Cubature Kalman Filtering: Theory & Applications

CUBATURE KALMAN FILTERING: THEORY & APPLICATIONS

BY

IENKARAN ARASARATNAM, B.Sc., M.A.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2009) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Cubature Kalman Filtering: Theory & Applications
AUTHOR:	Ienkaran Arasaratnam B.Sc. (Electronics & Telecommunications Engineering) University of Moratuwa, Sri Lanka M.A.Sc. (Electrical & Computer Engineering) McMaster University, Canada
SUPERVISOR:	Dr. Simon Haykin, Distinguished University Professor
NUMBER OF PAGES:	xii, 137

To my mother and the loving memory of my father

Abstract

Bayesian filtering refers to the process of sequentially estimating the current state of a complex dynamic system from noisy partial measurements using Bayes' rule. This thesis considers Bayesian filtering as applied to an important class of state estimation problems, which is describable by a discrete-time nonlinear state-space model with additive Gaussian noise. It is known that the conditional probability density of the state given the measurement history or simply the posterior density contains all information about the state. For nonlinear systems, the posterior density cannot be described by a finite number of sufficient statistics, and an approximation must be made instead.

The approximation of the posterior density is a challenging problem that has engaged many researchers for over four decades. Their work has resulted in a variety of approximate Bayesian filters. Unfortunately, the existing filters suffer from possible divergence, or the curse of dimensionality, or both, and it is doubtful that a single filter exists that would be considered effective for applications ranging from low to high dimensions. The challenge ahead of us therefore is to derive an approximate nonlinear Bayesian filter, which is theoretically motivated, reasonably accurate, and easily extendable to a wide range of applications at a minimal computational cost.

In this thesis, a new approximate Bayesian filter is derived for discrete-time nonlinear filtering problems, which is named the *cubature Kalman filter*. To develop this filter, it is assumed that the predictive density of the joint state-measurement random variable is Gaussian. In this way, the optimal Bayesian filter reduces to the problem of how to compute various multi-dimensional Gaussian-weighted moment integrals. To numerically compute these integrals, a third-degree spherical-radial cubature rule is proposed. This cubature rule entails a set of cubature points scaling linearly with the state-vector dimension. The cubature Kalman filter therefore provides an efficient solution even for high-dimensional nonlinear filtering problems. More remarkably, the cubature Kalman filter is the closest known approximate filter in the sense of completely preserving second-order information due to the maximum entropy principle. For the purpose of mitigating divergence, and improving numerical accuracy in systems where there are apparent computer roundoff difficulties, the cubature Kalman filter is reformulated to propagate the square roots of the error-covariance matrices.

The formulation of the (square-root) cubature Kalman filter is validated through three different numerical experiments, namely, tracking a maneuvering ship, supervised training of recurrent neural networks, and model-based signal detection and enhancement. All three experiments clearly indicate that this powerful new filter is superior to other existing nonlinear filters.

Acknowledgements

A journey is easier when we travel together. Interdependence is certainly more valuable than independence. This dissertation is the result of the last three years of work whereby I have been accompanied and supported by many people.

- First and foremost, I would like to thank my supervisor Professor Simon Haykin for providing me with an opportunity to undertake a research topic that is truly challenging, intriguing, and important in the field of estimation and control. I am deeply indebted to him for his superb advice and encouragement. He has been, and will always be, a role model for me.

- I would also like to express my gratitude to my two other committee members– Professor Thia Kirubarajan, and Professor Tom Hurd– for providing me with their valuable feedback all along the way.

- Thanks to all my wonderful friends for their help and camaraderie.

- Thanks to Cheryl Gies, Lola Brooks, Helen Jachana, and Cosmin Caroiu for being supportive during my stay at grad school.

- Finally, I deeply thank my family for love, faith, and unyielding support.
  God bless you all.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

The term 'filtering' is commonly used to refer to the process of estimating the target signal from a noisy received signal in some 'optimal' error criteria. Optimal filtering has been a focus of research in signal processing and control since the pioneering works of Kolmogorov [63] and Wiener [124] over half a century ago. In this thesis, we specifically consider Bayesian filtering as applied to state estimation problems described by discrete-time nonlinear dynamic systems with additive Gaussian noise. Here the term 'state' is typically used to describe the 'physical state' of a dynamic system.[1]

To fix ideas, let us consider a GPS (Global Positioning System) receiver mounted in a car. The GPS receiver is equipped to estimate the state vector components, namely, the position and velocity of the car. To illustrate the state-estimation mechanism

[1]The notion of 'state' is by no means confined to the physical state of a dynamic system. It is applicable to biological systems, economic systems, social systems, and others.

inside the GPS receiver, we write the motion equation describing how the state evolves in time as

$$(\text{state})_k = \big((\text{non})\text{linear function of state}\big)_{k-1} + (\text{process noise})_{k-1}, \quad (1.1)$$

where k is the time index indicating the value at k; the noise term is supposed to account for random perturbations due to the car's motion. The GPS receiver receives noisy measurements available in the form of a GPS signal, which typically contains the time-stamped locations of a set of satellites. Hence, we write another equation relating the temporally evolving state to the measurements as

$$(\text{measurement})_k = \big((\text{non})\text{linear function of state}\big)_k + (\text{measurement noise})_k. \quad (1.2)$$

Here the noise terms in (1.1) and (1.2) are assumed to be known up to second-order statistics.

For this scenario, we loosely define 'filtering' as the problem of estimating the current state of the car in an 'optimal' and consistent fashion as new GPS signals arrive sequentially in time.

The first landmark contribution to optimal filtering in discrete time was made by R. E. Kalman [62]. He formulated the recursive filtering solution to linear Gaussian problems using a state-space model similar to (1.1)-(1.2). Unfortunately, when the restricted assumptions of linearity and Gaussianity are violated, the Kalman filter fails to work. In real-world situations, however, dynamic systems are often nonlinear. Such nonlinear systems arise in many diverse applications from macroscopic ion channel

kinetic problems [83] to navigation and control of space capsules [74].

In 1964, Ho and Lee formulated a general stochastic filtering problem from a Bayesian estimation perspective, and derived a conceptual closed-form solution [47]. In their paper, they proved that the conditional probability density of the current state given the measurement history, or simply the posterior density, provides a complete statistical description of that state. From the posterior density, we can calculate any 'optimal' estimate of the state. In the sequential Bayesian filtering paradigm, the Bayesian filter sequentially updates the posterior density by acquiring information from newly arrived measurements. Ho and Lee went on to prove that the Kalman filter is a special case of a more generic Bayesian estimator.

For nonlinear systems, the posterior density cannot be described by a finite number of sufficient statistics, and we have to be content with an approximate filtering solution. Approximation of the Bayesian filter has engaged many researchers for over four decades, thereby coming up with a variety of solutions from the extended Kalman filter to particle filters. Unfortunately, the existing filters suffer from the curse of dimensionality, possible divergence or both, and it is doubtful that a single filter exists that would be considered effective for applications of both low and high dimensions. The challenge ahead of us therefore is to derive an approximate nonlinear Bayesian filter, which is theoretically motivated, reasonably accurate, and easily extendable to a wide range of applications at minimal computational cost.

1.2 Contributions

The main contribution of this thesis lies in the theoretical development of a new nonlinear Bayesian filter. The new filter is named the *cubature Kalman filter* (CKF)

[4]. In order to develop the CKF, the following two important steps are taken:

- The optimal Bayesian filter is reduced to the problem of how to compute multi-dimensional Gaussian-weighted integrals. This is accomplished by assuming that the predictive density of the joint state-measurement vector is Gaussian distributed.

- In order to numerically compute these integrals, a third-degree spherical-radial cubature rule is derived by combining a third-degree spherical rule and a systematically modified version of the first-degree generalized Gauss-Laguerre quadrature rule.

The CKF includes a number of striking properties:

- The computational cost of the CKF at each recursion cycle is a linear function of the number of function evaluations. The CKF therefore, provides an efficient solution even for high-dimensional nonlinear filtering problems.

- The CKF is derivative free. Hence, it can be applied even for some physics-based system models that include dead zones and look-up tables.

- Most remarkably, *given the second-order statistics of the state and innovations processes, due to the maximum entropy principle, the CKF is the closest known approximation to the Bayesian filter that could be designed for a nonlinear Gaussian filtering problem.*

Since the CKF committed to digital hardware may exhibit a divergent behavior or even complete failure due to numerical imprecision, a square-root version of the CKF

is also derived. The computational cost of this square-root formulation is of the same order as that of the CKF.

The formulation of this new filter is validated through a number of challenging computer experiments. Combined with the square-root formulation, the properties outlined above make the CKF an attractive choice for diverse applications. In particular, it is proven from the experimental results that the CKF may be the method of choice for training recurrent neural networks.

1.3 Organization of the Thesis

This thesis is organized as follows:

- Chapter 2 reviews the literature on discrete-time nonlinear filters. Specifically, based on computational aspects, nonlinear filters are discussed under two broad classes, namely, moment-matching algorithms and innovations-based algorithms.

- In Chapter 3, two different approaches to numerical integration– Product Rules, and Non-Product Rules– are briefly reviewed. An efficient non-product third-degree cubature rule for Gaussian-weighted integrals is finally derived.

- In Chapter 4, the cubature Kalman filter is developed for discrete-time nonlinear filtering problems using three extremely powerful ideas: (i) Bayes' rule, (ii) Statistical properties of the innovations process (iii) The cubature rule presented in Chapter 3. The computational cost of the CKF is computed in terms of *flops*.

- In Chapter 5, the square-root version of the CKF is developed. In order to accommodate the square-roots of the error covariances, the CKF is reformulated

using matrix factorization and the least-squares method. The additional computational cost incurred due to this square-root formulation is also computed.

- Chapter 6 evaluates the performance of the square-root cubature Kalman filter experimentally when applied to three challenging problems. They are (i) Tracking a maneuvering ship, (ii) Supervised training of recurrent neural networks for dynamic reconstruction of a chaotic time series, and (iii) Model-based signal detection and enhancement.

- Chapter 7 is divided into two parts: In the first part, a number of key attributes of the CKF are summarized. The second part outlines a few interesting research topics we would like to treat in future.

- Finally, three appendices are added to this thesis:

  – In Appendix A, the CKF developed for pure discrete-time problems is extended to filtering problems whose state-space model is described by a continuous-time process equation and a discrete-time measurement equation.

  – Appendix B presents the unscented filtering approach as it shares a number of common properties with the CKF.

  – In Appendix C, the theory of particle filtering is derived from a moment-matching perspective.

1.4 Related Publications

Many important results reported in this thesis can be found in the following articles:

- I. Arasaratnam and S. Haykin, "Cubature Kalman Filtering: A Powerful Tool for Aerospace Applications," accepted, *Int'l Radar Conf.*, Bordeaux, France, Oct. 2009.

- I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," forthcoming *IEEE Trans. Automatic Control*, vol. 54, June 2009.

- I. Arasaratnam and S. Haykin, "Nonlinear Bayesian Filters for Training Recurrent Neural Networks," Book Ch., *Advances in Artificial Intelligence*, Springer: Berlin/Heidelberg, A. Gelbukh and E. Morales, Eds., pp. 12-33, 2008.

- I. Arasaratnam and S. Haykin, "Square-Root Quadrature Kalman Filtering," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2589-2593, June 2008.

- I. Arasaratnam, S. Haykin and R. Elliott, "Discrete-Time Nonlinear Filtering Algorithms Using Gauss-Hermite Quadrature," *Proc. IEEE*, vol. 95, no. 5, pp. 953-977, May 2007.

Chapter 2

Literature Review

This chapter reviews a variety of existing solutions to filtering problems describable by a discrete-time nonlinear state-space model with additive Gaussian noise. Although it is understood from Figure 2.1 that this survey focuses only on the solutions of a subset of quite general and complex dynamic state-space models, many important real-world problems fall within this considered subset. For example, despite the fact that time is continuous in nature, we are interested in finding discrete-time filtering solutions for the following three main reasons:

- In many cases, we are provided with a sampled version of analog (continuous) processes. Moreover, we may encounter a sequence of events occurring naturally in discrete time, e.g., random-walk problems of statistics.

- The measurement models are commonly available in discrete time because sensors are digital devices. Sometimes, the discrete-time measurement models are unintentional. For example, the satellite navigation system provides GPS signals with a single position fix only as each satellite passes. Thus, the GPS

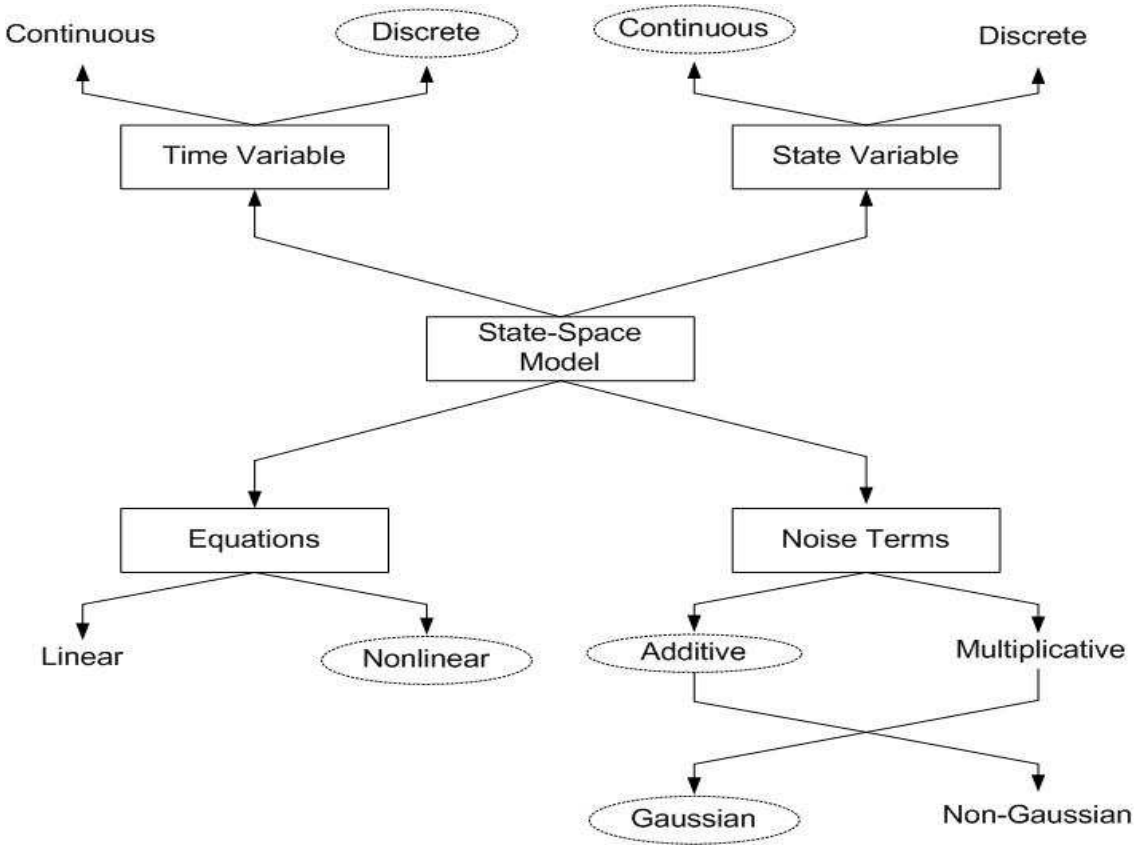


Figure 2.1: Setting of a generic state-space model; 'Equations' denote the state evolution function in (1.1), and the state-to-measurement mapping function in (1.2) without noise terms; circles denote the focus of this thesis

receiver can estimate its position only at discrete points in time.

- Finally, irrespective of the temporal description of a state-space model we tend to solve filtering problems using digital computers.

2.1 Optimal Bayesian Filter

This section presents a theoretically relevant discrete-time optimal Bayesian filter. Before setting the stage for the derivation of the Bayesian filter, consider the problem

of sequentially estimating the state $\mathbf{x}_k \in \mathbb{R}^{n_x}$ when it evolves in time according to the equation

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{q}_{k-1} \tag{2.1}$$

and is related to noisy measurements via

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{r}_k, \tag{2.2}$$

where

- $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{h} : \mathbb{R}^{n_x} \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ are known
- $\mathbf{z}_k \in \mathbb{R}^n$ is the measurement
- $\mathbf{q}_{k-1}$ and $\mathbf{r}_k$ are uncorrelated process and measurement Gaussian noise sequences with zero means and covariances Q_{k-1} and R_k, respectively
- $\mathbf{u}_k \in \mathbb{R}^m$ is a known input at time k; it may be derived from a compensator as shown in Figure 2.2.

The state-space model of this filtering problem embodies a pair of equations, namely, the process equation (2.1), and the measurement equation (2.2). The process equation describes the temporal evolution of the hidden state, whereas the measurement equation maps the hidden state to the measurements. The objective of the optimal filter is to recursively compute the posterior density $p(\mathbf{x}_k|D_k)$, where the history of input-measurement pairs available up to time k is denoted by $D_k = \{(\mathbf{z}_i, \mathbf{u}_i),\ i = 1, 2, \ldots k\}$. The posterior density provides a complete statistical

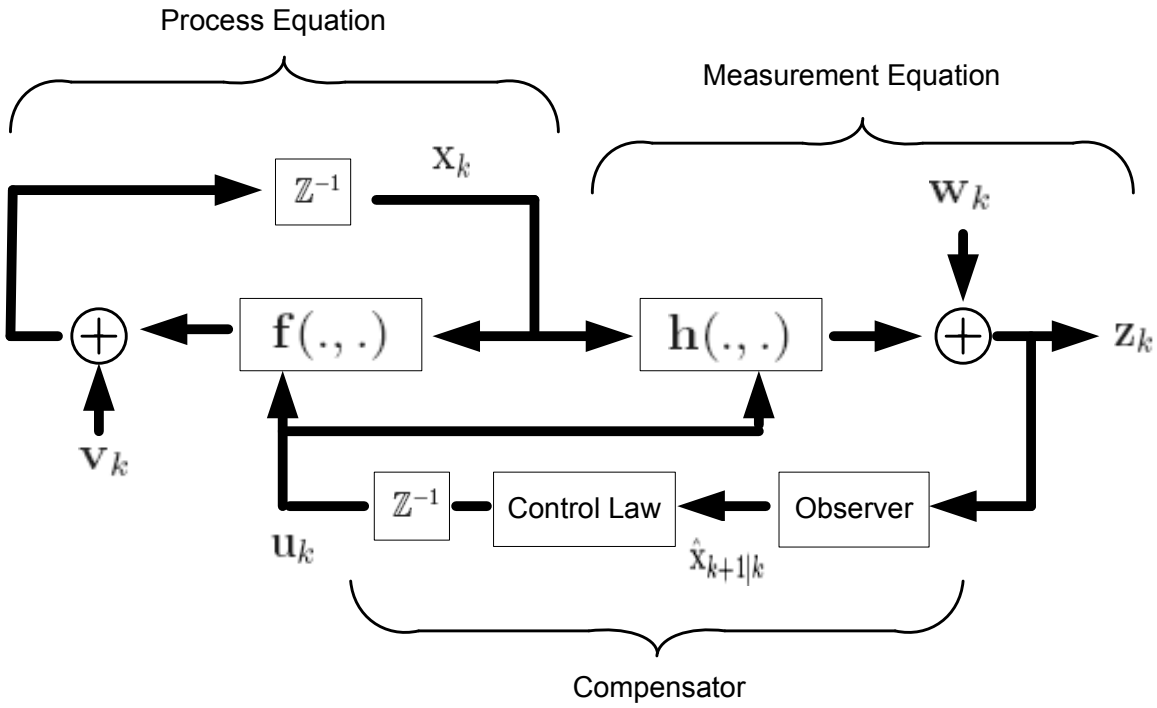


Figure 2.2: Signal-flow diagram of a dynamic state-space model driven by the feedback control input. The observer may employ a Bayesian filter. The label $\mathbb{Z}^{-1}$ denotes the unit delay.

description of the state at that time, and is computed in two basic update steps, namely, the time update, and the measurement update.

2.1.1 Time update

In this step, using Kolmogorov's forward equation [51], the old posterior density of the state is updated before receiving a new measurement. This operation finally leads to the predictive density of the current state

$$\begin{aligned} p(\mathbf{x}_k|D_{k-1}) &= \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k, \mathbf{x}_{k-1}|D_{k-1}) d\mathbf{x}_{k-1} \\ &= \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_{k-1}|D_{k-1}) p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) d\mathbf{x}_{k-1}, \end{aligned} \tag{2.3}$$

where $p(\mathbf{x}_{k-1}|D_{k-1})$ is the old posterior density at time $(k-1)$, and the state-transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is obtained from (2.1).

2.1.2 Measurement Update

Upon the receipt of a new measurement $\mathbf{z}_k$, using Bayes' rule, the predictive density is updated and the posterior density of the current state is obtained as follows:

$$\begin{aligned} p(\mathbf{x}_k|D_k) &= p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k, \mathbf{z}_k) \\ &= \frac{1}{c_k} p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k), \end{aligned} \tag{2.4}$$

where the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k)$ is obtained from (2.2) and the normalizing constant

$$\begin{aligned} c_k &= p(\mathbf{z}_k|D_{k-1}, \mathbf{u}_k) \\ &= \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k) d\mathbf{x}_k. \end{aligned}$$

To develop a recursive relationship between predictive and posterior densities in (2.4), the *natural condition of control*, which is described next, must be assumed.

Natural Condition of Control

The natural condition of control states that the control inputs have to satisfy the relationship [92]

$$p(\mathbf{u}_k|D_{k-1}, \mathbf{x}_k) = p(\mathbf{u}_k|D_{k-1}).$$

This condition therefore suggests that D_{k-1} has sufficient information to generate the input $\mathbf{u}_k$ as shown in Figure 2.2. To be more specific, the input $\mathbf{u}_k$ can be generated using $\hat{\mathbf{x}}_{k|k-1}$. Under this condition, it therefore holds that

$$p(\mathbf{x}_k|D_{k-1}, \mathbf{u}_k) = p(\mathbf{x}_k|D_{k-1}). \tag{2.5}$$

The natural condition of control is fulfilled when the control inputs are precomputed as in the case of an open-loop control system or a feedback control system as depicted in Figure 2.2. However, it may not be fulfilled in the situation where an observer and a controller function independently. For example, it is not fulfilled in a parameter-tracking problem where the controller is aware of the parameter more accurately than what the observer is estimating. *However, throughout this thesis, we assume that the natural condition of control is fulfilled.*

Now, let us return to the recursive solution to the filtering problem. Substituting (2.5) into (2.4) yields

$$p(\mathbf{x}_k|D_k) = \frac{1}{c_k} p(\mathbf{x}_k|D_{k-1}) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k), \tag{2.6}$$

as desired, where the 'new' normalizing constant

$$c_k = \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_k|D_{k-1}) p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k) d\mathbf{x}_k. \tag{2.7}$$

The recursive solution of the optimal Bayesian filter given by (2.3), (2.6) and (2.7) provides a unified approach for nonlinear filtering problems conceptually. From the posterior density (2.6), any statistic regarding $\mathbf{x}_k$ can be computed. For example, we

can provide an optimal estimate and the associated covariance (variance) according to some chosen criterion.

The signal-flow diagram in Figure 2.3 illustrates the interplay of various densities to produce a new posterior density from the old posterior density at each recursion cycle. This cycle is repeated to estimate the new state at time $(k+1)$. Unfortunately, in most cases, the posterior density is intractable for the following two reasons:

- For a multi-dimensional system, we must compute multi-dimensional integrals (2.3) and (2.7).
- Even after these integrals are computed, it may be difficult to propagate the posterior density through subsequent time steps. The reason is that there is no guarantee that the new posterior will remain closed with a finite summary statistic expressed in terms of (quasi-)moments.

Consequently, intense research in the past has resulted in a number of suboptimal solutions to nonlinear filtering problems. These suboptimal filtering solutions to the problem of recursively finding the posterior density may be classified in numerous ways. However, it is the measurements which provide new information for the purpose of updating the state. It is therefore reasonable to classify nonlinear filters based on how they process these measurements. In this chapter, all known nonlinear filters are accommodated under one of two broad classes of algorithms: moment-matching algorithms processing the raw measurements, and innovations-based algorithms processing the whitened measurements.

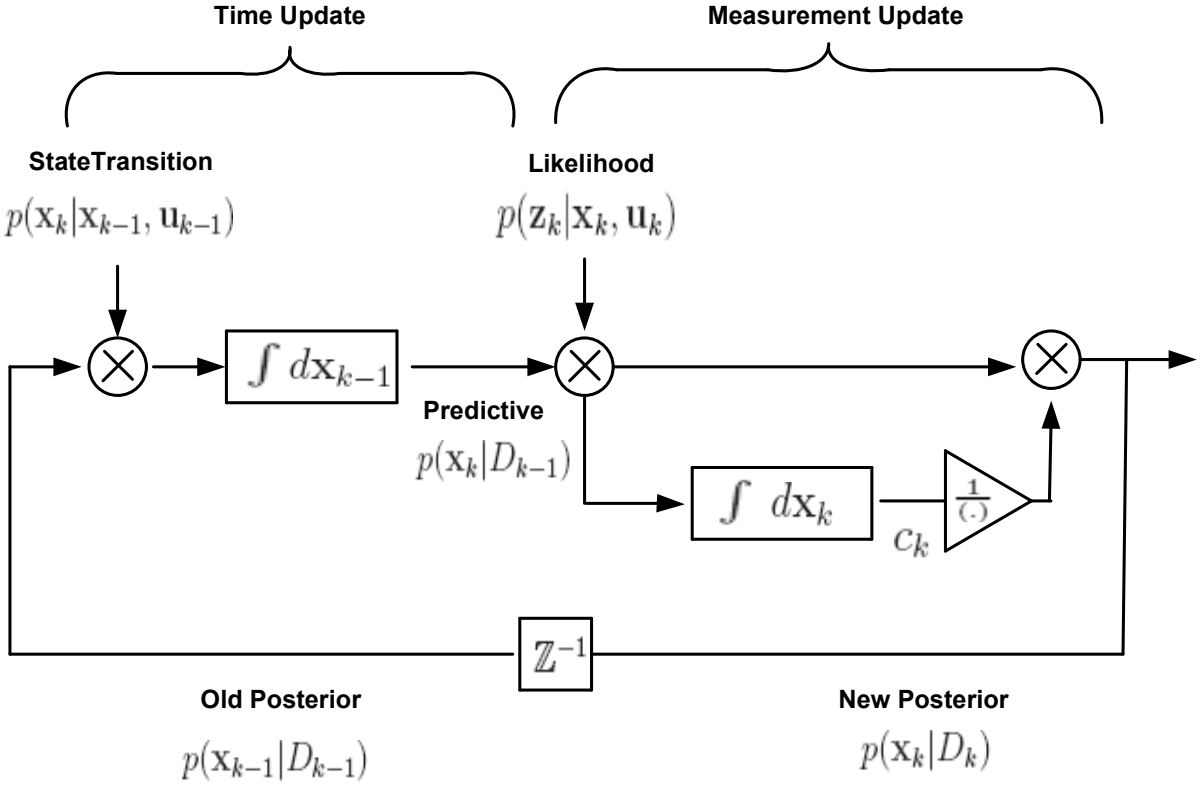


Figure 2.3: Optimal recursive Bayesian filter. The posterior density from the previous time step becomes our prior in the current time step in each recursion cycle.

