J. NORDLUND *(2002) Particle filters for positioning, navigation, and tracking,* IEEE Transactions on Signal Processing*, **50(2)**, 425–437.*

F. GUSTAFSSON AND G. HENDELBY *(2012) Some relations between extended and unscented Kalman filters,* IEEE Transactions on Signal Processing*, **60(2)**, 545–555.*

A.C. HARVEY *(1989)* Forecasting structural time series models and the Kalman filter*, Cambridge University Press.*

R. HOCHREITER AND G. PFLUG *(2007) Financial scenario generaton for stochastic multi-stage decision processes as facility location problems,* Annals of Operations Research*, **152(1)**, 257–272.*

J.C. HOFF *(1983) A practical guide to Box-Jenkins forecasting,* London: Lifetime learning publications*.*

R.A. HORN AND C.R. JOHNSON *(1985) Matrix Analysis, Cambridge University Press.*

K. HØYLAND, M. KAUT AND S.W. WALLACE *(2003) A heuristic for moment-matching scenario generation,* Computational optimization and applications*, **24(2-3)**, 169–185.*

K. HØYLAND AND S.W. WALLACE *(2001) Generating scenario trees for multistage decision problems,* Management Science*, **47(2)**, 295–307.*

N. ILICH *(2009) A matching algorithm for generation of statistically dependent random variables with arbitrary marginals,* European Journal of Operational Research*, **192(2)**, 468–478.*

M. ISARD AND A. BLAKE *(1996) Contour tracking by stochastic propagation of conditional density,* European Conference on Computer Vision, Cambridge, UK*, 343-356.*

K. ITO AND K. XIONG *(2000) Gaussian filters for nonlinear filtering problems,* IEEE Transactions on Automatic Control*, **45(5)**, 910–927.*

B.N. JAIN *(1975) Guaranteed error estimation in uncertain systems,* IEEE Transactions on Signal Processing*, **20(2)**, 230–232.*

E.T. JAYNES *(1957) Information theory and statistical mechanics,* Physical Review*, **106(4)**, 361–373.*

A. H. JAZWINSKI *(1970) Stochastic processes and filtering theory, Academic press.*

J.C. JIMENEZ AND T. OZAKI *(2003) Local linearization filters for nonlinear continuous-discrete state space models with multiplicative noise,* International Journal of Control*, **76(12)**, 1159–1170.*

S.J. JULIER *(2002) The scaled unscented transformation,* Proceedings of the American Control Conference, Anchorage, AL*, **6**, 4555–4559.*

S. J. JULIER *(1998) A skewed approach to filtering,* Proceedings of AeroSense: 12th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL*, **3373**, 54-65.*

S. JULIER, J. UHLMANN AND H. DURRANT-WHYTE *(1995) A new approach for filtering nonlinear systems,* Proceedings of the American Control Conference, Seattle, Washington USA*, **3**, 1628-1632.*

S. J. JULIER AND J. K. UHLMANN *(1997) A new extension of the Kalman filter to nonlinear systems,* Proceedings AeroSense: 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, **3(1)**, 182–193.*

S.J. JULIER AND J.K. UHLMANN *(2004) Unscented filtering and nonlinear estimation,* Proceedings of the IEEE*, **92(3)**, 401–422.*

G. KALLIANPUR *(1990)* Stochastic filtering theory*, Springer Verlag.*

R.E. KALMAN *(1960) A new approach to linear filtering and prediction problems,* Journal of Basic Engineering*, **82(Series D)**, 35-45.*

R.KARLSSON, T. SCHON AND F. GUSTAFSSON *(2005) Complexity analysis of the marginalized particle filter,* IEEE Transactions on Signal Processing*, **53(11)**, 4408–4411.*

M. KAUT AND S.W. WALLACE *(2007) Evaluation of scenario generation methods for stochastic programming,* Pacific Journal of Optimization, ***3(2)**, 257–271.*

M. KIEFFER AND E. WALTER *(2005) Interval analysis for guaranteed nonlinear parameter and state estimation,* Mathematical and Computer Modelling of Dynamic Systems, ***11(2)**, 171–181.*

G. KITAGAWA *(1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,* Journal of Computational and Graphical Statistics, ***5(1)**, 1–5.*