2.2 Moment-Matching Algorithms

The basic idea of moment-matching algorithms is to represent both the predictive and posterior densities in some form *a priori* (e.g., Gaussian, Gaussian-sums, orthogonal polynomials such as the Edge-worth series, spline functions, point-masses, particles, etc.), thereby setting the stage to recursively find its parameters or moments based on the two basic update equations (2.3) and (2.6). Of course, these methods directly use the raw measurements due to the presence of the likelihood function in (2.6). To elaborate on this idea, the classical approach introduced by Kushner in the late 1960s [67] is considered. Specifically, in [67, 68], Kushner approximates these two densities

as Gaussian. Under the Gaussian approximation, the moment-matching algorithm completes its recursion cycle in the two basic update steps described earlier.

Time Update

In this first step, the filter computes the predicted state or the mean of the predictive density as

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}[\mathbf{x}_k|D_{k-1}]. \tag{2.8}$$

Substituting (2.1) into (2.8) yields

$$\hat{\mathbf{x}}_{k|k-1} = \mathbb{E}\big[\mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1}) + \mathbf{v}_{k-1}|D_{k-1}\big]. \tag{2.9}$$

Because $\mathbf{v}_{k-1}$ is assumed to have zero mean, and uncorrelated with the past measurements, we get

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbb{E}\big[\mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})|D_{k-1}\big] \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})p(\mathbf{x}_{k-1}|D_{k-1})d\mathbf{x}_{k-1} \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1})\mathcal{N}(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1|k-1},P_{k-1|k-1})d\mathbf{x}_{k-1},\end{aligned} \tag{2.10}$$

where $\mathcal{N}(.,.)$ is the conventional symbol for a Gaussian density. Similarly, the error covariance is given by

$$\begin{aligned} P_{k|k-1} &= \mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T | D_{k-1}] \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathbf{f}^T(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}) d\mathbf{x}_{k-1} \\ &\qquad - \hat{\mathbf{x}}_{k|k-1} \hat{\mathbf{x}}_{k|k-1}^T + Q_{k-1}. \end{aligned} \tag{2.11}$$

Measurement Update

In this second step, by applying the measurement update equations (2.6) and (2.7), the moment-matching filter computes the mean and covariance of the Gaussian posterior density as

$$\hat{\mathbf{x}}_{k|k} = \frac{1}{c_k} \int_{\mathbf{R}^{n_x}} \mathbf{x}_k \mathcal{N}(\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k), R_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}) d\mathbf{x}_k, \tag{2.12}$$

$$\begin{aligned} P_{k|k} &= \frac{1}{c_k} \int_{\mathbf{R}^{n_x}} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T \mathcal{N}(\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k), R_k) \\ &\qquad \times \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}) d\mathbf{x}_k, \end{aligned} \tag{2.13}$$

where the normalizing constant

$$c_k = \int_{\mathbf{R}^{n_x}} \mathcal{N}(\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k), R_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}) d\mathbf{x}_k. \tag{2.14}$$

The set of integrals entailing (2.10), (2.11), (2.12), (2.13) and (2.14) can be approximated by numerical integration methods. For example, the Gauss-Hermite quadrature has been proposed in [68, 80, 120], whereas the Monte Carlo integration technique is employed in the so-called Gaussian particle filter [35]. *In so doing, the resulting*

moment-matching algorithm is expected to outperform the widely used EKF because it does not rely on a direct linearization of both the process and measurement functions.

In some nonlinear filtering problems, the conditional densities can be multi-modal or heavily skewed, and can no longer be approximated by a single Gaussian. To alleviate the errors due to the Gaussian assumption, the generalized Edgeworth series has been exploited to represent a wide class of densities in the literature [17, 22, 117, 119]. The generalized Edgeworth series consists of two multiplicative functions: (i) a Gaussian density (ii) a set of Hermite polynomials that are orthogonal to each other. Hence the posterior density can be characterized by quasi moments. The recursive relations can be established for a finite number of these quasi-moments. However, the main issue here is that a large number of terms are required to reasonably approximate a distinctly non-Gaussian density. It has also been observed that the behavior of the estimator is highly sensitive to the truncation of the infinite series [22, 118].

To this end, filtering solutions for the problem of how one could approximately represent the predictive and posterior densities using a limited amount of summary statistics have been discussed. At the extreme end of the moment-matching case, the following question arises: *Is it possible to propagate the whole posterior density, which can possibly be described by an infinite number of moments, recursively?* Surprisingly, the answer to this question is positive.

The simplest answer is known as the point-mass method. In the point-mass method, the densities are approximated by point masses located on a rectangular grid [13], Ch. 5 in [31]. Hence, the integrals for recursive Bayesian estimation can each be evaluated numerically as a discrete nonlinear convolution.

Another approach in [65] has proposed piecewise constant functions to approximately represent densities. This method is more sophisticated, and incurs less computational cost than the point mass method. More sophisticated interpolation schemes, such as different spline approximations, have also been investigated in the literature [14, 27, 123].

Finally, under this research direction, a relatively new class of algorithms called particle filters is discussed.[1]

Particle Filters

Particle filters are also suboptimal filters. Essentially, particle filtering without resampling embraces the idea of Monte Carlo integration [31, 40, 99]. Particle filters use a large number of weighted samples called particles to approximately represent the underlying posterior density. Since the basic form of this plain Monte Carlo estimation degenerates over time, a clever remedy in the form of resampling was introduced in the seminal paper by Gordon *et al.* [40]. Particle filters offer the following benefits:

- The basic theory of particle filtering is simple
- Coding all steps but resampling is relatively easy
- The greatest strength of particle filters lies in their flexibility– they can be applied to nonlinear filtering problems with noise terms being (non)additive and (non-)Gaussian.

Particle filters have gained popularity in many challenging and complicated systems, which commonly arise in econometrics, robotics, and financial mathematics. However,

[1]The reader may consult Appendix C for a detailed exposition of how particle filters fit into the moment-matching approach.

they suffer from the following serious limitations:

- Particle filters may require an enormous number of particles in the following contexts:
  - For filtering problems with a relatively high state-space dimension.
  - When more accurate measurements are available– This leads to a highly-peaked likelihood function that may not have a sufficient overlap with the predictive density. In order to mitigate the tendency of sampling particles from non-informative volume of the state space, more sophisticated sampling strategies such as Markov Chain Monte Carlo sampling may be required.
- The performance of particle filters crucially depends on the choice of the so-called proposal density. The optimal proposal density is the true posterior density, and is therefore unavailable. However, particle filters manage to perform well with suboptimal proposal densities [99].
- The efficiency of particle filters is significantly reduced by
  - random sampling and,
  - resampling.
- Particle filters are not immune to the *curse of dimensionality*, a term coined by Bellman five decades ago [12]– It is well known that the variance of the estimation error produced by a brute-force Monte Carlo sampling method that

numerically computes the integral $\int_{\mathbf{x}} f(\mathbf{x})dP(\mathbf{x})$ is given by

$$c\frac{\sigma^2}{N},$$

where

- c is the proportionality constant.
- N is the number of particles drawn from the probability density $P(\mathbf{x})$.
- σ^2 is the variance of the random variable $y = f(\mathbf{x})$.

It is however, argued in [24] that from the perspective of a particle filter that uses the Monte Carlo sampling method, c is not a constant; rather it is a variable that strongly depends on both time and the state-vector dimension.

- Finally, there is no rigorous guideline for choosing the 'optimal' number of particles. Of course, each step of the particle filtering algorithm can be refined for improved performance depending on the nature of a problem and experience of a designer.

Despite these problems, an intense research activity in the field of particle filtering has, over more than a decade, resulted in numerous improvements in both the statistical and computational efficiency (see [15] and the references therein). As a final remark, it is worth mentioning a quite common practical rule: *Only when a kit of analytical tools do not suffice for the problem at hand are the simulation-based methods necessary.* The bottom line: Particle filters may be an attractive choice only when all sorts of analytical filters, some of which are described next, fail.

2.3 Innovations-Based Algorithms

Before going into the details of innovations-based algorithms, this section starts with the definition of 'innovations'. The innovation at time k, denoted by $\boldsymbol{\epsilon}_k$, is obtained by subtracting the predicted measurement given the past measurement history D_{k-1} from the current measurement $\mathbf{z}_k$:

$$\boldsymbol{\epsilon}_k = \mathbf{z}_k - \mathbb{E}[\mathbf{z}_k|D_{k-1}]. \tag{2.15}$$

From (2.15), it is understood that $\boldsymbol{\epsilon}_k$ amounts to new information about the state that does not exist in D_{k-1}. Unlike moment-matching algorithms, which process the raw measurements, innovations-based algorithms are designed to process the innovations.

Suppose that the predictive density of the joint state-measurement vector is assumed to be Gaussian. In this case, all other conditional densities, namely, the predictive state density, the innovations density and the posterior density, become Gaussian. This implies that all conditional densities are closed in each recursion cycle, similarly to the Kalman filter, which nicely fits into innovations-based algorithms [41]. It is therefore, not surprising that the nonlinear filters embracing the innovations approach include simple matrix algebraic operations similar to those of the linear Kalman filter. The nonlinear filters exemplified by the extended Kalman filter and its variants [51], unscented Kalman filter [52], and quadrature Kalman filter [49], fall under this second category of algorithms. These filters differ from each other in the way in which the means and covariances of the conditional densities are computed. Curiously, the derivation of any one of these existing innovations-based nonlinear filters from the innovations perspective is absent from the literature. A

more detailed derivation of this powerful approach is deferred to Chapter 4.

Extended Kalman Filters

The extended Kalman filter (EKF) has been the workhorse for nonlinear filtering problems in signal processing [9, 81, 82, 100], control [29, 110, 116] and optimization [11, 95] for more than four decades. The EKF linearizes the nonlinear process and measurement functions using the first-order Taylor series expansion evaluated at the current best estimate of the state. This suggests that Kalman's original theory can be adopted to nonlinear systems. Though the EKF is popular for its simplicity, it suffers from the following limitations:

- It quickly diverges in 'highly' nonlinear systems owing to its limited approximation capability.
- The EKF often yields an underestimated error covariance because it does not account for a prior covariance matrix in its analytical linearization approach.
- Finally, its application is limited to differentiable functions only.

To improve the EKF, two different attempts have been made in the past:

- *The iterated EKF*: The basic idea of the iterated EKF is to linearize the measurement model around the updated state, rather than the predicted state [37]. This is achieved iteratively, and it involves the use of the current measurement.
- *The second-order EKF*: The basic idea of this algorithm is go beyond the first-order term of the Taylor series expansion, and retain the second-order terms as well [8].

The second-order EKF requires the computation of Jacobians and Hessians, procedures that are often numerically unstable, and computationally intensive. Particularly, the Hessian turns out to be a three-dimensional matrix with its entries being second-order partial derivatives. In some systems, they do not even exist, e.g., models representing abruptly changing behavior. In practice, the second-order EKF is not commonly used, and higher order approximations are almost never used.

As an analytical derivative-free alternative to the first-order EKF, the central-difference Kalman filter was proposed in [101], and later extended to the second order as well as the corresponding square-root solution for numerical stability in [89]. The original derivation in [89] involves the approximation of a nonlinear function using Stirling's second-order interpolation formula, and closely follows the second-order EKF theory. Of course, this second-order formula is identical to the second-order Taylor expansion of the function about the latest state with the Jacobian and Hessian being replaced by central differences [49, 5].[2]

Finally, under the innovations-based nonlinear filter family, two more members, namely, the quadrature Kalman filter and the unscented Kalman filter, are mentioned. The quadrature Kalman filter uses the Gauss-Hermite quadrature rule to numerically compute various Gaussian-weighted integrals arising in the Gaussian approximation to the nonlinear Bayesian filter [49]. The quadrature Kalman filter was derived from a statistical linear regression perspective, and expanded to facilitate its use in a non-Gaussian environment in [7]. For improved numerical stability, a square-root solution was proposed in [6].

[2]Nørgaard *et al.* in [89] have named the central-difference Kalman filter the 'divided-difference filter'. This naming is a misnomer because the divided-difference method is fundamentally different from the central-difference method. The central-difference Kalman filter is also referred to as the 'nprKF', where the prefix 'npr' stands for the initials of the three authors of [89], and the postfix 'KF' stands for 'Kalman Filter'.

A detailed account of the unscented Kalman filter is presented in Appendix A, which compares the unscented Kalman filter, with the CKF studied in this thesis.

So far, under the innovations class, a number of well-known algorithms based on the Gaussian assumption have been discussed. However, the posterior density resulting from nonlinear filtering may possibly be non-Gaussian with multiple modes. A complete statistical characterization requires more than the first two-order moments. As a systematic solution to represent the non-Gaussian posterior density, any one of the innovations-based filters can be generalized to approximate the posterior density as a mixture of Gaussians at the expense of increased computational complexity. The reason lies in the fact that any non-Gaussian density can arbitrarily closely be modeled by a Gaussian mixture [1]. A bank of filters running in parallel can be used to provide the state estimate as a weighted average of each filter estimate. The weights can be obtained from the innovations. However, the Gaussian mixture-based nonlinear filter suffers from the following two limitations:

- *Growing-memory problem.* The number of Gaussian components grows exponentially when the filter propagates over time. For a manageable implementation, the Gaussian mixture reduction techniques can be utilized at the expense of a certain amount of information loss [125].

- *Degeneration.* Remarkably, after a few time steps, all filter modules of the Gaussian mixture filter may tend to concentrate on a relatively small state-space volume, similar to particle filtering without resampling. To avoid this undesirable situation, a systematic reinitialization may often be required. Finding a systematic solution to avoid this reinitialization step is truly challenging.

Summary

This chapter presents the theory of the optimal Bayesian filter for an important class of state-estimation problems described by a discrete-time nonlinear state-space model with additive Gaussian noise. The Bayesian filter recursively propagates the posterior density, which embodies a complete statistical description of the current state. For nonlinear systems, however, the optimal posterior density is typically intractable, and an approximation must be made instead. A number of well-known approximate nonlinear filters are discussed with a special emphasis on their relative virtues and limitations.

Chapter 3

Theory of Cubature Rules

Consider a multi-dimensional weighted integral of the form

$$I(\mathbf{f}) \quad = \quad \int_{\mathscr{D}} \mathbf{f}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}, \tag{3.1}$$

where $\mathbf{f}(.)$ is some arbitrary function, $\mathscr{D} \subseteq \mathbb{R}^n$ is the region of integration, and the known weighting function $w(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathscr{D}$. For example, in a Gaussian-weighted integral, $w(\mathbf{x})$ is a Gaussian density and satisfies the nonnegativity condition in the entire region. Such integrals arise in many places, e.g., financial mathematics, computer graphics, etc. Unfortunately, they can seldom be computed analytically, especially in a multi-dimensional region. To compute them approximately, we seek numerical integration methods. The basic task of numerically computing the integral (3.1) is to find a set of points $\mathbf{x}_i$, and weights ω_i that approximates the integral $I(\mathbf{f})$ using a weighted sum of function evaluations:

$$I(\mathbf{f}) \quad \approx \quad \sum_{i=1}^{m} \omega_i \mathbf{f}(\mathbf{x}_i). \tag{3.2}$$

The methods used to find the weighted point set $\{\mathbf{x}_i, \omega_i\}$ are often described as *cubature rules*. Cubature rules can be divided into *product rules* and *non-product rules*, as described in the sequel.

3.1 Product Rules

For the simplest one-dimensional case, that is $n = 1$, the *quadrature rule* may be used to numerically compute (3.1) [114, 111]. In the Bayesian filtering context, if the weighting function $w(x)$ is in the form of a Gaussian density and the integrand $f(x)$ is well approximated by a polynomial in x, then the Gauss-Hermite quadrature rule can be used to numerically compute the Gaussian-weighted integral [7].

The quadrature rule may be extended to compute multi-dimensional integrals by successively applying it in a tensor-product of one-dimensional integrals. For example, consider an m-point/dimension quadrature rule that is exact for polynomials of degree up to d. To numerically compute an n-dimensional integral, a grid of m^n points is set up for function evaluations. The resulting product rule is exact for integrals whose integrands are monomials of degree up to d. The computational complexity of the product quadrature rule exponentially increases with n, and therefore suffers from the curse of dimensionality. In general, for $n > 5$, the product rule is not a reasonable choice to approximate integrals arising in the Bayesian filter.

3.2 Non-Product Rules

To mitigate the curse of dimensionality associated with product rules, non-product integration rules for integrals may be applied. Non-product rules choose points directly from the domain of integration. Some of the well-known non-product rules are:

- Randomized Monte Carlo methods [73]
- Quasi-Monte Carlo methods [85, 70]
- Lattice rules [107]
- Sparse grids [106, 36, 38, 91]
- Monomial-based cubature rules [115, 19]

The randomized Monte Carlo methods evaluate the integral using a set of equally-weighted sample points drawn randomly, whereas, in the quasi-Monte Carlo methods and lattice rules, the points are generated from a unit hyper-cube region using deterministically defined mechanisms. On the other hand, the sparse grids based on the Smolyak formula, in principle, combine a quadrature (univariate) routine for high-dimensional integrals; they detect important dimensions automatically, and place more grid points there. Although the non-product methods described so far are able to numerically compute a given integral with a prescribed accuracy, they do suffer from the curse of dimensionality to a certain extent. In particular, my work requires a numerical integration rule that

- yields reasonable accuracy;

- requires a small number of function evaluations; and
- is easily extendable to high dimensions.

For this reason, monomial-based cubature rules are considered as a 'good' choice.

3.2.1 Monomial-based Cubature Rules

In mathematics, the term *monomial* means a product of powers of variables. If only a single variable x is considered, this means that any monomial is either 1 or x^d, where d is a positive integer. If several variables are considered, say, x_1, x_2, and x_3, then each can individually be exponentiated so that the resulting monomial is of the form $x_1^{d_1} x_2^{d_2} x_3^{d_3}$, where d_1, d_2, and d_3 are nonnegative integers. Typically, a set of weighted cubature points is chosen so that the cubature rule is exact for a set of monomials of *degree* d or less, as shown by

$$\int_{\mathscr{D}} \mathscr{P}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} \;=\; \sum_{i=1}^{m} \omega_i \mathscr{P}(\mathbf{x}_i), \tag{3.3}$$

where the monomial $\mathscr{P}(\mathbf{x}) = x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}$ with d_j being nonnegative integers and $\sum_{j=1}^{n} d_j \leq d$. This suggests that higher the degree of the cubature rule, the more accurate its solution becomes. Moreover, the cubature rule is also exact for all linear combination of monomials of degree up to d as stated in the following proposition.

Proposition 3.1: *For a cubature rule to be exact for all linear combinations of monomials of degree up to d, it is sufficient and necessary that the cubature rule be exact for all monomials of degree up to d.*

Proof: Consider a function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ that is spanned by a vector subspace

consisting of monomial basis functions $\{\mathscr{P}_j(\mathbf{x})\}$. Hence, we write

$$\mathbf{f}(\mathbf{x}) = \sum_{j=1}^{\frac{(n+d)!}{n!d!}} c_j \mathscr{P}_j(\mathbf{x}), \tag{3.4}$$

where $\{c_j\}$ are some known coefficients; we have a number $\frac{(n+d)!}{n!d!}$ of monomials of degree up to d in an n-dimensional space.

Consider an integral of the form

$$I(\mathbf{f}) = \int_{\mathscr{D}} \mathbf{f}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}.$$

Using (3.4), we further simplify

$$\begin{aligned} I(\mathbf{f}) &= \int_{\mathscr{D}} \big(\sum_j c_j \mathscr{P}_j(\mathbf{x}) \big) w(\mathbf{x}) d\mathbf{x} \\ &= \sum_j c_j \int_{\mathscr{D}} \mathscr{P}_j(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{3.5}$$

To numerically compute (3.5), we consider a cubature rule of the form

$$C(\mathbf{f}) = \sum_{i=1}^{m} \omega_i \mathbf{f}(\mathbf{x}_i).$$

Using (3.4), we write

$$\begin{aligned} C(\mathbf{f}) &= \sum_{i=1}^{m} \omega_i \big(\sum_j c_j \mathscr{P}_j(\mathbf{x}_i) \big) \\ &= \sum_j c_j \big(\sum_{i=1}^{m} \omega_i \mathscr{P}_j(\mathbf{x}_i) \big). \end{aligned} \tag{3.6}$$

A comparison of (3.5) with (3.6) suggests that when the cubature rule is exact for $\{\mathscr{P}_j(\mathbf{x})\}$, then it also holds that $I(\mathbf{f}) = C(\mathbf{f})$, which proves the proposition. $\square$

To find the unknowns of the cubature rule of degree d, $\{\mathbf{x}_i, \omega_i\}$, we solve a set of moment equations. However, solving the system of moment equations may be more tedious with increasing polynomial degree and/or dimension of the integration domain. For example, an m-point cubature rule entails $m(n+1)$ unknown parameters from its points and weights. In general, a system of $\frac{(n+d)!}{n!d!}$ equations with respect to unknowns from distinct monomials of degree up to d may be formed. For the nonlinear system to have at least one solution (in this case, the system is said to be consistent), we must use at least as many unknowns as equations [19]. That is, m must be chosen to be $m \geq \frac{(n+d)!}{(n+1)!d!}$. Suppose that a cubature rule of degree three is needed when $n = 20$. In this case, we solve $\frac{(20+3)!}{20!3!} = 1771$ nonlinear moment equations; the resulting rule may consist of more than 85 ($> \frac{(20+3)!}{21!3!}$) weighted cubature points.

In order to reduce the size of the system of algebraically independent equations or equivalently the number of cubature points markedly, the *invariant theory* was proposed by Sobolev [108]. The reader may consult [19] and the references therein for a recent account of the invariant theory. The invariant theory discusses how to restrict the structure of a cubature rule by exploiting symmetries of the region of integration and the weighting function. For example, integration regions such as the unit hypercube, the unit hypersphere, and the unit simplex all exhibit some form of symmetry. Hence, it is reasonable to look for cubature rules sharing the same form of symmetry. For the case considered above ($n = 20$ and $d = 3$), using the invariant theory, a cubature rule consisting of $40(= 2n)$ cubature points can be constructed by

solving only a pair of moment equations as described in Section 3.3.2.

3.3 Cubature Rules for Gaussian-Weighted Integrals

This section presents a methodology for constructing a cubature rule for integrals whose integrands are all of the form *nonlinear function*$\times$*Gaussian*. For the ease of computational manipulations, however, the focus is first confined to an integral of the form

$$I(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) \exp(-\mathbf{x}^T\mathbf{x}) d\mathbf{x}. \tag{3.7}$$

To numerically compute (3.7), first, it is converted into a spherical-radial integration form. Subsequently, a third-degree spherical-radial rule is proposed. Later in this section, this spherical-radial cubature rule will be extended to numerically compute integrals whose weighting function is arbitrary Gaussian.

3.3.1 Transformation

The spherical-radial transformation involves a change of variables from the Cartesian vector $\mathbf{x} \in \mathbf{R}^n$ to the radius vector r, and the direction vector $\mathbf{y}$ as follows: Let $\mathbf{x} = r\mathbf{y}$ with $\mathbf{y}^T\mathbf{y} = 1$, so that $\mathbf{x}^T\mathbf{x} = r^2$ for $r \in [0, \infty)$. Afterwards, the integral (3.7) can be rewritten in the spherical-radial coordinate system as

$$I(\mathbf{f}) = \int_0^\infty \int_{U_n} \mathbf{f}(r\mathbf{y}) r^{n-1} \exp(-r^2) d\sigma(\mathbf{y}) dr, \tag{3.8}$$

where U_n is the surface of the sphere defined by $U_n = \{\mathbf{y} \in \mathbb{R}^n | \ \mathbf{y}^T\mathbf{y} = 1\}$ and $\sigma(.)$ is the spherical surface measure or the area element on U_n. We may thus write the *radial integral*

$$I = \int_0^\infty S(r) r^{n-1} \exp(-r^2) dr, \tag{3.9}$$

where $S(r)$ is defined by the *spherical integral* with the unit weighting function $w(\mathbf{y}) = 1$:

$$S(r) = \int_{U_n} \mathbf{f}(r\mathbf{y}) d\sigma(\mathbf{y}). \tag{3.10}$$

The spherical and the radial integrals are numerically computed by the spherical cubature rule (Subsection 3.3.2) and the Gaussian quadrature rule (Subsection 3.3.3), respectively.

Before proceeding further, a number of notations and definitions are used in the following sections/derivations:

- A cubature rule is said to be *fully symmetric* if the following two conditions hold [79]:

  1. $\mathbf{x} \in \mathscr{D}$ implies $\mathbf{y} \in \mathscr{D}$, where $\mathbf{y}$ is any point obtainable from $\mathbf{x}$ by permutations and/or sign changes of the coordinates of $\mathbf{x}$.
  2. $w(\mathbf{x}) = w(\mathbf{y})$ on the region $\mathscr{D}$. That is, all points in the fully symmetric set yield the same weight value.

  For example, in the one-dimensional space, a point $x \in \mathbb{R}$ in the fully symmetric set implies that $(-x) \in \mathbb{R}$ and $w(x) = w(-x)$.

- In a fully symmetric region, a point $\mathbf{u}$[1] is called a *generator* if

$$\mathbf{u} \;=\; (u_1, u_2, \ldots, u_r, 0, \ldots 0) \in \mathbb{R}^n, \quad u_i \geq u_{i+1} > 0, \quad i = 1, 2, \ldots (r-1).$$

- For brevity, $(n-r)$ zero coordinates are suppressed and the notation $[u_1, u_2, \ldots u_r]$ is used to represent a complete fully symmetric set of points that can be obtained by permutating and changing the sign of the generator $\mathbf{u}$ in all possible ways. Of course, the complete set entails $\frac{2^r n!}{(n-r)!}$ points when $\{u_i\}$ are all distinct. For example, $[1] \in \mathbb{R}^2$ represents the following set of points:

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.$$

  Here, the generator is $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

- The notation $[u_1, u_2, \ldots u_r]_i$ is used to denote the i-th point from the set $[u_1, u_2, \ldots u_r]$.

3.3.2 Spherical Cubature Rule

In this subsection, a third-degree spherical cubature rule is derived for numerically computing (3.10). Due to the invariant theory, this cubature rule can be written in the form

$$\int_{U_n} \mathbf{f}(\mathbf{y}) d\sigma(\mathbf{y}) \;\approx\; \omega \sum_{i=1}^{2n} \mathbf{f}[u]_i. \tag{3.11}$$

[1]The new $\mathbf{u}$ should not be confused with the control input $\mathbf{u}$ used in Chapter 2.

The point set due to $[u]$ is invariant under permutations and sign changes. For the above choice of the rule (3.11), the monomials $\{y_1^{d_1} y_2^{d_2} \dots y_n^{d_n}\}$ with $\sum_{i=1}^n d_i$ being an odd integer are integrated exactly. For this rule to be exact for all monomials of degree up to three, it remains to require that the rule is exact for all monomials for which $\sum_{i=1}^n d_i = 0, 2$. Equivalently, to find the unknown parameters u and ω, it suffices to consider monomials $\mathbf{f}(\mathbf{y}) = 1$, and $\mathbf{f}(\mathbf{y}) = y_1^2$ due to the fully symmetric cubature rule:

$$\mathbf{f}(\mathbf{y}) = 1 : \quad 2n\omega = \int_{U_n} d\sigma(\mathbf{y}) = A_n \tag{3.12}$$

$$\mathbf{f}(\mathbf{y}) = y_1^2 : \quad 2\omega u^2 = \int_{U_n} y_1^2 d\sigma(\mathbf{y}) = \frac{A_n}{n}, \tag{3.13}$$

where the surface area of the unit sphere [32]:

$$A_n = \frac{2\sqrt{\pi^n}}{\Gamma(n/2)},$$

with the gamma function $\Gamma(n) = \int_0^\infty x^{n-1} \exp(-x) dx$. The values on the right-hand side of (3.12)-(3.13) are based on the fact that the integration of a monomial of the form $\mathscr{P}(\mathbf{x}) = x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}$ over the n-dimensional unit sphere can be expressed in terms of the gamma function [32]:

$$\int_{U_n} \mathscr{P}(\mathbf{x}) d\sigma(\mathbf{x}) = \begin{cases} 0, & \text{if any } \tilde{d}_i \text{ is odd;} \\ \frac{2\Gamma(\tilde{d}_1)\Gamma(\tilde{d}_2)\dots\Gamma(\tilde{d}_n)}{\Gamma(\tilde{d}_1+\tilde{d}_2+\dots+\tilde{d}_n)}, & \text{if all } \tilde{d}_i \text{ are even,} \end{cases}$$

where $\tilde{d}_i = \frac{1}{2}(1 + d_i)$. Solving (3.12)-(3.13) yields

$$\begin{aligned} u^2 &= 1 \\ \omega &= \frac{A_n}{2n}. \end{aligned}$$

Therefore, the cubature points are located at the intersection of the unit sphere and its axes.

3.3.3 Radial Rule

In this subsection, a Gaussian quadrature is derived for numerically computing (3.9). First, to transform this integral into an integral for which the solution is familiar, another change of variable is made via $t = x^2$, and the following relationship is obtained:

$$\int_0^\infty f(x)x^{n-1}\exp(-x^2)dx = \frac{1}{2}\int_0^\infty \tilde{f}(t)t^{(\frac{n}{2}-1)}\exp(-t)dt, \tag{3.14}$$

where $\tilde{f}(t) = f(\sqrt{t})$. The integral on the right-hand side of (3.14) is now in the form of the well-known generalized Gauss-Laguerre formula. A first-degree Gauss-Laguerre rule is exact for $\tilde{f}(t) = 1, t$. Equivalently, the rule is exact for $f(x) = 1, x^2$; it is not exact for odd degree polynomials such as $f(x) = x, x^3$. Fortunately, when the radial-rule is combined with the spherical rule to compute the integral (3.7), the (combined) spherical-radial rule vanishes for all odd-degree polynomials; the reason is that the spherical rule vanishes by symmetry for any odd-degree polynomial (see (3.8)). Hence, the spherical-radial rule for (3.7) is exact for all odd degrees. Following this argument, for a spherical-radial rule to be exact for all third-degree polynomials

in $\mathbf{x} \in \mathbb{R}^n$, it suffices to consider the first-degree generalized Gauss-Laguerre rule of the form

$$\int_0^\infty \tilde{f}_i(t) t^{\frac{n}{2}-1} \exp(-t) dt \;=\; \omega_1 \tilde{f}_i(t_1), \quad i = 0, 1$$

where $\tilde{f}_0(t) = 1$ and $\tilde{f}_1(t) = t$. Hence, solving the moment equations

$$\begin{aligned} \tilde{f}_0(t) = 1: \qquad \omega_1 &= \int_0^\infty t^{\frac{n}{2}-1} \exp(-t) dt = \Gamma(\frac{n}{2}) \\ \tilde{f}_1(t) = t: \quad \omega_1 t_1 &= \int_0^\infty t^{\frac{n}{2}} \exp(-t) dt = \Gamma(\frac{n+1}{2}) = \frac{n}{2}\Gamma(\frac{n}{2}), \end{aligned}$$

yields $t_1 = \frac{n}{2}$ and $\omega_1 = \Gamma(\frac{n}{2})$. Consequently, the integral of interest (3.9) is approximated as follows:

$$\int_0^\infty f(x) x^{n-1} \exp(-x^2) dx \;\approx\; \frac{1}{2}\Gamma(\frac{n}{2}) f(\sqrt{\frac{n}{2}}). \tag{3.15}$$

3.3.4 Spherical-Radial Rule

In this subsection, two useful results are derived for (i) combining the spherical and radial rule obtained separately, and (ii) extending the spherical-radial rule for a Gaussian-weighted integral. The respective results are presented as two propositions:

Proposition 3.2: *Let the function* $\mathbf{f}(\mathbf{x})$ *be a monomial of degree d such that*

$$\mathbf{f}(\mathbf{x}) \;=\; x_1^{d_1} x_2^{d_2} \ldots x_n^{d_n},$$

where $\{d_i\}$ *are non-negative integers with* $\sum_{i=1}^{n} d_i = d$. *Next, consider the problem of finding a cubature rule for integrals of the form*:

$$I(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) \exp(-\mathbf{x}^T\mathbf{x}) d\mathbf{x}. \tag{3.16}$$

Let the m_s*-point spherical rule of degree* d *be used to numerically compute the spherical integral portion of* (3.16) (compare (3.10)):

$$S(r) = \int_{U_n} \mathbf{f}(r\mathbf{s}) d\sigma(\mathbf{s}) = \sum_{j=1}^{m_s} b_j \mathbf{f}(r\mathbf{s}_j).$$

Let the m_r*-point Gaussian quadrature rule of degree* $(d-1)/2$ *be used to numerically compute the radial integral portion of* (3.16) (compare (3.9)):

$$\int_0^\infty S(r) r^{n-1} \exp(-r^2) dr = \sum_{i=1}^{m_r} a_i S(r_i).$$

Then, the $(m_s \times m_r)$*-point spherical-radial cubature rule of degree* d *that numerically computes*

$$I(\mathbf{f}) = \sum_{j=1}^{m_s} \sum_{i=1}^{m_r} a_i b_j \mathbf{f}(r_i \mathbf{s}_j). \tag{3.17}$$

Proof: The integral of interest $I(\mathbf{f})$ in (3.16) is rewritten as

$$I(\mathbf{f}) = \int_{\mathbb{R}^n} x_1^{d_1} x_2^{d_2} \dots x_n^{d_n} \exp(-\mathbf{x}^T\mathbf{x}) d\mathbf{x}.$$

Making the change of variable as described in Subsection 3.3.1, we get

$$I(\mathbf{f}) = \int_0^\infty \int_{U_n} (ry_1)^{d_1}(ry_2)^{d_2}\ldots(ry_n)^{d_n} r^{n-1}\exp(-r^2)d\sigma(\mathbf{y})dr.$$

Decomposing the above integral into the radial and spherical integrals yields

$$I(\mathbf{f}) = \int_0^\infty r^{n+d-1}\exp(-r^2)dr \int_{U_n} y_1^{d_1} y_2^{d_2}\ldots y_n^{d_n} d\sigma(\mathbf{y}).$$

Applying the radial and spherical rules appropriately leads to

$$\begin{aligned} I(\mathbf{f}) &= \Big(\sum_{i=1}^{m_r} a_i r_i^d\Big)\Big(\sum_{j=1}^{m_s} b_j s_{j1}^{d_1} s_{j2}^{d_2}\ldots s_{jn}^{d_n}\Big) \\ &= \sum_{i=1}^{m_r}\sum_{j=1}^{m_s} a_i b_j (r_i s_{j1})^{d_1}(r_i s_{j2})^{d_2}\ldots(r_i s_{jn})^{d_n} \\ &= \sum_{i=1}^{m_r}\sum_{j=1}^{m_s} a_i b_j \mathbf{f}(r_i \mathbf{s}_j), \end{aligned}$$

which proves the proposition (see also Section 2.8 in [115]). □

Remark: The above proposition holds for monomials of degree less than d. Therefore, from Proposition 3.1, we may say that Proposition 3.2 holds for any arbitrary integrand that can be expressed as a linear combination of monomials of degree up to d.

Proposition 3.3: *Let the weighting functions* $w_1(\mathbf{x})$ and $w_2(\mathbf{x})$ be $w_1(\mathbf{x}) = \exp(-\mathbf{x}^T\mathbf{x})$ *and* $w_2(\mathbf{x}) = \mathcal{N}(\mathbf{x};\mu,\Sigma)$. *Then for every square matrix* $\sqrt{\Sigma}$ *such that* $\sqrt{\Sigma}\sqrt{\Sigma}^T = \Sigma$, we have

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})w_2(\mathbf{x})d\mathbf{x} = \frac{1}{\sqrt{\pi^n}}\int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{x}+\mu)w_1(\mathbf{x})d\mathbf{x}. \tag{3.18}$$

Proof: Consider the left-hand side of (3.18). Because Σ is a positive definite matrix, we factorize Σ to be $\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}^T$. Hence, we write

$$\begin{aligned}\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x};\mu,\Sigma)d\mathbf{x} &= \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\frac{1}{\sqrt{|2\pi\Sigma|}}\exp\big(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\big)d\mathbf{x} \\ &= \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\frac{1}{\sqrt{|2\pi\Sigma|}}\exp\Big(-\Big[(\sqrt{2\Sigma})^{-1}(\mathbf{x}-\mu)\Big]^T \\ &\qquad \times \Big[(\sqrt{2\Sigma})^{-1}(\mathbf{x}-\mu)\Big]\Big)d\mathbf{x}.\end{aligned}$$

Making a change of variable via $\mathbf{x} = (\sqrt{2\Sigma}\mathbf{y} + \mu)$ yields

$$\begin{aligned}\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x};\mu,\Sigma)d\mathbf{x} &= \int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{y}+\mu)\frac{1}{\sqrt{|2\pi\Sigma|}}\exp(-\mathbf{y}^T\mathbf{y})|\sqrt{2\Sigma}|d\mathbf{y} \\ &= \frac{1}{\sqrt{\pi^n}}\int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{y}+\mu)w_1(\mathbf{y})d\mathbf{y} \\ &= \frac{1}{\sqrt{\pi^n}}\int_{\mathbb{R}^n} \mathbf{f}(\sqrt{2\Sigma}\mathbf{x}+\mu)w_1(\mathbf{x})d\mathbf{x},\end{aligned}$$

which proves the proposition. □

In what follows, the third-degree spherical-radial cubature rules are derived for integrals having three different weighting functions:

- **Case 1**. First, consider the problem of finding a spherical-radial cubature rule for the integral:

$$I(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\exp(-\mathbf{x}^T\mathbf{x})d\mathbf{x}. \tag{3.19}$$

  By exploiting the results obtained in Subsections 3.3.2, 3.3.3 and Proposition 3.2, (3.19) can be approximated using the third-degree spherical-radial cubature

rule as shown by

$$I(\mathbf{f}) \approx \frac{\sqrt{\pi^n}}{2n}\sum_{i=1}^{2n}\mathbf{f}\big(\sqrt{\frac{n}{2}}[1]_i\big). \tag{3.20}$$

- **Case 2**. Next, consider the problem of numerically computing a standard Gaussian-weighted integral of the form

$$I_{\mathcal{N}}(\mathbf{f}) = \int_{\mathbb{R}^n}\mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x};\mathbf{0},\mathbf{I})d\mathbf{x}. \tag{3.21}$$

Substituting $\mathbf{I}$ and $\mathbf{0}$ for Σ and μ, respectively, in Proposition 3.3 yields

$$I_{\mathcal{N}}(\mathbf{f}) = \frac{1}{\sqrt{\pi^n}}\int_{\mathbb{R}^n}\mathbf{f}(\sqrt{2}\mathbf{x})\exp(-\mathbf{x}^T\mathbf{x})d\mathbf{x}.$$

This suggests that the cubature rule in (3.20) can be utilized to approximate $I_{\mathcal{N}}(\mathbf{f})$:

$$\begin{aligned} I_{\mathcal{N}}(\mathbf{f}) &\approx \frac{1}{\sqrt{\pi^n}}\Big(\frac{\sqrt{\pi^n}}{2n}\sum_{i=1}^{2n}\mathbf{f}\big(\sqrt{2}\sqrt{\frac{n}{2}}[1]_i\big)\Big) \\ &= \sum_{i=1}^{m}\omega_i\mathbf{f}(\xi_i), \end{aligned}$$

where $m = 2n$, and the cubature point-weight set in the Cartesian coordinate system is defined by

$$\begin{aligned} \xi_i &= \sqrt{\frac{m}{2}}[1]_i \\ \omega_i &= \frac{1}{m}, \qquad i = 1, 2, \ldots m. \end{aligned} \tag{3.22}$$

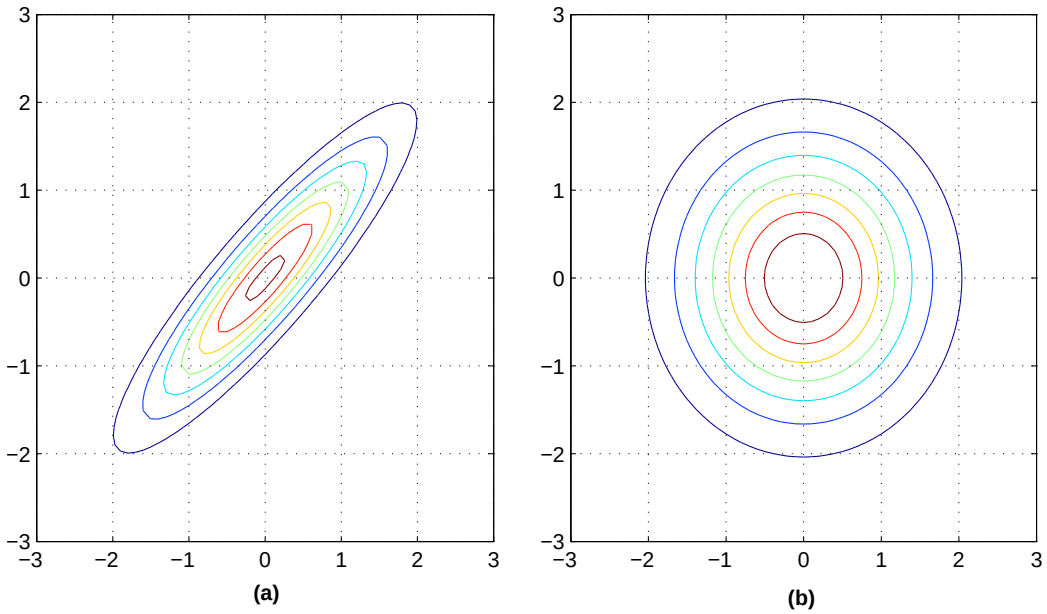


Figure 3.1: Contour plots of zero-mean Gaussian densities with covariances [1 0.9; 0.9 1] (Figure 1(a)) and [1 0; 0 1] (Figure 1(b)), respectively. Using the change of variables that transforms Figure 1(a) into Figure 1(b), we make the Gaussian-weighting function spread equally

This is an important result– In the next two chapters, this cubature point-weight set will be used to build a (square-root) cubature Kalman filter.

- **Case 3**. Finally, by making a change of variable via $\mathbf{x} = \sqrt{\Sigma}\mathbf{y} + \mu$, and following the steps in a manner similar to the proof of Proposition 3.3, it can be shown that

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x}; \mu, \Sigma)d\mathbf{x} \quad = \quad \int_{\mathbb{R}^n} \mathbf{f}(\sqrt{\Sigma}\mathbf{x} + \mu)\mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{I})d\mathbf{x}.$$

In light of the above equality relationship, the cubature point set defined in (3.22) can easily be exploited to numerically compute integrals whose weighting

functions are arbitrary Gaussian. That is,

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x};\mu,\Sigma)d\mathbf{x} \;\approx\; \sum_{i=1}^{m} \omega_i \mathbf{f}(\sqrt{\Sigma}\xi_i + \mu).$$

The change of variable $\mathbf{y} = (\sqrt{\Sigma})^{-1}(\mathbf{x} - \mu)$ eliminates the correlation of a non-standard Gaussian density and makes it to spread equally in all directions as depicted in Figure 3.1. This means that the cubature points are symmetrically distributed in the area of integration, and it is up to the nonlinear integrand function to decide which subset of cubature points should be covered at most. Of course, for improved numerical accuracy, it is important that cubature points overlap with the bulk of an integrand [68, 120].

Summary

In this chapter, a number of well-known numerical integration methods are briefly reviewed. Of those, monomial-based cubature rules can yield reasonably accurate results with a minimal amount of computations. To numerically compute integrals whose integrands are of the form (*nonlinear function* $\times$ *Gaussian*), a third-degree monomial-based cubature rule is derived. This cubature rule uses $2n$ number of equally weighted cubature points, where n denotes the dimension of the integration region. These cubature points are independent of the nonlinear integrand function, and can therefore be precomputed and stored to speed up a filter's execution.

Chapter 4

Cubature Kalman Filtering

In this chapter, as an approximate solution to the Bayesian filter, the cubature Kalman filter (CKF) is developed under the Gaussian assumption. The Gaussian is the most convenient and widely used density for the following reasons:

- It has many distinctive mathematical properties.
  - The Gaussian family is closed under a linear transformation and conditioning.
  - Uncorrelated jointly Gaussian random variables are independent.
- It approximates many physical random phenomena by virtue of the central limit theorem of probability (see Sections 5.7 and 6.7 in [112] for more details).
- Remarkably, according to the *maximum entropy principle*, given the first two order statistics of a hidden process, it is Gaussian that maximizes the information entropy criterion of that process (Ch. 3, [16]).

Consider the discrete-time state-space model given by (2.1) and (2.2). For this state-space model, due to the maximum entropy principle, we are motivated to assume that the predictive density of the joint state-measurement vector is Gaussian. Under this key assumption, it also holds that the predictive state density and the innovations density are Gaussian. More over, this assumption leads to a Gaussian posterior density, thereby allowing all conditional densities to be closed in every update cycle. This is the very assumption made in all innovations-based algorithms including the CKF. In this line of formulation, with a small number of multi-dimensional integrals at our disposal, the functional recursions of the Bayesian filter reduce to some matrix algebraic recursions. That is, the resulting approximate filter now operates only on the means and covariances of the conditional densities encountered in the time and measurement updates as described in the sequel.

4.1 Time Update

In the time update, the filter computes the mean $\hat{\mathbf{x}}_{k|k-1}$ and the associated covariance $P_{k|k-1}$ of the Gaussian predictive density as described in Subsection 2.2. For completeness, however, the final formulae are reproduced as follows (see (2.10)-(2.11)):

$$\hat{\mathbf{x}}_{k|k-1} = \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}) d\mathbf{x}_{k-1} \tag{4.1}$$

$$P_{k|k-1} = \int_{\mathbb{R}^{n_x}} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathbf{f}^T(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1}) d\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k|k-1} \hat{\mathbf{x}}_{k|k-1}^T + Q_{k-1}. \tag{4.2}$$

4.2 Innovations Process

This section describes the innovations process, which provides new information for the purpose of updating the predicted state estimate. Recall the definition of the innovation at time k, denoted by $\boldsymbol{\epsilon}_k$, from Chapter 2:

$$\boldsymbol{\epsilon}_k \;=\; \mathbf{z}_k - \mathbb{E}[\mathbf{z}_k|D_{k-1}]. \tag{4.3}$$

The innovations sequence $\{\boldsymbol{\epsilon}_i, i = 1, 2, \ldots k\}$ obtained from the measurement sequence $\{\mathbf{z}_i, i = 1, 2 \ldots k\}$ has a number of hallmark features, which may not be present in the raw measurements or the estimated states:

- *Zero mean.* The innovation has zero mean:

$$\mathbb{E}[\boldsymbol{\epsilon}_i] \;=\; \mathbf{0}$$

- *White sequence.* An important property of the innovations sequence is that it is white (also called uncorrelated or orthogonal):

$$\mathbb{E}[\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_{i-j}^T] \;=\; \mathbf{0}, \quad j < i, \quad j \neq 0$$

  We may therefore say that the innovation is the 'whitened' measurement.

- *Uncorrelated with past measurements.*

$$\mathbb{E}[\boldsymbol{\epsilon}_i \mathbf{z}_{i-j}^T] \;=\; \mathbf{0} \Rightarrow \mathrm{cov}[\boldsymbol{\epsilon}_i, \mathbf{z}_{i-j}] = \mathbf{0}, \quad 0 < j < i$$

- *Informationally equivalent.* Under the assumption that the predicted measurement $\hat{\mathbf{z}}_{k|k-1}$ is the linear minimum mean square error estimate of $\mathbf{z}_k$ given the past measurement history $D_{k-1} = \{\mathbf{z}_1,\ \mathbf{z}_2 \dots \mathbf{z}_{k-1}\}$, the sequence of measurements $\{\mathbf{z}_1, \mathbf{z}_2, \dots \mathbf{z}_k\}$ and its corresponding sequence of innovations $\{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \dots \boldsymbol{\epsilon}_k\}$ can be determined from each other by a causal and causally invertible linear transformation. In this case, it also indicates that the innovations sequence completely preserves information contained in the measurement sequence. Thus, we may write

$$\{\mathbf{z}_1, \mathbf{z}_2, \dots \mathbf{z}_k\} \ \rightleftharpoons \ \{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \dots \boldsymbol{\epsilon}_k\}.$$

The proofs of the above statistical properties of the innovations process can be found elsewhere [59, 60]. It is the combination of the above statistical properties that eases the development of many powerful algorithms. For example, it is worth mentioning a few innovations-based derivations such as the Kalman-Bucy filter, the Kalman smoother [55, 56], and the signal estimator based on second-order statistics (without a state-space model) [57, 86].

4.3 Measurement Update

The derivation of the CKF's measurement update is rooted in the assumption that the predictive density of the joint state-measurement is Gaussian, and the following lemma that helps fuse the current measurement with the predicted state estimate [9].

Lemma 4.1: *Let the two random variables* $(\mathbf{x}, \mathbf{z})$ *be jointly Gaussian*:

$$p(\mathbf{x}, \mathbf{z}) = \mathcal{N}\Big(\begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix}; \begin{pmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{z}} \end{pmatrix}, \begin{pmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{pmatrix} \Big).$$

Then the conditional density $p(\mathbf{x}|\mathbf{z})$ *is Gaussian distributed*:

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z})} = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}},\ P_{xx|z}),$$

where the conditional mean and its corresponding covariance are given by

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbb{E}[\mathbf{x}|\mathbf{z}] = \bar{\mathbf{x}} + P_{xz}P_{zz}^{-1}(\mathbf{z} - \bar{\mathbf{z}}) \\ P_{xx|z} &= \mathrm{cov}[\mathbf{x}|\mathbf{z}] = P_{xx}^{-1} - P_{xz}P_{zz}^{-1}P_{zx}. \end{aligned}$$

Proof: Because $(\mathbf{x}, \mathbf{z})$ is jointly Gaussian, the marginal density $p(\mathbf{z})$ is Gaussian distributed. Due to Bayes' rule, we write the conditional density

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}) &= \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z})} \\ &= \frac{\sqrt{|2\pi P_{zz}|}\exp(-\frac{1}{2}(\mathbf{y} - \bar{\mathbf{y}})^T P_{yy}^{-1}(\mathbf{y} - \bar{\mathbf{y}}))}{\sqrt{|2\pi P_{yy}|}\exp(-\frac{1}{2}(\mathbf{z} - \bar{\mathbf{z}})^T P_{zz}^{-1}(\mathbf{z} - \bar{\mathbf{z}}))}, \end{aligned} \tag{4.4}$$

where $\mathbf{y} = [\mathbf{x}^T\ \mathbf{z}^T]^T$. In order to reduce the non-zero mean random variables $\mathbf{x}$ and $\mathbf{z}$ to zero-mean variables, we make the change of variables via

$$\boldsymbol{\xi} = \mathbf{x} - \bar{\mathbf{x}} \tag{4.5}$$

$$\boldsymbol{\eta} = \mathbf{z} - \bar{\mathbf{z}}. \tag{4.6}$$

The exponent on the right-hand side of (4.4) has a quadratic form, as shown by

$$\begin{aligned} q &= \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix}^T \begin{pmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix} - \boldsymbol{\eta}^T P_{zz}^{-1} \boldsymbol{\eta} \\ &= \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix}^T \begin{pmatrix} T_{xx} & T_{xz} \\ T_{zx} & T_{zz} \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix} - \boldsymbol{\eta}^T P_{zz}^{-1} \boldsymbol{\eta}, \end{aligned} \tag{4.7}$$

where

$$T_{xx}^{-1} = P_{xx} - P_{xz} P_{zz}^{-1} P_{zx} \tag{4.8}$$

$$P_{zz}^{-1} = T_{zz} - T_{zx} T_{xx}^{-1} T_{zx} \tag{4.9}$$

$$T_{xx}^{-1} T_{xz} = -P_{xz} P_{zz}^{-1}. \tag{4.10}$$

The exponent q in (4.7) can further be expanded as follows:

$$\begin{aligned} q &= \boldsymbol{\xi}^T T_{xx} \boldsymbol{\xi} + \boldsymbol{\xi}^T T_{xz} \boldsymbol{\eta} + \boldsymbol{\eta}^T T_{zx} \boldsymbol{\xi} + \boldsymbol{\eta}^T T_{zz} \boldsymbol{\eta} - \boldsymbol{\eta}^T P_{zz}^{-1} \boldsymbol{\eta} \\ &= (\boldsymbol{\xi} + T_{xx}^{-1} T_{xz} \boldsymbol{\eta})^T T_{xx} (\boldsymbol{\xi} + T_{xx}^{-1} T_{xz} \boldsymbol{\eta}) + \boldsymbol{\eta}^T (T_{zz} - T_{zx} T_{xx}^{-1} T_{xz}) \boldsymbol{\eta} - \boldsymbol{\eta}^T P_{zz}^{-1} \boldsymbol{\eta} \\ &= (\boldsymbol{\xi} + T_{xx}^{-1} T_{xz} \boldsymbol{\eta})^T T_{xx} (\boldsymbol{\xi} + T_{xx}^{-1} T_{xz} \boldsymbol{\eta}) \end{aligned} \tag{4.11}$$

Substituting (4.5), (4.6), and (4.10) into the factor on the right-hand side of (4.11) yields

$$\boldsymbol{\xi} + T_{xx}^{-1} T_{xz} \boldsymbol{\eta} = \mathbf{x} - \bar{\mathbf{x}} - P_{xz} P_{zz}^{-1} (\mathbf{z} - \bar{\mathbf{z}}).$$

This suggests that the conditional density $p(\mathbf{x}|\mathbf{z})$ is Gaussian with the conditional

mean

$$\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{z}] = \bar{\mathbf{x}} + P_{xz}P_{zz}^{-1}(\mathbf{z} - \bar{\mathbf{z}})$$

and the corresponding conditional covariance

$$P_{xx|z} = \text{cov}[\mathbf{x}|\mathbf{z}] = T_{xx}^{-1} = P_{xx}^{-1} - P_{xz}P_{zz}^{-1}P_{zx}. \qquad \square$$

Returning to the measurement update, we have found that the innovations are a zero-mean *white* sequence. Under the reasonable assumption that the innovations are Gaussian, they become independent of each other. In this case, we write the innovations density

$$p(\boldsymbol{\epsilon}_k) = \mathcal{N}(\boldsymbol{\epsilon}_k; \mathbf{0}, P_{zz,k|k-1}), \tag{4.12}$$

where the innovations covariance

$$\begin{aligned} P_{zz,k|k-1} &= \mathbb{E}[(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T | D_{k-1}] \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\mathbf{h}^T(\mathbf{x}_k, \mathbf{u}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1})d\mathbf{x}_k \\ &\qquad - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T + R_k, \end{aligned} \tag{4.13}$$

with the predicted measurement

$$\begin{aligned} \hat{\mathbf{z}}_{k|k-1} &= \mathbb{E}[\mathbf{z}_k | D_{k-1}] \\ &= \int_{\mathbb{R}^{n_x}} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1})d\mathbf{x}_k. \end{aligned} \tag{4.14}$$

Rearranging (4.12) yields

$$\begin{aligned} p(\boldsymbol{\epsilon}_k) &= \mathcal{N}(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}; \mathbf{0}, P_{zz,k|k-1}) \\ &= \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}, P_{zz,k|k-1}). \end{aligned} \tag{4.15}$$

Of course, from (4.15), it is understood the innovations density $p(\boldsymbol{\epsilon}_k)$, and the filter likelihood density $p(\mathbf{z}_k|D_{k-1})$ are related by one-to-one transformation.