M.M. KONSTANTINOV, P.HR. PETKOV AND N.D. CHRISTOV *(1990) Perturbation analysis of matrix quadratic equations,* SIAM Journal of Scientific Statistical Computing, ***11(6)**, 1159–1163.*

M.M. KONSTANTINOV, P.HR. PETKOV AND N.D. CHRISTOV *(2009) Perturbation bounds for certain matrix expressions and numerical solution of matrix equations,* In numerical analysis and its applications, S. Margenov, L.G. Vulkov and J. Wasniewski (Eds.). Lecture notes in computer science, Springer-Verglag, Berlin, Heidelberg, ***5434**, 68–79.*

W. LEDERMANN, C. ALEXANDER AND D. LEDERMANN *(2011) Random orthogonal matrix simulation,* Linear Algebra and its Applications, ***434(6)**, 1444–1467.*

P.M. LURIE AND M.S. GOLDBERG *(1998) An approximate method for sampling correlated random variables from partially specified distributions,* Management Science, ***44(2)**, 203–218.*

H. MARKOWITZ *(1952) Portfolio selection,* Journal of Finance, ***7(1)**, 77–91.*

MATHWORKS INC. *(1995) LMI control toolbox manual.*

R. MERWE, A.DOUCET, N. FREITAS AND E.WAN *(2000) The unscented particle filter,* Technical Report CUED/F-INFENG/TR, ***380**.*

H. MITCHELL AND P. HOTEKAMER *(1998) Data assimilation using an ensemble Kalman filter technique,* Monthly Weather Review, ***126(3)**, 796–811.*

R.E. MOORE *(1959) Automatic error analysis in digital computation,* Technical Report LMSD-48421, Lockheed Missiles and Space Co, Palo Alto, CA.

T. MORI *(1988) Comments on "A matrix inequality associated with bounds on solutions of algebraic Riccati and Lyapunov equation",* IEEE Transactions on Automatic Control, ***33(11)**, 1088.*

N.E. NAHI *(1969) Optimal recursive estimation with uncertain observation,* IEEE Transactions on Information Theory, ***15(4)**, 457-462.*

M. NORGAARD, N.K. POULSEN AND O. RAVN *(2000) New developments in state estimation for nonlinear systems,* Automatica, ***36(11)**, 1627–1638.*

B. ØKSENDAL *(2003)* Stochastic differential equations, an introduction with applications*, Springer.*

A. PANKRATZ *(1983) Forecasting with univariate Box-Jenkins models:concepts and cases,* New York: Wiley.

I.R. PETERSEN AND M.R. JAMES *(1996) Performance analysis and controller synthesis for nonlinear systems with stochastic uncertainty constraints,* Automatica, ***32(7)**, 959–972.*

I.R. PETERSEN AND D.C. MCFARLANE *(1996) Optimal guaranteed cost filtering for uncertain discrete-time linear systems,* International Journal on Robust and Nonlinear Control, ***6(4)**, 267–280.*

I.R. PETERSEN AND A.V. SAVKIN *(1999) Robust Kalman filtering for signal and systems with large uncertainties,* Birkhauser, Boston.

P.HR. PETKOV, N.D. CHRISTOV AND M.M. KONSTANTINOV *(1991) Computational methods for linear control systems,* Prentice Hall, New York.

M.K. PITT AND N. SHEPHERD *(1999) Filtering via simulation: auxiliary particle filters,* Journal of American Statistical Association, ***94(446)**, 590–599.*

G.C. PFLUG *(2001) Scenario tree generation for multiperiod financial optimization by optimal discretization,* Mathematical Programming, ***89(Series B)**, 251–271.*

G.C. PFLUG *(2000) Some remarks on the Value-at-Risk and the Conditional Value-at-Risk,* Probabilistic Constrained Optimization: Methodology and Applications*, Ed. S. Uryasev, Kluwer Academic Publishers.*

K. PONOMAREVA, P. DATE AND Z. WANG *(2010) A new unscented Kalman filter with higher order moment-matching, Proceedings of Mathematical Theory of Networks and Systems (MTNS 2010), Budapest.*

K. PONOMAREVA AND P. DATE *(2012) Higher order sigma point filter: a new heuristic for nonlinear time series filtering,* Applied Mathematics and Computation Journal*, under review.*

K. PONOMAREVA AND P. DATE *(2012) An exact minimum variance filter for a class of discrete time systems with random parameter perturbations,* IEEE Transactions on Signal Processing*, under review.*