In order to develop an approximate Bayesian filter, it is further assumed that the predictive density of the joint state-measurement process can be approximated by Gaussian:

$$p\left(\begin{array}{cc} [\mathbf{x}_k^T & \mathbf{z}_k^T]^T | D_{k-1} \end{array} \right) = \mathcal{N}\left(\left(\begin{array}{c} \mathbf{x}_k \\ \mathbf{z}_k \end{array} \right); \left(\begin{array}{c} \hat{\mathbf{x}}_{k|k-1} \\ \hat{\mathbf{z}}_{k|k-1} \end{array} \right), \left(\begin{array}{cc} P_{k|k-1} & P_{xz,k|k-1} \\ P_{xz,k|k-1}^T & P_{zz,k|k-1} \end{array} \right) \right), \tag{4.16}$$

where the cross covariance

$$P_{xz,k|k-1} = \int_{\mathbb{R}^{n_x}} \mathbf{x}_k \mathbf{h}^T(\mathbf{x}_k, \mathbf{u}_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, P_{k|k-1}) d\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \hat{\mathbf{z}}_{k|k-1}^T. \tag{4.17}$$

Due to Bayes' rule, the posterior density is written as

$$p(\mathbf{x}_k|D_k) = p(\mathbf{x}_k|\mathbf{z}_k, D_{k-1}) = \frac{p(\mathbf{x}_k, \mathbf{z}_k|D_{k-1})}{p(\mathbf{z}_k|D_{k-1})}. \tag{4.18}$$

Upon the receipt of a new measurement $\mathbf{z}_k$, and substituting (4.15) and (4.16) into

(4.18) according to Lemma 4.1 yields

$$p(\mathbf{x}_k|D_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, P_{k|k}), \tag{4.19}$$

where

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + W_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \tag{4.20}$$

$$P_{k|k} = P_{k|k-1} - W_k P_{zz,k|k-1} W_k^T \tag{4.21}$$

$$W_k = P_{xz,k|k-1} P_{zz,k|k-1}^{-1}. \tag{4.22}$$

The signal-flow diagram in Figure 4.1 summarizes the steps involved in the recursion cycle of the CKF. The CKF solves the problem of how to compute Gaussian-weighted integrals whose integrands are all of the form (*nonlinear function* $\times$ *Gaussian density*) present in (4.1), (4.2), (4.13), (4.14), and (4.17) using the third-degree spherical-radial cubature point set $\{(\xi_i, \omega_i),\ i = 1, 2, \ldots 2n_x\}$ presented in Subsection 3.3.4.

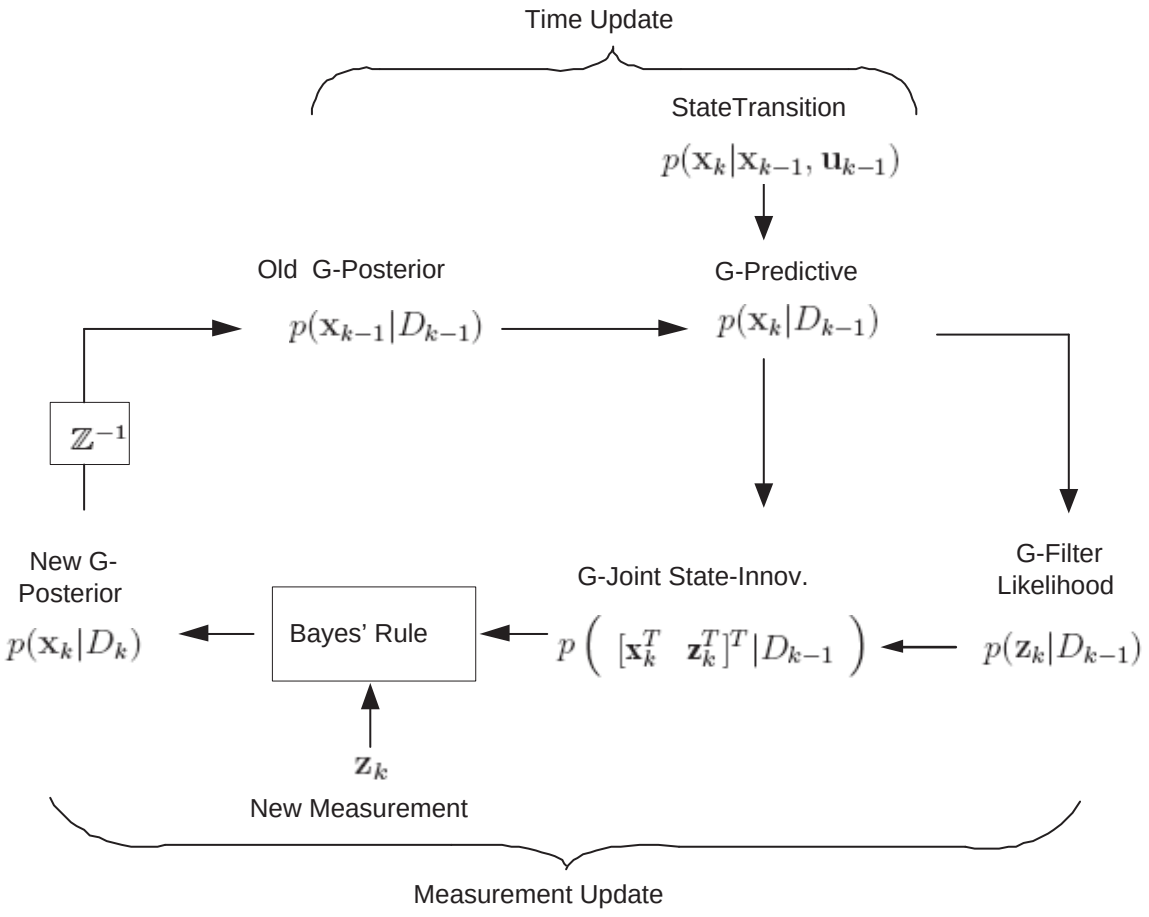


Figure 4.1: Signal-flow diagram of the CKF, where 'G-' stands for 'Gaussian-'.

CKF: Time Update

1. Assume at time k that the posterior density

$$p(\mathbf{x}_{k-1}|D_{k-1}) = \mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1})$$

is known. Factorize

$$P_{k-1|k-1} = S_{k-1|k-1}S_{k-1|k-1}^T. \tag{4.23}$$

2. Evaluate the cubature points (i=1,2,...,m, where $m = 2n_x$)

$$X_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \hat{\mathbf{x}}_{k-1|k-1}. \tag{4.24}$$

3. Evaluate the propagated cubature points (i=1,2,...,m)

$$X^*_{i,k|k-1} = \mathbf{f}(X_{i,k-1|k-1}, \mathbf{u}_{k-1}). \tag{4.25}$$

4. Estimate the predicted state

$$\hat{\mathbf{x}}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^*_{i,k|k-1}. \tag{4.26}$$

5. Estimate the predicted error covariance

$$P_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^*_{i,k|k-1}X^{*T}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}^T_{k|k-1} + Q_{k-1}. \tag{4.27}$$

CKF: Measurement Update

1. Factorize

$$P_{k|k-1} = S_{k|k-1}S_{k|k-1}^T. \tag{4.28}$$

2. Evaluate the cubature points (i=1,2,...,m)

$$X_{i,k|k-1} = S_{k|k-1}\xi_i + \hat{\mathbf{x}}_{k|k-1}. \tag{4.29}$$

3. Evaluate the propagated cubature points (i=1,2,...,m)

$$Z_{i,k|k-1} = \mathbf{h}(X_{i,k|k-1}, \mathbf{u}_k). \tag{4.30}$$

4. Estimate the predicted measurement

$$\hat{\mathbf{z}}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1}. \tag{4.31}$$

5. Estimate the innovations covariance matrix

$$P_{zz,k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1}Z_{i,k|k-1}^T - \hat{\mathbf{z}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T + R_k. \tag{4.32}$$

6. Estimate the cross-covariance matrix

$$P_{xz,k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X_{i,k|k-1}Z_{i,k|k-1}^T - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{z}}_{k|k-1}^T. \tag{4.33}$$

7. Estimate the cubature Kalman gain

$$W_k = P_{xz,k|k-1} P_{zz,k|k-1}^{-1}. \tag{4.34}$$

8. Estimate the updated state

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + W_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}). \tag{4.35}$$

9. Estimate the corresponding error covariance

$$P_{k|k} = P_{k|k-1} - W_k P_{zz,k|k-1} W_k^T. \tag{4.36}$$

4.4 Do We Need Higher-Degree Cubature Rules?

This section emphasizes the importance of a third-degree cubature rule over higher-degree (more than three) cubature rules, when they are embedded in cubature Kalman filtering.

- *Sufficient approximation*: A higher-degree rule will translate to higher accuracy based on two assumptions:
  - The weighting function (conditional density) is known to be a Gaussian density exactly.
  - The integrand is well-behaved in the sense of being approximated by a higher-degree monomial.

From the nonlinear filtering perspective, these two requirements are hardly met. For this reason, *the estimation accuracy of a nonlinear filtering problem becomes more complicated than the notion of accuracy associated with a degree of a cubature rule*. To further elaborate, suppose that the posterior random variable $\mathbf{y}$ and a prior Gaussian random variable $\mathbf{x}$ are related to each other via a nonlinear function

$$\mathbf{y} = \mathbf{f}(\mathbf{x}).$$

When the posterior random variable $\mathbf{y}$ is forced to be Gaussian, despite the nonlinear nature of $\mathbf{f}(.)$, the implicit assumption is that $\mathbf{f}(.)$ can well be approximated by some low-degree monomials (typically up to a second or third degree) in the vicinity of the prior mean. The reason for this approximation is attributed to one of the important properties of the Gaussian family that it is closed under a linear transformation (see the numerical experiment at the end of this section).

- *Numerically attractive computation*: The theoretical lower bound for the number of cubature points of a third-degree cubature rule is $2n$, where n is the dimension of an integration region [93]. Hence, the proposed spherical-radial cubature rule can be considered as a *hyper-efficient* third-degree cubature rule. Because the number of points or function evaluations in the proposed cubature rules scales linearly n, it may be considered as a practical step for easing the curse of dimensionality. In contrast, the theoretical lower bound for a fifth degree cubature rule is in the order of n^2, suggesting that the cubature filter using the

higher-degree cubature rule suffers from the curse of dimensionality [84].

The proposed third-degree spherical-radial cubature rule entails $2n$ equal, positive weights, whereas the construction of a fifth degree rule or any higher-degree rule results in a set of cubature points, some of which have negative weights.[1] When the cubature filter is developed using a cubature rule with negative weights, it suffers from numerical instability. More importantly, it may be impossible for any one to formulate a square-root solution that completely solves the numerical instability issue (see also Appendix B, which describes a number of limitations of the unscented filter due to the inclusion of a negatively weighted point).

In the final analysis, even though it may be tempting to use a higher-degree cubature rule for the CKF's improved accuracy, its use may cause numerical problems and the filter's estimation accuracy may marginally be improved at the expense of an increased computational cost.

4.4.1 Motivating Numerical Example

Consider a Gaussian random variable $\mathbf{z}_p = [r \ \theta]^T$ in the polar coordinate with the mean $\bar{\mathbf{z}}_p$ and covariance Σ_p. The first component of this random variable r corresponds to the 'range' and the second θ to the 'angle' in radians. The random variable $\mathbf{z}_p$ in the polar coordinate is converted to another random variable $\mathbf{z}_c = [x \ y]^T$ in the

[1] According to [96] and Section 1.5 of [115], a 'good' cubature rule has the following two properties: (i) all the cubature points lie inside the region of integration, and (ii) all the cubature weights are positive.

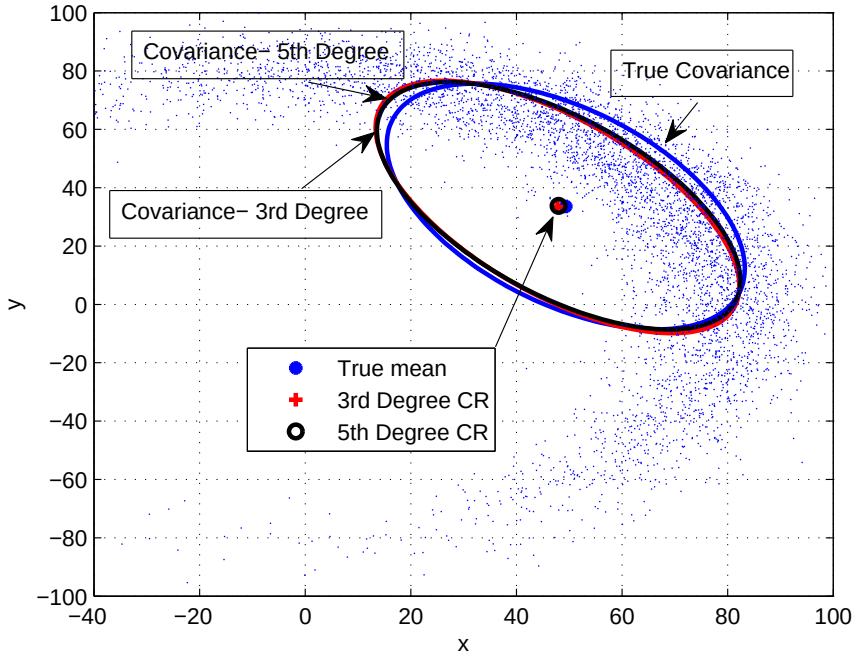


Figure 4.2: First two order statistics of a nonlinearly transformed Gaussian random variable computed by the third and fifth degree cubature rules

Cartesian coordinate using the following nonlinear transformation:

$$\mathbf{z}_c = \mathbf{g}(\mathbf{z}_p) = \begin{pmatrix} r\cos\theta \\ r\sin\theta \end{pmatrix}$$

The objective of this experiment is to numerically compute the second-order statistics of the random variable $\mathbf{z}_c$, given by

$$\begin{aligned} \mathbb{E}[\mathbf{z}_c] &= \int_{\mathbf{z}_p} \mathbf{g}(\mathbf{z})\mathcal{N}(\mathbf{z}; \overline{\mathbf{z}}_p, \Sigma_p)d\mathbf{z} \\ \text{cov}[\mathbf{z}_c] &= \int_{\mathbf{z}_p} \big(\mathbf{g}(\mathbf{z}) - \mathbb{E}[\mathbf{z}_c]\big)\big(\mathbf{g}(\mathbf{z}) - \mathbb{E}[\mathbf{z}_c]\big)^T \mathcal{N}(\mathbf{z}; \overline{\mathbf{z}}_p, \Sigma_p)d\mathbf{z}, \end{aligned}$$

using two different cubature rules, namely,

- the third-degree cubature rule, and
- the fifth degree cubature rule.

Let the mean range $\bar{r}$ be 80 units and the mean angle $\bar{\theta}$ be 0.61 radians, with $\Sigma_p = \text{diag}([60\ 0.6])$. The Monte Carlo method with 5000 samples is used to obtain the true estimate.

Figure 4.2 shows the results of this transformation. The cloud of random samples represents the true density of the random variable $\mathbf{z}_c$. Observe that the transformed variable $\mathbf{z}_c$ is clearly non-Gaussian. It also shows the true mean and covariance (a 2σ-ellipse) of $\mathbf{z}_c$ together with the estimated results employing the cubature rules of degrees three and five. As can be seen from Figure 4.2, the mean values estimated using the third and fifth degree cubature rules lie on top of the true mean. Moreover, the third degree rule yields a covariance contour which is seen to be more closer to that of the fifth degree rule even though both contours slightly deviate from the true contour.

In conclusion, from the cubature filter's perspective, the fifth degree cubature rule does not yield a significant improvement over the third degree rule.

4.5 Computational Cost

Suppose that the processor of an embedded control system has a limited processing speed. In this case, before committing the CKF to this processor, the knowledge of its computational cost is of paramount importance. The unit of the computational cost is typically measured in terms of *flops* (FLoating-point OPeration) counts or simply

flops. One *flop* is defined as one addition, substraction, multiplication, or division of two floating-point numbers. For instance, the product of an $m \times n$ matrix by an $n \times p$ matrix requires mnp multiplications and $mp(n-1)$ additions. Hence, the total computational cost of this matrix multiplication is $mp(2n-1)$ *flops*.

Let n_x, n_z and m denote the state-vector dimension, measurement-vector dimension, and the number of cubature points, respectively. The total computational cost of the CKF is given by

$$\begin{aligned} C(n_x, n_z, m) &= 4mn_x^2 + \frac{2}{3}n_x^3 + mn_x(2n_z+3) + 6n_x^2 \\ &\quad + mn_z(2n_z+1) + n_x n_z(6n_z+1) + \frac{14}{3}n_x \\ &\quad + n_z^3 + 5n_z^2 + \frac{1}{2}n_z - 3 \ \textit{flops}, \end{aligned} \tag{4.37}$$

which is dominated by the outer products of matrices and Cholesky factorizations. In deriving this total computational cost, (i) *flops* required for problem-specific function evaluations are not considered, and (ii) costs such as indexing, fetches and stores, input/output etc., are ignored. The latter cost is difficult to compute. However, provided a state-space model, the *flops* required for the function evaluations can be computed, and added to (4.37).

For a typical filtering problem, it is reasonable to assume n_x to be greater than n_z. Moreover, for the CKF using the third-degree cubature rule, m turns out that $m = 2n_x$. In this case, the implementation of the CKF requires $\frac{26}{3}n_x^3$ *flops* approximately. Note that the computational cost of the EKF also grows cubically in terms of n_x, but the scaling factor is less than that of the CKF [121]. This means that the CKF is computationally slightly more expensive than the EKF.

Remark:

The CKF reduces to the Kalman filter in a linear setting for the following reason: The class of linear functions can be thought of as a linear combination of state-vector elements, and hence belongs to monomials of degree one. Because a cubature rule of degree three can exactly compute integrals whose integrands are all monomials of degree one, the CKF functions exactly similar to the Kalman filter. In this case, the cubature Kalman gain W_k in (4.22) becomes equivalent to the Kalman gain. However, the CKF is computationally three times more expensive than the Kalman filter, which requires only $3n_x^3$ *flops* [61].

Summary

In this chapter, the cubature Kalman filter (CKF) is derived by taking the following two key steps: (i) The theory of the optimal Bayesian filter is reduced to the problem of how to compute various multi-dimensional Gaussian-weighted moment integrals by assuming the predictive density of the joint state-measurement vector to be Gaussian distributed (ii) To numerically compute the Gaussian-weighted moment integrals, a third-degree monomial-based cubature rule, presented in Chapter 3, is employed. To the best of the author's knowledge, this is the first time that the cubature Kalman filtering theory is systematically derived for nonlinear filtering under the above mentioned assumption. It is also justified that a third-degree is adequate for the CKF to perform satisfactorily. The computational cost of the CKF grows cubically with the state-vector dimension.

Chapter 5

Square-Root Cubature Kalman Filtering

The first publicly known use of the extended Kalman filter was made at NASA Ames Research Center in the early 1960s for circumlinear navigation and control of the Apollo space capsule [74]. The key to this successful operation on a small fixed-point on-board computer is attributed to the square-root formulation of the extended Kalman filter. This chapter discusses the need for a similar square-root extension of the CKF. The *square-root cubature Kalman filter* (SCKF) is then derived by systematically manipulating the error covariance matrices. Finally, the additional cost incurred due to this square-root formulation is computed.

5.1 Motivation

The two basic properties of an error covariance matrix are symmetry and positive definiteness. It is important that these two properties be preserved in each update

cycle. In practice, due to errors introduced by arithmetic operations performed on finite word-length digital computers, these two properties are often lost. Specifically, the loss of positive definiteness may probably be more hazardous as it stops the CKF from running continuously. In each update cycle of the CKF, the following numerically sensitive operations may catalyze to destroy the properties of a covariance matrix:

- Matrix square rooting (see (4.23) and (4.28))
- Matrix inversion (see (4.34)
- Matrix squared forms, which are responsible for amplifying roundoff errors (see (4.27), (4.32) and (4.33))
- Substraction of two positive-definite matrices present in the error covariance update (see (4.36))

Moreover, some nonlinear filtering problems may be intrinsically ill-conditioned. In the literature on Bayesian filtering, various ad-hoc methods have been introduced. Some of them include [26]:

- Measurement update with a sequence of scalar measurements in a preferred order
- Decoupled or quasi-decoupled covariances
- Symmetrization of covariances based on the formula $\mathbf{P}' = \frac{\mathbf{P}+\mathbf{P}^T}{2}$
- Computation of only upper triangular entries of covariances
- Tikhonov regularization

- Joseph's covariance update
- Use of large process and measurement noise covariances
- Use of a doubled-precision arithmetic

In contrast, as a systematic solution to preserve the properties of covariances and to mitigate ill effects that may eventually lead to a divergent behavior or even complete breakdown, square-root algorithms which propagate the square-roots of various error covariance matrices have been proposed [61]. Following this line of thinking, we can also structurally reformulate the CKF to develop a square-root version. The SCKF essentially propagates the square roots of the predictive and posterior error covariances, and offers the following benefits as compared to the CKF:

- *Preservation of symmetry and positive (semi)definiteness.* It is proven in [41, 10] that the use of a forced symmetry on the solution of the matrix Ricatti equation improves the apparent numerical stability of the Kalman filter. Since the CKF embraces the Kalman filter theory, the CKF is also expected to improve its stability. Because the underlying meaning of the covariance is embedded in positive definiteness, meaningful conclusions about a filter's performance from its covariance can be drawn, e.g., the *trace* of the covariance yields a confidence measure about the filter's state estimate.

- *Improved numerical accuracy.* In the nonlinear filtering context, errors are likely to be introduced to filter computations from various sources. For example, roundoff errors and catastrophic cancelation errors may occur owing to a large number of nonlinear function evaluations when they are computed in a system with finite arithmetic precision. These errors tend to accumulate over time,

causing the filter to diverge eventually. It is known that the condition number of the covariance matrix P, $\kappa(P)$, and the condition number of the square-root of P, $\kappa(S)$, are related to each other via

$$\kappa(S) = \sqrt{\kappa(P)}.$$

Due to a significantly smaller condition number of S, the final results involving S are expected to be less erroneous than that involving P [39].

- *Doubled-order precision*, whereby the square-root approach yields twice the effective precision of the conventional approach [10]. Loosely speaking, the results given by the square-root approach in single precision are close to those given by the conventional approach in double precision. This property is extremely important because the computational time of single precision operations is substantially less than that of double precision operations.

- *Availability of square roots*. The covariance matrix is likely to turn out to be non-positive definite in a system with finite precision when [61]:

  – highly accurate or precise measurements are processed or

  – some state constraints are imposed.

  In the constrained state estimation case, a set of linearly combined state components is assumed to be known. In this case, the posterior error covariance becomes positive semidefinite; it contains some zero eigen values, reflecting that the filter has perfect estimates along the corresponding eigen vectors. This is undesirable in practical terms. In the SCKF, however, a square root of the

error covariance matrix is available explicitly. For this reason, the SCKF can run continuously even in this undesirable situation.

5.2 Derivation of SCKF

In this section, the SCKF is developed using the least-squares method for the cubature Kalman gain and matrix triangular factorization or triangularization for the covariance updates. The least-squares method avoids the explicit computation of matrix inversion; whereas triangularization essentially computes a triangular square-root factor of the covariance without square rooting a squared-matrix form of the covariance [39]. To elaborate the latter, a covariance matrix P of the form

$$P \;=\; AA^T, \tag{5.1}$$

is considered, where $P \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times m}$ is a 'fat' matrix with $m \geq n$. Hence, the CKF preserves the properties of a covariance matrix, namely, its symmetry and positive-definiteness, provided that no computational errors are made, which is probably not true in practice. Although A in (5.1) can be considered as a square root of P, for computational reasons a triangular matrix of dimension $n \times n$ is desired. This is accomplished by applying a matrix triangularization algorithm:

$$S \;=\; \mathbf{Tria}(A),$$

where $S \in \mathbb{R}^{n\times n}$ is the desired triangular matrix, and '**Tria**' denotes any one of tringularization algorithms. For example, to transform A into S the *QR decomposition*

may be used.[1] When A^T is decomposed into an orthogonal matrix $Q \in \mathbb{R}^{m \times n}$ and an upper triangular matrix $R \in \mathbb{R}^{n \times n}$ such that $A^T = QR$, we get

$$P = AA^T = R^T Q^T Q R = R^T R = SS^T,$$

where $S = R^T$. Note that in computing the square-root covariance S, Q is discarded, and only the upper triangular matrix R is exploited. Since S is a triangular matrix, its sparseness can be exploited for an efficient computation, and reduced storage space.

Next, the SCKF algorithm is summarized. The steps are explicitly written only when they differ from the CKF algorithm presented in Chapter 4. In the SCKF algorithm, the symbol / is used to denote the matrix *right division* operator, which applies the *back substitution* algorithm for an upper triangular matrix S and the *forward substitution* algorithm for a lower triangular matrix S. The matrix right division is used to find the Kalman gain of the SCKF. Moreover, the symbols $S_{Q,k}$, and $S_{R,k}$ are used to denote the square roots of the process noise covariance Q_k, and measurement noise covariance R_k, respectively:

SCKF: Time Update

---

1. Skip the factorization step (4.23) because the square root of the error covariance, $S_{k-1|k-1}$, is available. Compute from (4.24) to (4.26).

2. Estimate the square root of the predicted error covariance

$$S_{k|k-1} = \mathbf{Tria}([\mathscr{X}^*_{k|k-1} \quad S_{Q,k-1}]), \tag{5.2}$$

[1] The QR decomposition has been chosen as one of the top ten algorithms of the computer age [18].

where $S_{Q,k-1}$ denotes a square root of Q_{k-1} such that $Q_{k-1} = S_{Q,k-1}S_{Q,k-1}^T$, and the weighted-centered (prior mean is subtracted off) matrix

$$\begin{aligned}\mathscr{X}_{k|k-1}^* &= \frac{1}{\sqrt{m}}[X_{1,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1} \quad X_{2,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1} \dots \\ &\qquad X_{m,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1}]. \end{aligned} \tag{5.3}$$

SCKF: Measurement Update

1. Skip the factorization step (4.28) because the square root of the error covariance, $S_{k|k-1}$, is available. Compute from (4.29) to (4.31).

2. Estimate the square root of the innovations covariance matrix

$$S_{zz,k|k-1} = \mathbf{Tria}([\mathscr{Z}_{k|k-1} \quad S_{R,k}]) \tag{5.4}$$

where $S_{R,k}$ denotes a square-root factor of R_k such that $R_k = S_{R,k}S_{R,k}^T$, and the weighted-centered matrix

$$\mathscr{Z}_{k|k-1} = \frac{1}{\sqrt{m}}[Z_{1,k|k-1} - \hat{\mathbf{z}}_{k|k-1} \quad Z_{2,k|k-1} - \hat{\mathbf{z}}_{k|k-1} \dots Z_{m,k|k-1} - \hat{\mathbf{z}}_{k|k-1}] \tag{5.5}$$

3. Estimate the cross-covariance matrix

$$P_{xz,k|k-1} = \mathscr{X}_{k|k-1}\mathscr{Z}_{k|k-1}^T, \tag{5.6}$$

where the weighted-centered matrix

$$\mathscr{X}_{k|k-1} = \frac{1}{\sqrt{m}}[X_{1,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \quad X_{2,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \ldots X_{m,k|k-1} - \hat{\mathbf{x}}_{k|k-1}]. \tag{5.7}$$

4. Estimate the square-root cubature Kalman gain

$$W_k = (P_{xz,k|k-1}/S^T_{zz,k|k-1})/S_{zz,k|k-1}. \tag{5.8}$$

5. Estimate the updated state $\hat{\mathbf{x}}_{k|k}$ as in (4.35).

6. Estimate the square root of the corresponding error covariance

$$S_{k|k} = \mathbf{Tria}([\mathscr{X}_{k|k-1} - W_k\mathscr{Z}_{k|k-1} \quad W_k S_{R,k}]). \tag{5.9}$$

All of the above steps in the SCKF are straightforward except the square-root error covariance $S_{k|k}$, which is derived as follows: Substituting (4.34) into (4.36) yields

$$P_{k|k} = P_{k|k-1} - P_{xz,k|k-1}W_k^T. \tag{5.10}$$

After multiplying by $P_{zz,k|k-1}$ on both sides of (4.34), we get

$$P_{xz,k|k-1} = W_k P_{zz,k|k-1}. \tag{5.11}$$

Multiplying by W_k^T on both sides of (5.11), and taking transpose yields

$$W_k P_{xz,k|k-1}^T = W_k P_{zz,k|k-1} W_k^T. \tag{5.12}$$

Adding (5.10) and (5.12) together yields

$$P_{k|k} = P_{k|k-1} - P_{xz,k|k-1} W_k^T + W_k P_{zz,k|k-1} W_k^T - W_k P_{xz,k|k-1}^T. \tag{5.13}$$

Using the fact that $P_{k|k-1} = \mathscr{X}_{k|k-1} \mathscr{X}_{k|k-1}^T$, and substituting (5.4), and (5.6) into (5.13) appropriately yields

$$\begin{aligned} P_{k|k} &= \mathscr{X}_{k|k-1} \mathscr{X}_{k|k-1}^T - \mathscr{X}_{k|k-1} \mathscr{Z}_{k|k-1}^T W_k^T + W_k (\mathscr{Z}_{k|k-1} \mathscr{Z}_{k|k-1}^T + S_{R,k} S_{R,k}^T) W_k^T \\ &\quad - W_k \mathscr{Z}_{k|k-1} \mathscr{X}_{k|k-1}^T \\ &= [\mathscr{X}_{k|k-1} - W_k \mathscr{Z}_{k|k-1} \quad W_k S_{R,k}][\mathscr{X}_{k|k-1} - W_k \mathscr{Z}_{k|k-1} \quad W_k S_{R,k}]^T. \end{aligned} \tag{5.14}$$

The matrix $P_{k|k}$ in (5.14) is the Joseph covariance in disguise [7]. Applying matrix triangularization on (5.14) leads to (5.9).

5.3 Computational Cost due to SCKF

Assuming that the modified Gram Schmidt-based QR decomposition is used as a triangularization algorithm, the computational cost of the SCKF in each recursion

cycle is given by

$$\begin{aligned} C(n_x, n_z, m) &= 6mn_x^2 + 2n_x^3 + mn_x(4n_z + 9) + 2n_x^2 n_z \\ &\quad + \frac{1}{2}mn_z(5n_z + 11) + n_x n_z(4n_z + 1) + n_x \\ &\quad + \frac{7}{3}n_z^3 + \frac{1}{2}n_z^2 + \frac{19}{6}n_z \ \textit{flops}, \end{aligned}$$

where the notations denote the same meaning as in Section 4.5. The above cost is dominated by the matrix triangularization algorithm. When $n_x \gg n_z$, the SCKF approximately costs $14n_x^3$ *flops* in each recursion cycle. As expected, the SCKF performs more computations than what the CKF does. The SCKF will be a desired alternative to the CKF if the application of interest demands stability, and the computational burden is not a major consideration. However, the cost of the SCKF may be reduced significantly by taking the following steps:

- Carefully manipulating the sparsity of square-root covariances owing to their triangular form. For example, in (4.29), the product term $S_{k|k-1}\xi_i$ can be computed by simply pulling out 'some' non-zero elements of the i-th column vector of $S_{k|k-1}$, and scaling by $\sqrt{n_x}$. In so doing, the fact that ξ_i is a scaled elementary vector is used.

- Coding a triangularization algorithm for distributed processor-memory architectures [88].

Summary

We have witnessed many successful applications of the Kalman filter due to its square-root formulation. Although the CKF closely embraces the idea of Kalman filtering, it

is more sensitive to limited precision than the Kalman filter or its extended versions. The reason is that the CKF, by its nature, explicitly requires the square-roots of the error covariances. In this chapter, the CKF is reformulated to recursively propagate the square-roots of the error covariances. In a limited precision system, the resulting square-root filter significantly improves numerical stability and accuracy at the expense of an increased computational cost. The square-root filter performs nearly 60% more computations than what the CKF does.

Chapter 6

Experimental Evaluations

This chapter evaluates the performance of the CKF against other presently known approximate Bayesian filters when applied to the following three nonlinear estimation problems: (i) Target Tracking (ii) Supervised Training of Recurrent Neural Networks (iii) Model-based Signal Processing. In all these experiments, the square-root version of the CKF is employed for its improved numerical stability.

6.1 Tracking a Maneuvering Ship

Scenario. In this first experiment, the problem of tracking a ship moving in an area bounded by a shore line was considered. The shore line was assumed to be a circle of radius r with its center at the origin. The ship was assumed to move at a constant velocity perturbed by white Gaussian noise when it was not 'too far away' from the origin; whereas when it drifted outside of the radius r from the origin, and was headed towards the shore, a gentle turning force pushed it back towards the origin. The model is interesting in that (i) it exhibits significant nonlinear behavior near the shore, and

(ii) it is an example where the various forms of the EKF including the CDKF fail completely.

Following the scenario described in [68], the kinematic of this maneuvering motion was modeled in a discrete-time interval of δ by the following four equations:

$$\begin{aligned} x_{1,t+\delta} &= x_{1,t} + \delta x_{3,t} + q_1 v_{1,t} \\ x_{2,t+\delta} &= x_{2,t} + \delta x_{4,t} + q_1 v_{2,t} \\ x_{3,t+\delta} &= x_{3,t} + f_1(\mathbf{x}_t) + q_2 v_{3,t} \\ x_{4,t+\delta} &= x_{4,t} + f_2(\mathbf{x}_t) + q_2 v_{4,t}, \end{aligned}$$

where

- The state of the ship $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]^T$; x_1 and x_2 denote positions whereas x_3 and x_4 denote velocities in the x and y directions, respectively
- The noise samples, $\{v_i, i = 1, 2 \ldots 4\}$, were assumed to be independent and drawn from the standard Gaussian; $\{q_i, i = 1, 2\}$ were their noise intensities
- The functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are given by

$$f_1(\mathbf{x}) = \begin{cases} \frac{-\lambda x_1}{\sqrt{x_1^2+x_2^2}}, & \sqrt{x_1^2+x_2^2} \geq r \text{ and } x_1x_3 + x_2x_4 \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

$$f_2(\mathbf{x}) = \begin{cases} \frac{-\lambda x_2}{\sqrt{x_1^2+x_2^2}}, & \sqrt{x_1^2+x_2^2} \geq r \text{ and } x_1x_3 + x_2x_4 \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Note that the term $(x_1x_3 + x_2x_4)$ in the last bullet is positive only if the distance of the ship from the origin increases. The functions $\frac{-\lambda x_1}{\sqrt{x_1^2+x_2^2}}$ and $\frac{-\lambda x_2}{\sqrt{x_1^2+x_2^2}}$ should not be

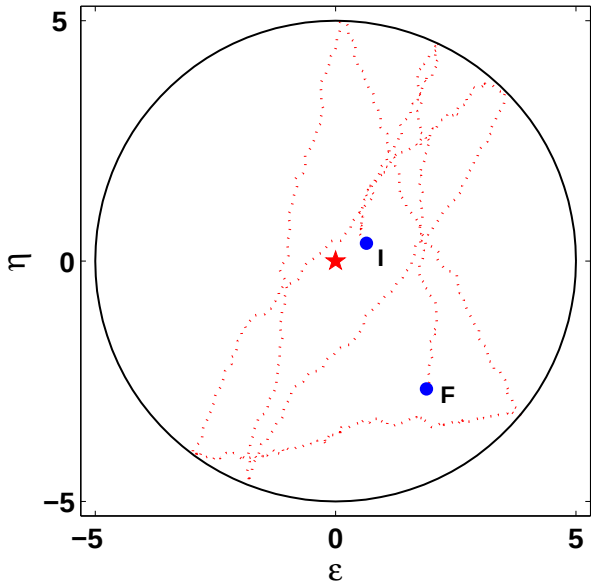


Figure 6.1: Representative trajectory of the ship (I - initial point, F - final point); ★ - Radar location; circle represents the shore line

confused with the notion of deceleration; they denote turning forces with λ denoting the strength of the turning force. When the above constraints are active for a very short period of one or two time steps, the ship does not slow down its motion; rather, it tends to veer and moves away from the shore. Figure 6.1 shows a representative trajectory of the ship, in which it wanders from its initial position, and veers sharply when approaching the shore.

A radar was fixed at the origin of the plane, and equipped to measure the range, r, and the bearing, θ, at a measurement-time interval Δ. The measurement equation is therefore given by

$$\begin{pmatrix} r_k \\ \theta_k \end{pmatrix} = \begin{pmatrix} \sqrt{x_{1,k}^2 + x_{2,k}^2} \\ \tan^{-1}(\frac{x_{2,k}}{x_{1,k}}) \end{pmatrix} + \mathbf{w}_k,$$

where the measurement noise $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, R)$ with $R = \text{diag}[\sigma_r^2 \quad \sigma_\theta^2]$.

Data:

$$\begin{aligned} r &= 5 \\ \Delta &= 0.05 \\ \lambda &= 2 \\ q_1 &= q_2 = 0.01 \\ \sigma_r &= 0.01 \\ \sigma_\theta &= \frac{0.5\pi}{180} \text{ radians} \end{aligned}$$

The initial state estimate $\hat{\mathbf{x}}_{0|0}$ was fixed at $[1\ 1\ 1\ 1]^T$ and the associated covariance $P_{0/0}$ at $10I_4$. The true initial state $\mathbf{x}_0$ was randomly chosen from the unit hyper cube with its center at the origin in each run; the total number of radar scans per run was 800. The units of time and space were assumed to be normalized. For example, when the unit of time is, say 20 minutes, then the radar is assumed to acquire its measurement every 1 minute ($= 20 \times 0.05$). The above parameters were chosen according to [68].

To track the maneuvering ship, the following nonlinear filters were employed:

- Particle filter with resampling after each update cycle (1000 particles)
- Extended Kalman filter (EKF)
- Unscented Kalman filter (UKF)
- Central-difference Kalman filter (CDKF)
- Cubature Kalman filter (CKF)

In order to closely follow the motion of the ship, each filter was assumed to predict five intermediate steps without measurement updates between two consecutive measurements. For a fair comparison, 50 independent trajectories were made. All the above filters were initialized with the same condition in each run.

Performance metrics. To compare various nonlinear filter performances, the root-mean square errors (RMSEs) of the position and velocity were used. The RMSE in position at time k is defined by

$$\text{RMSE}_{\text{pos}}(k) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\big((x_{1,k}(n)-\hat{x}_{1,k}(n))^2+(x_{2,k}(n)-\hat{x}_{2,k}(n))^2\big)},$$

where $(x_{1,k}(n), x_{2,k}(n))$ and $(\hat{x}_{1,k}(n), \hat{x}_{2,k}(n))$ are the true and estimated positions in the n-th Monte Carlo simulation run and $N = 50$. Similarly, the RMSE in velocity can also be written.

Observations. Figures 6.2(a) and 6.2(b) show the RMSEs of various filters in the position and velocity, respectively, during the period of 500 time units. Owing to a number of limitations outlined in Chapter 2 and Appendix B, the EKF diverged, and the UKF often broke down. For this reason, their results are not presented here. From Figures 6.2(a) and 6.2(b), we see that the CKF tracks better than the CDKF and the particle filter all along the trajectory except a short period of time, during which we see some overshoots owing to nonlinear effects around some maneuvers, but the CKF recovers quickly and tracks the trajectory afterwards. Though such errors occurred in a few simulated tracks, they dominated the average error in that short period. Overall, in this low-dimensional problem, the CKF tracks more accurately than others.

Algorithm	Time (s)
EKF	0.2
CDKF	1.0
Particle filter	57.6
CKF	0.6

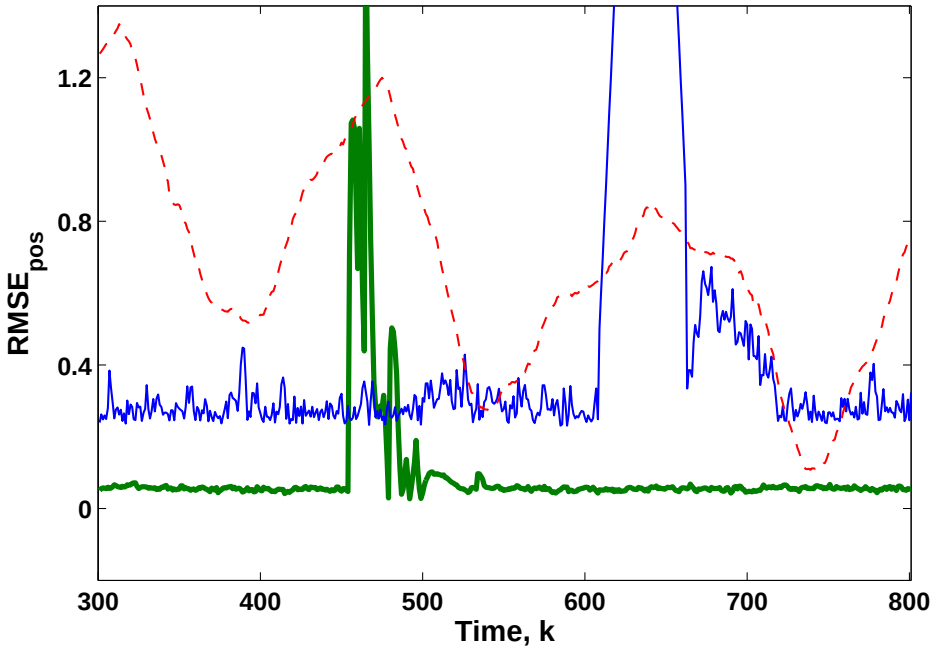
Table 6.1: Comparison of computational times (per track) averaged over 50 simulated tracks

The computational times of the EKF, the CDKF, the CKF, and the particle filter are tabulated in Table 6.1 when implemented on a 2.6-GHz Intel Core2 Duo processor using MATLAB. As expected, the CDKF requires slightly higher computational time than the CKF partly because the CDKF uses a set of $(2n + 1)$ interpolating points, where n is the state-vector dimension. The particle filter requires a computational time of 100 times that of the CKF, approximately.

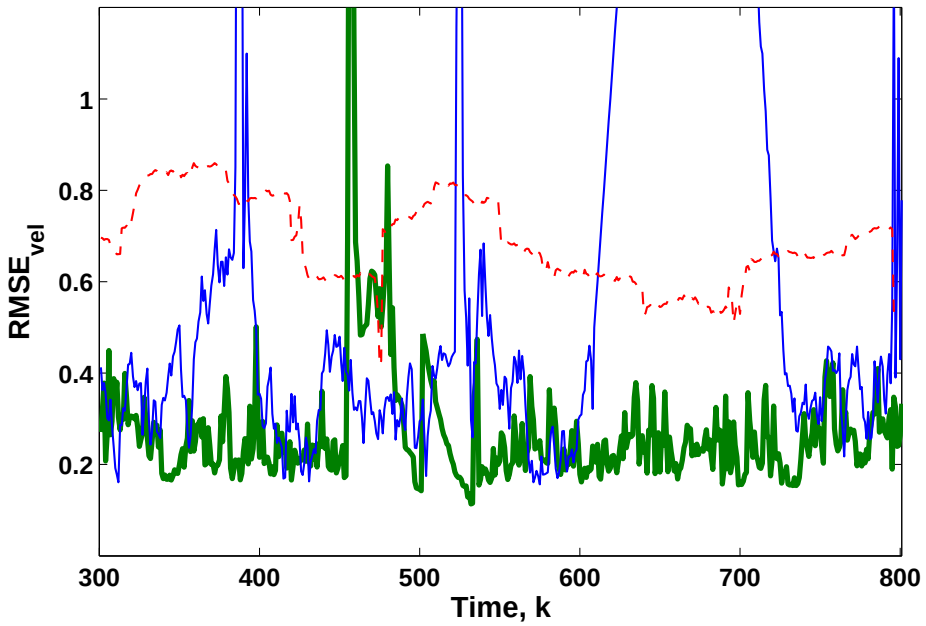
On the basis of a tracking performance-computational time tradeoff, the CKF is the best filter for this tracking problem.

Remark:

The CKF assumes continuous dynamics. When there is a discontinuity in the dynamics of the problem at hand, the underlying theory of the CKF is no longer applicable. To mitigate this problem, we may resort to the use of smoothing. For the tracking problem under consideration, when the ship gets closer to the shore, its motion does not exactly follow the dynamics described by the state-space model because of a set of constraints. In order to improve the tracking performance, a fixed-lag smoother may be employed at the cost of a constant delay (see also Ch. 7 for a discussion on cubature Kalman smoothing).



(a) RMSE in position.



(b) RMSE in velocity.

Figure 6.2: Root mean-squared error (RMSE) averaged over 50 independent trajectories (dotted- Particle filter, solid thin- CDKF, solid thick- CKF)

6.2 Supervised Training of Recurrent Neural Networks

Recurrent neural networks (RNNs) are well known for their use in chaotic dynamic reconstruction, among other applications. Chaotic dynamics are commonly observed in a wide variety phenomena from molecular vibrations to satellite motions. In most cases, the underlying governing equations of chaotic dynamics are difficult to obtain. In such cases, they may be replaced with RNNs. In this experiment, Bayesian filter-trained RNNs were considered. The well-known chaotic Mackey-Glass system was used to generate both the training and test data. The EKF, the CDKF, and the CKF were employed for the purpose of supervised training of RNNs. Particle filters were not considered for the following reason: The supervised training of RNNs involves a large number of weights to be estimated. Hence, an enormous amount of particles is required to completely capture this huge state-space volume as outlined in Chapter 2. Simply put, particle filters are computationally quite demanding for this application.

Chaotic Mackey-Glass Attractor. The Mackey-Glass equation is often used to model the production of white-blood cells in Leukemia patients, and given by the delay differential equation [75]:

$$\frac{du_t}{dt} = 0.1u_t + \frac{0.2u_{t-\Delta}}{1+u_{t-\Delta}^{10}}, \tag{6.1}$$

where the delay $\Delta = 30$. To sample the time-series at discrete time steps, (6.1) was numerically integrated using the forth-order Runge-Kutta method with a sampling period of $T = 6$ s, and initial condition $u_t = 0.9$, for $0 \leq t \leq \Delta$. Given a chaotic

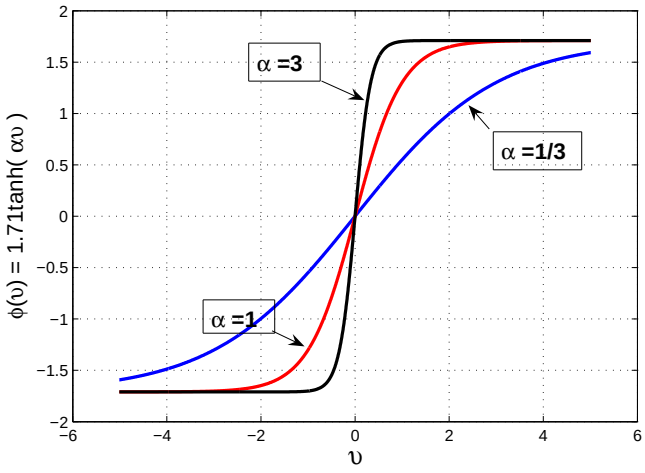


Figure 6.3: Effect of α on the shape of the activation function $\varphi(v) = 1.71\tanh(\alpha v)$.

system, it is known that the next data sample $u_{k+\tau}$ can be predicted from a properly chosen time sequence $\mathbf{u}_k = [u_k \quad u_{k-\tau} \ldots u_{k-[d_E-2]\tau} \quad u_{k-[d_E-1]\tau}]$, where d_E and τ are called the embedding dimension and the embedding delay, respectively. For the chaotic Mackey-Glass system, d_E and τ were chosen to be seven and one, respectively.

RNN Architecture. Bayesian filter-trained RNNs were used to predict the chaotic Mackey-Glass time-series data. The structure of a RNN was chosen to have seven inputs representing an embedding of the observed time-series, one output, and one self-recurrent hidden layer with five neurons. Hence, the RNN has a total of 71 connecting weights (bias included). The linear activation function was used by the output neuron, whereas all the hidden neurons used a hyperbolic tangent function of the form

$$\varphi(v) \quad = \quad 1.71 \tanh(\alpha v),$$

where α was assumed to take values ranging from 1/3 to 3. As shown in Figure 6.3,

the hyperbolic tangent function is 'mildly' nonlinear (that is, close to a linear function) around its origin when $\alpha = 1/3$. Its nonlinearity increases with α, and behaves closely similar to a switch when $\alpha = 3$.