K. PONOMAREVA, P. DATE AND D. ROMAN *(2012) A new algorithm for moment-matching scenario generation with application to financial portfolio optimization,* European Journal of Operational Research*, under review.*

P.K. RAJASEKARAN, N. SATYANARAYANA AND M.D. SINRATH *(1971) Optimum linear estimation of stochastic signals in the presence of multiplicative noise,* IEEE Transactions on Aerospace and Electronic Systems*, **7(3)**, 462–468.*

B. RISTIC, S. ARULAMPALAM AND N. GORDON *(2004) Beyond the Kalman Filter, Artech House Radar Library.*

R.T. ROCKAFELLAR AND S. URYASEV *(2002) Conditional value-at-risk for general loss distributions,* Journal of Banking and Finance*, **26(7)**, 1443–1471.*

U. SHAKED AND C.E. DE SOUZA *(1995) Robust minimum variance filtering,* IEEE Transactions on signal processing*, **43(11)**, 2474–2483.*

C.E. SHANNON *(1948) A mathematical theory of communication,* Bell System Technical Journal*, **27(3-4)**, 379–423, 623–659.*

D. SIMON *(2006) Optimal state estimation: Kalman, $\mathcal{H}_\infty$ and nonlinear approaches,* Wiley-Blackwell, Hoboken, New Jersey.

J.E. SMITH *(1993) Moment Methods for Decision Analysis,* Management Science*, **39(3)**, 340–358.*

H. W. SORENSON *(1985) Kalman filtering: theory and application, IEEE press.*

J.G. SUN *(1998) Perturbation theory for aglebraic Riccati equations,* SIAM Journal on Matrix Analysis and Applications*, **19(1)**, 39–65.*

R.F. SUOTO, J.Y. ISHIHARA AND G.A. BORGES *(2009) A robust extended Kalman filter for discrete-time systems with uncertain dynamics, measurements and correlated Noise,* Proceedings of American Control Conference, St. Louis, Missouri, USA.

D. TENNE AND T. SINGH *(2003) The higher order unscented filter,* Proceedings of American Control Conference, Buffalo, NY, ***3**, 2441–2446.*

N. TOPALOGLOU AND H. VLADIMIROU AND S.A. ZENIOS *(2002) CVaR models with selective hedging for international asset allocation,* Journal of banking and finance, ***26(7)**, 1535–1561.*

J.K. TUGNAIT *(1981) Stability of optimum linear estimators of stochastic signals in white multiplicative noise,* IEEE Transactions on Automatic Control, ***26(3)**, 757–761.*

W. Vandaele (1983) Applied time series and Box-Jenkins models, New York: Academic Press.

S.W. WALLACE AND W.T. ZIEMBA *(2005) Applications of Stochastic Programming, SIAM.*

Z. WANG, F. YANG AND X. LIU *(2006) Robust filtering for systems with stochastic non-linearities and deterministic uncertainties,* Journal of Systems and Control Engineering, ***220(3)**, 171–182.*

Y.WU, D.HU, M.WU, AND X. HU *(2005) Unscented Kalman filtering for additive noise case: augmented versus non-augmented,* IEEE Signal Processing Letters, ***12(5)**, 357–360.*

L. XIE AND Y.C. SOH *(1994) Robust Kalman filtering for uncertain systems,* Systems and Control Letters, ***22(2)**, 123–129.*

L. XIE, L. LU, D. ZHANG AND H. ZHANG *(2003) Robust filtering for uncertain discrete-time systems: an improved LMI approach,* Proceedings of the 42^{nd} IEEE Conference on Decision and Control, Maui, Hawaii.

F. YANG, Z. WANG AND Y.S. HUNG *(2002) Robust Kalman filtering for discrete time-varying uncertain systems with multiplicative noises,* IEEE Transactions on Automatic Control, ***47(7)**, 1179–1183.*

J. ZHU, J. PARK AND K.S. LEE *(2007) Robust Kalman filter of discrete-time Markovian jump system with parameter and noise uncertainty,* Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization, Beijing, China.

X. ZHU, Y.C. SOH AND L. XIE *(2002) Design and analysis of discrete-time robust Kalman filters,* Automatica, ***38(6)**, 1069–1077.*