State-Space Model. To estimate the weight parameters using a Bayesian filter, they are typically assumed to be Gaussian random variables. Specifically, the weight variables are assumed to follow the first-order noisy autoregressive model, and the state-space model can therefore be written as

$$\begin{aligned} \mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{q}_{k-1} \\ d_k &= W_o \varphi\big(W_r \mathbf{x}_{k-1} + W_i \mathbf{u}_k\big) + r_k, \end{aligned}$$

where

- The process noise $\mathbf{q}_k$ is assumed to be zero-mean Gaussian with covariance Q_{k-1}
- The measurement noise r_k is assumed to be zero-mean Gaussian with variance R_k
- The internal state of the RNN or the output of the hidden layer at time $(k-1)$ is denoted by $\mathbf{x}_{k-1}$ (Figure 6.4)
- The desired output d_k acts as the measurement
- W_i, W_r and W_o are input, recurrent and output weight matrices of appropriate dimensions; the weight vector $\mathbf{w}_k$ is obtained by grouping elements from W_i, W_r and W_o in 'some' orderly fashion

Data. A chaotic time sequence of length 1000 was generated, the first half of which was used for training and the rest for testing. To train the RNN using the

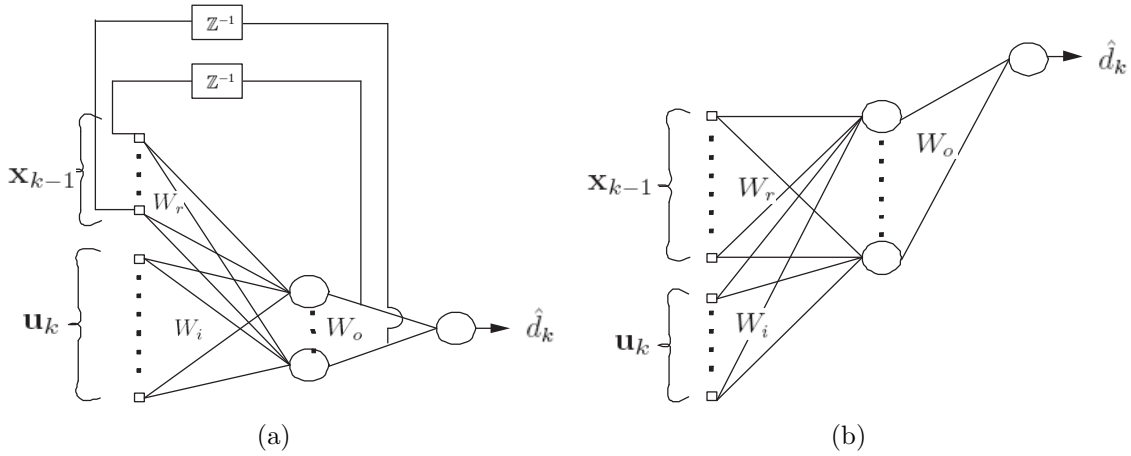


Figure 6.4: Schematic diagrams (a). Original RNN (b). Unfolded RNN of unity truncation depth.

CKF, 10 epochs/run were made. Each epoch was obtained from a 107 time-step long subsequence, starting from a randomly selected point. That is, each epoch consisted of 100 examples, all of which were gleaned by sliding a window of length eight over the subsequence. The weights were initialized to be zero-mean Gaussian with a diagonal covariance of $0.5I_w$; Q_{k-1} was made to decay such that $Q_{k-1} = (\frac{1}{\lambda} - 1)P_{k-1|k-1}$, where $\lambda \in (0, 1)$ is the "forgetting factor" as defined in the recursive least-squares algorithm [44]; this approximately assigns exponentially decaying weights to past measurements; λ was fixed at 0.9995, and R_k at 5×10^{-3} across the entire epoch; the state of the RNN at $t = 0$, $\mathbf{x}_0$, was assumed to be zero.

Unlike the CKF, which relies on integration, the EKF and the CDKF use gradient information, which in turn necessitate the use of the truncated backpropagation through time method. To unfold the recurrent loop of the neural network, a truncation depth of unity was found to be sufficient in this experiment (see Figure 6.4).

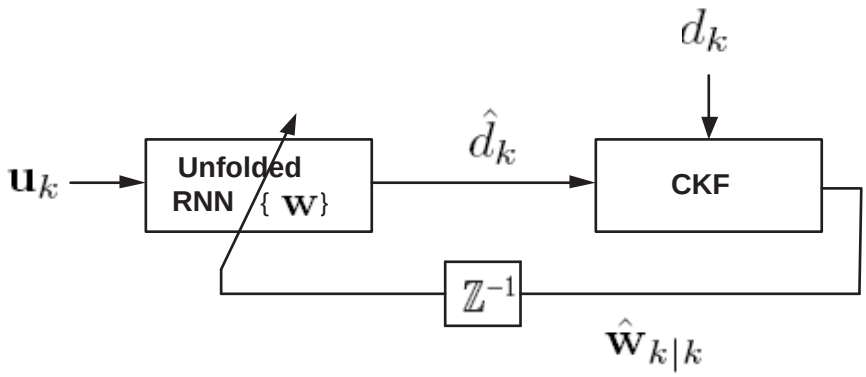


Figure 6.5: CKF-based supervised training of RNN.

Figure 6.5 illustrates how the CKF sequentially updates the weights from the input-output pair during a training phase.

Performance Metric. During the test phase, RNNs were initialized with a 20 time-step long test sequence and allowed to run autonomously using their own output for the next 100 steps. To fairly compare the performance of various filter-trained RNNs, 50 independent training runs were made for each value of α. As a performance metric, the ensemble-averaged cumulative absolute error, which is defined by

$$e_k = \frac{1}{50}\sum_{r=1}^{50}\sum_{i=1}^{k}|d_i^{(r)} - \hat{d}_i^{(r)}|; \quad k = 1, 2, \ldots 100,$$

was used.

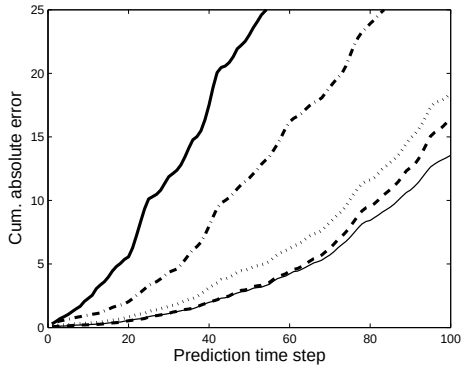
Observations. The long-term accumulative prediction error is expected to increase exponentially with time for the following two reasons:

- Chaotic systems are highly sensitive even to a slight perturbation in their present state, popularly referred to as the butterfly effect [90].

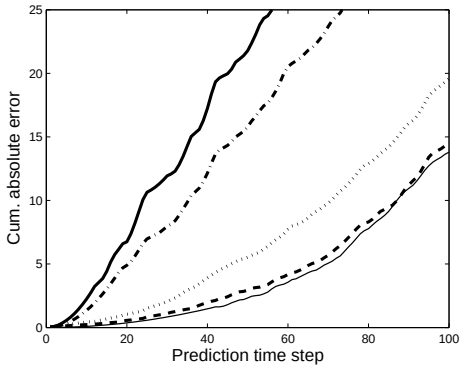
- The prediction error is amplified at each time step due to the closed loop structure.

From Figures 6.6(a) and 6.6(b), it is observed that the RNNs trained with the EKF and the CDKF break down at $\alpha = 2$ and beyond. The CKF-trained RNN performs reasonably well even when $\alpha = 3$, for which the hyperbolic tangent function is 'severely' nonlinear (Figure 6.6(c)). The reason is that the CKF tends to find a better local minimum of the cost function in the weight space than the EKF or the CDKF.

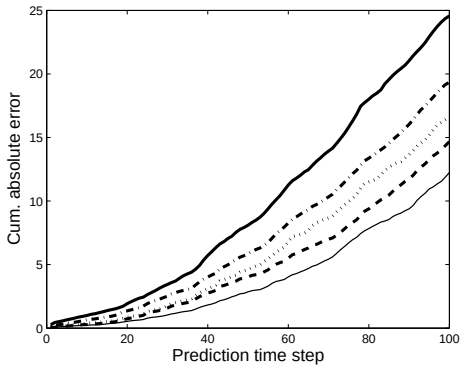
To visualize whether the CKF-trained RNN has captured the true dynamics of the chaotic time series, the phase plot– a three-dimensional diagram with its axes denoting the RNN outputs $\hat{d}_{k+2}, \hat{d}_{k+1}$, and $\hat{d}_k$– was constructed. The desired result is that the RNN closely approximate the true dynamics of the Mackey-Glass system. Figures 6.7(a), 6.7(b) and 6.7(c) show the phase plots of the true dynamics, and the reconstructed dynamics when $\alpha = 1/3$ and $\alpha = 3$, respectively. When $\alpha = 1/3$ the reconstructed phase plot closely resembles the true phase plot as desired; whereas it is not exactly the case when $\alpha = 3$.



(a) EKF-trained RNN.

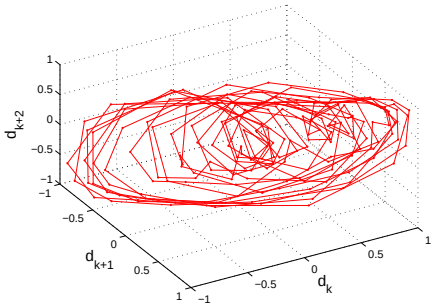


(b) CDKF-trained RNN.

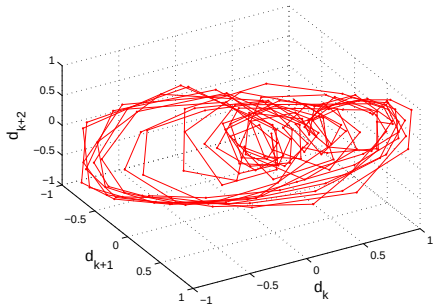


(c) CKF-trained RNN.

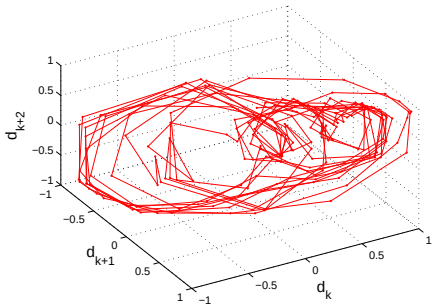
Figure 6.6: Effect of nonlinearity on the autonomous-prediction performance. Nonlinearity is controlled by the parameter α, and the prediction performance is measured by the ensemble-averaged cumulative absolute error criterion ($\alpha = 1/3$ (solid-thin), 2/3 (dashed), 1 (dotted), 2 (dash-dot), and 3 (solid-thick))



(a) True Mackey-Glass phase plot



(b) Reconstructed plot when $\alpha = \frac{1}{3}$



(c) Reconstructed plot when $\alpha = 3$

Figure 6.7: Comparison of two different reconstructed phase plots with the true plot

6.3 Model-Based Signal Processing

In the second experiment, the empirical model of the chaotic Mackey-Glass system was built from the clean input-output data. In contrast, provided the noisy measurements of a dynamic system, the objective of this third experiment was to build a nonlinear empirical model of the dynamic system from noisy measurements for the following purposes:

- To denoise a given test signal (signal enhancement)
- To statistically decide whether the denoised test signal belongs to the empirical model (signal detection)

In this experiment, the idea of directly training RNNs in the supervised mode must be abandoned because the desired (teacher) output is noisy. A similar situation arises in many important real-life applications such as speech signal enhancement, image processing, decoding of symbols transmitted through a noisy wireless channel, and fault diagnosis. To achieve the above objectives, a systematic filtering setup is important.

Cooperative Filtering for Signal Enhancement

The objective of cooperative filtering is to construct an empirical model using (pseudo-) clean data extracted from the noisy data. To accomplish this objective, two distinct estimators, namely, the signal estimator and the weight (parameter) estimator, are coupled to operate in a cooperative manner (see Figure 6.8). At each time instant, the weight parameters of the RNN are estimated from the latest

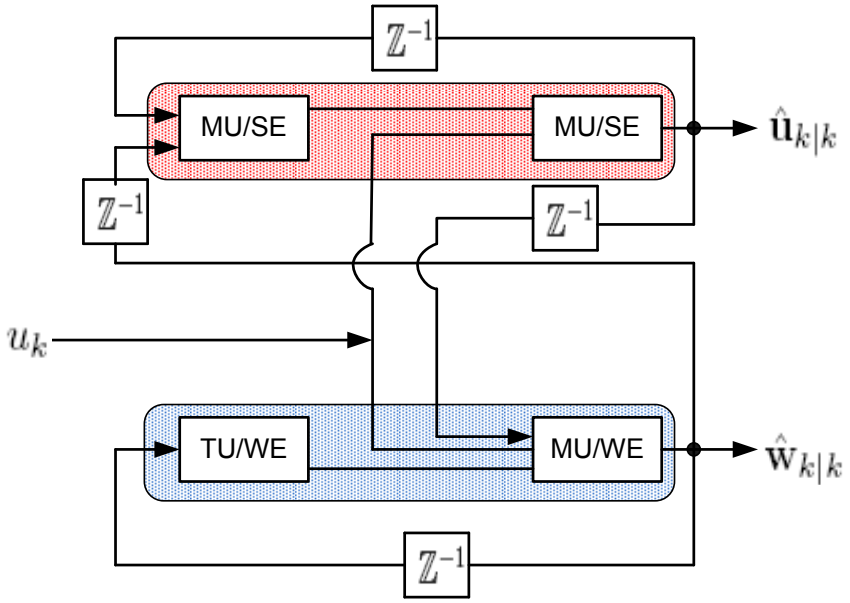


Figure 6.8: Cooperative filtering illustrating interactions between the signal estimator (SE) and the weight estimator (WE); the labels TU and MU denote 'Time Update' and 'Measurement Update', respectively)

signal estimate, and the signal itself is estimated from the latest weight estimate, appropriately.

Data. To generate a noisy time series, the chaotic Mackey-Glass time series was considered again, but it was corrupted by additive white Gaussian noise this time. The signal-to-noise ratio was fixed at 10 dB. As in the second experiment, the architecture of a RNN was chosen to be 7-5R-1, and a hyperbolic tangent function of the form $\varphi(\upsilon) = \tanh(\upsilon)$ was used.

State-Space Models. The dynamic state-space model for the signal estimator can be written as

$$\mathbf{u}_k = \mathbf{f}(\mathbf{u}_{k-1}, \hat{\mathbf{w}}_{k-1|k-1}, \mathbf{x}_{k-2}) + [1 \quad 0 \ldots 0] v_{k-1} \tag{6.2}$$

$$z_k = [1 \quad 0 \ldots 0]\mathbf{u}_k + e_k, \tag{6.3}$$

where

- $\mathbf{u}_k = [u_k \;\; u_{k-1} \ldots u_{k-6}]$ denotes the data window to be estimated
- The state transition function

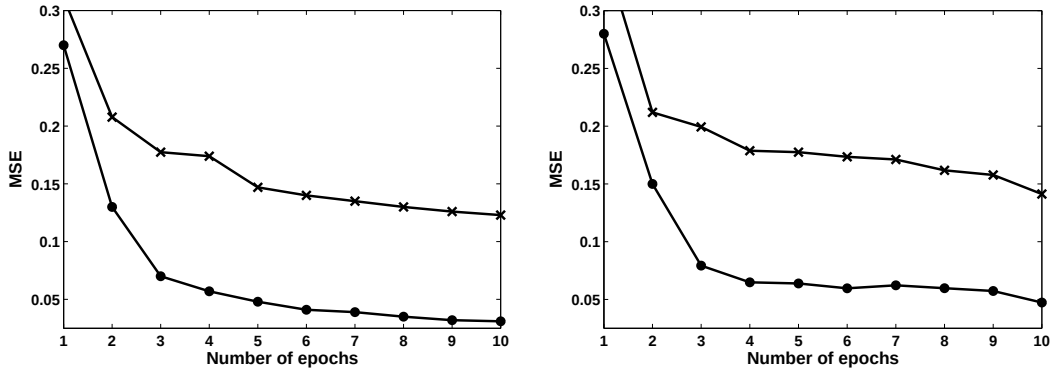
$$\mathbf{f}(.,.,.) = \begin{pmatrix} \hat{W}_{o,k-1|k-1}\varphi\big(\hat{W}_{r,k-1|k-1}\mathbf{x}_{k-2} + \hat{W}_{i,k-1|k-1}\mathbf{u}_{k-1}\big) \\ \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \begin{pmatrix} u_{k-1} \\ \vdots \\ u_{k-6} \end{pmatrix} \end{pmatrix}$$

- The measurement noise e_k was assumed to be $e_k \sim \mathcal{N}(0, \sigma_e^2)$, where the variance σ_e^2 was computed from the prescribed value of the signal-to-noise ratio
- The process noise v_{k-1} was assumed to be $v_{k-1} \sim \mathcal{N}(0, \sigma_v^2)$, where the variance σ_v^2 was fixed to be 10% of σ_e^2; the final result was not sensitive to this choice of percentage as long as it was below 100%
- The initial signal estimate was assumed to be zero with unity covariance.

To set the stage for the state-space model of the weight estimator, (6.3) is rewritten in terms of $\mathbf{w}$ as follows:

$$\begin{aligned} z_k = \mathbf{u}_k[1] + e_k &= W_o\varphi\big(W_r\mathbf{x}_{k-2} + W_i\mathbf{u}_{k-1}\big) + v_{k-1} + e_k \\ &\approx W_o\varphi\big(W_r\mathbf{x}_{k-2} + W_i\hat{\mathbf{u}}_{k-1|k-1}\big) + r_k, \end{aligned}$$

where the measurement noise $r_k \sim \mathcal{N}(0, \sigma_e^2 + \sigma_v^2)$. The state-space model of the weight



(a) Training Phase (b) Test Phase

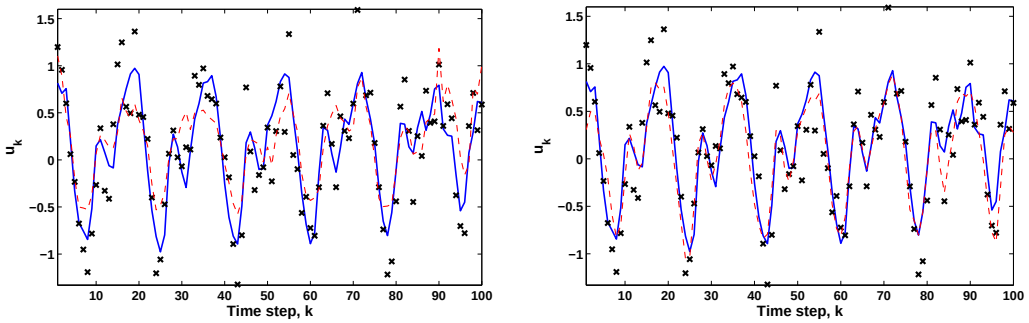
Figure 6.9: Ensemble-averaged (over 50 runs) Mean-Squared Error (MSE) Vs. number of epochs (x- EKF, filled circle- CKF).

estimator is therefore given by

$$\begin{aligned} \mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{q}_{k-1} \\ z_k &= W_o \varphi\big(W_r \mathbf{x}_{k-2} + W_i \hat{\mathbf{u}}_{k-1|k-1}\big) + r_k. \end{aligned}$$

As shown in Figure 6.8, the cooperative filtering system functions only with inputs in a manner similar to unsupervised training.

To fairly compare the performance of the CKF-trained RNN against the EKF and the CDKF-trained RNNs, 50 independent training and test runs were made, each of which consisted of 10 epochs. Each training epoch consisted of a subsequence of 100 examples. During the test phase (that is, at the end of each training epoch here), the trained RNN was presented with a test sequence of length 100. Thus, the ensemble-averaged (over 50 runs) MSE was computed in the course of training and test phases (see Figures 6.9(a) and 6.9(b)). In the course of the test phase, the



(a) EKF (b) CKF

Figure 6.10: Representative test signal before and after cleaning (x- noisy signal (or measurements), dotted thin- signal after cleaning, thick- original clean signal)

weight estimator remained turned off. As shown in Figures 6.9(a) and 6.9(b), the CKF improves performance by a discernable margin in both the training and test phases.

Figures 6.10(a) and 6.10(b) show the representative cleaned test signals obtained from the EKF and the CKF, respectively, at the end of the tenth epoch. The CKF significantly improves the quality of the signal as compared to the EKF.

Signal Detection

Motivated by the problem of detecting targets buried in sea clutter [122, 43], the third experiment was further augmented to deal with a signal detection scenario. To systematically perform signal detection, the consistency check– making a statistical decision whether the test signal is consistent with the trained model– based on the normalized innovations squared (NIS) statistic of signal estimators was introduced.

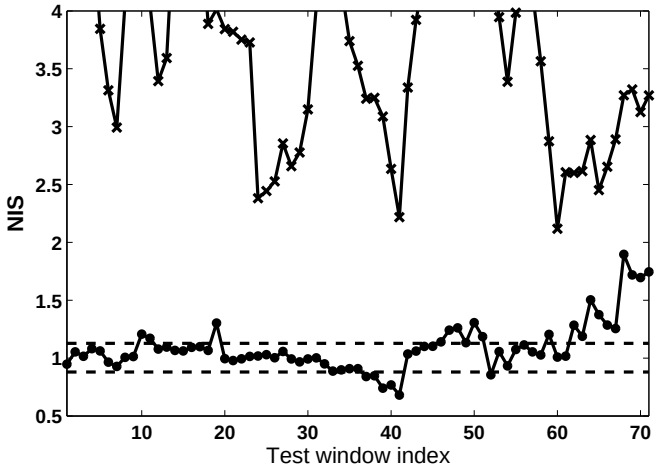


Figure 6.11: Normalized innovations squared (NIS) statistic Vs. test window index (x- EKF, filled circle- CKF, dotted thick- 95% confidence intervals).

Under the hypothesis that the test signal is consistent, the NIS statistic, defined by

$$\epsilon_k = [\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}]^T P^{-1}_{zz,k|k-1}[\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}],$$

is a realization of the chi-squared distribution with n_z degrees of freedom, where n_z is the dimension of the measurement vector [9].

In this experiment, the NIS statistic of the test data was computed as follows: The test data of length 100 was divided into a number of overlapping data windows of length $K = 10$. Two adjacent windows were separated by one time step. Thus, we were able to obtain 71 data windows. The ensemble-averaged (over $N = 50$ runs) NIS statistic for all these windows were then computed. For example, the NIS statistic of the first window was computed as

$$\bar{\epsilon}(1) = \frac{1}{NK} \sum_{n=1}^{N=50} \sum_{k=1}^{K=10} \epsilon_k(n).$$

To accept the hypothesis for the consistency at 95% confidence level, the confidence interval was computed from [72]:

$$I \approx [\frac{1}{2NK}(\sqrt{(2NK-1)} - 1.96)^2, \frac{1}{2NK}(\sqrt{(2NK-1)} + 1.96)^2].$$

In this experiment, the confidence interval is shown by the dotted lines in Figure 6.11. The desired result is that the NIS statistic lie inside those confidence intervals more than 95% of the time. As can be seen from Figure 6.11, the CKF provides a reliable detection result. The CKF result indicates that the test signal belongs to the trained model with 95% confidence approximately.

Summary

In this chapter, the formulation of the square-root cubature Kalman filter (SCKF) is successfully validated through three different filtering problems. In all these problems, the SCKF significantly outperforms other presently known nonlinear filters.

Chapter 7

Synopsis and Future Research

7.1 Synopsis

For linear Gaussian dynamic systems, the classic Kalman filter provides the optimal estimate of the state in the minimum mean-squared error sense. In general, however, real-world systems are nonlinear, non-Gaussian or both. In this context, it is difficult to obtain a closed-form solution for the state estimate, and therefore some approximations must be made. Finding an approximate, yet more accurate nonlinear filtering solution has been the subject of intensive research since the original formulation of the Kalman filter in 1960. Unfortunately, the presently known nonlinear filtering algorithms may experience divergence, the curse of dimensionality or both.

In this thesis, an approximate Bayesian filter is derived for discrete-time nonlinear filtering problems, which is named the *cubature Kalman filter* (CKF). Under the assumption that the predictive density of the joint state-measurement vector is Gaussian, the Bayesian filter reduces to the problem of how to compute multi-dimensional

integrals, whose integrands are all of the form *nonlinear function* $\times$ *Gaussian density*. We are therefore emboldened to say that the cubature rule is the method of choice for solving this problem efficiently. Unfortunately, the cubature rule has been overlooked in the literature on nonlinear filters for more than four decades. In order to numerically compute these integrals, a third-degree spherical-radial cubature rule is proposed.

A few striking properties of the CKF are summarized as follows:

- *Derivative Free*: The cubature rule is derivative free. This useful property helps broaden the applicability of the CKF to situations where it is difficult to compute Jacobians and Hessians, e.g., physics-based models that include dead zones, look-up tables, etc.

- *Efficient Cubature Rule*: The third-degree cubature rule has a theoretical lower bound of $2n$ cubature points, where n is the dimension of the region of integration. The proposed spherical-radial rule also entails $2n$ points. It is therefore considered a *hyper-efficient* cubature rule.

- *Closest Known Solution*: Suppose that the statistics of the state and measurement processes are known up to their second order. According to the maximum entropy principle also called Jaynes' principle [50], it is Gaussian that could be used to model those processes (including their joint density) at best. Based on this line of thinking, we may say that the CKF is the closest known approximation to the nonlinear Bayesian filter in a "Gaussian environment". It is also interesting to note that the CKF functions similarly to the Kalman filter in a linear setting.

- *Reasonable Computational Cost*: Because the proposed cubature rule entails $2n$ cubature points, we are required to compute $2n$ function evaluations at each update cycle. Hence, when n increases, the computational cost in terms of the function evaluations increases linearly whereas it scales cubically in terms of *flops*, similarly to the extended Kalman filter. This suggests that the CKF reduces the effect of the curse of dimensionality, but, by itself, it is not a complete remedy for the problem of dimensionality. Because the cubature point set is independent of the integrands, it may be computed off-line and stored to speed up the filter's execution.

It is the combination of these properties that makes the CKF an attractive choice for many diverse applications. For example, consider the supervised training of recurrent neural networks. For this application, the CKF offers the following benefits over the conventional backpropagation algorithm [5]:

- The CKF converges faster.
- The CKF is well-suited to handle noisy and nonstationary training data.
- Grounded in the fact that it incorporates prior information, the CKF has a built-in ability to regularize the solution of an ill-posed inverse problem.

Among several members of the Bayesian filter family, the CKF has another special attribute– its theory is rooted in integration as opposed to the differentiation adopted in the EKF and its variants. This offers a couple of key benefits. First, during the training phase, a recurrent loop needs to be unfolded so that the unfolded network has a truncation depth of unity. Therefore, the burden of finding an optimal depth is eliminated. Second, the CKF does not suffer from the vanishing gradient problem.

For these reasons, the CKF may be considered as the method of choice for training RNNs as shown in Chapter 6.

In general, substantial performance benefits are expected to gain by applying the CKF to many applications, where the EKF has been the method of choice in the past, without significantly increasing the computational cost.

7.2 Directions for Future Work

There is much room for further work on problems related to nonlinear estimation. The theory of cubature Kalman filtering may provide a basis for contributions. A number of interesting problems to be treated in the years ahead include:

- Nonlinear smoothing
- Extension to deal with non-Gaussianity
- Robust nonlinear filtering
- Stability analysis
- CKF-based cooperative filtering as a substitute for the EM algorithm
- CKF-integrated control

7.2.1 Nonlinear Smoothing

The next logical step for nonlinear filtering is to develop nonlinear smoothing algorithms. Smoothing is the estimation of the state at any time within a given window of measurements. For linear-Gaussian systems, the optimal smoothing density is

tractable [97, 33]. For nonlinear systems, however, the situation is different– only the theoretically relevant equations to the smoothing density exist [71].

Recall that the CKF is built on the following three extremely powerful ideas:

- Bayes' rule;
- the statistical properties of the innovations process; and
- the cubature rule for numerical integrations.

Now the question becomes: *How shall we exploit these ideas to develop a nonlinear smoother*? It is hoped that the conditional densities arising in the nonlinear smoother could well be approximated by Gaussian. Consequently, the proposed cubature rule may be used to numerically compute their moments. For the moment, let the name of the resulting smoother be the *cubature Kalman smoother* (CKS). The development of the CKS will have to address two significant issues:

- It is not clear yet how the properties of the innovations process could fully be exploited.
- To improve the numerical stability, the CKS need to be extended so that it operates on the square-root covariances only.

On the practical side, the CKS may, like the CKF, find use in many applications. Specifically, I would like to apply the CKS to the following two problems:

- *Supervised Training of RNNs*: In Chapter 6, the epoch-based training is used to build a RNN-based empirical model. This conventional training method may be replaced with a *repetitive filtering-smoothing* type procedure. This procedure will traverse through the data a number of times and move forward as a new set

of data is presented. This new procedure is expected to outperform the ad-hoc epoch-based training in terms of accuracy and efficiency.

- *Optimal Control*: Dynamic programming often arises in optimal control problems [77]. However, the closed-form solution to dynamic programming is inherently burdensome. The CKS may be viewed as a potential approximate solution to dynamic programming.

7.2.2 Extension to Deal with Non-Gaussianity

We have found that the CKF is a good candidate to tackle nonlinear Gaussian filtering problems. In order to facilitate its use in a non-Gaussian environment, it could also be expanded through the Gaussian-mixture approximation. The reason is that any non-Gaussian noise distribution can be expressed as, or approximated sufficiently well by, a finite sum of Gaussian densities. For an exposition of this similar idea using the Gauss-Hermite quadrature filter, see [6]. As outlined in Chapter 2, however, we could expect two serious limitations associated with this Gaussian mixture-based filtering approach, namely, degeneration and the growing-memory problem. Of these, the problem of degeneration is more serious, and should be the focus of further research.[1]

7.2.3 Robust Filtering

The CKF is built under the assumption that a well-defined dynamic state-space model exists. Unfortunately, this assumption limits the utility of the CKF. In practice, uncertainties arise in many forms including

[1]For linear filtering in a non-Gaussian noise environment, two different methods, namely, the Masreliez filter [76], and the Kalman-Levy filter [109], have been proposed in the literature.

- Incorrect or incomplete assumptions about the initial conditions.
- System model mismatch.
- Uncertainty in the noise models

Of these, the most commonly encountered form of uncertainty results from poor or incomplete knowledge of the noise statistics [42]. Usually, the process and measurement noise sources are assumed to be zero-mean Gaussians, and as a result, the remaining task involves finding more accurate estimates of the noise covariances. The measurement noise covariance is relatively easy to define because it explicitly depends on the precision of a measuring instrument. However, the process noise covariance may be more difficult to define as it must accurately account for random disturbances of a system in operation, which are often non-stationary.

A logical approach to tackling these uncertainties is to develop a more robust design of the CKF. The robust cubature filter must ensure that the error signals are bounded within some tolerance limits despite the effects of uncertainties on the system. In the past, researchers have proposed a number of solutions to the various uncertainties that can occur in different contexts. They include:

- H_∞ filters [104];
- Variable structure filters combining the Kalman filter with variable structure control [42]

These algorithms assume the modelled systems to be linear. Making use of these robust linear filtering techniques for nonlinear settings would therefore be of practical significance.

7.2.4 Stability Analysis

Another issue that deserves a future treatment lies in the stability analysis of the CKF. The stability analysis is of practical interest for the following reasons [98]:

- It suggests under what conditions the estimation errors are bounded; hence, we may predict in advance if the CKF would break down for a given problem before committing it to hardware.
- It helps to take precautious measures against potential blow ups.

The stability analysis for a general nonlinear setup should include:

- Numerical errors stemming from the use of the cubature rule.
- Errors due to the approximation of a possibly non-Gaussian posterior density by a Gaussian density.
- The cumulative effect of errors when propagated recursively over time.

7.2.5 CKF-based Cooperative Filtering and EM

The Expectation-Maximization (EM) algorithm is a well-known method for learning the parameters of a probabilistic model, where the model depends on hidden state (latent) variables [78, 28]. This learning scenario arises in many applications that include data clustering in machine learning, medical image reconstruction, and computer vision. As its name suggests, the EM algorithm alternates between two steps– (i) The E-step, in which the marginal likelihood function– the conditional probability density of the hidden states (also called the latent variables) given the parameters– is updated. (ii) The M-step, in which the parameters of interest are determined by

maximizing the marginal likelihood. These two steps are cooperatively updated by feeding their respective latest estimates as the other's input. At the end of a number of iterations of the EM algorithm, we have a probability density of the state variables together with a point estimate for the parameters.

The EM algorithm with these two interacting steps is comparable to what is happening inside cooperative filtering with the signal and weight estimators as described in Chapter 6. However, a distinguishing feature of the EM algorithm is that it is not fully Bayesian, whereas cooperative filtering takes a fully Bayesian approach. Cooperative filtering operates on the probability densities of the state variables as well as the parameters. It is therefore hoped that the EM algorithm could be improved by taking either one of the following two steps:

- Replacing the E-step or the M-step of the EM algorithm with the CKF.
- Replacing the whole EM algorithm with the CKF-based cooperative filtering algorithm.

The rationale for the claim of improved accuracy is attributed to the following fact: It is well-known in the estimation community that the parameters should be treated as some random variables instead of some unknown numbers because this idea would allow the use of Bayes' rule in incorporating 'prior' knowledge of the parameters (see also [9], pp. 105-106)(see also [9], pp. 105-106). Future research should attempt to test this idea against a number of challenging problems.

7.2.6 CKF-Integrated Control Algorithms

Many common industrial processes include two key elements: the observer and the controller [102, 21]. For linear closed-loop systems to achieve stability, the observer

and the controller can be designed independently due to the well-known *separation principle*. Since the system state is typically unobservable, the observer is employed to provide an estimate of the unobservable state and the associated confidence on this estimate. The controller then uses the observer's state estimate to generate a control input (see Figure 2.2).

For nonlinear systems, the situation is very different. The separation principle does not hold in general. Of course, the controller performance is intimately linked to the accuracy of the state estimate. The conventional nonlinear state observers include the EKF, the extended Luenberger observer and the nonlinear moving horizon estimator. Future research should investigate the impact of the CKF against these existing observers.

Appendix A

Cubature Kalman Filtering for Continuous-Discrete Systems

In this appendix, the CKF is extended to deal with a state estimation problem whose state-space model is described by a continuous-time process equation and a discrete-time measurement equation. In many practical problems, the process equation is often derived from the underlying physics of a dynamic system, and expressed in the form of a set of differential equations, e.g., the motion of a body as it enters the atmosphere from a high altitude [8]. Since the sensors are digital devices, the measurements are available at discrete times. In this 'hybrid' setup, we have a state-space model with a continuous-time process equation and a discrete-time measurement equation. In the sequel, a more accurate time update is therefore derived whereas the measurement update remains the same as described for the standard CKF [3].

Time Update

Consider a state process generated from

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)dt + \sqrt{Q}d\boldsymbol{\beta}_t, \tag{A.1}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state of the system to be estimated at time t; $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is some known function; $\boldsymbol{\beta}_t$ denotes the standard Brownian motion with increments $d\boldsymbol{\beta}_t$ being independent of $\mathbf{x}_t$; and Q is the diffusion matrix. It is known that the temporal evolution of the probability density of the state at time t obeys the famous Fokker-Planck equation, also called Kolmogorov's forward equation [51]. A closed-form solution to the Fokker-Planck equation is difficult to obtain in general, and numerical methods are often the only avenue.

A continuous-time process equation can be treated to be a limiting case of its discrete-time counterpart with substantially small time intervals between successive states. Because the process equation in (A.1) holds in the sense of Itô [87], it is rewritten as

$$\begin{aligned} \mathbf{x}_t &= \mathbf{x}_s + \int_s^t \mathbf{f}(\mathbf{x}_\tau)d\tau + \int_s^t \sqrt{Q}d\boldsymbol{\beta}\tau \\ &\approx \mathbf{x}_s + \int_s^t \mathbf{f}(\mathbf{x}_\tau)d\tau + \sqrt{Q}(\boldsymbol{\beta}_t - \boldsymbol{\beta}_s), \end{aligned} \tag{A.2}$$

where $0 \leq s \leq t$, and $(t - s)$ is assumed to be 'small'. By numerically integrating over a small interval of δ, we get

$$\mathbf{x}_{k+\delta} = \mathbf{F}(\mathbf{x}_k) + \sqrt{\delta Q}\mathbf{q}_k, \tag{A.3}$$

where

- The mapping function $\mathbf{F}(.)$ can be obtained from the Euler, Milhstein, or the Runge-Kutta approximation method [64]. Suppose that the forth-order Runge-Kutta method is employed to numerically integrate (A.2) over the interval δ. In this case, we have

$$\mathbf{F}(\mathbf{x}_k) = \mathbf{x}_k + \frac{\delta}{6}(K_1 + 2K_2 + 2K_3 + K_4),$$

where

$$\begin{aligned} K_1 &= \mathbf{f}(\mathbf{x}_k) \\ K_2 &= \mathbf{f}(\mathbf{x}_k + \frac{\delta}{2}K_1) \\ K_3 &= \mathbf{f}(\mathbf{x}_k + \frac{\delta}{2}K_2) \\ K_4 &= \mathbf{f}(\mathbf{x}_k + \delta K_3) \end{aligned}$$

- The last term on the right hand side of (A.3) is due to the definition of Itô's stochastic integral.

- $\mathbf{q}_k = (\boldsymbol{\beta}_{k+\delta} - \boldsymbol{\beta}_k)$ is the standard Gaussian random variable.

To predict the state more accurately before receiving the measurement at time $(k+1)$, several integration steps are performed within the measurement-time interval $[k, k+1]$. The time update steps of the continuous-discrete time filter are now described as follows:

Time Update: From time step k to $(k+1)$

1. Divide the time interval $[k, k+1]$ into n equal subintervals, each of length δ.

2. Given $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, P_{k|k})$ at time $t = k$, recursively estimate the predicted state at the i-th intermediate time step as $(i = 1, 2, \ldots n)$:

$$\begin{aligned} \hat{\mathbf{x}}_{k+i\delta|k} &= \mathbb{E}\big[\mathbf{x}_{k+i\delta}|D_k\big] \\ &\approx \int_{\mathbb{R}^{n_x}} F(\mathbf{x})\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k+(i-1)\delta|k}, P_{k+(i-1)\delta|k})d\mathbf{x} \end{aligned} \tag{A.4}$$

3. Recursively estimate the corresponding predicted error covariance $(i = 1, 2, \ldots n)$:

$$\begin{aligned} P_{k+i\delta|k} &= \mathbb{E}\big[(\mathbf{x}_{k+i\delta} - \hat{\mathbf{x}}_{k+i\delta})(\mathbf{x}_{k+i\delta} - \hat{\mathbf{x}}_{k+i\delta})^T|D_k\big] \\ &\approx \delta Q + \int_{\mathbb{R}^{n_x}} (F(\mathbf{x}) - \hat{\mathbf{x}}_{k+i\delta})(F(\mathbf{x}) - \hat{\mathbf{x}}_{k+i\delta})^T \\ &\qquad \times \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k+(i-1)\delta|k}, P_{k+(i-1)\delta|k})d\mathbf{x}. \end{aligned} \tag{A.5}$$

To numerically integrate (A.4)-(A.5), the third-degree spherical-radial cubature rule may be used. When $i = n$ or at time $(k+1)$, the Gaussian predicted state variable with mean $\hat{\mathbf{x}}_{k+1|k}$ and covariance $P_{k+1|k}$ is obtained, which is finally updated using the measurement $\mathbf{z}_{k+1}$ according to the measurement update of the standard CKF.

Appendix B

Comparison of UKF with CKF

Of all innovations-based nonlinear Bayesian filters, it is the unscented Kalman filter (UKF) [52, 53], which shares a number of important common properties with the CKF, e.g., a set of deterministic weighted points, a weighted sum of these points for computing means and covariances, etc. The purpose of this appendix therefore is to compare the UKF with the CKF. To elaborate the approach taken in the UKF, an n-dimensional random variable $\mathbf{x}$ having a symmetric prior density $\Pi(\mathbf{x})$ with the mean μ and covariance Σ is considered. Obviously, the Gaussian is a special case of a symmetric density. Then a set of $(2n+1)$ sample points and weights, $\{\mathcal{X}_i, \omega_i\}_{i=0}^{2n}$ are chosen to satisfy the following moment-matching conditions:

$$\begin{aligned} \sum_{i=0}^{2n} \omega_i \mathcal{X}_i &= \mu \\ \sum_{i=0}^{2n} \omega_i \big(\mathcal{X}_i - \mu_x\big)\big(\mathcal{X}_i - \mu_x\big)^T &= \Sigma. \end{aligned}$$

Among many candidate sets, one symmetrically distributed sample point set, hereafter called the *sigma-point set*, is picked up as follows:

$$\begin{aligned} \mathcal{X}_0 = \mu, \qquad \omega_0 &= \frac{\kappa}{n+\kappa} \\ \mathcal{X}_i = \mu + (\sqrt{(n+\kappa)\Sigma})_i, \qquad \omega_i &= \frac{1}{2(n+\kappa)} \\ \mathcal{X}_{n+i} = \mu - (\sqrt{(n+\kappa)\Sigma})_i, \qquad \omega_{n+i} &= \frac{1}{2(n+\kappa)}, \end{aligned}$$

where $i = 1, 2, \ldots n$ and the i-th column of a matrix $\mathbf{A}$ is denoted by $(\mathbf{A})_i$; the parameter κ is used to scale the spread of sigma points from the prior mean μ, hence the name *scaling parameter*. Due to its symmetry, the sigma-point set matches the skewness. Moreover, to capture the kurtosis of the prior density closely, it is suggested that κ be fixed at $(3-n)$ (Appendix I of [52],[53]). This choice preserves moments up to the fifth order exactly in the simple one-dimensional Gaussian case. In summary, the sigma-point set is chosen to capture a number of low-order moments of the prior density $p(\mathbf{x})$ as correctly as possible.

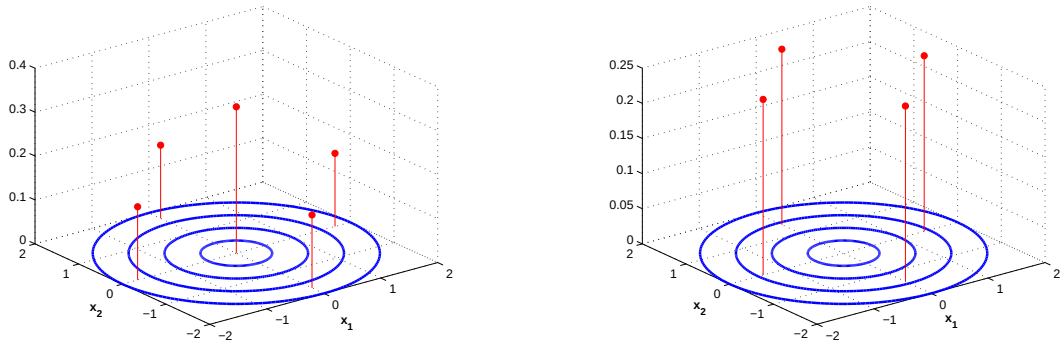
Then the *unscented transformation* is introduced as a method of computing posterior statistics of $\mathbf{y} \in \mathbf{R}^m$ that are related to $\mathbf{x}$ by a nonlinear transformation $\mathbf{y} = \mathbf{f}(\mathbf{x})$. It approximates the mean and the covariance of $\mathbf{y}$ by a weighted sum of projected

sigma points in the $\mathbf{R}^m$ space, as shown by

$$\begin{aligned}
\mathbb{E}[\mathbf{y}] &= \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\Pi(\mathbf{x})d\mathbf{x} \\
&\approx \sum_{i=0}^{2n} \omega_i \mathcal{Y}_i \qquad \text{(B.1)} \\
\text{cov}[\mathbf{y}] &= \int_{\mathbb{R}^n} (\mathbf{f}(\mathbf{x}) - \mathbb{E}[\mathbf{y}])(\mathbf{f}(\mathbf{x}) - \mathbb{E}[\mathbf{y}])^T \Pi(\mathbf{x})d\mathbf{x} \\
&\approx \sum_{i=0}^{2n} \omega_i (\mathcal{Y}_i - \mathbb{E}[\mathbf{y}])(\mathcal{Y}_i - \mathbb{E}[\mathbf{y}])^T, \qquad \text{(B.2)}
\end{aligned}$$

where $\mathcal{Y}_i = \mathbf{f}(\mathcal{X}_i),\ i = 0, 1, \ldots 2n$. The unscented transformation approximating the mean and the covariance, in particular, is correct up to a p-th order nonlinearity when the sigma-point set correctly captures the first p order prior moments. This can be proved by comparing the Taylor expansion of the nonlinear function $\mathbf{f}(\mathbf{x})$ up to p-order terms and the statistics computed by the unscented transformation; here, $\mathbf{f}(\mathbf{x})$ is expanded about the true (prior) mean μ, which is related to $\mathbf{x}$ via $\mathbf{x} = \mu + \mathbf{e}$ with the perturbation error $\mathbf{e} \sim \mathcal{N}(0, \Sigma)$ [52].

The UKF and the CKF share a common property– both use a set of weighted symmetric points. Figure B.1, shows the spread of the weighted sigma-point set and the proposed cubature-point set, respectively in the two-dimensional space; the points and their weights are denoted by the location and the height of the stems, respectively. However, as shown in Figure B.1, for the sigma-point set the stem at the center is highly significant as it carries more weight, whereas the cubature-point set does not have a stem at the center. Hence, they are fundamentally different. Of course, the cubature-point set built into the new CKF is determined with a totally different philosophy in mind:



(a) Sigma point set for the UKF

(b) Third-degree spherical-radial cubature point set for the CKF

Figure B.1: Two different kinds of point sets distributed in the two-dimensional space

- The prior statistic of $\mathbf{x}$ is assumed to be Gaussian instead of a more general symmetric density.

- Subsequently, the problem of how to compute the posterior statistic of $\mathbf{y}$ more accurately is tackled, specifically the first two-order moments of $\mathbf{y}$.

In this line of thinking, the solution for the problem at hand boils down to the efficient third-order cubature rule. Unlike the sigma-point approach, the derivation of a point set for the prior density, and the computation of posterior statistics are not treated as two disjoint problems.

Moreover, suppose that a given function $\mathbf{f}(.)$ can be written as a linear combination of monomials of degree at most three and some other higher odd-degree monomials. Then it is guaranteed that the error incurred in computing the integral of $\mathbf{f}(.)$ with respect to a Gaussian weighting function using the cubature rule vanishes. As in the sigma-point approach, the function $\mathbf{f}(.)$ does not need to be well approximated by the Taylor polynomial about a prior mean.

Finally, the following limitations of the sigma-point set built into the UKF are mentioned, which are not present in the cubature-point set built into the CKF:

- *Numerical inaccuracy.* Traditionally, there has been more emphasis on cubature rules having desirable numerical quality criterion than on the efficiency. It is proven that the cubature rule implemented in a finite-precision arithmetic machine introduces a large amount of roundoff errors when the stability factor defined by $\frac{\sum_i |\omega_i|}{\sum_i \omega_i}$, is larger than unity [115, 36]. Let us look at the formulas (B.1)-(B.2) in the unscented transformation from the numerical integration perspective as described in [69, 48]. In this case, when n goes beyond three, we have $\sum_{i=0}^{2n} |\omega_i| = (\frac{2n}{3} - 1)$ and $\sum_{i=0}^{2n} \omega_i = 1$. Hence, the stability factor scales linearly with n, thereby inducing significant perturbations in numerical estimates for moment integrals.

- *Unavailability of a square-root solution.* The square-rooting operation (or the Cholesky factorization) on the covariance matrices are performed as the first step of both the time and measurement updates in each cycle of both the UKF and the CKF. From the implementation point of view, this is one of the most numerically sensitive operations. In a square-root filter, this is completely avoided as described in Chapter 5. Unfortunately, it may be impossible for any one to formulate a square-root UKF that enjoys numerical advantages similar to the SCKF. The reason is that when a negatively weighted sigma point is used to update any matrix, the resulting down-dated matrix may possibly be non-positive definite. Hence, errors may crop up when executing the 'pseudo' square-root version of the UKF in a limited word-length system (see the pseudo square-root version of the UKF in [121]).

- *Filter instability.* Given no computational errors due to limited word length, the presence of the negative weight in the sigma point set may still prohibit us from writing a covariance matrix P in a squared form such that $P = SS^T$. To state it in another way, the UKF-computed covariance matrix is not always guaranteed to be positive definite. Hence, the unavailability of a square-root covariance causes the UKF to halt its operation. This is disastrous in practical terms. To improve stability, some heuristic solutions such as fudging the covariance matrix artificially (Appendix III of [52]) and the scaled unscented transformation [54] are proposed.

Note that the parameter κ of the sigma-point set may be forced to be zero to match the cubature-point set. Unfortunately, in the past there has been no mathematical justification or motivation for choosing κ to be zero for its associated interpretabilities. Instead, more secondary scaling parameters apart from κ have been introduced in an attempt to incorporate knowledge of prior statistics in the unscented transformation [54].

To sum up, the cubature approach is more accurate, and more principled in mathematical terms than the sigma-point approach.

Appendix C

Particle Filtering: A Moment-Matching Perspective

The purpose of this appendix is to prove with mathematical rigor that particle filters fit into the moment-matching approach. Because particle filtering is rooted in Monte Carlo integration, first, the basic theory of Monte Carlo integration is briefly reviewed. Suppose that we would like to empirically find the expected value of some function $\Phi(\mathbf{x})$ written as

$$\mathbb{E}[\Phi(\mathbf{x})] = \int_{\mathbf{X}} \Phi(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{C.1}$$

where $p(\mathbf{x})$ is the probability density function of the random variable $\mathbf{x}$. Assume that an ensemble with a set of N i.i.d samples $\{\mathbf{x}^i, i = 1, 2, \ldots N\}$ can be generated from $p(\mathbf{x})$. In this case, $p(\mathbf{x})$ is approximated in a discrete form as

$$p(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}^i), \tag{C.2}$$

where $\delta(.)$ is the Dirac delta function. Substituting (C.2) into (C.1) we have the Monte Carlo empirical estimate

$$\begin{aligned}\hat{\mathbb{E}}[\Phi(\mathbf{x})] &= \frac{1}{N}\int_X \Phi(\mathbf{x})\sum_{i=1}^{N}\delta(\mathbf{x}-\mathbf{x}^i)dx \\ &= \frac{1}{N}\sum_{i=1}^{N}\int_X \Phi(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}^i)dx \\ &= \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{x}^i). \end{aligned} \tag{C.3}$$

The error of the Monte Carlo estimate $e = (\mathbb{E}[.] - \hat{\mathbb{E}}[.])$ is of the order $O(\frac{1}{\sqrt{N}})$ meaning that the error reduces as the number of samples N increases. Note that the function $\Phi(\mathbf{x})$ becomes $\mathbf{x}$ and $\mathbf{x}\mathbf{x}^T$ when we compute the mean and covariance of $\mathbf{x}$, respectively.

Now, let us return to the particle filtering theory. Suppose that a set of equally weighted samples or particles $\{\mathbf{x}_{k-1}^i,\ i = 1, 2, \ldots N\}$ is available from the (old) posterior density at time $(k-1)$, $p(\mathbf{x}_{k-1}|D_{k-1})$. Particle filters essentially propagate these particles through the optimal Bayesian filtering solution (2.3)-(2.6), and obtain a new set of particles $\{\mathbf{x}_k^i,\ i = 1, 2, \ldots N\}$ that approximates the (new) posterior density at time k, $p(\mathbf{x}_k|D_k)$. For simplicity, the state-space model (2.1)-(2.2) is considered without control inputs. Next, the particle filtering mechanism is discussed under two basic update steps: the time update and the measurement update.

Time Update

In this first update step, the predictive density is written as (see (2.3)):

$$p(\mathbf{x}_k|D_{k-1}) = \int p(\mathbf{x}_{k-1}|D_{k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})d\mathbf{x}_{k-1}, \tag{C.4}$$

The formula that computes any moment of the predictive state variable is written as

$$\hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] = \int \Phi(\mathbf{x}_k) p(\mathbf{x}_k|D_{k-1}) d\mathbf{x}_k. \tag{C.5}$$

Substituting (C.4) into (C.5), and using Fubini's theorem, we get

$$\hat{\mathbb{E}}[\Phi(\mathbf{x})] = \int\int \Phi(\mathbf{x}_k) p(\mathbf{x}_{k-1}|D_{k-1}) p(\mathbf{x}_k|\mathbf{x}_{k-1}) d\mathbf{x}_{k-1} d\mathbf{x}_k.$$

Because an equally weighted set of samples $\{\mathbf{x}_{k-1}^i,\ i = 1, 2, \ldots N\}$, approximating the old posterior density $p(\mathbf{x}_{k-1}|D_{k-1})$, is available, using the concept of Monte Carlo integration, we write

$$\hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] \approx \frac{1}{N} \sum_{i=1}^{N} \int \Phi(\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) d\mathbf{x}_k,$$

where the state-transition density

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{q}_{k-1}) p(\mathbf{q}_{k-1}|D_{k-1}) d\mathbf{q}_{k-1}.$$

Because the process noise term $\mathbf{q}_{k-1}$ is assumed to be independent of D_{k-1}, we have

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{q}_{k-1}) p(\mathbf{q}_{k-1}) d\mathbf{q}_{k-1}. \tag{C.6}$$

Moreover, given $\mathbf{x}_{k-1}^i$ and $\mathbf{q}_{k-1}$, the state $\mathbf{x}_k$ is deterministic due to the state equation (2.1). Hence, we write

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) = \int \delta(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}^i) - \mathbf{q}_{k-1}) p(\mathbf{q}_{k-1}) d\mathbf{q}_{k-1}. \tag{C.7}$$

Substituting (C.7) into (C.6) yields

$$\begin{aligned}\hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] &\approx \frac{1}{N}\sum_{i=1}^{N}\int\Big[\int\Phi(\mathbf{x}_k)\delta(\mathbf{x}_k-\mathbf{f}(\mathbf{x}_{k-1}^i)-\mathbf{q}_{k-1})d\mathbf{x}_k\Big]p(\mathbf{q}_{k-1})d\mathbf{q}_{k-1}\\ &= \frac{1}{N}\sum_{i=1}^{N}\int\Phi(\mathbf{f}(\mathbf{x}_{k-1}^i)+\mathbf{q}_{k-1})p(\mathbf{q}_{k-1})d\mathbf{q}_{k-1}. \qquad \text{(C.8)}\end{aligned}$$

Suppose a set of i.i.d samples $\{\mathbf{q}_{k-1}^j,\ j=1,2,\dots M\}$ can be drawn from the noise density $p(\mathbf{q}_{k-1})$. Equation (C.8) is further simplified as

$$\begin{aligned}\hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] &\approx \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\Phi(\mathbf{f}(\mathbf{x}_{k-1}^i)+\mathbf{q}_{k-1}^j)\\ &= \frac{1}{MN}\sum_{s=1}^{MN}\Phi(\mathbf{x}_k^s), \qquad \text{(C.9)}\end{aligned}$$

where $\mathbf{x}_k^s = \mathbf{f}(\mathbf{x}_{k-1}^i)+\mathbf{q}_{k-1}^j$. A comparison of (C.5) with (C.9) suggests that the equally weighted set of samples $\{\mathbf{x}_k^s,\ s=1,2,\dots MN\}$ can be thought of as a set of samples drawn from the predictive density $p(\mathbf{x}_k|D_{k-1})$.

Measurement Update

In the measurement update, using Bayes' rule, the posterior density is written as (see (2.6)):

$$p(\mathbf{x}_k|D_k) = \frac{\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|D_{k-1})}{\int\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|D_{k-1})dx_k}, \qquad \text{(C.10)}$$

where, for the measurement equation (2.2), the measurement-likelihood function is given by

$$\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k), R_k).$$

The formula that computes any moment of the posterior state variable is given by

$$\hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] = \int \Phi(\mathbf{x}_k) p(\mathbf{x}_k|D_k) d\mathbf{x}_k \tag{C.11}$$

Substituting (C.10) into (C.11) yields

$$\hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] = \frac{\int \Phi(\mathbf{x}_k)\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|D_{k-1}) d\mathbf{x}_k}{\int \mathcal{L}(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|D_{k-1}) d\mathbf{x}_k}.$$

Because the set of samples $\{\mathbf{x}_k^s,\ s = 1, 2, \ldots MN\}$ are available from the predictive density, due to Monte Carlo integration, we write

$$\begin{aligned} \hat{\mathbb{E}}[\Phi(\mathbf{x}_k)] &\approx \frac{\frac{1}{MN}\sum_{s=1}^{MN} \Phi(\mathbf{x}_k^s)\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k^s)}{\frac{1}{MN}\sum_{s=1}^{MN} \mathcal{L}(\mathbf{z}_k|\mathbf{x}_k^s)} \\ &= \sum_{s=1}^{MN} w^s \Phi(\mathbf{x}_k^s), \end{aligned} \tag{C.12}$$

where the normalized weight

$$\begin{aligned} w^s &= \frac{\mathcal{L}(\mathbf{z}_k|\mathbf{x}_k^s)}{\sum_{s=1}^{MN} \mathcal{L}(\mathbf{z}_k|\mathbf{x}_k^s)} \\ &= \frac{\mathcal{N}(\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k^s), R_k)}{\sum_{s=1}^{MN} \mathcal{N}(\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k^s), R_k)}, \quad s = 1, 2, \ldots MN. \end{aligned}$$

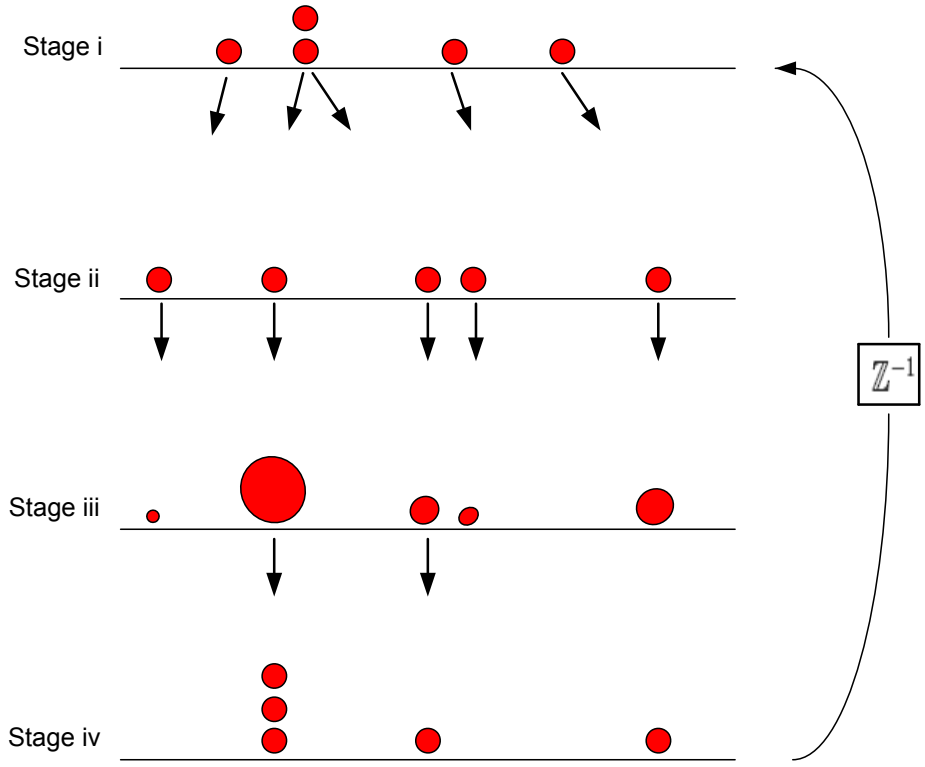


Figure C.1: Propagation of particles in the one-dimensional state-space through four different stages. Stage i: Particles represent the posterior density at time $(k-1)$; Stage ii: Particles represent the predictive density at time k after the time-update step; Stage iii: Particles represent the posterior density at time k after the measurement-update step; Stage iv: Particles represent the posterior density at time k after resampling. The size of a circle amounts to the corresponding weight of that particle

A comparison of (C.12) with (C.11) suggests that the weighted sample set $\{(\mathbf{x}_k^s, w^s),\ s = 1, 2, \ldots MN\}$ can be thought of as a sample set drawn from the posterior density $p(\mathbf{x}_k|D_k)$.

To put pieces together, in each recursion cycle, the set of samples $\{\mathbf{x}_k^s,\ s = 1, 2, \ldots MN\}$ when carrying equal weights of $\frac{1}{MN}$ approximates the predictive density $p(\mathbf{x}_k|D_{k-1})$; whereas it approximates the posterior density $p(\mathbf{x}_k|D_{k-1})$ when the set of weights $\{w^s,\ s = 1, 2, \ldots MN\}$ are associated appropriately. In choosing these

two sets of weighted samples, the central idea is this: *They must tend to accurately match as many low-order moments of the predictive and posterior densities as possible.* Figure C.1 illustrates the key steps involved in particle filtering. Resampling is appended as the final step (stage iv) of each recursion cycle to mitigate the degeneration of weighted samples.

Remarks

- When propagating a set of weighted samples over time, particle filters suffer from the growing-memory problem. To avoid this issue, M is fixed at unity. To be more specific, each propagated particle is perturbed by a randomly drawn process noise sample in the time-update step.

- In the measurement update, to numerically compute any moment of the posterior state variable, samples from the predictive density instead of the posterior density are typically used. This is therefore comparable to the mechanism of importance sampling. However, it should be noted the purpose of importance sampling, which is to reduce the variance of an estimate, is not achieved in particle filters. The reason is that the predictive density carries less accurate information about the current state than the posterior density.

Bibliography

[1] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automatic Cont.*, vol. 17, no. 4, pp. 439-448, Aug. 1972.

[2] B. D. O. Anderson and J. B. Moore, *Optimal filtering,* Eaglewood Cliffs, NJ: Prentice Hall, 1979.

[3] I. Arasaratnam and S. Haykin, "Cubature Kalman filtering: A powerful tool for aerospace applications," under review, *Int'l Radar Conf. 2009*, Bordeaux, France, Oct. 2009.

[4] I. Arasaratnam and S. Haykin, "Cubature Kalman filters", forthcoming *IEEE Trans. Automatic Cont.*, vol. 54, June 2009.

[5] I. Arasaratnam and S. Haykin, *Nonlinear Bayesian Filters for Training Recurrent Neural Networks*, Book Ch., *Advances in Artificial Intelligence*, A. Gelbukh and E. Morales, Eds., pp. 12-33, Springer 2008.

[6] I. Arasaratnam and S. Haykin, "Square-root quadrature Kalman filtering", *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2589-2593, June 2008.

[7] I. Arasaratnam, S. Haykin and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature", *Proc. IEEE*, vol. 95, no. 5, pp. 953-977, May 2007.

[8] M. Athans, R. P. Wishner and A. Bertolini, "Suboptimal state estimation for continuous -time nonlinear systems from discrete noise measurements," *IEEE Trans. Automatic Cont.*, vol. 13, pp. 504-514, 1968.

[9] Y. Bar Shalom, X. R. Li and T. Kirubarajan, *Estimation with applications to tracking and navigation,* NY: Wiley & Sons, 2001.

[10] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, NY: Academic Press, 1977.

[11] D. P. Bertsekas, "Incremental least squares methods and the extended Kalman filter," *SIAM J. Optimization*, vol. 6, pp. 807−822, 1996.

[12] R. E. Bellman, *Adaptive control processes*, NJ: Princeton Univ. Press, 1961.

[13] R. S. Bucy and K. D. Senne, "Digital synthesis of nonlinear filters," *Automatica,* vol. 24, no. 6, pp. 789-801, 1974.

[14] R. S. Bucy and H. Youssef, "Nonlinear filter representation via spline functions," *5th Symp. Nonlinear Estimation*, pp. 51-60, 1974.

[15] O. Cappe, S. J. Godsill and E. Moulines, "An overview of existing methods and recent advances in sequential Monte carlo," *Proc. IEEE*, vol. 95, no. 5, May 2007.

[16] D. E. Catlin, *Estimation, control and the discrete Kalman filter*, NY: Springer-Verlag, 1989.

[17] S. Challa, Y. Bar-Shalom and V. Krishnamurthy, "Nonlinear filtering via generalized Edgeworth series and Gauss Hermite Quadrature," *IEEE Trans. Signal Processing,* vol. 48, no. 6, pp. 1816-1820, June 2000.

[18] B. Cipra, "The Best of the 20th Century: Editors Name Top 10 Algorithms," *SIAM News*, vol. 33, no. 4, May 2000.

[19] R. Cools, "Constructing cubature formulas: the science behind the art," *Acta Numerica 6*, pp. 1 − 54, 1997.

[20] R. Cools, "Advances in multidimensional integration," *J. of Comput. & Applied Math.*, vol. 149, no. 1, pp. 1-12, Dec. 2002.

[21] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, Chapman & Hall, 2004.

[22] C. O. Culver, "Optimal Estimation for Nonlinear Stochastic Systems," Ph.D. dissertation, Dept. Aero. & Astro., MIT, Cambridge, 1969.

[23] F. Daum, "Nonlinear filters: Beyond the Kalman filter," *Aerospace and Electronic Sys. Magazine*, vol. 20, no. 8, pp. 57-69, Aug. 2005.

[24] F. Daum and J. Huang, "Curse of dimensionality and particle filters", *Proceedings, IEEE Aerospace Conf.*, vol.4, pp. 1979-1993, Mar. 2003.

[25] F. Daum, "Exact finite-dimensional nonlinear filters," *IEEE Trans. Autom. Contr.,* vol. 31, pp. 616−622, 1986.

[26] F. Daum and R. Fitzgerald, "Decoupled Kalman filters for phased array radar tracking," *IEEE Trans. Autom. Contr.,* vol. 28, no. 3, pp. 269-283, 1983.

[27] R. J. P. de Figueiredo and J. G. Jan, "Spline filters," *2nd symp. on nonlinear estimation*, pp. 127-138, 1971.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. of the Royal Stat. Soc.: Series B*, vol. 39, no. 1, pp. 138, Nov. 1977.

[29] D. Dochain, "State and parameter estimation in chemical and biochemical processes: a tutorial," *J. of Process Contr.,* vol. 13, pp. 801-818, 2003.

[30] R. C. Dorf and R. H. Bishop, *Modern control systems*, 10th ed., NJ: Prentice Hall, 2005.

[31] A. Doucet, J. de Freitas and N. Gordon, *Sequential Monte Carlo in Pracice,* Cambrige University Press, 2001.

[32] G. B. Folland, "How to integrate a polynomial over a sphere," *The American Math. Monthly*, vol. 108, no. 5, pp. 446-448, May 2001.

[33] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Trans. Autom. Control*, vol. 14, no. 4, pp. 387−390, Aug. 1969.

[34] P. Frost and T. Kailath, "An Innovations Approach to Least-Squares Estimation-Part III: Nonlinear Estimation in White Gaussian Noise," *IEEE Trans. Autom. Contr.,* vol. AC-16, no. 3, June 1971.

[35] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592-2602, Oct. 2003.

[36] A. Genz and B. D. Keister, "Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight," *J. Comput. Appl. Math.*, vol. 71, no. 2, pp. 299-309, 1996.

[37] A. Gelb, ed., *Applied Optimal Estimation*, MIT press, 1974.

[38] T. Gerstner and M. Griebel, "Numerical integration using sparse grids" *Numerical Algorithms*, vol. 18, no. 4, pp. 209 − 232, 1998.

[39] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Baltimore: The John Hopkins Univ. Press 1996.

[40] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc.-F,* vol. 140, pp. 107-113, April 1993.

[41] M. Grewal and A. Andrews, *Kalman filtering: Theory and practice using Matlab*, 2nd ed., NY: Wiley, 2001.

[42] S. Habibi and R. Burton, "The variable structure filter," *ASME J. Dyn. Sys, Meas. and Cont.*, vol. 125, no. 3, p. 287-293, Sept. 2003.

[43] S. Haykin and B. X. Li, "Detection of signals in chaos," *Proc. IEEE*, vol. 83, pp. 95−122, Jan. 1995.

[44] S. Haykin, *Adaptive filter theory*, 3rd ed., NJ: Prentice Hall, 1996.

[45] S. Haykin, *Neural networks and learning machines*, 3rd ed., NJ: Prentice Hall, 2008.

[46] S. Haykin, ed., *Kalman filtering and neural networks*, NY: Wiley, 2001.

[47] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Automatic Cont.*, vol. AC-9, pp. 333-339, Oct. 1964.

[48] D. Hu, Y. Wu, M. Wu and X. Hu, "A numerical integration perspective on Gaussian filters," *IEEE Trans. Signal Process.*, vol. 54, no. 8, pp. 2910-2922, Aug. 2006.

[49] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Automat. Control*, vol. 45, no. 5, pp. 910-927, May 2000.

[50] E.T. Jaynes, "Information theory and statistical mechanics," *Phys. Rev. 106*, no. 4, pp. 620-630, 1957.

[51] A. H. Jazwinski, *Stochastic processes and filtering theory*, NY: Academic, 1970.

[52] S. J. Julier, J.K. Ulhmann and H. F. Durrant-Whyte, "A new method for non-linear transformation of means and covariances in filters and estimators," *IEEE Trans. Automatic Cont.,* vol. 45, pp. 472-482, Mar. 2000.

[53] S. J. Julier and J. K. Ulhmann, "Unscented filtering and nonlinear estimation", *Proc. IEEE*, vol. 92, no. 3, Mar. 2004.

[54] S. J. Julier and J. K. Uhlmann, "The scaled unscented transformation," *Proc. IEEE American Control Conf.*, pp. 4555 −4559, USA, May 2002.

[55] T. Kailath, "An innovations approach to least-squares estimation- Part I: Linear filtering in additive white noise," *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 64-54, Dec. 1968.

[56] T. Kailath and P. Frost, "An innovations approach to least-squares estimation-Part II: Linear smoothing in additive white noise," *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 655-660, Dec. 1968.

[57] T. Kailath and R. Geesey, "An innovations approach to least squares estimation-Part IV: Recursive estimation given lumped covariance functions," *IEEE Trans. Inf. Theory*, vol. 16, no. 6, pp. 720-727, Dec. 1971.

[58] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 146-181, Mar. 1974.

[59] T. Kailath, A. H. Sayed and B. Hassibi, *Linear Estimation*, NJ: Prentice Hall, 2000.

[60] E. W. Kamen and J. K. Su, *Introduction to optimal estimation*, Berlin: Springer, 1999.

[61] P. Kaminski, A. Bryson and S. Schmidt, "Discrete square root filtering: A survey of current techniques," *IEEE Trans. Automat. control*, vol. ac-16, Dec. 1971.

[62] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, pp. 34-45, Mar. 1960.

[63] A. N. Kolmogorov, "Sur linterpolation et extrapolation des suite stationaries," *C. R. Acad. Sci.*, vol. 208, pp. 20432045, 1939.

[64] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations,* Springer: Berlin, 1999.

[65] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximation," *Automatica*, vol. 24, no. 6, pp. 789-801, 1988.

[66] H. J. Kushner, "On the differential equations satisfied by conditional probability densities of Markov processes, with applications," *J. SIAM Control Ser. A*, vol. 2, no. 1, 1964.

[67] H. J. Kushner, "Approximations to optimal nonlinear filters," *IEEE Trans. Automatic Cont.*, AC-12, pp. 546-556, Oct. 1967.

[68] H. J. Kushner and A. S. Budhiraja, "A nonlinear filtering algorithm based on an approximation of the conditional distribution," *IEEE Trans. Autom. Control*, vol. 45, no. 3, Mar. 2000.

[69] U. N. Lerner, "Hybrid Bayesian networks for reasoning about complex systems," Ph.D. dissertation, Computer Sci. Dept., Stanford Univ., Stanford, CA, 2002.

[70] P. Lecuyer and C. Lemieux, *Recent advances in randomized quasi- Monte Carlo methods*, Ch. 20 in *Modeling uncertainty: An examination of stochastic theory, methods and applications*, MA: Kluwer Academic, 2002.

[71] C. T. Leondes, J. B. Peller and E. B. Stear, "Nonlinear smoothing theory," *IEEE Trans. Systems Sci. Cybern.*, vol. 6, no. 1, pp. 63-71, Jan. 1970.

[72] X. R. Li, Z. Zhao, and V. P. Jilkov, "Practical measures and test for credibility of an estimator," *Proc. on Estimation, Tracking, and Fusion: A Tribute to Y. Bar-Shalom*, Monterey, CA, pp. 481495, May 2001.

[73] J. S. Liu, *Monte Carlo strategies in scientific computing,* Springer, 2001.

[74] L. A. McGee, and S. F. Schmidt, "Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry," *NASA Techmcal Memorandum 86847*, Nov. 1985.

[75] M. C. Mackey and L. Glass , "Oscillation and chaos in physiological control systems," *Science,* vol. 197, no. 43, pp. 287-289, July 1977.

[76] C. J. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 107-110, 1975.

[77] P. Maybeck, *Stochastic Models, Estimation and Control*, vol. III, NY: Academic, 1979.

[78] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, NY: Wiley & Sons, 1996.

[79] J. McNamee and F. Stenger, "Construction of fully symmetric numerical integration formulas," *Numer. Math.*, vol. 10, pp. 327 − 344, 1966.

[80] S. R. McReynolds, "Multidimensional Hermite-Gaussian quadrature formulae and their application to nonlinear estimation," *Proc. 6th Symp. Nonlinear Est.*, pp. 188 − 191, 1975.

[81] R. Mehra, "A comparison of several non-linear filters for re-entry vehicle tracking," *IEEE Trans. Autom. Contr.*, vol. AC-16, no. 4, pp. 307-319, 1971.

[82] J. M. Mendel, *Lessons in Digital Estimation Theory*, NJ: Prentice-Hall, 1986.

[83] L. S. Milescu, G. Akk, and F. Sachs, "Maximum likelihood estimation of ion

channel kinetics from macroscopic currents," *Biophysical J.*, vol. 88, pp. 24942515, April 2005.

[84] H. M. Möller, "On the construction of the number of nodes in cubature formula in Numerische Integration," *Int'l Ser. Numer. Math. 45*, pp. 221230, 1979,

[85] H. Niederreiter, "Quasi-Monte Carlo methods and pseudo-random numbers" *Bull. Amer. Math. Soc.*, vol. 84, pp. 957 − 1041, 1978.

[86] S. Nakamoria, R. C. Aguilab, A. H. Carazoc and J. L. Perezc, "New design of estimators using covariance information with uncertain observations in linear discrete-time systems," *Applied Math. and Comp.*, vol. 135, no. 2-3, pp. 429-441, Mar. 2003.

[87] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, NY: Springer, 2003.

[88] D. P. O'Leary and P. Hitman, "Parallel QR factorization by householder and Gram-Schmidt algorithms", *Parallel Computing*, vol. 16, pp. 99-112, 1990.

[89] M. Nørgaard, N. K. Poulsen and O. Ravn, "New developments in state estimation of nonlinear systems," *Automatica*, vol. 36, pp. 1627-1638, 2000.

[90] E. Ott, *Chaos in dynamical systems*, 2nd ed., Cambridge University Press, 2002.

[91] K. Petras, "Fast calculation of coefficients in the Smolyak algorithm," *Numerical Alg.*, vol. 26, no. 2, pp. 93 − 109, 2001.

[92] V. Peterka, "Bayesian approach to system identification," in *Trends and progress*

in system identification, P. Eykhoff, ed., pp. 239-304, Pergamon Press, Oxford, 1981.

[93] G. M. Philips, "A survey of one-dimensional and multi-dimensional numerical integration," *Comp. Physics Comm.*, vol. 20, pp. 17-27, 1980.

[94] W. H. Press and S. A. Teukolsky, "Orthogonal polynomials and Gaussian quadrature with nonclassical weighting functions," *Computers in Physics*, pp. 423-426, july/aug. 1990.

[95] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 279 − 297, 1994.

[96] P. Rabinowitz and N. Richter, "Perfectly symmetric two-dimensional integration formulas with minimal numbers of points," *Math. of Computation*, vol. 23, no. 108, pp. 765-779, Oct., 1969.

[97] H. E. Rauch, F. Tung and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, no. 8, pp. 1445 − 1450, 1965.

[98] K. Reif, S. Gunther, E. Yaz and R. Unbehauen, "Stochastic stability of the discrete-time extended Kalman filter," *IEEE Trans. Autom. Cont.*, vol. 44, no. 4, Apr. 1999.

[99] B. Ristic, S. Arulampalam and N. Gordon, *Beyond Kalman Filters: Particle Filters for Applications,* Artech House, 2004.

[100] A. P. Sage and J. L. Melsa, *Estimation Theory With Applications to Communications and Control,* NY: McGraw-Hill, 1971.

[101] T. S. Schei, "A Finite difference method for linearizing in nonlinear estimation algorithms," *Automatica,* vol. 33, no. 11, pp. 2051-2058, 1997.

[102] D. E. Seborg and M. A. Henson, eds.,*Nonlinear Process Control*, NY: Prentice Hall, 1996.

[103] M. Simandl, J. Kralovec and T. Söderstörm, "Advanced point-mass method for nonlinear state estimation," *Automatica*, vol. 42, no. 7, pp. 1133-1145, 2006.

[104] D. Simon, *Optimal State Estimation: Kalman, H Infinity and Nonlinear Approaches*, NJ: Wiley, 2006.

[105] H. Singer, "Generalized Gauss-Hermite filtering," *J. Adv. in Stat. Analysis,* vol. 92, pp. 179-195, May 2008.

[106] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet. Math. Dokl. 4*, pp. 240-243, 1963.

[107] I. H. Sloan and S. Joe, *Lattice methods for multiple integration*, Oxford: Oxford Univ. Press, 1994.

[108] S.L. Sobolev, "Formulas of mechanical cubature on the surface of a sphere," *Sibirsk. Math.* vol. Z.3, pp. 769 − 796, 1962 (Russian).

[109] D. Sornette, and K. Ide, "The Kalman-Levy filter," *Physica D*, vol. 151, pp. 142174, 2001.

[110] R. F. Stengel, *Optimal Control and Estimation*, NY: Dover Pub., 1994.

[111] J. Stoer and R. Bulirsch, *Introduction to Numerical Analaysis*, 3rd ed., Springer, 2002.

[112] C. Stone, *A course in probability and statistics,* Duxbury Press, 1996.

[113] R. L. Stratonovich, *Conditional Markov Processes and Their Application to the Theory of Optimal Control*, New York: American Elsevier, 1968.

[114] A. H. Stroud, *Gaussian quadrature formulas*, NJ: Prentice Hall, 1966.

[115] A. H. Stroud, *Approximate calculation of multiple integrals*, NJ: Prentice Hall, 1971.

[116] Y. Song and J. W. Grizzle, "The extended Kalman filter as a local asymptotic observer for discrete-time nonlinear systems," *J. Math. Syst. Estim. Contr.*, vol. 5, pp. 59-78, 1995.

[117] H. W. Sorenson and A. R. Stubberud, "Nonlinear filtering by approximation of *a posteriori* density," *Int. J. Contr.*, vol. 8, no. 1, pp. 33-51, 1968.

[118] H. W. Sorenson,"Recursive Estimation for Nonlinear Dyanmic Systems," ch. 6 in *Bayesian Analysis of Time Series and Dynamic Models*, J. C. Spall, ed., NY: Marcel Dekker, 1988.

[119] K. Srinivasan, "State Estimation by orthogonal expansion of probability distribution," *IEEE Trans. Automatic Contr.*, vol. AC-15, no. 1, pp. 3-10, Feb. 1970.

[120] W. Tam and D. Hatzinakos, "An efficient radar tracking algorithm using multidimensional Gauss-Hermite quadratures," *IEEE Int'l. Conf. Acoust. Speech Signal Process. 5*, pp. 3777 − 3780, 1997.

[121] R. van der Merwe and E. Wan, "The square-root unscented Kalman filter for

state and parameter estimation," *Proc. Int'l Conf. Acous. Speech and Signal Process. (ICASSP'01)*, vol. 6, pp. 3461-3464, 2001.

[122] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part III*, Wiley & Sons, 2001.

[123] A. H. Wang and R. L. Klein, "Implementation of nonlinear estimation using monospline, " *13th IEEE conf. Decision and Control,* pp. 1305-1307, 1976.

[124] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*, Cambridge, MA: MIT Press, 1949.

[125] J. L. Williams, "Gaussian mixture reduction for tracking multiple maneuvering targets in clutter," M.S.E.E. Thesis, Air Force Inst. of Tech., Wright-Patterson Air Force Base, OH., 2003 (Available online: http://ssg.mit.edu/ jlwil/).

[126] K. Xiong, H. Y. Zhang and C. W. Chan, " Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261- 270, 2006.